



RESEARCH CENTER

FIELD

Activity Report 2012

Section Software

Edition: 2013-04-24

1. ABS Project-Team	9
2. ABSTRACTION Project-Team	12
3. ACES Project-Team (section vide)	16
4. ADAM Project-Team	17
5. ALEA Project-Team	20
6. ALF Project-Team	21
7. ALGORILLE Project-Team	24
8. ALICE Project-Team	26
9. ALPAGE Project-Team	28
10. AMIB Project-Team	33
11. AOSTE Project-Team	35
12. APICS Project-Team	38
13. ARIC Team	43
14. ARLES Project-Team	47
15. ASAP Project-Team	51
16. ASCLEPIOS Project-Team	53
17. ASCOLA Project-Team	54
18. ASPI Project-Team (section vide)	57
19. ATEAMS Project-Team	58
20. ATHENA Project-Team	62
21. ATLANMOD Team	63
22. AVALON Team	68
23. AVIZ Project-Team	71
24. AXIS Project-Team	76
25. AYIN Team	80
26. BACCHUS Team	81
27. BAMBOO Project-Team	86
28. BANG Project-Team	90
29. BEAGLE Team	91
30. BIGS Project-Team	92
31. BIOCORE Project-Team	93
32. BIPOP Project-Team	94
33. BONSAI Project-Team	97
34. BYMOORE Exploratory Action	100
35. CAD Team	101
36. CAGIRE Team	102
37. CAIRN Project-Team	104
38. CALVI Project-Team	109
39. CAMUS Team	110
40. CARMEL Project-Team	113
41. CARMEN Team (section vide)	116

42. CARTE Project-Team	117
43. CASCADE Project-Team	118
44. CASSIS Project-Team	119
45. CASTOR Team	122
46. CELTIQUE Project-Team	124
47. CEPAGE Project-Team	125
48. CIDRE Project-Team	127
49. CLASSIC Project-Team (section vide)	128
50. CLIME Project-Team	129
51. COFFEE Project-Team	131
52. COMETE Project-Team	132
53. COMMANDS Project-Team	134
54. COMPSYS Project-Team	136
55. CONCHA Project-Team	142
56. CONTRAINTES Project-Team	146
57. CONVECS Team	149
58. COPRIN Project-Team	152
59. CORIDA Project-Team	154
60. CORTEX Project-Team	155
61. CQFD Project-Team (section vide)	157
62. DAHU Project-Team (section vide)	158
63. DANTE Team	159
64. DART Project-Team	160
65. DEDUCTEAM Team	161
66. DEFI Project-Team	162
67. DEMAR Project-Team	165
68. DIONYSOS Project-Team	166
69. DISCO Project-Team	167
70. DISTRIBCOM Project-Team	170
71. DOLPHIN Project-Team	172
72. DRACULA Project-Team	174
73. DREAM Project-Team	175
74. DYLISS Team	177
75. E-MOTION Project-Team	179
76. ESPRESSO Project-Team	181
77. EXMO Project-Team	184
78. FLOWERS Project-Team	186
79. FLUMINANCE Project-Team	203
80. FOCUS Project-Team	205
81. FORMES Team	207
82. FUN Team	210

83. GALAAD Project-Team	211
84. GALEN Team	213
85. GALLIUM Project-Team	215
86. GAMMA3 Project-Team (section vide)	216
87. GANG Project-Team (section vide)	217
88. GECO Team	218
89. GENSCALE Team	219
90. GEOMETRICA Project-Team	221
91. GEOSTAT Project-Team	223
92. GRACE Team	224
93. GRAND-LARGE Project-Team	225
94. GRAPHIK Project-Team	231
95. HIEPACS Project-Team	233
96. HIPERCOM Project-Team	237
97. I4S Team	238
98. IBIS Project-Team	239
99. IMAGINE Team	240
100. IMARA Project-Team	243
101. IMEDIA2 Team	244
102. IN-SITU Project-Team	245
103. INDES Project-Team	254
104. IPSO Project-Team (section vide)	257
105. KERDATA Project-Team	258
106. LAGADIC Project-Team	260
107. LEAR Project-Team	265
108. LFANT Project-Team	266
109. LOGNET Team	269
110. MACS Project-Team	277
111. MADYNES Project-Team	279
112. MAESTRO Project-Team (section vide)	280
113. MAGIQUE-3D Project-Team	281
114. MAGNOME Project-Team	282
115. MAGRIT Project-Team	284
116. MAIA Project-Team	285
117. MANAO Team	286
118. MARELLE Project-Team	287
119. MASAIE Project-Team (section vide)	288
120. MASCOTTE Project-Team	289
121. MATHRISK Team	292
122. MAVERICK Team	295
123. MAXPLUS Project-Team	299

124. MC2 Project-Team	301
125. MCTAO Team (section vide)	304
126. MESCAL Project-Team	305
127. METISS Project-Team	308
128. MEXICO Project-Team	310
129. MICMAC Project-Team (section vide)	312
130. MIMETIC Team	313
131. MINT Project-Team	315
132. MISTIS Project-Team	317
133. MNEMOSYNE Team	320
134. MOAIS Project-Team	322
135. MODAL Project-Team	327
136. MODEMIC Project-Team	330
137. MOISE Project-Team	331
138. MORPHEME Team	333
139. MORPHEO Team	334
140. MOSTRARE Project-Team	336
141. MUTANT Project-Team	337
142. MYRIADS Project-Team	338
143. NACHOS Project-Team	342
144. NANO-D Team	344
145. NECS Project-Team	345
146. NEUROMATHCOMP Project-Team	347
147. NON-A Project-Team (section vide)	349
148. NUMED Project-Team	350
149. OAK Team	351
150. OASIS Project-Team	353
151. OPALE Project-Team	355
152. ORPAILLEUR Project-Team	357
153. PAREO Project-Team	362
154. PARIETAL Project-Team	363
155. PARKAS Project-Team	364
156. PAROLE Project-Team	368
157. PARSIFAL Project-Team	372
158. PERCEPTION Team	374
159. PHOENIX Project-Team	376
160. PI.R2 Project-Team	380
161. PLANETE Project-Team	383
162. POEMS Project-Team	390
163. POLSYS Project-Team	391
164. POMDAPI Project-Team	392

165. POP ART Project-Team	393
166. POPIX Exploratory Action	398
167. POTIOC Team	399
168. PRIMA Project-Team	400
169. PROSECCO Project-Team	407
170. RAP Project-Team (section vide)	409
171. REALOPT Project-Team	410
172. REGAL Project-Team	411
173. REGULARITY Project-Team	413
174. REO Project-Team	414
175. REVES Project-Team	415
176. RMOD Project-Team	417
177. ROMA Team	421
178. RUNTIME Project-Team	422
179. S4 Project-Team	427
180. SAGE Project-Team	428
181. SARDES Project-Team	433
182. SCIPORT Team	436
183. SCORE Team	438
184. SECRET Project-Team (section vide)	439
185. SECSI Project-Team	440
186. SELECT Project-Team	441
187. SEMAGRAMME Team	442
188. SEQUEL Project-Team	445
189. SERPICO Team	446
190. SHACRA Project-Team	451
191. SIERRA Project-Team	454
192. SIMPAF Project-Team	456
193. SIROCCO Project-Team	458
194. SISYPHE Project-Team	461
195. SMIS Project-Team	463
196. SOCRATE Team	465
197. STARS Team	466
198. STEEP Exploratory Action	471
199. TAO Project-Team	472
200. TASC Project-Team	474
201. TEXMEX Project-Team	476
202. TOCCATA Team	481
203. TOSCA Project-Team	486
204. TREC Project-Team	487
205. TRIO Project-Team	488

206. TRISKELL Project-Team	491
207. TYPICAL Project-Team	494
208. URBANET Team	495
209. VEGAS Project-Team	496
210. VERIDIS Project-Team	498
211. VERTECS Project-Team	499
212. VIRTUAL PLANTS Project-Team	500
213. VISAGES Project-Team	504
214. VR4I Team	508
215. WAM Project-Team	510
216. WILLOW Project-Team	513
217. WIMMICS Team	515
218. ZENITH Project-Team	519

ABS Project-Team

4. Software

4.1. Software

This section briefly comments on all the software distributed by ABS. On the one hand, the software released in 2012 is briefly described as the context is presented in the sections dedicated to new results. On the other hand, the software made available before 2012 is briefly specified in terms of applications targeted.

In any case, the website advertising a given software also makes related publications available.

4.1.1. *addict: Stoichiometry Determination for Mass Spectrometry Data*

Participants: Deepesh Agarwal, Frédéric Cazals, Noël Malod-Dognin.

Context. Our work on the stoichiometry determination (SD) problem for noisy data in structural proteomics is described in section 5.2.1. The *addict* software suite not only implements our algorithms DP++ and DIOPHANTINE, but also important algorithms to determine the so-called Frobenius number of a vector of protein masses, and also to estimate the number of solutions of a SD problem, from an unbounded knapsack problem.

Distribution. Binaries for the *addict* software suite are made available from <http://team.inria.fr/abs/software/voratom/>.

4.1.2. *vorpatch and compatch: Modeling and Comparing Protein Binding Patches*

Participants: Frédéric Cazals, Noël Malod-Dognin.

Context. Modeling protein binding patches is a central problem to foster our understanding of the stability and of the specificity of macro-molecular interactions. We developed a binding patch model which encodes morphological properties, allows an atomic-level comparison of binding patches at the geometric and topological levels, and allows estimating binding affinities—with state-of-the-art results on the protein complexes of the binding affinity benchmark. Given a protein complex, *vorpatch* compute the binding patches, while the program *compatch* allows comparing two patches.

Distribution. Binaries for *VORPATCH* and *COMPATCH* are available from <http://team.inria.fr/abs/software/vorpatch-compatch>.

4.1.3. *voratom: Modeling Protein Assemblies with Toleranced Models*

Participants: Frédéric Cazals, Tom Dreyfus.

Context. Large protein assemblies such as the Nuclear Pore Complex (NPC), chaperonin cavities, the proteasome or ATP synthases, to name a few, are key to numerous biological functions. Modeling such assemblies is especially challenging due to their plasticity (the proteins involved may change along the cell cycle), their size, and also the flexibility of the sub-units. To cope with these difficulties, a reconstruction strategy known as Reconstruction by Data Integration (RDI), aims at integrating diverse experimental data. But the uncertainties on the input data yield equally uncertain reconstructed models, calling for quantitative assessment strategies.

To leverage these reconstruction results, we introduced Toleranced Model (TOM) framework, which inherently accommodates uncertainties on the shape and position of proteins. The corresponding software package, *VORATOM*, includes programs to (i) perform the segmentation of (probability) density maps, (ii) construct toleranced models, (iii) explore toleranced models (geometrically and topologically), (iv) compute Maximal Common Induced Sub-graphs (MCIS) and Maximal Common Edge Sub-graphs (MCES) to assess the pairwise contacts encoded in a TOM.

Distribution. Binaries for the software package VORATOM are made available from <http://team.inria.fr/abs/software/voratom/>.

4.1.4. *wsheller: Selecting Water Layers in Solvated Protein Structures*

Participants: Frédéric Cazals, Christine Roth.

Context. Given a snapshot of a molecular dynamics simulation, a classical problem consists of *quenching* that structure—minimizing the potential energy of the solute together with selected layers of solvent molecules. The program `wsheller` provides a solution to the water layer selection, and incorporates a topological control of the layers selected.

Distribution. Binaries for `wsheller` are available from <http://team.inria.fr/abs/software/wsheller>.

4.1.5. *intervor: Modeling Macro-molecular Interfaces*

Participant: Frédéric Cazals.

In collaboration with S. Lorient (The GEOMETRY FACTORY)

Context. Modeling the interfaces of macro-molecular complexes is key to improve our understanding of the stability and specificity of such interactions. We proposed a simple parameter-free model for macro-molecular interfaces, which enables a multi-scale investigation—from the atomic scale to the whole interface scale. Our interface model improves the state-of-the-art to (i) identify interface atoms, (ii) define interface patches, (iii) assess the interface curvature, (iv) investigate correlations between the interface geometry and water dynamics / conservation patterns / polarity of residues.

Distribution. The following website <http://team.inria.fr/abs/software/intervor> serves two purposes: on the one hand, calculations can be run from the website; on the other hand, binaries are made available. To the best of our knowledge, this software is the only publicly available one for analyzing Voronoi interfaces in macro-molecular complexes.

4.1.6. *vorlume: Computing Molecular Surfaces and Volumes with Certificates*

Participant: Frédéric Cazals.

In collaboration with S. Lorient (The GEOMETRY FACTORY, France)

Context. Molecular surfaces and volumes are paramount to molecular modeling, with applications to electrostatic and energy calculations, interface modeling, scoring and model evaluation, pocket and cavity detection, etc. However, for molecular models represented by collections of balls (Van der Waals and solvent accessible models), such calculations are challenging in particular regarding numerics. Because all available programs are overlooking numerical issues, which in particular prevents them from qualifying the accuracy of the results returned, we developed the first certified algorithm, called `vorlume`. This program is based on so-called certified predicates to guarantee the branching operations of the program, as well as interval arithmetic to return an interval certified to contain the exact value of each statistic of interest—in particular the exact surface area and the exact volume of the molecular model processed.

Distribution. Binaries for `Vorlume` is available from <http://team.inria.fr/abs/software/vorlume>.

4.1.7. *ESBTL: the Easy Structural Biology Template Library*

Participant: Frédéric Cazals.

In collaboration with S. Lorient (The GEOMETRY FACTORY, France) and J. Bernauer (Inria AMIB, France).

Context. The ESBTL (Easy Structural Biology Template Library) is a lightweight C++ library that allows the handling of PDB data and provides a data structure suitable for geometric constructions and analyses.

Distribution. The C++ source code is available from <http://esbtl.sourceforge.net/http://esbtl.sourceforge.net/>.

4.1.8. A_purva: Comparing Protein Structure by Contact Map Overlap Maximization

Participant: Noël Malod-Dognin.

In collaboration with N. Yanev (University of Sofia, and IMI at Bulgarian Academy of Sciences, Bulgaria), and R. Andonov (Inria Rennes - Bretagne Atlantique, and IRISA/University of Rennes 1, France).

Context. Structural similarity between proteins provides significant insights about their functions. Maximum Contact Map Overlap maximization (CMO) received sustained attention during the past decade and can be considered today as a credible protein structure measure. The solver A_purva is an exact CMO solver that is both efficient (notably faster than the previous exact algorithms), and reliable (providing accurate upper and lower bounds of the solution). These properties make it applicable for large-scale protein comparison and classification.

Distribution. The software is available from <http://apurva.genouest.org><http://apurva.genouest.org>.

ABSTRACTION Project-Team

5. Software

5.1. The Apron Numerical Abstract Domain Library

Participants: Antoine Miné [correspondent], Bertrand Jeannet [team PopArt, Inria-RA].

Convex polyhedra, Intervals, Linear equalities, Numerical abstract domain, Octagons.

The **APRON** library is dedicated to the static analysis of the numerical variables of a program by abstract interpretation. Its goal is threefold: provide ready-to-use numerical abstractions under a common API for analysis implementers, encourage the research in numerical abstract domains by providing a platform for integration and comparison of domains, and provide a teaching and demonstration tool to disseminate knowledge on abstract interpretation.

The **APRON** library is not tied to a particular numerical abstraction but instead provides several domains with various precision versus cost trade-offs (including intervals, octagons, linear equalities and polyhedra). A specific C API was designed for domain developers to minimize the effort when incorporating a new abstract domain: only few domain-specific functions need to be implemented while the library provides various generic services and fallback methods (such as scalar and interval operations for most numerical data-types, parametric reduced products, and generic transfer functions for non-linear expressions). For the analysis designer, the **APRON** library exposes a higher-level API with C, C++, OCaml, and Java bindings. This API is domain-neutral and supports a rich set of semantic operations, including parallel assignments (useful to analyze automata), substitutions (useful for backward analysis), non-linear numerical expressions, and IEEE floating-point arithmetic.

The **APRON** library is freely available on the web at <http://apron.cri.ensmp.fr/library>; it is distributed under the LGPL license and is hosted at **InriaGForge**. Packages exist for the Debian and Fedora Linux distributions. In order to help disseminate the knowledge on abstract interpretation, a simple inter-procedural static analyzer for a toy language is included. An on-line version is deployed at <http://pop-art.inrialpes.fr/interproc/interprocweb.cgi>.

The **APRON** library is developed since 2006 and currently consists of 130 000 lines of C, C++, OCaml, and Java.

Current and past external library users include the Constraint team (LINA, Nantes, France), the Proval/Démon team (LRI Orsay, France), the Analysis of Computer Systems Group (New-York University, USA), the Sierum software analysis platform (Kansas State University, USA), NEC Labs (Princeton, USA), EADS CCR (Paris, France), IRIT (Toulouse, France), ONERA (Toulouse, France), CEA LIST (Saclay, France), VERIMAG (Grenoble, France), ENSMP CRI (Fontainebleau, France), the IBM T.J. Watson Research Center (USA), the University of Edinburgh (UK).

In 2012, **APRON** has been used in several researches conducted within or in collaboration with the Abstraction project-team: the design of a sufficient-condition generator [23] and the design of a constraint solver based on abstract domains [25].

5.2. The Astrée Static Analyzer of Synchronous Software

Participants: Patrick Cousot [project scientifique leader, correspondent], Radhia Cousot, Jérôme Feret, Laurent Mauborgne, Antoine Miné, Xavier Rival.

Absence of runtime error, Abstract interpretation, Static analysis, Verifier.

ASTRÉE is a static analyzer for sequential programs based on abstract interpretation [41], [32], [42], [34].

The **ASTRÉE** static analyzer [31], [46][1] www.astree.ens.fr aims at proving the absence of runtime errors in programs written in the C programming language.

ASTRÉE analyzes structured C programs, with complex memory usages, but without dynamic memory allocation nor recursion. This encompasses many embedded programs as found in earth transportation, nuclear energy, medical instrumentation, and aerospace applications, in particular synchronous control/command. The whole analysis process is entirely automatic.

ASTRÉE discovers all runtime errors including:

- undefined behaviors in the terms of the ANSI C99 norm of the C language (such as division by 0 or out of bounds array indexing);
- any violation of the implementation-specific behavior as defined in the relevant Application Binary Interface (such as the size of integers and arithmetic overflows);
- any potentially harmful or incorrect use of C violating optional user-defined programming guidelines (such as no modular arithmetic for integers, even though this might be the hardware choice);
- failure of user-defined assertions.

The analyzer performs an abstract interpretation of the programs being analyzed, using a parametric domain (**ASTRÉE** is able to choose the right instantiation of the domain for wide families of software). This analysis produces abstract invariants, which over-approximate the reachable states of the program, so that it is possible to derive an *over*-approximation of the dangerous states (defined as states where any runtime error mentioned above may occur) that the program may reach, and produces alarms for each such possible runtime error. Thus the analysis is sound (it correctly discovers *all* runtime errors), yet incomplete, that is it may report false alarms (*i.e.*, alarms that correspond to no real program execution). However, the design of the analyzer ensures a high level of precision on domain-specific families of software, which means that the analyzer produces few or no false alarms on such programs.

In order to achieve this high level of precision, **ASTRÉE** uses a large number of expressive abstract domains, which allow expressing and inferring complex properties about the programs being analyzed, such as numerical properties (digital filters, floating-point computations), Boolean control properties, and properties based on the history of program executions.

ASTRÉE has achieved the following two unprecedented results:

- **A340–300**. In Nov. 2003, **ASTRÉE** was able to prove completely automatically the absence of any RTE in the primary flight control software of the Airbus A340 fly-by-wire system, a program of 132,000 lines of C analyzed in 1h20 on a 2.8 GHz 32-bit PC using 300 MB of memory (and 50mn on a 64-bit AMD Athlon 64 using 580 MB of memory).
- **A380**. From Jan. 2004 on, **ASTRÉE** was extended to analyze the electric flight control codes then in development and test for the A380 series. The operational application by Airbus France at the end of 2004 was just in time before the A380 maiden flight on Wednesday, 27 April, 2005.

These research and development successes have led to consider the inclusion of **ASTRÉE** in the production of the critical software for the A350. **ASTRÉE** is currently industrialized by **AbsInt Angewandte Informatik GmbH** and is **commercially available**.

5.3. The AstréeA Static Analyzer of Asynchronous Software

Participants: Patrick Cousot [project scientifique leader, correspondent], Radhia Cousot, Jérôme Feret, Antoine Miné, Xavier Rival.

Absence of runtime error, Abstract interpretation, Data races, Interference, Memory model, Parallel software, Static analysis, Verifier.

ASTRÉE A is a static analyzer prototype for parallel software based on abstract interpretation [43], [44], [36]. It started with support from **THÉSÉE** ANR project (2006–2010) and is continuing within the **ASTRÉE A** project (2012–2015).

The **ASTRÉE**A prototype www.astreea.ens.fr is a fork of the **ASTRÉE** static analyzer (see 5.2) that adds support for analyzing parallel embedded C software.

ASTRÉEA analyzes C programs composed of a fixed set of threads that communicate through a shared memory and synchronization primitives (mutexes, FIFOs, blackboards, etc.), but without recursion nor dynamic creation of memory, threads nor synchronization objects. **ASTRÉE**A assumes a real-time scheduler, where thread scheduling strictly obeys the fixed priority of threads. Our model follows the ARINC 653 OS specification used in embedded industrial aeronautic software. Additionally, **ASTRÉE**A employs a weakly-consistent memory semantics to model memory accesses not protected by a mutex, in order to take into account soundly hardware and compiler-level program transformations (such as optimizations). **ASTRÉE**A checks for the same run-time errors as **ASTRÉE**, with the addition of data-races.

Compared to **ASTRÉE**, **ASTRÉE**A features: a new iterator to compute thread interactions, a refined memory abstraction that takes into account the effect of interfering threads, and a new scheduler partitioning domain. This last domain allows discovering and exploiting mutual exclusion properties (enforced either explicitly through synchronization primitives, or implicitly by thread priorities) to achieve a precise analysis.

ASTRÉEA is currently being applied to analyze a large industrial avionic software: 1.6 MLines of C and 15 threads, completed with a 2,500-line model of the ARINC 653 OS developed for the analysis. The analysis currently takes a few tens of hours on a 2.9 GHz 64-bit intel server using one core and generates around 1,200 alarms. The low computation time (only a few times larger than the analysis time by **ASTRÉE** of synchronous programs of a similar size and structure) shows the scalability of the approach (in particular, we avoid the usual combinatorial explosion associated to thread interleavings). Precision-wise, the result, while not as impressive as that of **ASTRÉE**, is quite encouraging. The development of AstréeA continues within the scope of the **ASTRÉE**A ANR project (Section 8.1.1.2).

5.4. The OpenKappa modeling platform

Participants: Monte Brown [Harvard Medical School], Vincent Danos [University of Edinburgh], Jérôme Feret [Correspondent], Walter Fontana [Harvard Medical School], Russ Harmer [Paris VII], Jean Krivine [Paris VII].

Causal traces, Model reduction, Rule-based modelling, Simulation, Static analysis.

OPENKAPPA is a collection of tools to build, debug and run models of biological pathways. It contains a compiler for the Kappa Language [52], a static analyzer [51] (for debugging models), a simulator [50], a compression tool for causal traces [49],[20], and a model reduction tool [4], [48], [53].

OPENKAPPA is developed since 2007 and, the OCaml version currently consists of 46 000 lines of OCaml. Software are available in OCaml and in Java. Moreover, an Eclipse pluggin is available.

OPENKAPPA is freely available on the web at <http://kappalanguage.org> under the LGPL license. Discussion groups are also available on line.

Current external users include the ETH Zürich, the UNAM-Genomics Mexico team. It is used as pedagogical material in graduate lessons at Harvard Medical School, and at the Interdisciplinary Approaches to Life science (AIV) Master Program (Université de Médecine Paris-Descartes).

5.5. Translation Validation

Participant: Xavier Rival [correspondent].

Abstract interpretation, Certified compilation, Static analysis, Translation validation, Verifier.

The main goal of this software project is to make it possible to certify automatically the compilation of large safety critical software, by proving that the compiled code is correct with respect to the source code: When the proof succeeds, this guarantees that no compiler bug did cause incorrect code be generated. Furthermore, this approach should allow to meet some domain specific software qualification criteria (such as those in DO-178 regulations for avionics software), since it allows proving that successive development levels are correct with respect to each other *i.e.*, that they implement the same specification. Last, this technique also justifies the use of source level static analyses, even when an assembly level certification would be required, since it establishes separately that the source and the compiled code are equivalent.

The compilation certification process is performed automatically, thanks to a prover designed specifically. The automatic proof is done at a level of abstraction which has been defined so that the result of the proof of equivalence is strong enough for the goals mentioned above and so that the proof obligations can be solved by efficient algorithms.

The current software features both a C to Power-PC compilation certifier and an interface for an alternate source language frontend, which can be provided by an end-user.

5.6. Zarith

Participants: Antoine Miné [Correspondent], Xavier Leroy [Inria Paris-Rocquencourt], Pascal Cuoq [CEA LIST].

Arbitrary precision integers, Arithmetic, OCaml.

ZARITH is a small (10K lines) OCaml library that implements arithmetic and logical operations over arbitrary-precision integers. It is based on the GNU MP library to efficiently implement arithmetic over big integers. Special care has been taken to ensure the efficiency of the library also for small integers: small integers are represented as Caml unboxed integers and use a specific C code path. Moreover, optimized assembly versions of small integer operations are provided for a few common architectures.

ZARITH is an open-source project hosted at OCamlForge (<http://forge.ocamlcore.org/projects/zarith>) and distributed under a modified LGPL license.

ZARITH is currently used in the **ASTRÉE** analyzer to enable the sound analysis of programs featuring 64-bit (or larger) integers. It is also used in the Frama-C analyzer platform developed at CEA LIST and Inria Saclay.

ACES Project-Team (section vide)

ADAM Project-Team

5. Software

5.1. Introduction

We report on the major software systems that are developed by our research group. FRASCATI, PowerAPI, SPACES relate to the first research direction. AppliIDE and CALICO relate to the second one. Finally, FRACTAL is a general purpose component framework that serves as a foundation for most of our work around reconfigurable middleware.

5.2. AppliIDE

Participants: Laurence Duchien, Clément Quinton [correspondant].

AppliIDE is directly connected to the work of Carlos Parra's PhD thesis [116] and Ubinov ADT's work which covers the definition and implementation of a Context-Aware Dynamic Software Product Line (DSPL) named CAPucine. It provides a set of tools for the selection of features, metamodel transformation and code generation for mobile applications [119]. The current implementation of AppliIDE addresses transformation from CAPucine metamodel towards SCA metamodel, and Spoon EMF metamodel. The transformations were formerly written with Acceleo tool, which is a dedicated language for transformation, enhancing the readability. AppliIDE metamodels are based on the Eclipse Modeling Framework. Code generators are all written in Acceleo.

Inria Evaluation Committee Criteria for Software Self-Assessment: A-3, SO-3, SM-3, EM-2, SDL-2. AppliIDE is registered with the APP (Agence pour la Protection des Programmes) under reference IDDN.FR.001.500004.000.S.A.2010.000.10600.

5.3. CALICO

Participants: Laurence Duchien [correspondant], Antonio de Almeida Souza Neto.

CALICO is an agile development framework for the design and evolution of safe component-based and service-oriented software that has been developed in the context of Guillaume Waignier's PhD thesis [128].

Agile software development relies on an iterative and incremental development cycle that allows the software architect to iterate between the design of the architecture and the debug of the software in its execution context. At each iteration, the architect can evolve its software and check the consistency of its evolution through the execution of static and dynamic analysis tools. During the design and the evolution of the system, the architect can use a set of metamodels to specify the structure of the architecture and its various quality of services requirement. During the deployment, CALICO instantiates the system on the target runtime platform from the models specified and keeps them synchronized with the software during its execution. Through this means, the architect has a conceptual view, which allows him to reason on the critical software properties during its evolution. Moreover, in order to check these evolutions, CALICO provides a unifying framework, which allows reuse of many static analysis tools of software architectures and dynamic debugging tools, that were scattered in different existing platforms. Thus, each change can be statically analyzed on the conceptual view before being propagated to the software system. Dynamic analyses are based on data values available during the execution only. The capture of these values is done through automatic instrumentation of the software system.

Globally, CALICO enables reliable evolution even if the underlying platforms do not natively provide this support. The current version handles four component-based and service-oriented platforms (FRASCATI, FRACTAL, OPENCCM, OPENCOM). Moreover, the benchmarks that we have performed show that CALICO is usable for the design and development of safe applications up to 10,000 components and services, which corresponds to the maximal load of most runtime platforms.

Inria Evaluation Committee Criteria for Software Self-Assessment: A-3, SO-4, SM-3, EM-3, SDL-4. CALICO is an open source software available at <http://calico.gforge.inria.fr>.

5.4. Fractal

Participants: Philippe Merle [correspondant], Romain Rouvoy, Lionel Seinturier.

FRACTAL is a modular, extensible, and reflective component framework. The FRACTAL toolchain can be used to design, implement, deploy and reconfigure any kind of software and middleware system. FRACTAL has initially been designed by both Inria and France Telecom R&D.

Inria Evaluation Committee Criteria for Software Self-Assessment: A-4, SO-4, SM-3-up, EM-3-up, SDL-4-up, DA-3, CD-4, MS-4, TPM-4. FRACTAL is a project of the OW2 consortium for open-source middleware. Web site: <http://fractal.ow2.org>. License LGPL. Some of the research activities around FRACTAL [91], [90], [124] are on top cited publications of the CBSE research community [109]. The ADAM project-team members are among the top committers of the project with 33.8% of all commits and they are the principal contributors for several modules including AOKell [124], Fraclet, the Inria ODL F4E [95], [96], Juliac, Koch. Philippe Merle is the leader of the OW2 FRACTAL project.

5.5. FraSCaTi

Participants: Philippe Merle [correspondant], Christophe Munilla, Romain Rouvoy, Lionel Seinturier.

FRASCATI is a service-oriented component-based middleware platform implementing OASIS *Service Component Architecture* (SCA) specifications. The main originality of OW2 FRASCATI is to bring FRACTAL-based reflectivity to SCA, *i.e.*, any FRASCATI software component is equipped with both the SOA capabilities brought by SCA and the reflective capabilities (*i.e.*, introspection and reconfiguration) brought by FRACTAL. Various micro-benchmarks have shown that FRASCATI reflectivity is achieved without hindering its performance relative to the de facto reference SCA implementation, *i.e.*, Apache Tuscany. Non-functional concerns (logging, transaction, security, etc.), so called intents in SCA terms, are also programmed as FRASCATI components and are (un)woven on business components dynamically at runtime, this is based on aspect-oriented concepts defined in FAC [117]. OW2 FRASCATI supports various implementation technologies (SCA Composite, Java, WS-BPEL, Spring Framework, OSGi, Fractal ADL, native C library, Apache Velocity templates, and seven scripting languages as BeanShell, FScript, Groovy, JavaScript, JRuby, Jython, XQuery) for programming services or integrating legacy code, various binding protocols (SOAP, REST, JSON-RPC, UPnP, HTTP servlets, Java RMI, JMS, JGroups) and interface definition languages (WSDL, Java, WADL) for inter-operating with existing services. OW2 FRASCATI provides management tools like standalone, Web-based, and JMX-based graphical consoles and a dedicated scripting language for reconfiguring SCA applications. The whole OW2 FRASCATI platform is itself built as a set of reflective SCA components.

Inria Evaluation Committee Criteria for Software Self-Assessment: A-4-up, SO-4, SM-4-up, EM-3-up, SDL-4-up, DA-4, CD-4, MS-4, TPM-4. FRASCATI is a project of the OW2 consortium for open-source middleware. Web site: <http://frascati.ow2.org>. 208 Kloc (mainly Java). Registered with the APP (Agence pour la Protection des Programmes) under reference FR.001.050017.000.S.P.2010.000.10000. License: LGPL. Embedded into several industrial software systems: EasySOA, Petals Link EasyViper, EasyBPEL, EasyESB, OW2 PEtALS, OW2 Scarbo. Various demonstrators built during funded projects: ANR SCOrWare, FP7 SOA4All, ANR ITEmIS, ANR SALTY, ANR SocEDA, FUI Macchiato, FUI EasySOA, ADT Galaxy and ADT Adapt. Main publications: [19], [123], [111], [112], [98], [97].

5.6. PowerAPI

Participants: Aurélien Bourdon, Adel Nouredine, Romain Rouvoy [correspondant].

PowerAPI is a Scala-based library for monitoring energy at the process-level. It is based on a modular and asynchronous event-driven architecture using the Akka library. PowerAPI differs from existing energy process-level monitoring tool in its pure software, fully customizable and modular aspect which let users precisely define what they want to monitor, without plugging any external device. PowerAPI offers an API which can be used to express requests about energy spent by a process, following its hardware resource utilization (in terms of CPU, memory, disk, network, etc.). Its applications cover energy-driven benchmarking [52], energy hotspots and bugs detection [53], [75] and real-time distributed system monitoring.

PowerAPI is registered with the APP (Agence pour la Protection des Programmes) under reference IDDN.FR.001.400015.000.S.P.2012.000.10000. License: AGPL.

5.7. SPACES

Participants: Russel Nzekwa, Daniel Romero [correspondant], Romain Rouvoy, Lionel Seinturier.

SPACES is a context mediation middleware that follows the *REpresentational State Transfer* (REST) principles [100]. The current implementation of SPACES is based on the COSMOS context framework [93], [121] and the COMANCHE web server [91]. Both COSMOS and COMANCHE are based on the FRACTAL component model and the JULIA implementation [91].

The main features of the current SPACES implementation are presented below:

1. *Ubiquitous connectors:* SPACES defines connectors that encapsulate the distribution concern. These connectors expose the COSMOS context nodes as REST resources with logical associated URLs, and enable interactions between consumers and producers via different communication protocols and the discovery of the available context sources. The current SPACES implementation supports interaction using the HTTP and twitter [108] protocols. For discovery, the implementation uses the Service Location Protocol (SLP) [102].
2. *Context Representation:* Following the REST principles, SPACES supports multiple representations of the context information: JSON [94], XML and Java serialization.
3. *Quality of context (QoC) information:* The QoC properties are incorporated as service attributes in the SLP advertisements of the context information.
4. *Context selection:* The restrictions in terms of QoC of the required context information are expressed as LDAP filters [125]. SPACES benefits from the LDAP based queries of SLP to select the context providers.

We use XStream 1.3.13 [89] and JSON-lib 2.2.34 [88] to serialize context information as XML and JSON documents. For SLP and twitter we employ jSLP 1.0.0 [120] and twitter-4j 2.0.6 [129].

SPACES is registered with the APP (Agence pour la Protection des Programmes) under reference IDDN 10-500002-000.

ALEA Project-Team

5. Software

5.1. BiiPS software

BiiPS is a general software, developed by Adrien Todeschini, for Bayesian inference with interacting particle systems, a.k.a. sequential Monte Carlo (SMC) methods. It aims at popularizing the use of these methods to non-statistician researchers and students, thanks to its automated “black box” inference engine.

It borrows from the BUGS/JAGS software, widely used in Bayesian statistics, the statistical modeling with graphical models and the language associated with their descriptions.

Unlike MCMC methods used by BUGS/JAGS, SMC methods are more adapted to dynamic problems (tracking, signal filtering, etc).

A beta version of the software can be downloaded from the website of the [BiiPS project](#). This software has been presented at the international workshop [BayesComp](#) in Kyoto, the international conference [ISBA](#) in Tokyo, the conference on [Premières Rencontres R](#) in Bordeaux, and the [international workshop on efficient simulation in finance](#) in Paris.

ALF Project-Team

5. Software

5.1. Panorama

The ALF team is developing several software prototypes for research purposes: compilers, architectural simulators, programming environments,

Among the many prototypes developed in the project, we describe here **ATMI**, a microarchitecture temperature model for processor simulation, **STiMuL**, a temperature model for steady state studies, **ATC**, an address trace compressor, **HAVEGE**, an unpredictable random number generator and **tiptop**, a user-level Linux utility that collects data from hardware performance counters for running tasks, software developed by the team.

5.2. ATMI

Participant: Pierre Michaud.

Microarchitecture temperature model **Contact :** Pierre Michaud

Status : Registered with APP Number IDDN.FR.001.250021.000.S.P.2006.000.10600, Available under GNU General Public License

Research on temperature-aware computer architecture requires a chip temperature model. General purpose models based on classical numerical methods like finite differences or finite elements are not appropriate for such research, because they are generally too slow for modeling the time-varying thermal behavior of a processing chip.

We have developed an ad hoc temperature model, ATMI (Analytical model of Temperature in Microprocessors), for studying thermal behaviors over a time scale ranging from microseconds to several minutes. ATMI is based on an explicit solution to the heat equation and on the principle of superposition. ATMI can model any power density map that can be described as a superposition of rectangle sources, which is appropriate for modeling the microarchitectural units of a microprocessor.

Visit <http://www.irisa.fr/alf/ATMI> or contact Pierre Michaud.

5.3. STiMuL

Participant: Pierre Michaud.

Microarchitecture temperature modeling

Status: Registered with APP Number IDDN.FR.001.220013.000.S.P.2010.000.31235, Available under GNU General Public License

Some recent research has started investigating the microarchitectural implications of 3D circuits, for which the thermal constraint is stronger than for conventional 2D circuits.

STiMuL can be used to model steady-state temperature in 3D circuits consisting of several layers of different materials. STiMuL is based on a rigorous solution to the Laplace equation [6]. The number and characteristics of layers can be defined by the user. The boundary conditions can also be defined by the user. In particular, STiMuL can be used along with thermal imaging to obtain the power density inside an integrated circuit. This power density could be used for instance in a dynamic simulation oriented temperature modeling such as ATMI.

STiMuL is written in C and uses the FFTW library for discrete Fourier transforms computations.

Visit <http://www.irisa.fr/alf/stimul> or contact Pierre Michaud.

5.4. ATC

Participant: Pierre Michaud.

Address trace compression **Contact :** Pierre Michaud

Status: registered with APP number IDDN.FR.001.160031.000.S.P.2009.000.10800, available under GNU LGPL License.

Trace-driven simulation is an important tool in the computer architect's toolbox. However, one drawback of trace-driven simulation is the large amount of storage that may be necessary to store traces. Trace compression techniques are useful for decreasing the storage space requirement. But general-purpose compression techniques are generally not optimal for compressing traces because they do not take advantage of certain characteristics of traces. By specializing the compression method and taking advantages of known trace characteristics, it is possible to obtain a better tradeoff between the compression ratio, the memory consumption and the compression and decompression speed.

ATC is a utility and a C library for compressing/decompressing address traces. It implements a new lossless transformation, Bytesort, that exploits spatial locality in address traces. ATC leverages existing general-purpose compressors such as gzip and bzip2. ATC also provides a lossy compression mode that yields higher compression ratios while preserving certain important characteristics of the original trace.

Visit <http://www.irisa.fr/alf/atc> or contact Pierre Michaud.

5.5. HAVEGE

Participant: André Seznec.

Unpredictable random number generator

Contact : André Seznec

Status : Registered with APP Number IDDN.FR.001.500017.001.S.P.2001.000.10000. Available under the LGPL license.

An unpredictable random number generator is a practical approximation of a truly random number generator. Such unpredictable random number generators are needed for cryptography. HAVEGE (HARdware Volatile Entropy Gathering and Expansion) is a user-level software unpredictable random number generator for general-purpose computers that exploits the continuous modifications of the internal volatile hardware states in the processor as a source of uncertainty [12]. HAVEGE combines on-the-fly hardware volatile entropy gathering with pseudo-random number generation.

The internal state of HAVEGE includes thousands of internal volatile hardware states and is merely unmonitored. HAVEGE can reach an unprecedented throughput for a software unpredictable random number generator: several hundreds of megabits per second on current workstations and PCs.

The throughput of HAVEGE favorably competes with usual pseudo-random number generators such as `rand()` or `random()`. While HAVEGE was initially designed for cryptology-like applications, this high throughput makes HAVEGE usable for all application domains demanding high performance and high quality random number generators, e.g., Monte Carlo simulations.

Visit <http://www.irisa.fr/alf/HAVEGE> or contact André Seznec.

5.6. Tiptop

Participant: Erven Rohou.

Performance, hardware counters, analysis tool.

Status: Registered with APP (Agence de Protection des Programmes). Available under GNU General Public License v2.

Tiptop is a new simple and flexible user-level tool that collects hardware counter data on Linux platforms (version 2.6.31+). The goal is to make the collection of performance and bottleneck data as simple as possible, including simple installation and usage. In particular, we stress the following points.

- Installation is only a matter of compiling the source code. No patching of the Linux kernel is needed, and no special-purpose module needs to be loaded.
- No privilege is required, any user can run *tiptop* — non-privileged users can only watch processes they own, ability to monitor anybody's process opens the door to side-channel attacks.
- The usage is similar to *top*. There is no need for the source code of the applications of interest, making it possible to monitor proprietary applications or libraries. And since there is no probe to insert in the application, understanding of the structure and implementation of complex algorithms and code bases is not required.
- Applications do not need to be restarted, and monitoring can start at any time (obviously, only events that occur after the start of *tiptop* are observed).
- Events can be counted per thread, or per process.
- Any expression can be computed, using the basic arithmetic operators, constants, and counter values.
- A configuration file lets users define their preferred setup, as well as custom expressions.

Tiptop is written in C. It can take advantage of libncurses when available for pseudo-graphic display.

For more information, please contact Erven Rohou.

ALGORILLE Project-Team

5. Software

5.1. Introduction

Software is a central part of our output. In the following we present the main tools to which we contribute. We use the [Inria software self-assessment](#) catalog for a classification.

5.2. parXXL

Participants: Jens Gustedt, Stéphane Vialle.

ParXXL is a library for large scale computation and communication that executes fine grained algorithms on coarse grained architectures (clusters, grids, mainframes). It is one of the software bases of the InterCell project and has been proven to be a stable support, there. It is available under a GPLv2 at <http://parxxl.gforge.inria.fr/>. ParXXL is not under active development anymore, but still maintained in the case of bugs or portability problems.

Software classification: A-3, SO-4, SM-3, EM-2, SDL-4, DA-4, CD-4, MS-2, TPM-2

5.3. Distem

Participants: Tomasz Buchert, Emmanuel Jeanvoine, Lucas Nussbaum, Luc Sarzyniec.

Wrekavoc and Distem are distributed system emulators. They enable researchers to evaluate unmodified distributed applications on heterogeneous distributed platforms created from an homogeneous cluster: CPU performance and network characteristics are altered by the emulator.

Wrekavoc was developed until 2010, and we then focused our efforts on **Distem**, that shares the same goals with a different design. Distem is available from <http://distem.gforge.inria.fr/> under GPLv3.

Software classification: A-3-up, SO-4, SM-3-up, EM-3, SDL-4, DA-4, CD-4, MS-4, TPM-4.

5.4. SimGrid

Participants: Martin Quinson, Marion Guthmuller, Paul Bédaride, Lucas Nussbaum.

SimGrid is a toolkit for the simulation of distributed applications in heterogeneous distributed environments. The specific goal of the project is to facilitate research in the area of parallel and distributed large scale systems, such as Grids, P2P systems and clouds. Its use cases encompass heuristic evaluation, application prototyping or even real application development and tuning. SimGrid has an active user community of more than one hundred members, and is available under GPLv3 from <http://simgrid.gforge.inria.fr/>.

Software classification: A-4-up, SO-4, SM-4, EM-4, SDL-5, DA-4, CD-4, MS-3, TPM-4.

5.5. ORWL and P99

Participant: Jens Gustedt.

ORWL is a reference implementation of the Ordered Read-Write Lock tools as described in [4]. The macro definitions and tools for programming in C99 that have been implemented for ORWL have been separated out into a toolbox called P99. ORWL is intended to become opensource, once it will be in a publishable state. P99 is available under a QPL at <http://p99.gforge.inria.fr/>.

Software classification: A-3-up, SO-4, SM-3, EM-3, SDL (P99: 4, ORWL: 2-up), DA-4, CD-4, MS-3, TPM-4

5.6. Kadeploy

Participants: Luc Sarzyniec, Emmanuel Jeanvoine, Lucas Nussbaum.

Kadeploy is a scalable, efficient and reliable deployment (provisioning) system for clusters and grids. It provides a set of tools for cloning, configuring (post installation) and managing cluster nodes. It can deploy a 300-nodes cluster in a few minutes, without intervention from the system administrator. It plays a key role on the Grid'5000 testbed, where it allows users to reconfigure the software environment on the nodes, and is also used on a dozen of production clusters both inside and outside INRIA. It is available from <http://kadeploy3.gforge.inria.fr/> under the Cecill license.

Software classification: A-4-up, SO-3, SM-4, EM-4, SDL-4-up, DA-4, CD-4, MS-4, TPM-4.

ALICE Project-Team

4. Software

4.1. Graphite

Participants: Dobrina Boltcheva, Phuong Ho, Bruno Lévy, David Lopez, Romain Merland, Vincent Nivoliens, Jeanne Pellerin, Nicolas Ray.

Graphite is a research platform for computer graphics, 3D modeling and numerical geometry. It comprises all the main research results of our “geometry processing” group. Data structures for cellular complexes, parameterization, multi-resolution analysis and numerical optimization are the main features of the software. Graphite is publicly available since October 2003. It is hosted by Inria GForge since September 2008 (1000 downloads in two months). Graphite is one of the common software platforms used in the frame of the European Network of Excellence **AIMShape**.

4.2. MicroMegs

Participant: Samuel Hornus.

Micromegas is a 3D modeler, developed as a plugin of Graphite, dedicated to molecular biology. It is developed in cooperation with the Fourmentin Guilbert foundation and has recently been renamed "GraphiteLife-Explorer". Biologists need simple spatial modeling tools to help in understanding the role of objects' relative position in the functioning of the cell. In this context, we develop a tool for easy DNA modeling. The tool generates DNA along any user-given curve, open or closed, allows fine-tuning of atoms' position and, most importantly, exports to PDB.

In 2012, its development has been actively pursued by Samuel Hornus in the first trimester. The software is freely downloadable. A paper describing will appear in the broad journal PLOS One [9]. A poster was also presented at the European Conference on Computational Biology in september 2012.

4.3. CGAL package for Delaunay triangulations

Participant: Samuel Hornus.

This year was devolved also to finishing touches on the CGAL package for Delaunay triangulations (3rd submission to the CGAL editorial board).

Following the reviews for the second submission, Samuel Hornus has collaborated with Olivier Devillers (Inria Sophia Antipolis) to put the finishing touches to a new CGAL package for Delaunay triangulation in any dimension. It provides exact construction of Delaunay triangulations, supporting both the addition and deletion of vertices. The code takes the form of a collection of C++ template classes to ensures high performance when specializing the code to a given euclidian dimension.

4.4. OpenNL - Open Numerical Library

Participants: Thomas Jost, Bruno Lévy, Nicolas Ray, Rhaleb Zayer.

OpenNL is a standalone library for numerical optimization, especially well-suited to mesh processing. The API is inspired by the graphics API OpenGL, this makes the learning curve easy for computer graphics practitioners. The included demo program implements our LSCM [5] mesh unwrapping method. It was integrated in **Blender** by Brecht Van Lommel and others to create automatic texture mapping methods. OpenNL is extended with two specialized modules :

- **CGAL parameterization package:** this software library, developed in cooperation with Pierre Alliez and Laurent Saboret, is a **CGAL** package for mesh parameterization.
- **Concurrent Number Cruncher:** this software library extends OpenNL with parallel computing on the GPU, implemented using the CUDA API.

4.5. Intersurf

Participants: Xavier Cavin, Nicolas Ray.

Intersurf is a plugin of the VMD (Visual Molecular Dynamics) software. VMD is developed by the Theoretical and Computational Biophysics Group at the Beckmann Institute at University of Illinois. The Intersurf plugin is released with the official version of VMD since the 1.8.3 release. It provides surfaces representing the interaction between two groups of atoms, and colors can be added to represent interaction forces between these groups of atoms. We plan to include in this package the new results obtained this year in molecular surface visualization by Matthieu Chavent.

4.6. LibSL

Participants: Anass Lasram, Sylvain Lefebvre.

LibSL is a Simple library for graphics. Sylvain Lefebvre continued development of the LibSL graphics library (under CeCill-C licence, filed at the APP). LibSL is a toolbox for rapid prototyping of computer graphics algorithms, under both OpenGL, DirectX 9/10, Windows and Linux. The library is actively used in both the REVES / Inria Sophia-Antipolis and the Alice / Inria Nancy Grand-Est teams.

ALPAGE Project-Team

5. Software

5.1. Syntax

Participants: Pierre Boullier [correspondant], Benoît Sagot.

See also the web page <http://syntax.gforge.inria.fr/>.

The (currently beta) version 6.0 of the SYNTAX system (freely available on Inria GForge) includes various deterministic and non-deterministic CFG parser generators. It includes in particular an efficient implementation of the Earley algorithm, with many original optimizations, that is used in several of Alpage's NLP tools, including the pre-processing chain SXPipe and the LFG deep parser SXLFG. This implementation of the Earley algorithm has been recently extended to handle probabilistic CFG (PCFG), by taking into account probabilities both during parsing (beam) and after parsing (n -best computation). SYNTAX 6.0 also includes parsers for various contextual formalisms, including a parser for Range Concatenation Grammars (RCG) that can be used among others for TAG and MC-TAG parsing.

Direct NLP users of SYNTAX for NLP, outside Alpage, include Alexis Nasr (Marseilles) and other members of the (now closed) SEQUOIA ANR project, Owen Rambow and co-workers at Columbia University (New York), as well as (indirectly) all SXPipe and/or SXLFG users. The project-team VASY (Inria Rhône-Alpes) is one of SYNTAX' user for non-NLP applications.

5.2. System DyALog

Participant: Éric Villemonte de La Clergerie [maintainer].

DYALOG on Inria GForge: <http://dyalog.gforge.inria.fr/>

DYALOG provides an environment to compile and execute grammars and logic programs. It is essentially based on the notion of tabulation, i.e. of sharing computations by tabulating traces of them. DYALOG is mainly used to build parsers for Natural Language Processing (NLP). It may nevertheless be used as a replacement for traditional PROLOG systems in the context of highly ambiguous applications where sub-computations can be shared.

The current release **1.13.0** of DYALOG is freely available by FTP under an open source license and runs on Linux platforms for x86 and architectures and on Mac OS intel (both 32 and 64bits architectures).

The current release handles logic programs, DCGs (*Definite Clause Grammars*), FTAGs (*Feature Tree Adjoining Grammars*), FTIGs (*Feature Tree Insertion Grammars*) and XRCGs (*Range Concatenation Grammars* with logic arguments). Several extensions have been added to most of these formalisms such as intersection, Kleene star, and interleave operators. Typed Feature Structures (TFS) as well as finite domains may be used for writing more compact and declarative grammars [120].

C libraries can be used from within DYALOG to import APIs (`mysql`, `libxml`, `sqlite`, ...).

DYALOG is largely used within ALPAGE to build parsers but also derivative softwares, such as a compiler of Meta-Grammars (cf. 5.3). It has also been used for building a parser from a large coverage French TIG/TAG grammar derived from a Meta-Grammar. This parser has been used for the Parsing Evaluation campaign EASy, the two Passage campaigns (Dec. 2007 and Nov. 2009), cf. [117], [118], and very large amount of data (700 millions of words) in the SCRIBO project.

DYALOG and other companion modules are available on Inria GForge.

5.3. Tools and resources for Meta-Grammars

Participant: Éric Villemonte de La Clergerie [maintainer].

mgcomp, *MGTOOLS*, and *FRMG* on Inria GForge: <http://mgkit.gforge.inria.fr/>

DYALOG (cf. 5.2) has been used to implement *mgcomp*, Meta-Grammar compiler. Starting from an XML representation of a MG, *mgcomp* produces an XML representation of its TAG expansion.

The current version **1.5.0** is freely available by FTP under an open source license. It is used within ALPAGE and (occasionally) at LORIA (Nancy) and at University of Pennsylvania.

The current version adds the notion of namespace, to get more compact and less error-prone meta-grammars. It also provides other extensions of the standard notion of Meta-Grammar in order to generate very compact TAG grammars. These extensions include the notion of *Guarded nodes*, i.e. nodes whose existence and non-existence depend on the truth value of a guard, and the use of the regular operators provided by DYALOG on nodes, namely disjunction, interleaving and Kleene star. The current release provides a dump/restore mechanism for faster compilations on incremental changes of a meta-grammars.

The current version of *mgcomp* has been used to compile a wide coverage Meta-Grammar FRMG (version 2.0.1) to get a grammar of around 200 TAG trees [122]. Without the use of guarded nodes and regular operators, this grammar would have more than several thousand trees and would be almost intractable. FRMG has been packaged and is freely available.

To ease the design of meta-grammars, a set of tools have been implemented, mostly by Éric de La Clergerie, and collected in *MGTOOLS* (version **2.2.2**). This package includes a converter from a compact format to a XML pivot format, an Emacs mode for the compact and XML formats, a graphical viewer interacting with Emacs and XSLT stylesheets to derive HTML views. A new version is under development to provide an even more compact syntax and some checking mechanisms to avoid frequent typo errors.

The various tools on Metagrammars are available on Inria GForge. FRMG is used directly or indirectly (through a Web service or by requiring parsed corpora) by several people and actions (ANR Rhapsodie, ANR Chronoline, ...)

5.4. The Bonsai PCFG-LA parser

Participants: Marie Candito [correspondant], Djamé Seddah.

Web page:

http://alpage.inria.fr/statgram/frdep/fr_stat_dep_parsing.html

Alpage has developed as support of the research papers [74], [65], [66], [11] a statistical parser for French, named Bonsai, trained on the French Treebank. This parser provides both a phrase structure and a projective dependency structure specified in [4] as output. This parser operates sequentially: (1) it first outputs a phrase structure analysis of sentences reusing the Berkeley implementation of a PCFG-LA trained on French by Alpage (2) it applies on the resulting phrase structure trees a process of conversion to dependency parses using a combination of heuristics and classifiers trained on the French treebank. The parser currently outputs several well known formats such as Penn treebank phrase structure trees, Xerox like triples and CONLL-like format for dependencies. The parsers also comes with basic preprocessing facilities allowing to perform elementary sentence segmentation and word tokenisation, allowing in theory to process unrestricted text. However it is believed to perform better on newspaper-like text. The parser is available under a GPL license.

5.5. The MICA parser

Participants: Benoît Sagot [correspondant], Marie Candito, Pierre Boullier, Djamé Seddah.

Web page:

<http://mica.lif.univ-mrs.fr/>

MICA (Marseille-Inria-Columbia- AT&T) is a freely available dependency parser [57] currently trained on English and Arabic data, developed in collaboration with Owen Rambow and Daniel Bauer (Columbia University) and Srinivas Bangalore (AT&T). MICA has several key characteristics that make it appealing to researchers in NLP who need an off-the-shelf parser, based on Probabilistic Tree Insertion Grammars and on the SYNTAX system. MICA is fast (450 words per second plus 6 seconds initialization on a standard high-end machine) and has close to state-of-the-art performance (87.6% unlabeled dependency accuracy on the Penn Treebank).

MICA consists of two processes: the supertagger, which associates tags representing rich syntactic information with the input word sequence, and the actual parser, based on the Inria SYNTAX system, which derives the syntactic structure from the n -best chosen supertags. Only the supertagger uses lexical information, the parser only sees the supertag hypotheses.

MICA returns n -best parses for arbitrary n ; parse trees are associated with probabilities. A packed forest can also be returned.

5.6. Alpage's linguistic workbench, including SxPipe

Participants: Benoît Sagot [correspondant], Rosa Stern, Marion Baranes, Damien Nouvel, Virginie Mouilleron, Pierre Boullier, Éric Villemonte de La Clergerie.

See also the web page <http://lingwb.gforge.inria.fr/>.

Alpage's linguistic workbench is a set of packages for corpus processing and parsing. Among these packages, the SxPipe package is of a particular importance.

SxPipe [97] is a modular and customizable chain aimed to apply to raw corpora a cascade of surface processing steps. It is used

- as a preliminary step before Alpage's parsers (e.g., FRMG);
- for surface processing (named entities recognition, text normalization...).

Developed for French and for other languages, SxPipe includes, among others, various named entities recognition modules in raw text, a sentence segmenter and tokenizer, a spelling corrector and compound words recognizer, and an original context-free patterns recognizer, used by several specialized grammars (numbers, impersonal constructions, quotations...). In 2012, SxPipe has received a renewed attention in four directions:

- Support of new languages, and most notably German (although this is still at a very preliminary stage of development);
- Analysis of unknown words, in particular in the context of the ANR project EDyLex and of the collaboration with *viavoo*; this involves in particular (i) new tools for the automatic pre-classification of unknown words (acronyms, loan words...) (ii) new morphological analysis tools, most notably automatic tools for constructional morphology (both derivational and compositional), following the results of dedicated corpus-based studies;
- Development of new local grammars for detecting new types of entities, such as chemical formulae or dimensions, in the context of the PACTE project.

5.7. MElt

Participants: Benoît Sagot [correspondant], Pascal Denis.

MElt is a part-of-speech tagger, initially trained for French (on the French TreeBank and coupled with the *Lefff*), English [78], Spanish, Kurmanji Kurdish [125] and Persian [106], [107]. It is state-of-the-art for French. It is distributed freely as a part of the Alpage linguistic workbench.

In 2012, MElt has underwent two major upgrades:

- It has been successfully trained and used on Italian [35], Spanish [26] and German data. In particular, a statistical parsing architecture for Italian that used MElt in a pre-processing step has obtained the best results in the EVALITA shared task on Italian parsing [35].
- MElt can now be called within a wrapper developed for handling noisy textual data such as user-generated content produced on Web 2.0 platforms (forums, blogs, social media); more precisely, this wrapper is able to "clean" such data, then tag it using MElt, and finally transfer MElt annotations from the "cleaned" data, which could be annotated more easily, to the original noisy data. This architecture has proved useful on French for creating the French Social Media Bank [37], [36]. On English, it has played an important role within both variants of the Alpage parsing architecture that were ranked 2nd and 3rd at the SANCL shared task on parsing user-generated content, organized by Google [38].

5.8. The Alexina framework: the Lefff syntactic lexicon, the Aleda entity database and other Alexina resources

Participants: Benoît Sagot [correspondant], Laurence Danlos.

See also the web page <http://gforge.inria.fr/projects/alexina/>.

Alexina is Alpage's Alexina framework for the acquisition and modeling of morphological and syntactic lexical information. The first and most advanced lexical resource developed in this framework is the *Lefff*, a morphological and syntactic lexicon for French.

Historically, the *Lefff* 1 was a freely available French morphological lexicon for verbs that has been automatically extracted from a very large corpus. Since version 2, the *Lefff* covers all grammatical categories (not just verbs) and includes syntactic information (such as subcategorization frames); Alpage's tools, including Alpage's parsers, rely on the *Lefff*. The version 3 of the *Lefff*, which has been released in 2008, improves the linguistic relevance and the interoperability with other lexical models.

Other Alexina lexicons exist, at various stages of development, in particular for Spanish (the *Leffe*), Polish, Slovak, English, Galician, Persian, Kurdish, Italian and since this year for German, as well as for Latin and Maltese verbs. These lexicons are used in various tools, including instances of the MElt POS-tagger.

Alexina also hosts *Aleda* [114], [33] a large-scale entity database currently developed for French but under development for English, Spanish and German, extracted automatically from Wikipedia and Geonames. It is used among others in the SxPipe processing chain and its NP named entity recognition, as well as in the NOMOS named entity linking system.

5.9. The free French wordnet WOLF

Participants: Benoît Sagot [correspondant], Marion Richard, Sarah Beniamine.

The WOLF (Wordnet Libre du Français) is a wordnet for French, i.e., a lexical semantic database. The development of WOLF started in 2008 [99], [100]. At this time, we focused on benefiting from available resources of three different types: general and domain-specific bilingual dictionaries, multilingual parallel corpora and Wiki resources (Wikipedia and Wiktionaries). This work was achieved in a large part in collaboration with Darja Fišer (University of Ljubljana, Slovenia), in parallel with the development of a free Slovene wordnet, sloWNet. However, it was also impacted by specific collaborations, e.g., on adverbial synsets [101].

2012 results concerning the WOLF are described in the corresponding section.

The WOLF is freely available under the Cecill-C license. It has already been used in various experiments, within and outside Alpage.

5.10. Automatic construction of distributional thesauri

Participant: Enrique Henestroza Anguiano [correspondant].

FREDISTis a freely-available (LGPL license) Python package that implements methods for the automatic construction of distributional thesauri.

We have implemented the context relation approach to distributional similarity, with various context relation types and different options for weight and measure functions to calculate distributional similarity between words. Additionally, FREDISTis highly flexible, with parameters including: context relation type(s), weight function, measure function, term frequency thresholds, part-of-speech restrictions, filtering of numerical terms, etc.

Distributional thesauri for French are also available, one each for adjectives, adverbs, common nouns, and verbs. They have been constructed with FreDist and use the best settings obtained in an evaluation. We use the *L'Est Republicain* corpus (125 million words), *Agence France-Presse* newswire dispatches (125 million words) and a full dump of the French Wikipedia (200 million words), for a total of 450 million words of text.

5.11. Tools and resources for time processing

Participant: Laurence Danlos [correspondant].

Alpage developed the *French TimeBank*, a freely-available corpus annotated with ISO-TimeML-compliant temporal information (dates, events and relations between events).

5.12. System EasyRef

Participants: Éric Villemonte de La Clergerie [maintainer], Corentin Ribeyre.

A collaborative WEB service EASYREF has been developed, in the context of ANR action Passage, to handle syntactically annotated corpora. EASYREF may be used to view annotated corpus, in both EASY or PASSAGE formats. The annotations may be created and modified. Bug reports may be emitted. The annotations may be imported and exported. The system provides standard user right management. The interface has been designed with the objectives to be intuitive and to speed edition.

EASYREF relies on an Model View Controller design, implemented with the Perl Catalyst framework. It exploits WEB 2.0 technologies (i.e. AJAX and JavaScript).

Version 2 has been used by ELDA and LIMSI to annotate a new corpus of several thousands words for PASSAGE.

A preliminary version 3 has been developed by François Guérin and revised by Éric de La Clergerie, relying on Berkeley DB XML to handle very large annotated corpora and to provide a complete query language expanded as XQuery expressions. EASYREF is maintained under Inria GForge.

AMIB Project-Team

4. Software

4.1. Varna

Participants: Yann Ponty [correspondant], Alain Denise.

VARNA is a tool for the automated drawing, visualization and annotation of the secondary structure of RNA, designed as a companion software for web servers and databases. VARNA implements four drawing algorithms, supports input/output using the classic formats *dbn*, *ct*, *bpseq* and *RNAML* and exports the drawing, either as a bitmap (*JPEG*, *PNG*) or as a vector picture (*SVG*, *EPS* and *XFIG*). It also allows manual modification and structural annotation of the resulting drawings using either an interactive *point and click* approach, within a web server or through command-line arguments. VARNA is a free software distributed under the terms of the GPLv3.0 license and available at <http://varna.lri.fr>.

VARNA is currently used by RNA scientists (Cited by 92 research articles since its presentation in Fall of 2009, according to Google scholar), web servers such as the BOULDEALE (<http://www.microbio.me/boulderale/>), TFOLD (<http://tfold.ibisc.univ-evry.fr/TFold/>), CYLOFOLD (<http://cylofold.abcc.ncifcrf.gov/>) webservers, and by databases such as IRESITE (<http://iresite.org/>), SRNATARBASE (<http://ccb.bmi.ac.cn/srnatarbase/>) and RFAM (<http://rfam.sanger.ac.uk/>), the main source of sequence/structure data for RNA scientist, to display secondary structures. It is also used as an integrated component within JALVIEW, arguably one of the leading sequence alignment editor (<http://www.jalview.org/>), and Y. Ponty co-supervised with Jim Procter (University of Dundee, Jalview Project Leader) two internships (including a Google Summer of Code) in the summer of 2012 to further the interactions between the two software.

4.2. SPFlow

Participants: Jiuqiang Chen, Sarah Cohen-Boulakia [correspondant], Christine Froidevaux [correspondant].

SPFLOW is a scientific workflow rewriting tool. SPFlow aims at transforming complex workflow structures (non series-parallel structures) into provenance-equivalent simple workflow structures (series-parallel structures). SPFlow takes as an input a file representing one scientific workflow from Taverna and produces another file in which the structure of the original workflow is made series-parallel while ensuring that both workflows have the same provenance (more information available at [32], [39]). The tool is freely available at <http://www.lri.fr/~chenj/SPFlow>.

4.3. GeneValorization

Participants: Bryan Brancotte, Sarah Cohen-Boulakia [correspondant].

High-throughput technologies provide fundamental information concerning thousands of genes. Many of the current research laboratories daily use one or more of these technologies and end-up with lists of genes. Assessing the originality of the results obtained includes being aware of the number of publications available concerning individual or multiple genes and accessing information about these publications. Faced with the exponential growth of publications available and number of genes involved in a study, this task is becoming particularly difficult to achieve. We introduce GENEVALORIZATION, a web-based tool which gives a clear and handful overview of the bibliography available corresponding to the user input formed by (i) a gene list (expressed by gene names or ids from ENTREZGENE) and (ii) a context of study (expressed by keywords). From this input, GENEVALORIZATION provides a matrix containing the number of publications with co-occurrences of gene names and keywords. Graphics are automatically generated to assess the relative importance of genes within various contexts. Links to publications and other databases offering information on genes and keywords are also available. To illustrate how helpful GENEVALORIZATION is, we have considered the gene list of the OncotypeDX prognostic marker test. it is available at <http://bioguide-project.net/gv>.

4.4. HSIM

Participant: Patrick Amar [correspondant].

HSIM is a simulation tool for studying the dynamics of biochemical processes in a virtual bacteria. The model is given using a language based on probabilistic rewriting rules that mimics the reactions between biochemical species. HSIM is a stochastic automaton which implements an entity-centered model of objects. This kind of modelling approach is an attractive alternative to differential equations for studying the diffusion and interaction of the many different enzymes and metabolites in cells which may be present in either small or large numbers.

The new version of HSIM includes a Stochastic Simulation Algorithm *a la* Gillespie that can be used with the same model in a standalone way or in a mixed way with the entity-centered algorithm. This new version offers also the possibility to export the model in SciLab for a ODE integration. Last, HSIM can export the differential equations system, equivalent to the model, to LaTeX for pretty-printing.

This software is freely available at <http://www.lri.fr/~pa/Hsim>; A compiled version is available for the Windows, Linux and MacOSX operating systems.

4.5. Cartaj

Participant: Alain Denise [correspondant].

CARTAJ is a software that automatically predicts the topological family of three-way junctions in RNA molecules, from their secondary structure only. The Cartaj software <http://cartaj.lri.fr> that implements our method can be used online. It is also meant for being part of RNA modelling softwares and platforms. The methodology and the results of CARTAJ are presented in [14].

AOSTE Project-Team

5. Software

5.1. TimeSquare

Participants: Charles André, Nicolas Chleq, Julien Deantoni, Frédéric Mallet [correspondant].

TimeSquare is a software environment for the modeling and analysis of timing constraints in embedded systems. It relies specifically on the Time Model of the MARTE UML profile (see section 3.2), and more accurately on the associated Clock Constraint Specification Language (CCSL) for the expression of timing constraints.

TimeSquare offers four main functionalities:

1. graphical and/or textual interactive specification of logical clocks and relative constraints between them;
2. definition and handling of user-defined clock constraint libraries;
3. automated simulation of concurrent behavior traces respecting such constraints, using a Boolean solver for consistent trace extraction;
4. call-back mechanisms for the traceability of results (animation of models, display and interaction with waveform representations, generation of sequence diagrams...).

In practice TimeSquare is a plug-in developed with Eclipse modeling tools. The software is registered by the *Agence pour la Protection des Programmes*, under number IDDN.FR.001.170007.000.S.P.2009.001.10600. It can be downloaded from the site <http://timesquare.inria.fr/>. It has been integrated in the **OpenEmbeDD** ANR RNTL platform, and other such actions are under way.

5.2. K-Passa

Participants: Jean-Vivien Millo [correspondant], Robert de Simone.

This software is dedicated to the simulation, analysis, and static scheduling scheduling of Event/Marked Graphs, SDF and KRG extensions. A graphical interface allows to edit the Process Networks and their time annotations (*latency*, ...). Symbolic simulation and graph-theoretic analysis methods allow to compute and optimize static schedules, with best throughputs and minimal buffer sizes. In the case of KRG the (ultimately k-periodic) routing patterns can also be provided and transformed for optimal combination of switching and scheduling when channels are shared. KPASSA also allows for import/export of specific description formats such as UML-MARTE, to and from our other TimeSquare tool.

The tool was originally developed mainly as support for experimentations following our research results on the topic of Latency-Insensitive Design. This research was conducted and funded in part in the context of the CIM PACA initiative, with initial support from ST Microelectronics and Texas Instruments.

KPASSA is registered by the *Agence pour la Protection des Programmes*, under the number IDDN.FR.001.310003.000.S.P.2009.000.20700. it can be downloaded from the site <http://www-sop.inria.fr/aoste/index.php?page=software/kpassa>.

5.3. SynDEx

Participants: Maxence Guesdon, Yves Sorel [correspondant], Cécile Stentzel, Meriem Zidouni.

SynDEx is a system level CAD software implementing the AAA methodology for rapid prototyping and for optimizing distributed real-time embedded applications. Developed in OCaml it can be downloaded free of charge, under Inria copyright, from the general SynDEx site <http://www.syndex.org>.

The AAA methodology is described in section 3.3 . Accordingly, SYNDEX explores the space of possible allocations (spatial distribution and temporal scheduling), from application elements to architecture resources and services, in order to match real-time requirements; it does so by using schedulability analyses and heuristic techniques. Ultimately it generates automatically distributed real-time code running on real embedded platforms. The last major release of SYNDEX (V7) allows the specification of multi-periodic applications.

Application algorithms can be edited graphically as directed acyclic task graphs (DAG) where each edge represents a data dependence between tasks, or they may be obtained by translations from several formalisms such as Scicos (<http://www.scicos.org>), Signal/Polychrony (<http://www.irisa.fr/espresso/Polychrony>), or UML2/MARTE models (http://www.omg.org/technology/documents/profile_catalog.htm).

Architectures are represented as graphical block diagrams composed of programmable (processors) and non-programmable (ASIC, FPGA) computing components, interconnected by communication media (shared memories, links and busses for message passing). In order to deal with heterogeneous architectures it may feature several components of the same kind but with different characteristics.

Two types of non-functional properties can be specified for each task of the algorithm graph. First, a period that does not depend on the hardware architecture. Second, real-time features that depend on the different types of hardware components, ranging amongst *execution and data transfer time, memory, etc.* Requirements are generally constraints on deadline equal to period, latency between any pair of tasks in the algorithm graph, dependence between tasks, etc.

Exploration of alternative allocations of the algorithm onto the architecture may be performed manually and/or automatically. The latter is achieved by performing real-time multiprocessor schedulability analyses and optimization heuristics based on the minimization of temporal or resource criteria. For example while satisfying deadline and latency constraints they can minimize the total execution time (makespan) of the application onto the given architecture, as well as the amount of memory. The results of each exploration is visualized as timing diagrams simulating the distributed real-time implementation.

Finally, real-time distributed embedded code can be automatically generated for dedicated distributed real-time executives, possibly calling services of resident real-time operating systems such as Linux/RTAI or Osek for instance. These executives are deadlock-free, based on off-line scheduling policies. Dedicated executives induce minimal overhead, and are built from processor-dependent executive kernels. To this date, executive kernels are provided for: TMS320C40, PIC18F2680, i80386, MC68332, MPC555, i80C196 and Unix/Linux workstations. Executive kernels for other processors can be achieved at reasonable cost following these examples as patterns.

5.4. SAS

Participants: Daniel de Rauglaudre [correspondant], Yves Sorel.

The SAS (Simulation and Analysis of Scheduling) software allows the user to perform the schedulability analysis of periodic task systems in the monoprocessor case.

The main contribution of SAS, when compared to other commercial and academic softwares of the same kind, is that it takes into account the exact preemption cost between tasks during the schedulability analysis. Beside usual real-time constraints (precedence, strict periodicity, latency, etc.) and fixed-priority scheduling policies (Rate Monotonic, Deadline Monotonic, Audsley⁺⁺, User priorities), SAS additionally allows to select dynamic scheduling policy algorithms such as Earliest Deadline First (EDF). The resulting schedule is displayed as a typical Gantt chart with a transient and a permanent phase, or as a disk shape called "dameid", which clearly highlights the idle slots of the processor in the permanent phase.

For a schedulable task system under EDF, when the exact preemption cost is considered, the period of the permanent phase may be much longer than the least common multiple (LCM) of the periods of all tasks, as often found in traditional scheduling theory. Specific effort has been made to improve display in this case. The classical utilization factor, the permanent exact utilization factor, the preemption cost in the permanent phase, and the worst response time for each task are all displayed when the system is schedulable. Response times of each task relative time can also be displayed (separately).

SAS is written in OCaml, using CAMLP5 (syntactic preprocessor) and OLIBRT (a graphic toolkit under X). Both are written by Daniel de Rauglaudre. It can be downloaded from the site <http://pauillac.inria.fr/~ddr/sas-dameid/>.

APICS Project-Team

5. Software

5.1. RARL2

Participant: Martine Olivi [corresponding participant].

Status: Currently under development. A stable version is maintained.

This software is developed in collaboration with Jean-Paul Marmorat (Centre de mathématiques appliquées (CMA), École des Mines de Paris).

RARL2 (Réalisation interne et Approximation Rationnelle L2) is a software for rational approximation (see section 3.3.2.2) <http://www-sop.inria.fr/apics/RARL2/rarl2-eng.html>.

The software RARL2 computes, from a given matrix-valued function in $\overline{H}^{2m \times l}$, a local best rational approximant in the L^2 norm, which is *stable and of prescribed McMillan degree* (see section 3.3.2.2). It was initially developed in the context of linear (discrete-time) system theory and makes an heavy use of the classical concepts in this field. The matrix-valued function to be approximated can be viewed as the transfer function of a multivariable discrete-time stable system. RARL2 takes as input either:

- its internal realization,
- its first N Fourier coefficients,
- discretized (uniformly distributed) values on the circle. In this case, a least-square criterion is used instead of the L^2 norm.

It thus performs model reduction in case 1) and 2) and frequency data identification in case 3). In the case of band-limited frequency data, it could be necessary to infer the behavior of the system outside the bandwidth before performing rational approximation (see 3.2.2). An appropriate Moebius transformation allows to use the software for continuous-time systems as well.

The method is a steepest-descent algorithm. A parametrization of MIMO systems is used, which ensures that the stability constraint on the approximant is met. The implementation, in matlab, is based on state-space representations.

The number of local minima can be rather high so that the choice of an initial point for the optimization can play a crucial role. Two methods can be used: 1) An initialization with a best Hankel approximant. 2) An iterative research strategy on the degree of the local minima, similar in principle to that of Rarl2, increases the chance of obtaining the absolute minimum by generating, in a structured manner, several initial conditions.

RARL2 performs the rational approximation step in our applications to filter identification (see section 4.3) as well as sources or cracks recovery (see section 4.2). It was released to the universities of Delft, Maastricht, Cork and Brussels. The parametrization embodied in RARL2 was also used for a multi-objective control synthesis problem provided by ESTEC-ESA, The Netherlands. An extension of the software to the case of triple poles approximants is now available. It provides satisfactory results in the source recovery problem and it is used by FindSources3D (see section 5.6).

5.2. RGC

Participant: Fabien Seyfert [corresponding participant].

Status: A stable version is maintained.

This software is developed in collaboration with Jean-Paul Marmorat (Centre de mathématiques appliquées (CMA), École des Mines de Paris).

The identification of filters modelled by an electrical circuit that was developed by the team (see section 4.3) led us to compute the electrical parameters of the underlying filter. This means finding a particular realization (A, B, C, D) of the model given by the rational approximation step. This 4-tuple must satisfy constraints that come from the geometry of the equivalent electrical network and translate into some of the coefficients in (A, B, C, D) being zero. Among the different geometries of coupling, there is one called “the arrow form” [53] which is of particular interest since it is unique for a given transfer function and is easily computed. The computation of this realization is the first step of RGC. Subsequently, if the target realization is not in arrow form, one can nevertheless show that it can be deduced from the arrow-form by a complex-orthogonal change of basis. In this case, RGC starts a local optimization procedure that reduces the distance between the arrow form and the target, using successive orthogonal transformations. This optimization problem on the group of orthogonal matrices is non-convex and has many local and global minima. In fact, there is not even uniqueness of the filter realization for a given geometry. Moreover, it is often relevant to know all solutions of the problem, because the designer is not even sure, in many cases, which one is being handled. The assumptions on the reciprocal influence of the resonant modes may not be equally well satisfied for all such solutions, hence some of them should be preferred for the design. Today, apart from the particular case where the arrow form is the desired form (this happens frequently up to degree 6) the RGC software provides no guarantee to obtain a single realization that satisfies the prescribed constraints. The software Dedale-HF (see section 5.4), which is the successor of RGC, solves with guarantees this constraint realization problem.

5.3. PRESTO-HF

Participant: Fabien Seyfert [corresponding participant].

Status: Currently under development. A stable version is maintained.

PRESTO-HF: a toolbox dedicated to lowpass parameter identification for microwave filters http://www-sop.inria.fr/apics/personnel/Fabien.Seyfert/Presto_web_page/presto_pres.html. In order to allow the industrial transfer of our methods, a Matlab-based toolbox has been developed, dedicated to the problem of identification of low-pass microwave filter parameters. It allows one to run the following algorithmic steps, either individually or in a single shot:

- determination of delay components caused by the access devices (automatic reference plane adjustment),
- automatic determination of an analytic completion, bounded in modulus for each channel,
- rational approximation of fixed McMillan degree,
- determination of a constrained realization.

For the matrix-valued rational approximation step, Presto-HF relies on RARL2 (see section 5.1), a rational approximation engine developed within the team. Constrained realizations are computed by the RGC software. As a toolbox, Presto-HF has a modular structure, which allows one for example to include some building blocks in an already existing software.

The delay compensation algorithm is based on the following strong assumption: far off the passband, one can reasonably expect a good approximation of the rational components of S_{11} and S_{22} by the first few terms of their Taylor expansion at infinity, a small degree polynomial in $1/s$. Using this idea, a sequence of quadratic convex optimization problems are solved, in order to obtain appropriate compensations. In order to check the previous assumption, one has to measure the filter on a larger band, typically three times the pass band.

This toolbox is currently used by Thales Alenia Space in Toulouse, Thales airborne systems and a license agreement has been recently negotiated with TAS-España. XLim (University of Limoges) is a heavy user of Presto-HF among the academic filtering community and some free license agreements are currently being considered with the microwave department of the University of Erlangen (Germany) and the Royal Military College (Kingston, Canada).

5.4. Dedale-HF

Participant: Fabien Seyfert [corresponding participant].

Status: Currently under development. A stable version is maintained.

Dedale-HF is a software dedicated to solve exhaustively the coupling matrix synthesis problem in reasonable time for the users of the filtering community. For a given coupling topology, the coupling matrix synthesis problem (C.M. problem for short) consists in finding all possible electromagnetic coupling values between resonators that yield a realization of given filter characteristics (see section 6.3). Solving the latter problem is crucial during the design step of a filter in order to derive its physical dimensions as well as during the tuning process where coupling values need to be extracted from frequency measurements (see Figure 3).

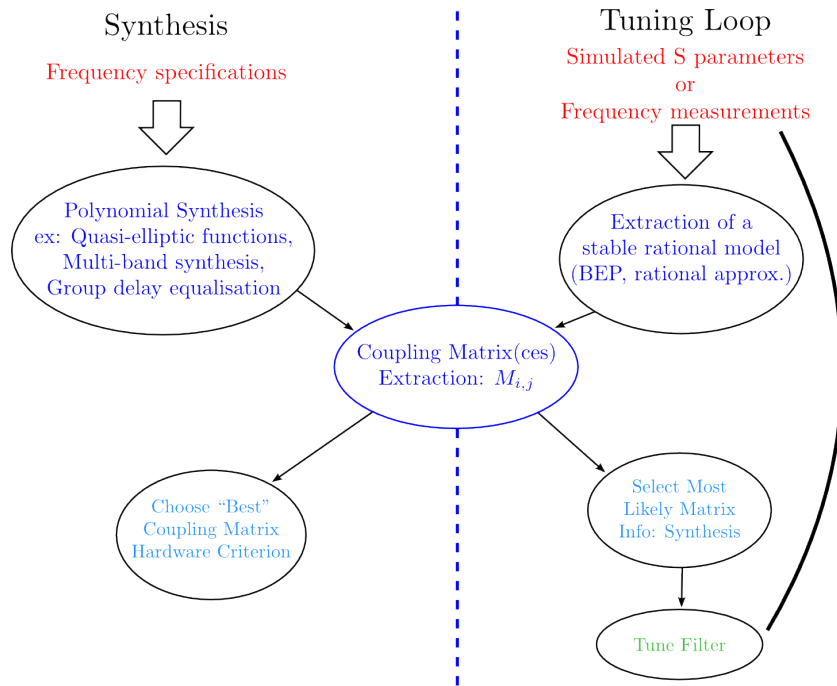


Figure 3. Overall scheme of the design and tuning process of a microwave filter.

Dedale-HF consists in two parts: a database of coupling topologies as well as a dedicated predictor-corrector code. Roughly speaking each reference file of the database contains, for a given coupling topology, the complete solution to the C.M. problem associated to particular filtering characteristics. The latter is then used as a starting point for a predictor-corrector integration method that computes the solution to the C.M. problem of the user, *i.e.* the one corresponding to user-specified filter characteristics. The reference files are computed off-line using Groebner basis techniques or numerical techniques based on the exploration of a monodromy group. The use of such a continuation technique combined with an efficient implementation of the integrator produces a drastic reduction, by a factor of 20, of the computational time.

Access to the database and integrator code is done via the web on <http://www-sop.inria.fr/apics/Dedale/WebPages>. The software is free of charge for academic research purposes: a registration is however needed in order to access full functionality. Up to now 90 users have registered world wide (mainly: Europe, U.S.A, Canada and China) and 4000 reference files have been downloaded.

A license of this software has been sold end 2011, to TAS-Espagna to tune filter, with topologies with multiple solutions. The usage of Dedale-HF is here considered together with Presto-HF.

5.5. easyFF

Participant: Fabien Seyfert.

Status: A stable version is maintained.

This software has been developed by Vincent Lunot (Taiwan Univ.) during his Ph.d. He still continues to maintain it.

EasyFF is a software dedicated to the computation of complex, and in particular multi-band, filtering functions. The software takes as input, specifications on the modulus of the scattering matrix (transmission and rejection), the filter's order and the number of transmission zeros. The output is an "optimal" filtering characteristic in the sense that it is the solution of an associated min-max Zolotarev problem. Computations are based on a Remez-type algorithm (if transmission zeros are fixed) or on linear programming techniques if transmission zeros are part of the optimization [11].

5.6. FindSources3D

Participant: Juliette Leblond [corresponding participant].

Status: Currently under development. A stable version is maintained.

This software is developed in collaboration with Maureen Clerc and Théo Papadopoulo from the Athena EPI, and with Jean-Paul Marmorat (Centre de mathématiques appliquées (CMA), École des Mines de Paris).

FindSources3D is a software dedicated to source recovery for the inverse EEG problem, in 3-layer spherical settings, from pointwise data (see <http://www-sop.inria.fr/apics/FindSources3D/>). Through the algorithm described in [16] and section 4.2, it makes use of the software RARL2 (section 5.1) for the rational approximation step in plane sections. The data transmission preliminary step ("cortical mapping") is solved using boundary element methods through the software OpenMEEG (its CorticalMapping features) developed by the Athena Team (see <http://www-sop.inria.fr/athena/software/OpenMEEG/>). A first release of FindSources3D is now available, which will be demonstrated and distributed within the medical teams we are in contact with (see Figure 4, CeCILL license, APP version 1.0: IDDN.FR.001.45009.S.A.2009.000.10000).

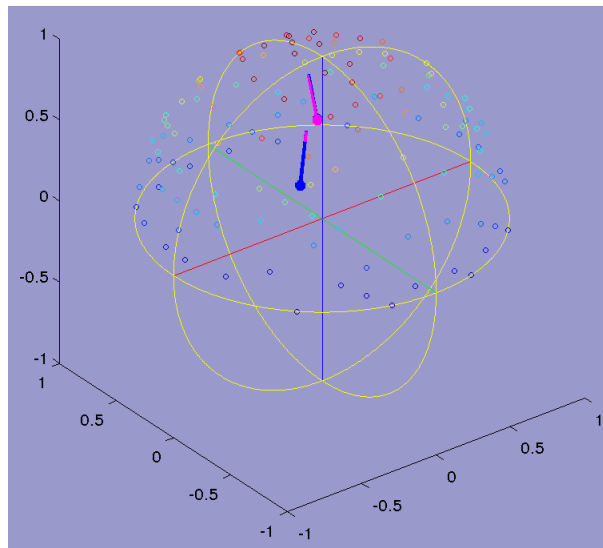


Figure 4. Potential values at electrodes on a sphere (scalp), recovered 2 sources (FindSources3D).

5.7. Sollya

Participant: Sylvain Chevillard [corresponding participant].

Status: Currently under development. A stable version is maintained.

This software is developed in collaboration with Christoph Lauter (LIP6) and Mioara Joldeş (Uppsala University, Sweden).

Sollya is an interactive tool where the developers of mathematical floating-point libraries (libm) can experiment before actually developing code. The environment is safe with respect to floating-point errors, *i.e.* the user precisely knows when rounding errors or approximation errors happen, and rigorous bounds are always provided for these errors.

Amongst other features, it offers a fast Remez algorithm for computing polynomial approximations of real functions and also an algorithm for finding good polynomial approximants with floating-point coefficients to any real function. It also provides algorithms for the certification of numerical codes, such as Taylor Models, interval arithmetic or certified supremum norms.

It is available as a free software under the CeCILL-C license at <http://sollya.gforge.inria.fr/>.

ARIC Team

5. Software

5.1. Overview

AriC software and hardware realizations are accessible from the web page <http://www.ens-lyon.fr/LIP/AriC/ware.html>. We describe below only those which progressed in 2012.

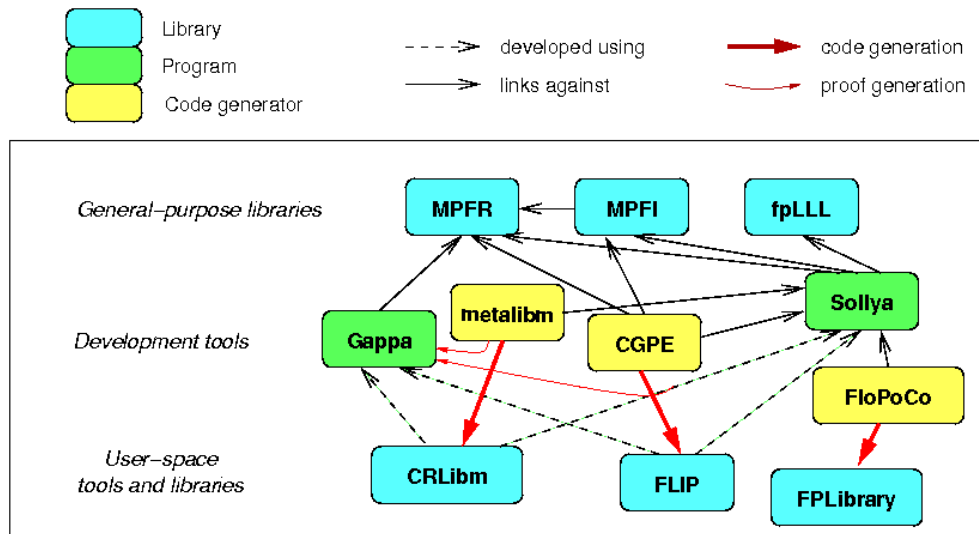


Figure 1. Relationships between some AriC developments.

5.2. FloPoCo

Participants: Florent de Dinechin [correspondant], Matei Istoan.

The purpose of the FloPoCo project is to explore the many ways in which the flexibility of the FPGA target can be exploited in the arithmetic realm. FloPoCo is a generator of operators written in C++ and outputting synthesizable VHDL automatically pipelined to an arbitrary frequency.

In 2012, the diverging multiplier implementations in FloPoCo were unified using a common *bit-heap* framework. In addition, several new operators were added.

FloPoCo also now offers state-of-the-art random generators written by David Thomas at Imperial College.

Versions 2.3.1 and 2.4.0 were released in 2012.

Among the known users of FloPoCo are U. Cape Town, U.T. Cluj-Napoca, Imperial College, U. Essex, U. Madrid, U. P. Milano, T.U. Muenchen, T. U. Kaiserslautern, U. Paderborn, CalTech, U. Pernambuco, U. Perpignan, U. Tokyo, Virginia Tech U. and several companies.

URL: <http://flopoco.gforge.inria.fr/>

- Version: 2.3.0 (december 2011)
- APP: IDDN.FR.001.400014.000.S.C.2010.000.20600 (version 2.0.0)
- License: specific, GPL-like.
- Type of human computer interaction: command-line interface, synthesizable VHDL output.
- OS/Middleware: Linux, Windows/Cygwin.
- Required library or software: MPFR, flex, Sollya.
- Programming language: C++.
- Documentation: online and command-line help, API in doxygen format, articles.

5.3. GNU MPFR

Participants: Vincent Lefèvre [correspondant], Paul Zimmermann [Caramel, Inria Nancy - Grand Est].

GNU MPFR is an efficient multiple-precision floating-point library with well-defined semantics (copying the good ideas from the IEEE-754 standard), in particular correct rounding in 5 rounding modes. GNU MPFR provides about 80 mathematical functions, in addition to utility functions (assignments, conversions...). Special data (*Not a Number*, infinities, signed zeros) are handled like in the IEEE-754 standard.

MPFR was one of the main pieces of software developed by the old SPACES team at Loria. Since late 2006, with the departure of Vincent Lefèvre to Lyon, it has become a joint project between the Caramel (formerly SPACES then CACAO) and the AriC (formerly Arénaire) project-teams. MPFR has been a GNU package since 26 January 2009.

An MPFR-MPC developers meeting took place from 25 to 27 June 2012 in Bordeaux. GNU MPFR 3.1.1 was released on 3 July 2012.

The main changes done in the AriC project-team for the future versions are tcc support, more automation for the releases, new functions to operate on groups of flags, and bug fixes.

URL: <http://www.mpfr.org/>

GNU MPFR is now on the Ohloh community platform for free and open source software: <https://www.ohloh.net/p/gnu-mpfr>

- ACM: D.2.2 (Software libraries), G.1.0 (Multiple precision arithmetic), G.4 (Mathematical software).
- AMS: 26-04 Real Numbers, Explicit machine computation and programs.
- APP: no longer applicable (copyright transferred to the Free Software Foundation).
- License: LGPL version 3 or later.
- Type of human computer interaction: C library, callable from C or other languages via third-party interfaces.
- OS/Middleware: any OS, as long as a C compiler is available.
- Required library or software: **GMP**.
- Programming language: C.
- Documentation: API in texinfo format (and other formats via conversion); algorithms are also described in a separate document.

5.4. Exhaustive Tests for the Correct Rounding of Mathematical Functions

Participant: Vincent Lefèvre.

The search for the worst cases for the correct rounding (hardest-to-round cases) of mathematical functions (exp, log, sin, cos, etc.) in a fixed precision (mainly double precision) using Lefèvre’s algorithm is implemented by a set of utilities written in Perl, with calls to Maple/intpakX for computations on intervals and with C code generation for fast computations. It also includes a client-server system for the distribution of intervals to be tested and for tracking the status of intervals (fully tested, being tested, aborted).

The Perl scripts have been improved to detect various errors from Maple and in particular, restart Maple automatically when the license server is not reachable.

5.5. FLIP: Floating-point Library for Integer Processors

Participants: Claude-Pierre Jeannerod [correspondant], Jingyan Jourdan-Lu.

FLIP is a C library for the efficient software support of binary32 IEEE 754-2008 floating-point arithmetic on processors without floating-point hardware units, such as VLIW or DSP processors for embedded applications. The current target architecture is the VLIW ST200 family from STMicroelectronics (especially the ST231 cores). This year, we have extended the DP2 operator (fused dot product in dimension two) and its specializations, initially designed for rounding to nearest, to directed rounding modes. We have also worked on the implementation of the simultaneous computation of sine and cosine, with proven 1-ulp accuracy and in the same latency as the evaluation of sine alone.

URL: <http://flip.gforge.inria.fr/>

- ACM: D.2.2 (Software libraries), G.4 (Mathematical software)
- AMS: 26-04 Real Numbers, Explicit machine computation and programs.
- APP: IDDN.FR.001.230018.S.A.2010.000.10000
- License: CeCILL v2
- Type of human computer interaction: C library callable, from any C program.
- OS/Middleware: any, as long as a C compiler is available.
- Required library or software: none.
- Programming language: C

5.6. FPLLL: A Lattice Reduction Library

Participants: Xavier Pujol, Damien Stehlé [correspondant].

fpLLL contains several algorithms on lattices that rely on floating-point computations. This includes implementations of the floating-point LLL reduction algorithm, offering different speed/guarantees ratios. It contains a “wrapper” choosing the estimated best sequence of variants in order to provide a guaranteed output as fast as possible. In the case of the wrapper, the succession of variants is oblivious to the user. It also includes a rigorous floating-point implementation of the Kannan-Fincke-Pohst algorithm that finds a shortest non-zero lattice vector, and the BKZ reduction algorithm.

The fpLLL library is used or has been adapted to be integrated within several mathematical computation systems such as Magma, Sage and PariGP. It is also used for cryptanalytic purposes, to test the resistance of cryptographic primitives.

Versions 4.0.0 and 4.0.1 were released in 2012, implementing the BKZ reduction algorithm.

URL: <http://xpujol.net/fplll/>

- ACM: D.2.2 (Software libraries), G.4 (Mathematical software)
- APP: Procedure started
- License: LGPL v2.1
- Type of human computer interaction: C++ library callable, from any C++ program.
- OS/Middleware: any, as long as a C++ compiler is available.
- Required library or software: MPFR and GMP.
- Programming language: C++.
- Documentation: available in html format on **URL:** <http://xpujol.net/fplll/fplll-doc.html>

5.7. Symbolic-numeric Computations with Linear ODEs

Participant: Marc Mezzarobba.

NumGfun is a Maple package for performing numerical and “analytic” computations with the solutions of linear ordinary differential equations with polynomial coefficients. Its main features include the numerical evaluation of these functions with rigorous error bounds and the computation of symbolic bounds on solutions of certain recurrences. NumGfun is distributed as part of gfun, itself part of the Algolib bundle. It is used by the [Dynamic Dictionary of Mathematical Functions](#) to provide its numerical evaluation features. NumGfun 0.6, released in 2012, provides new feature for the numerical solution of so-called regular singular connection problems, and many small improvements.

URL: <http://marc.mezzarobba.net/#code-NumGfun>

- ACM: D.2.2 (Software libraries), G.4 (Mathematical software)
- APP: cf. gfun
- License: LGPL v2.1
- Type of human computer interaction: Maple library, usable interactively or from Maple code.
- OS/Middleware: any platform supporting Maple.
- Required library or software: [Maple](#), [gfun](#).
- Programming language: Maple
- Documentation: available as Maple help pages and in pdf format.

5.8. SIPE: Small Integer Plus Exponent

Participant: Vincent Lefèvre.

SIPE (Small Integer Plus Exponent) is a mini-library in the form of a C header file, to perform computations in very low precisions with correct rounding to nearest in radix 2. The goal of such a tool is to do proofs of algorithms/properties or computations of tight error bounds in these precisions by exhaustive tests, in order to try to generalize them to higher precisions. The currently supported operations are the addition, subtraction, multiplication, FMA, minimum/maximum/comparison functions (of the signed numbers or in magnitude), and conversions.

A new macro SIPE_2MUL, returning the rounded result and the error of a multiplication, has been added.

A test program and scripts to perform timing comparisons with hardware IEEE-754 floating-point and with GNU MPFR are available, together with a discussion on the technical and algorithmic choices behind SIPE and timing results. [39]

- ACM: D.2.2 (Software libraries), G.4 (Mathematical software).
- AMS: 26-04 Real Numbers, Explicit machine computation and programs.
- License: LGPL version 2.1 or later.
- Type of human computer interaction: C header file.
- OS/Middleware: any OS.
- Required library or software: GCC compiler.
- Programming language: C.
- Documentation: Research report Inria RR-7832.
- URL: <http://www.vinc17.net/software/sipe.h>

ARLES Project-Team

5. Software

5.1. Introduction

In order to validate our research results, our research activities encompass the development of related prototypes as surveyed below.

5.2. iCONNECT – Emergent Middleware Enablers

Participant: Valérie Issarny [correspondent].

As part of our research work on Emergent Middleware, we have implemented Enablers (or Enabler functionalities) that make part of the overall CONNECT architecture realizing Emergent Middleware in practice [2]. The focus of ARLES work is on the: *Discovery enabler* that builds on our extensive background in the area of interoperable pervasive service discovery; and *Synthesis enabler* that synthesizes mediators that allow networked systems that have compatible functionalities to interact despite mismatching interfaces and/or behaviors.

The CONNECT Discovery Enabler is the component of the overall CONNECT architecture that handles discovery of networked systems (NSs), stores their descriptions (NS models), and performs an initial phase of matchmaking to determine which pairs of systems are likely to be able to interoperate. Such pairs are then passed to the Synthesis Enabler so that mediators can be generated. The Discovery Enabler is written in Java and implements several legacy discovery protocols including DPWS and UPnP.

The proposed solution to mediator synthesis assumes an ontology-based system description *à la* OWL-S, which is made available by the Discovery Enabler, possibly helped by machine learning. The semantically-annotated interfaces of systems that need to communicate are then processed to compute the semantic mapping between their respective operations using a constraint solver. The resulting mapping serves generating a mediator process that further coordinates the behaviors of the systems and guarantees their successful interaction. The mediator is deployed on a dedicated engine able to parse and compose middleware messages and convert them to fit each system expectations regarding the type and order of these messages.

The CONNECT Enablers have been integrated and experimented with by the CONNECT consortium to effectively enable Emergent Middleware. Part of them are available for download under an open source license at the CONNECT Web site at <https://www.connect-forever.eu/software.html>.

5.3. Service-oriented Middleware for Pervasive Computing

Participants: Nikolaos Georgantas [correspondent], Valérie Issarny [correspondent].

In the past years, we have built a strong foundation of service-oriented middleware to support the pervasive computing vision. This specifically takes the form of a family of middlewares, all of which have been released under the open source LGPL license:

- **WSAMI - A Middleware Based on Web Services for Ambient Intelligence:** WSAMI (Web Services for AMbient Intelligence) is based on the Web services architecture and allows for the deployment of services on wireless handheld devices like smartphones and PDAs.
URL: <http://www-rocq.inria.fr/arles/download/ozone/index.htm>
- **Ariadne - A Protocol for Scalable Service Discovery in MANETs:** Ariadne enriches WSAMI with the Ariadne service discovery protocol, which has been designed to support decentralized Web service discovery in multi-hop mobile ad hoc networks (MANETs). Ariadne enables small and resource-constrained mobile devices to seek and find complementary, possibly mobile, Web services needed to complete specified tasks in MANETs, while minimizing the traffic generated and tolerating intermittent connectivity.

URL: <http://www-rocq.inria.fr/arles/download/ariadne/index.html>

- **MUSDAC - A Middleware for Service Discovery and Access in Pervasive Networks:** The Multi-protocol Service Discovery and Access (MUSDAC) middleware platform enriches WSAMI so as to enable the discovery and access to services in the pervasive environment, which is viewed as a loose and dynamic composition of independent networks. MUSDAC manages the efficient dissemination of discovery requests between the different networks and relies on specific plug-ins to interact with the various middleware used by the networked services.

URL: <http://www-rocq.inria.fr/arles/download/ubisec/index.html>

- **INMIDIO - An Interoperable Middleware for Ambient Intelligence:** INMIDIO (INteroperable MiddleWare for service Discovery and service InteractiOn) dynamically resolves middleware mismatch. More particularly, INMIDIO identifies the interaction middleware and also the discovery protocols that execute on the network and translates the incoming/outgoing messages of one protocol into messages of another, target protocol.

URL: <http://www-rocq.inria.fr/arles/download/inmidio/index.html>

- **COCOA - A Semantic Service Middleware:** COCOA is a comprehensive approach to semantic service description, discovery, composition, adaptation and execution, which enables the integration of heterogeneous services of the pervasive environment into complex user tasks based on their abstract specification. Using COCOA, abstract user tasks are realized by dynamically composing the capabilities of services that are currently available in the environment.

URL: <http://gforge.inria.fr/projects/amigo/>

- **ubiSOAP - A Service Oriented Middleware for Seamless Networking:** ubiSOAP brings multi-radio, multi-network connectivity to services through a comprehensive layered architecture: (i) the multi-radio device management and networking layers together abstract multi-radio connectivity, selecting the optimal communication link to/from nodes, according to quality parameters; (ii) the communication layer allows for SOAP-based point-to-point and group-based interactions in the pervasive network; and (iii) the middleware services layer brings advanced distributed resource management functionalities customized for the pervasive networking environment.

URL: <http://www.ist-plastic.org>.

5.4. xSOM – Service-oriented middleware for the Future Internet

Participant: Nikolaos Georgantas [correspondent].

Building on our long experience on service-oriented middleware (SOM) for pervasive environments (see § 5.3 above) and given the evolution of such environments towards the Future Internet and the Internet of Things, we have already implemented early results of our related research into an extensible SOM (xSOM) for the Future Internet. xSOM aims at enabling large-scale dynamic compositions of services and things, while being highly extensible for accommodating the extreme heterogeneity of such compositions. xSOM currently supports two major functionalities: (i) a protocol bus-based solution to seamless integration of heterogeneous interaction paradigms for services and things; and (ii) a solution to discovery, access and data fusion over large populations of things.

Regarding (i), we have introduced a protocol interoperability solution comprising representative abstract connector types for the client/server (CS), publish/subscribe (PS) and tuple space (TS) paradigms, as well as their mapping to a higher-level generic application (GA) connector type. We apply these connector abstractions to introduce an enhanced bus paradigm, the eXtensible Service Bus (XSB). XSB features richer interaction semantics than common Enterprise Service Bus (ESB) implementations and incorporates special consideration for semantics-preserving cross-integration of CS, PS and TS. We have carried out a realization of XSB—first on the PEtALS ESB, and then on EasyESB—where we provide templates for systematic and highly facilitated building of binding components for heterogeneous systems (services and things) that are plugged into the XSB. To demonstrate the applicability of our approach, we have implemented a smoke-detection-and-alert system integrating a JMEDS DPWS Web Service (CS), a JMS system based on Apache ActiveMQ (PS), and a Jini JavaSpaces system (TS).

Regarding (ii), we support data queries over large populations of things, notably smartphones, which are getting increasingly ubiquitous and embed a rich collection of sensors. xSOM enables: (i) high-level programming of things on top of heterogeneous smartphone sensors; (ii) thing discovery and access dealing with large numbers of networked things; and (iii) on-the-fly composition of such things and fusion of their data in response to queries. In target settings, e.g., at the scale of a city, not all phones need to register for reporting their data (e.g., ambient sound level); some smartly distributed sampling is sufficient. This enables efficient scalable coverage of the entire city with only a subset of the large phone population being registered. The phone's things registration manager includes a probabilistic decision algorithm for selective registration based on the truncated Lévy walk mobility model. The registration decision is based on the actual density of already registered phones, the coverage quality requirements, and the coverage of the estimated path that the user will take for the next few minutes. We have implemented a demonstrative application enabling a user to know "how lively is this city spot at this moment" based on retrieving and aggregating smartphone ambient sound level data.

Our software will soon be released under open source license as part of the newly launched OW2 initiative on "Future Internet of Software Services" (http://www.ow2.org/view/Future_Internet/).

5.5. Srijan: Data-driven Macroprogramming for Sensor Networks

Participant: Animesh Pathak [correspondent].

Macroprogramming is an application development technique for wireless sensor networks (WSNs) where the developer specifies the behavior of the system, as opposed to that of the constituent nodes. As part of our work in this domain, we are working on *Srijan*, a toolkit that enables application development for WSNs in a graphical manner using data-driven macroprogramming.

It can be used in various stages of application development, *viz.*,

1. Specification of application as a task graph,
2. Customization of the auto-generated source files with domain-specific imperative code,
3. Specification of the target system structure,
4. Compilation of the macroprogram into individual customized runtimes for each constituent node of the target system, and finally
5. Deployment of the auto generated node-level code in an over-the-air manner to the nodes in the target system.

The current implementation of *Srijan* targets both the Sun SPOT sensor nodes and larger nodes with J2SE. Most recently, *Srijan* also includes rudimentary support for incorporating Web services in the application being designed. The software is released under open source license, and available as an Eclipse plug-in at <http://code.google.com/p/srijan-toolkit/>.

5.6. Yarta: Middleware for supporting Mobile Social Applications

Participant: Animesh Pathak [correspondent].

With the increased prevalence of advanced mobile devices (the so-called "smart" phones), interest has grown in *Mobile Social Ecosystems* (MSEs), where users not only access traditional Web-based social networks using their mobile devices, but are also able to use the context information provided by these devices to further enrich their interactions. We are developing a middleware framework for managing mobile social ecosystems, having a multi-layer middleware architecture consisting of modules, which will provide the needed functionalities, including:

- Extraction of social ties from context (both physical and virtual),
- Enforcement of access control to protect social data from arbitrary access,
- A rich set of MSE management functionalities, using which mobile social applications can be developed.

Our middleware adopts a graph-based model for representing social data, where nodes and arcs describe socially relevant entities and their connections. In particular, we exploit the Resource Description Framework (RDF), a basic Semantic Web standard language that allows representing and reasoning about social vocabulary, and creating an interconnected graph of socially relevant information from different sources.

The current implementation of the Yarta middleware targets both desktop/laptop nodes running Java 2 SE, as well as Android smart phones. The software is released under open source license at <https://gforge.inria.fr/projects/yarta/>.

5.7. iBICOOP: Mobile Data Management in Multi-* Networks

Participant: Valérie Issarny [correspondent].

Building on the lessons learned with the development of pervasive service oriented middleware and of applications using them, we have been developing the custom iBICOOP middleware. iBICOOP specifically aims at assisting the development of advanced mobile, collaborative application services by supporting interactions between mobile users.

Briefly, the iBICOOP middleware addresses the challenges of easily accessing content stored on mobile devices, and consistent data access across multiple mobile devices by targeting both fixed and mobile devices, leveraging their characteristics (e.g., always on and unlimited storage for home/enterprise servers, ad hoc communication link between mobile devices), and by leveraging the capabilities of all available networks (e.g., ad hoc networks, Internet, Telecoms infrastructure networks). It also relies on Web and Telecoms standards to promote interoperability.

The base architecture of the iBICOOP middleware consists of core modules on top of which we can develop applications that may arise in the up-coming multi-device, multi-user world:

- The *Communication Manager* provides mechanisms to communicate over different available network interfaces of a device — Bluetooth, WiFi, Cellular — and also using different technologies e.g., Web services, HTTP/TCP sockets, ad hoc mode.
- The *Security Manager* uses well-established techniques of cryptography and secure communication to provide necessary security.
- The *Partnership Manager* provides device or user information in the form of *profiles*.
- iBICOOP relies on service location protocols for *naming and discovery* of nearby services on currently active network interfaces that support IP multicast.
- Besides normal file managing tasks, the *Local File Manager* gives the user clear cues to the files that have been replicated across multiple devices or shared among different users by using different icons.

The iBICOOP middleware has been licensed by AMBIENTIC (<http://www.ambientic.com/>), a start-up that specifically develops innovative mobile distributed services on top of the iBICOOP middleware that allows for seamless interaction and content sharing in today's multi-* networks.

URL: <https://www-roc.inria.fr/arles/index.php/ongoing-research-projects/74-data-sharing-and-replication-in-pervasive-networks>

ASAP Project-Team

5. Software

5.1. WhatsUp: A Distributed News Recommender

Participants: Antoine Boutet, Davide Frey, Arnaud Jegou, Anne-Marie Kermarrec.

Contact:	Antoine Boutet
Licence:	Open Source
Presentation:	A Distributed News Recommender
Status:	Beta version

This work has led to the development of WhatsUp, a distributed recommendation system aimed to distribute instant news in a large scale dynamic system. WhatsUp has two parts, an embedded application server in order to exchange with other peers in the system and a fully dynamic web interface for displaying news and collecting opinions about what the user reads. Underlying this web-based application lies Beep, a biased epidemic dissemination protocol that delivers news to interested users in a fast manner while limiting spam. Beep is parametrized on the fly to manage the orientation and the amplification of news dissemination. Every user forwards the news of interest to a randomly selected set of users with a preference towards those that have similar interests (orientation). The notion of interest does not rely on any explicit social network or subscription scheme, but rather on an implicit and dynamic overlay capturing the commonalities between users with respect to what they are interested in. The size of the set of users to which a news is forwarded depends on the interest of the news (amplification). A centralized version of WhatsUp is already up and running and the decentralized one is still in beta version.

5.2. GossipLib: effective development of gossip-based applications

Participants: Davide Frey, Heverson Borba Ribeiro, Anne-Marie Kermarrec.

Contact:	Davide Frey
Licence:	Open Source
Presentation:	Library for Gossip protocols
Status:	released version 0.7alpha

GossipLib is a library consisting of a set of JAVA classes aimed to facilitate the development of gossip-based application in a large-scale setting. It provides developers with a set of support classes that constitute a solid starting point for building any gossip-based application. GossipLib is designed to facilitate code reuse and testing of distributed applications: it thus provides the implementation of a number of standard gossip protocols that may be used out of the box or extended to build more complex protocols and applications. These include for example the peer-sampling protocols for overlay management.

GossipLib also provides facility for the configuration and deployment of applications as final-product but also as research prototype in environments like PlanetLab, clusters, network emulators, and even as event-based simulation. The code developed with GossipLib can be run both as a real application and in simulation simply by changing one line in a configuration file.

5.3. YALPS

Participants: Davide Frey, Heverson Borba Ribeiro, Anne-Marie Kermarrec.

Contact:	Davide Frey
Licence:	Open Source
Presentation:	Library for Gossip protocols
Status:	released version 0.3alpha

YALPS is an open-source Java library designed to facilitate the development, deployment, and testing of distributed applications. Applications written using YALPS can be run both in simulation and in real-world mode without changing a line of code or even recompiling the sources. A simple change in a configuration file will load the application in the proper environment. A number of features make YALPS useful both for the design and evaluation of research prototypes and for the development of applications to be released to the public. Specifically, YALPS makes it possible to run the same application as a simulation or in a real deployment without a single change in the code. Applications communicate by means of application-defined messages which are then routed either through UDP/TCP or through YALPS's simulation infrastructure. In both cases, YALPS's communication layer offers features for testing and evaluating distributed protocols and applications. Communication channels can be tuned to incorporate message losses or to constrain their outgoing bandwidth. Finally, YALPS includes facilities to support operation in the presence of NATs and firewalls using relaying and NAT-traversal techniques.

The work has been done in collaboration with Maxime Monod (EPFL).

5.4. HEAP: Heterogeneity-aware gossip protocol.

Participants: Davide Frey, Arnaud Jegou, Anne-Marie Kermarrec.

Contact:	Davide Frey
Licence:	Open Source
Presentation:	Java Application
Status:	release & ongoing development

This work has been done in collaboration with Vivien Quéma (CNRS Grenoble), Maxime Monod and Rachid Guerraoui (EPFL), and has led to the development of a video streaming platform based on HEAP, *HEterogeneity-Aware gossip Protocol*. The platform is particularly suited for environment characterized by heterogeneous bandwidth capabilities such as those comprising ADSL edge nodes. HEAP is, in fact, able to dynamically leverage the most capable nodes and increase their contribution to the protocol, while decreasing by the same proportion that of less capable nodes. During the last few months, we have integrated HEAP with the ability to dynamically measure the available bandwidth of nodes, thereby making it independent of the input of the user.

ASCLEPIOS Project-Team

4. Software

4.1. SOFA

Participants: Hervé Delingette [correspondant], Brina Goyette, Federico Spadoni, Stéphanie Marchesseau, Hugo Talbot.

SOFA is an Open Source framework primarily targeted at real-time simulation, with an emphasis on medical simulation. It is mostly intended for the research community to help develop newer algorithms, but can also be used as an efficient prototyping tool. based on an advanced software architecture, it allows to:- create complex and evolving simulations by combining new algorithms with algorithms already included in SOFA- modify most parameters of the simulation (deformable behavior, surface representation, solver, constraints, collision algorithm, etc.) by simply editing an XML file- build complex models from simpler ones using a scene-graph description- efficiently simulate the dynamics of interacting objects using abstract equation solvers- reuse and easily compare a variety of available methods. It is mainly developed by the Inria team projects Shaman, Evasion and Asclepios.

See also the web page <http://www.sofa-framework.org/>.

- ACM: J.2 Physics, J.3 LIFE AND MEDICAL SCIENCES
- Software benefit:- Simulation of the human body
- License: GPL
- License: LGPL
- Type of human computer interaction: console, opengl, qt
- OS/Middleware: linux, windows, mac
- Required library or software: Qt - GPL - GLEW - BSD/MIT - Tinyxml - zlib
- Programming language: C/C++
- Documentation: - each function of the core API and each class in the SOFA modules - doxygen
- ACM: J.3
- Programming language: C/C++

4.2. MedInria

Participants: Benoît Bleuzé, Florian Vichot, Hakim Fadil, Loic Cadour, Agata Krason, Maxime Sermesant [correspondant], Nicolas Toussaint.

MedInria is a free collection of softwares developed by the Asclepios research project in collaboration with the Athena, Parietal and Visages Inria research projects. It aims at providing to clinicians state-of-the-art algorithms dedicated to medical image processing and visualization. Efforts have been made to simplify the user interface, while keeping high-level algorithms. MedInria is available for Microsoft windows XP/Vista/7, Linux Fedora Core, MacOSX, and is fully multithreaded.

The first release of Medinria 2.0 was done in April 2012.

See also the web page <http://med.inria.fr>.

- Version: 2.0
- Keywords: Medical Image Processing
- License: Proprietary Licence
- Type of human computer interaction: QT
- OS/Middleware: Windows - Linux - MacOSX
- Required library or software: DTI Track (Proprietary), vtkInria3D (CeCillB), Baladin (Proprietary)
- Programming language: C++

ASCOLA Project-Team

5. Software

5.1. AWED

Participants: Mario Südholt [correspondent], Ismael Mejia.

Aspect-oriented programming, distributed programming, event-based programming, invasive patterns

The model of Aspects With Explicit Distribution (AWED) supports the modularization of crosscutting functionalities of distributed applications. It addresses the problem that common aspect systems do not provide features for distributed programming. It notably features three main aspect abstractions: remote pointcuts, remotely-executed advice, and distributed aspects.

The AWED system has also been employed in the CESSA project proposal (see Sec. 8.1) as a basis for our work on the secure evolution of service-oriented architectures.

AWED is available at <http://awed.gforge.inria.fr>.

5.2. btrCloud (and Entropy)

Participants: Jean-Marc Menaud [correspondent], Rémy Pottier, Clotilde Massot, Guillaume Le Louët, Thierry Bernard, Frédéric Dumont.

Orchestration, virtualization, energy, autonomic system, placement, cloud computing, cluster, data center, scheduler, grid

btrCloud is a virtual machine manager for clusters and provides a complete solution for the management and optimization of virtualized data center. btrCloud (acronym of better cloud) is composed of three parts.

The analysis function enables operatives and people in charge to monitor and analyze how a data-center works, be it on a daily basis or on the long run and predict future trends. This feature includes a performances, an analysis and a trends board.

btrCloud, by the integration of btrScript, provides (semi-)automated, VM lifecycle management, including provisioning, resource pool management, VM tracking, cost accounting, and scheduled deprovisioning. Key features include a thin client interface, template-based provisioning, approval workflows, and policy-based VM placement.

Finally, Several kinds of optimizations are currently available, such as energy and load balancing. The former can help save up to around 20% of the data-center energy consumption, of course depending on the context. The latter enhances provides optimal quality of service for the applications that are hosted in the virtualized data-center.

btrCloud is available at <http://www.btrcloud.org>.

5.3. ECaesarJ, EJava and EScala

Participants: Jacques Noyé [correspondent], Jurgen Van Ham.

Symmetric AOP, features, software product lines, inheritance, virtual classes, propagating mixin composition, event-based programming, events, declarative events, state machines, CaesarJ, Java, Scala

ECaesarJ is a language developed in the context of the European project AMPLE, as joint work with the *Technische Universität Darmstadt* (TUD). The basic objective was to provide support for directly mapping the high-level features defined by a software product line onto implementation-level features, beyond standard feature-oriented programming. But the language has much wider applications. ECaesarJ can actually be seen as a language which smoothly integrates Object-Oriented Programming, Feature-Oriented Programming, Aspect-Oriented Programming, and Event-based Programming.

It is an extension of Java with *virtual classes* and *propagating mixin composition* (as its ancestor CaesarJ, developed at TUD), but also *declarative events* and *state machines*. Unlike AspectJ, ECaesarJ does not include a class-like concept of aspect. Instead, it deals with pointcuts and pieces of advice as (implicit) events and event handlers, which are standard class members. This makes it possible to use standard inheritance to reuse and refine them. Explicit events can also be used when events must be explicitly triggered as in traditional event-based programming. Finally, in the same way as pointcuts can be composed using logical operators, *declarative events* can be defined as a composition of other events.

This provides a symmetric version of AOP where virtual classes can be used to deal with structural aspects whereas events can be used to deal with behavioral aspects.

In ECaesarJ, a class can also include, as class members, state transitions. Combining this with virtual classes makes it possible to define, at the programming language level, refinable hierarchical state machines. The combination of state machines and events provides, in particular, effective language support for the State design pattern as well as a form of Event-based AOP.

EJava and EScala are more recent developments of the same ideas applied to Java and Scala, respectively. EJava benefits from Java tooling with an eclipse plugin developed with the Spoofox Language Workbench. Unlike EJava and ECaesarJ, EScala makes it possible to dynamically register and unregister event handlers. It also benefits from a more efficient, compiler-based, implementation. As ECaesarJ, EScala is joint work with TUD.

Prototype implementations of these languages are available through <http://ecaesarj.gforge.inria.fr/>.

5.4. FPath and FScript

Participants: Thomas Ledoux [correspondent], Frederico Alvares.

dynamic reliable reconfiguration, self-adaptive components, Fractal, autonomic computing

FPath and FScript are two domain-specific languages (DSLs) dealing respectively with the navigation and the dynamic reconfiguration of Fractal architectures. *FPath* is a DSL for querying Fractal architectures. It is restricted to the introspection of architectures by browsing elements identified by their properties or location in the architecture. This focused domain allows FPath to offer a very concise and readable syntax and ensures correctness properties by construction (e.g. any query terminates in a finite time). *FScript* is a DSL dedicated to the reconfiguration of Fractal component architectures. It enables reconfiguration scripts to modify a Fractal architecture. Like FPath, FScript guarantees several properties by construction, e.g. termination of scripts by excluding the possibility of infinite loops. Moreover the FScript interpreter supports a transactional model of reconfigurations and the preservation of the ACID properties.

An adaptation of FPath/FScript to FraSCAti, a component framework providing runtime support for the Service Component Architecture (SCA), has been developed by the Inria Adam project-team. In that way, software architects are able to navigate using FPath notation through FraSCAti architectures and to reconfigure them with FScript. We have used this adaptation in our recent work [11][31] for reconfiguring cloud applications in order to reduce the energy footprint in cloud infrastructures.

FScript and its extensions are available under the LGPL license at <http://fractal.ow2.org/fscript>.

5.5. WildCAT

Participants: Thomas Ledoux [correspondent], Frederico Alvares.

monitoring, context-aware applications, complex event processing

WildCAT is a generic Java framework for context-aware applications. It permits the monitoring of large-scale applications by allowing developers to easily organize and access resources through a hierarchical organization backed with a powerful SQL-like language to inspect sensors data and to trigger actions upon particular conditions. WildCAT proposes two modes to inspect the resources: a pull mode relies on synchronous communication and a push one relies on asynchronous communication. In the pull mode, developers programmatically get and set attributes. In the push mode, developers register listeners on queries expressed over the events generated by the backend.

WildCAT has been developed by the team in the last years. We have used WildCAT in our recent work [11] for allowing cloud applications to listen events notification fired by the cloud infrastructure (e.g. whenever the pricing policy of cloud resources changes) or to detect changes on the application activity (e.g. to detect whenever the number of requests sharply increases/decreases) in order to launch the reconfiguration of cloud applications.

WildCAT is available under GPL v2 at <http://wildcat.ow2.org>.

ASPI Project-Team (section vide)

ATEAMS Project-Team

4. Software

4.1. Derric

Participants: Tijs van der Storm, Jeroen van den Bos [correspondent].

Characterization: A-2-up3, SO-4, SM-2-up3, EM-3, SDL-3-up4, OC-DA-3-CD-3-MS-3-TPM-3.

WWW: <http://www.derric-lang.org>

Objective: Encapsulate all the variability in the construction of so-called “carving” algorithms, then generate the fastest and most accurate implementations. Carving algorithms recover information that has been deleted or otherwise scrambled on digital media such as hard-disks, usb sticks and mobile phones.

Users: Digital forensic investigation specialists

Impact: Derric has the potential of revolutionizing the carving area. It does in 1500 lines of code what other systems need tens of thousands of lines for with the same accuracy. Derric will be an enabler for faster, more specialized and more successful location of important evidence material.

Competition: Derric competes in a small market of specialized open-source and commercial carving tools.

Engineering: Derric is a Rascal program of 1.5 kloc designed by two persons.

Publications: [8][32], [14]

In 2012 Derric was validated on a large body of image files taken from wikipedia, and a new approach to software optimization via model transformation was developed for optimizing Derric code. We released Derric 1.0 in 2012.

4.2. Basic Voting Theory

Participants: Jan van Eijck [correspondent], Floor Sietsma.

Characterization: A1, SO-3, SM-1, EM-1, SDL-2, OC-DA-3-CD-3-MS-3-TPM-3.

WWW: <http://homepages.cwi.nl/~jve/software>

Objective: Demonstrate the basic concepts of voting theory.

Users: Students and researchers interested in voting theory.

Impact: This is a demonstrator and a tool for teaching.

Competition: None.

Engineering: Haskell program.

4.3. Rascal

Participants: Paul Klint, Jurgen Vinju [correspondent], Tijs van der Storm, Jeroen van den Bos, Mark Hills, Bert Lisser, Atze van der Ploeg, Vadim Zaytsev, Anastasia Izmaylova, Michael Steindorfer, Ali Afrozeh.

Characterization: A5, SO-4, SM-4, EM-4, SDL-4-up5, OC-DA-3-CD-3-MS-3-TPM-3.

WWW: <http://www.rascal-mpl.org>

Objective: Provide a completely integrated programming language parametric meta programming language for the construction of any kind of meta program for any kind of programming language: analysis, transformation, generation, visualization.

Users: Researchers in model driven engineering, programming languages, software engineering, software analysis, as well as practitioners that need specialized tools.

Impact: Rascal is making the mechanics of meta programming into a non-issue. We can now focus on the interesting details of the particular fact extraction, model, source analysis, domain analysis as opposed to being distracted by the engineering details. Simple things are easy in Rascal and complex things are manageable, due to the integration, the general type system and high-level programming features.

Competition: There is a plethora of meta programming toolboxes and frameworks available, ranging from plain parser generators to fully integrated environments. Rascal is distinguished because it is a programming language rather than a specification formalism and because it completely integrates different technical domains (syntax definition, term rewriting, relational calculus). For simple tools, Rascal competes with scripting languages and for complex tools it competes context-free general parser generators, with query engines based on relational calculus and with term rewriting and strategic programming languages.

Engineering: Rascal is about 100 kLOC of Java code, designed by a core team of three and with a team of around 8 phd students and post-docs contributing to its design, implementation and maintenance. The goal is to work towards more bootstrapping and less Java code as the project continues.

Publications: [23], [22], [11], [21], [22]

4.3.1. Novelties

- Statically typed access to external data-sources [21]. This includes access to CVS files, spreadsheets, databases, etc.
- Significant improvements to online documentation and inter-active tutor environment.
- Full transparent support for Unicode codepoints.
- Added language-supported quickcheck-style random testing facility (by Wietse Venema, intern), including bridge to JUnit testing framework and IDE support.
- Revived access libraries to CVS, SVN and Git VCSs.
- Added support for JSON export and import, towards Rascal webservice.
- Totally re-implemented and extended debugging interface.
- Priority and associativity mechanism for context-free grammars was completed, such that it can not be used to accidentally remove sentences from a language anymore.
- Reimplementation of the except disambiguation filter with much higher efficiency.
- Improved module import times.
- Reimplemented URI encoding/decoding mechanism for correctness and portability.
- Added semi-automated exam generation and grading feature to the Rascal tutor environment.
- Experimented with strategies for error recovery in context-free general top-down parser.
- Added MissGrant and SuperAwesomeFighter language workbench demonstrations.
- Structured re-design of menus and menu options in the IDE

- Added bindings to Apache statistics libraries
- Created Rascalopedia, a glossary of concepts and terms that are relevant for metaprogrammers. The descriptions are aiming at under-graduate students.
- Two previously designed programmable transformation languages for grammars in a broad sense: the unidirectional XBGF and the bidirectional Ξ BGF — have been reimplemented as libraries in Rascal.
- Improved general stability and efficiency.

4.4. IDE Meta-tooling Platform

Participants: Jurgen Vinju [correspondent], Michael Steindorfer.

IMP, the IDE meta tooling platform is an Eclipse plugin developed mainly by the team of Robert M. Fuhrer at IBM TJ Watson Research institute. It is both an abstract layer for Eclipse, allowing rapid development of Eclipse based IDEs for programming languages, and a collection of meta programming tools for generating source code analysis and transformation tools.

Characterization: A5, SO-3, SM4-up5, EM-4, SDL-5, DA-2-CD-2-MS-2-TPM-2

WWW: <http://www.eclipse.org/imp>

Objective: The IDE Meta Tooling Platform (IMP) provides a high-level abstraction over the Eclipse API such that programmers can extend Eclipse with new programming languages or domain specific languages in a few simple steps. IMP also provides a number of standard meta tools such as a parser generator and a domain specific language for formal specifications of configuration parameters.

Users: Designers and implementers of IDEs for programming languages and domain specific languages. Also, designers and implementers of meta programming tools.

Impact: IMP is popular among meta programmers especially for it provides the right level of abstraction.

Competition: IMP competes with other Eclipse plugins for meta programming (such as Model Driven Engineering tools), but its API is more general and more flexible. IMP is a programmers framework rather than a set of generators.

Engineering: IMP is a long-lived project of many contributors, which is managed as an Eclipse incubation project at eclipse.org. Currently we are moving the project to Github to explore more different ways of collaboration.

Publications: [3]

4.5. Ensō

Participant: Tijs van der Storm [correspondent].

Characterization: A5, SO-4, SM-3-up-4, EM-2-up-4, SDL-4, OC-DA-4-CD-4-MS-4-TPM-4

WWW: <http://www.enso-lang.org>

Objective: Together with Prof. Dr. William R. Cook of the University of Texas at Austin, and Alex Aloh, Tijs van der Storm has been designing and implementing a new programming system, called Ensō. Ensō is theoretically sound and practical reformulation of model-based development. It is based on model-interpretation as opposed to model transformation and code generation. Currently, the system already supports models for schemas (data models), web applications, context-free grammars, diagram editors and security.

Users: All programmers.

Impact: Ensō has the potential to revolutionize the activity of programming. By looking at model driven engineering from a completely fresh perspective, with as key ingredients interpreters and partial evaluation, it may make higher level (domain level) program construction and maintenance as effective as normal programming.

Competition: Ensō competes as a programming paradigm with model driven engineering tools and generic programming and languages that provide syntax macros and language extensions.

Engineering: Ensō is less than 7000 lines of (bootstrapped) Ruby code.

4.6. Turing language

Participants: Davy Landman [correspondent], Tijs van der Storm, Jeroen van den Bos, Vadim Zaytsev, Paul Klint.

Characterization: A3, SO-2, SM-1, EM-1, SDL-5, DA-3-CD-3-MS-3-TPM-3

WWW: <http://www.legoturingmachine.org>

Objective: This software is used to program the Lego Turing Machine which was built as a piece for the Turing Centennial exposition at CWI. The software features a full fledged Eclipse based IDE for a small programming language which is compiled to Turing machine instructions that run on a Lego machine.

Users: People interested in learning about computation and programming languages.

Impact: the Lego Turing Machine and its software have reached more than 3 million people via the internet (slashdot, vimeo, youtube) and all Dutch national newspapers.

Competition: none.

Engineering: the hardware is Lego and the software is fully generated from Rascal syntax definitions and IDE construction functions.

4.7. Lua AiR

Participant: Riemer van Rozen [correspondent].

Characterization: A2-up, SO-4, SM-2-up, EM-3-up, SDL-4, DA-3-CD-3-MS-3-TPM-3

WWW: https://github.com/cwi-swat/Lua_AiR

Objective: This system provides IDE integrated support for static analysis of Lua code. Lua is a scripting language used in game development.

Users: Game programmers and game designers

Impact: Lua AiR is currently a research prototype designed to experiment with and validate the static analysis of a highly dynamic scripting language.

Competition: none.

Engineering: Lua AiR is fully implemented in Rascal.

ATHENA Project-Team

5. Software

5.1. OpenMEEG

Participants: Théodore Papadopoulo, Maureen Clerc, Alexandre Gramfort [Telecom ParisTech].

OpenMEEG provides state-of-the-art tools for low-frequency bio-electromagnetism, notably solving forward problems related to EEG and MEG [5]. It implements the symmetric BEM which provides excellent accuracy and versatility. OpenMEEG is a free open software written in C++. It can be accessed either through a command line interface or through Python/Matlab interfaces.

OpenMEEG is multiplatform (Linux, MacOS, Windows) and it is distributed under the French opensource license CeCILL-B. See also the web page <http://www-sop.inria.fr/athena/software/OpenMEEG/>.

5.2. Diffusion MRI

Participants: Aurobrata Ghosh, Rachid Deriche.

The algorithms previously developed within the ODYSSEE Project team and related to the Diffusion Tensor and Q-Ball imaging are available upon request from the Inria source forge (<https://gforge.inria.fr>). One can use all the estimation and visualization tools developed, ranging from estimation, regularization, segmentation to Q-ball estimation, fiber ODF estimation and tractography algorithms. New visualization tools for Q-Ball images represented by spherical harmonic decomposition have also been developed.

The software library comprises geometric and variational methods devised to estimate, regularize, segment and perform tractography in DT (Diffusion Tensor) and HARDI (High Angular Resolution) MRI images. The library is multi-platform (Linux, Windows and OS X) and is embedded into two open-source high level languages, TCL and Python.

5.3. medInria

Participants: Jaime Garcia Guevara, Théodore Papadopoulo.

The Athena team is involved along with the research teams Asclepios, Parietal and Visages in the development of medInria a free software platform dedicated to medical data visualization and processing.

It aims at providing to clinicians and researchers state-of-the-art algorithms developed at Inria and elsewhere (for the future), through an intuitive user interface. medInria offers from standard to cutting-edge processing functionalities for medical images such as 2D/3D/4D image visualization, image registration, diffusion MR processing and tractography.

Athena contribution so far consists in various improvements on the core application as well as several plugins which will be available in the next version: advanced dMRI visualization and processing (integration of the Diffusion MRI library depicted in the previous section), M/EEG signal visualisation (by integrating code from the software AnyWave developed at by Bruno Colombet and J.-M. Badier [Inserm UMR 1106 and Aix-Marseille University](#)).

See also the web page <http://med.inria.fr>.

- Version: 2.0.1
- Keywords: Medical Image Processing and Visualization
- License: Proprietary Licence (soon open source for the core application)
- Multiplatform: Windows - Linux - MacOSX
- Programming language: C++

ATLANMOD Team

5. Software

5.1. The ATL Model Transformation Language

URL: <http://www.eclipse.org/m2m/atl/>

With an eye on the normative work of the OMG (MOF, OCL, QVT, etc.), a new conceptual framework has been developed based on a second generation model transformation language called ATL. Although ATL influenced the OMG standard, the approach is more general as discussed in [8]. In 2004 IBM gave an Eclipse innovation award to the ATL project. In 2007 Eclipse recognized ATL as one central solution for model transformation and promoted it to the M2M project (see *Eclipse.org/m2m*). There are more than 200 industrial and academic sites using ATL today, and several Ph.D. thesis in the world are based on this work.

In 2011 we started a new evolution phase for ATL. Our mid-term plan is making of ATL the leading solution for building autonomous reactive transformation systems, i.e. transformation networks that can autonomously manage a set of dataflows among the application models.

Following this line, we first implemented a new refinement mode for ATL, to support in-place transformations. This extension allows the dynamic manipulation of models while keeping them connected to runtime applications. Next, we presented a lazy execution algorithm for ATL. With it, the elements of the target model are generated only when and if they are accessed. This extension allows to build reactive transformation systems that react to requests of model elements, by triggering the necessary computation. Our lazy version of ATL enables also transformations that generate infinite target models, extending the application space of the model-transformation paradigm.

The latest (still ongoing) work in this direction is the development of a full reactive ATL engine, able to activate the minimal computation for responding to updates or request on the involved models. This engine is studied to scale up with large ATL networks. In this line we also introduced an algorithm for simplifying ATL transformation chains.

5.2. MoDisco (Model Discovery)

URL: <http://www.eclipse.org/modisco/>

MoDisco is an open source Eclipse project that provides a generic and extensible framework dedicated to the elaboration of Model Driven Reverse Engineering (MDRE) solutions. Gathering contributions from both academics and industrials, the goal of the project is to federate common efforts in the model-based transformation of legacy software systems implemented using different technologies (e.g.; Java, COBOL, C). The first principle is to discover models out of legacy artifacts, representing appropriately all the relevant information, to be then used as part of reverse engineering processes for software understanding, evolution or modernization. Targeted scenarios include software (technical or architectural) migration of large legacy systems, but also retro-documentation, refactoring, quality assurance, etc. Within this context, MoDisco has collaborations with the OMG Architecture Driven Modernization (ADM) Task Force, for which the project provides several reference implementations of its standards: Knowledge Discovery Metamodel (KDM), Software Measurement Metamodel (SMM) and Abstract Syntax Tree Metamodel (ASTM).

The MoDisco framework [12] is composed of a set of Eclipse plugins, and relies on the de-facto standard Eclipse Modeling Framework (EMF) for model handling. Thanks to its modular architecture, it allows completely covering the three steps of a standard MDRE approach: 1) Discovery (i.e. extracting a complete model of the source code), 2) Understanding (i.e. browsing and providing views on this model for a given purpose) and 3) Transformation (evolving the model towards a new technology, architecture, etc). More specifically, as part of its *Infrastructure* layer, MoDisco offers the set of generic (i.e.; legacy technology-independent) reusable components really useful to build the core of MDRE solutions: Discovery Manager and Workflow for MDRE task orchestration, Model Browser for advanced navigation in complex models, model extension and customization capabilities for understanding (e.g.; views definition), etc. As part of its *Technologies* layer, it provides an advanced support for the Java, JEE and XML technologies, including complete metamodels, corresponding model discoverers, transformations, code generators, customizations, query libraries, etc.

MoDisco (or some of its components) is being used by different partners including other academics, industrials (e.g.; Sodifrance on several of their real modernization projects for their customers) or Eclipse projects (e.g.; Eclipse-MDT Papyrus as developed by CEA). Moreover, the Eclipse-EMFT EMF Facet project has been initiated as a MoDisco spin-off, in order to externalize some features which are not actually specific to reverse engineering problems and thus may be reused in many different contexts (cf. corresponding EMF Facet section).

The initiative continues to be developed within the context of the European FP7-ICT project named ARTIST², and also to a lower extent within the context of the French FUI 13 project named TEAP.

5.3. Community-driven language development

URL: <http://code.google.com/a/eclipselabs.org/p/collaboro/>

Software development processes are collaborative in nature. Neglecting the key role of end-users leads to software that does not satisfy their needs. This collaboration becomes specially important when creating Domain-Specific Languages (DSLs), which are (modeling) languages specifically designed to carry out the tasks of a particular domain. While end-users are actually the experts of the domain for which a DSL is developed, their participation in the DSL specification process is still rather limited nowadays.

Thus, Collaboro is an approach to make language development processes more participative, meaning that both developers and users of the language can collaborate together to create and evolve it. The tool has been developed as an Eclipse plugin, whose features currently implemented are:

- Version view to navigate through the Proposals of a version. For each Proposal, the solutions and comments are shown.
- Collaboration view to show the data related to a Collaboration selected in the version view. This view also shows the changes to apply if the selected element is a Solution.
- The user can login to the Collaboro system and create proposals, solutions and comments by right-clicking in the version view. The user can also vote for/against the collaborations.
- Decision engine based on a total agreement (i.e., all the community users must vote for the collaboration). The decision engine can be launch by using the bar menu.
- Notation engine and Notation view to render SVG snapshots of the DSL concrete syntax.

5.4. Virtual EMF (Model Virtualization)

URL: <http://code.google.com/a/eclipselabs.org/p/virtual-emf/>

²<http://www.artist-project.eu/>

Virtual EMF is an Eclipse plugin built on top of EMF that enables the creation and manipulation of *virtual models*, i.e., models whose elements do not contain concrete data, but are rather proxies to elements contained in other models. The idea is related to that of model composition, as it aims capturing the (often overlapping) concepts as one single global model. This is a frequently faced problem as, in complex scenarios, modelers often have to deal with a large number of heterogeneous and interrelated models. Most times, the view a specific kind of user requires does not correspond to any of these models but is a combination of cross-domain information scattered among several ones.

Current composition techniques rely on the materialization of the composed model, an approach that poses some important limitations in terms of (i)*efficiency*, as they do not scale (the data duplication mechanism they use implies in extra memory usage and time-consuming generation of the composed model), (ii)*synchronization*, as updates in the composed model are not propagated to the original ones (or vice-versa) thus losing consistency, or even (iii)*interoperability*, as in some cases the composed model requires a specific API/tool to be handled.

Virtual EMF allows overcoming the limitations above. A virtual model provides to tools/users the *illusion* of working with a regular model whereas, in fact, all model access and manipulation requests are transparently redirected to its set of *virtualized* models. It serves as a centralized and transparent access point to a set of interconnected models, allowing users to easily compose, weave and link them. It provides the following beneficial properties:

- *Interoperability*: it behaves as a normal model. Therefore, compatibility with existing EMF-based solutions/tools (e.g. models transformations, model editors, ...) is guaranteed;
- *Synchronization*: changes are automatically and transparently propagated between virtual and original models;
- *Scalability*: support for very big models;
 - low memory usage: no data duplication, direct access to original model elements;
 - faster generation time: no need for (time-consuming) information cloning operations (e.g. executing a model transformation);
- *Genericity*: support for several types of inter-model relationships (e.g. merge, association, filter) and extension capabilities for their semantics.

Virtual EMF is available as an open source project on Eclipse Labs. It has been contributed by the AtlanMod team to the CESAR project. The initiative continues to be developed within the context of the French FUI 13 project named TEAP [TODO Put ref to TEAP <http://www.atlanpole.fr/Atlanpole-Digital-Innovation/liste-des-news/TEAP-Projet-Collaboratif-d-Innovation>].

5.5. EMFtoCSP

URL: <http://code.google.com/a/eclipselabs.org/p/emftocsp/>

EMFtoCSP is a tool for the verification of precisely defined conceptual models and metamodels. For these models, the definition of the general model structure (using UML or EMF) is supplemented by OCL constraints. The Eclipse Modeling Development Tools (MDT ³) provides mature tool support for such OCL-annotated models with respect to model definition, transformation, and validation.

However, an additional important task that is not supported by Eclipse MDT is the assurance of model quality. A systematical assessment of the correctness of such models is a key issue to ensure the quality of the final application. EMFtoCSP fills this gap by provided support for automated model verification in Eclipse.

³<http://www.eclipse.org/modeling/mdt/?project=ocl>

Essentially, the EMFtoCSP is a sophisticated bounded model finder that yields instances of the model that conform not only to the structural definition of the model (e.g., the multiplicity constraints), but also to the OCL constraints. Based on this core, several correctness properties can be verified:

1. Satisfiability – is the model able to express our domain? For this check, the minimal number of instances and links can be specified to ensure non-trivial instances.
2. Unsatisfiability – is the model unable to express undesirable states? To verify this, we add further constraints to the model that state undesired conditions. Then we can check if it is impossible to instantiate the amended model.
3. Constraint subsumption – is one constraint already implied by others (and could therefore be removed)?
4. Constraint redundancy – do different constraints express the same fact (and could therefore be removed)?

To solve these search problems, EMFtoCSP translates the EMF/OCL (resp. UML/OCL) model into a constraint satisfaction problem and employs the Eclipse CLP solver ⁴ to solve it. This way, constraint propagation is exploited to tackle the (generally NP-hard) search.

The tool is a continuation of the UMLtoCSP approach [48] developed previously by Jordi Cabot, Robert Clarisó and Daniel Riera. It provides a generic plugin framework for Eclipse to solve OCL-annotated models using constraint logic programming. Apart from already supported Ecore and UML metamodels, further metamodels can be added easily in the future. Similarly, other constraint solving back-ends can be integrated. It is provided under the Eclipse Public License.

5.6. EMF Facet

URL: <http://www.eclipse.org/modeling/emft/facet/>

EMF Facet is an open source Eclipse project, under the Eclipse Public License (EPL), that provides a generic and extensible framework dedicated to the dynamic and non-intrusive extension of models. It can be used to extend already existing metamodels with additional concepts and properties, the corresponding models being then transparently augmented, reduced or modified accordingly at runtime. Such a metamodel extension is called a facet, and can be specified on top of any metamodel in EMF Ecore. The underlying mechanism is based on the runtime execution of queries on the models corresponding to the *faceted* metamodels. Facets are notably particularly relevant for obtaining different views on existing models without having to actually alter them with any extra data.

The EMF Facet framework is composed of several Eclipse plugins, and relies on the de-facto standard Eclipse Modeling Framework (EMF) for model handling. The facet definitions are stored as facet models, allowing them to be exchanged and reused in various contexts. The queries can be implemented using any suitable query language (e.g.; ATL, OCL, Java, XPath), as far as the corresponding adapters exist and are correctly registered within the framework. The proposed tooling includes dedicated editors for creating, editing and saving both facet and query definitions, the implemented support for Java, OCL and ATL queries, a Table Editor for visualizing query results. An advanced support for the model display customization (e.g.; icons, colors, fonts) is also provided as part of the framework.

EMF Facet is currently intensively used in MoDisco for extracting and displaying different specific views from large models of legacy systems. Its extension and customization capabilities are actually integrated into several MoDisco components, such as notably the MoDisco Model Browser. However, different other integration possibilities will be also explored in the future.

The initiative continues to be developed within the context of the European FP7-ICT project ARTIST.

⁴<http://eclipseclp.org/>

5.7. Industrialization strategy for research prototypes

Research labs, as a source of innovation, are potential key actors of the Software Engineering market. However, an important collaborative effort with the other players in the software industry is still needed in order to actually transfer the corresponding techniques or technologies from the research lab to a company. Based on the AtlanMod concrete experience with the previously mentioned open source tools/projects, we have extracted a pragmatic approach [3] for transforming the results of scientific experimentation into practical industrial solutions.

While dealing with innovation, this approach is also innovation-driven itself, as the action is actually conducted by the research lab via a technology transfer. Three different partners are directly involved in this process, using open source as the medium for maintaining a constant interaction between all of them:

- **Use Case Provider.** Usually a company big enough to have to face real complex industrial scenarios which need to be solved (at least partially) by applying new innovative principles and techniques;
- **Research Lab.** Usually a group from a research institute (public or private) or university evaluating the scientific relevance of the problems, identifying the research challenges and prototyping possible solutions;
- **Technology Provider.** Usually a small or medium company, with a particular technical expertise on the given domain or Software Engineering field, building and delivering the industrial version of the designed solutions;

From our past and current experience, three main characteristics of this industrialization *business model* can be highlighted:

- **Win-win situation.** Each partner can actually focus on its core activity while also directly benefiting from the results obtained by the others (notably the research lab can continue to do research);
- **Application-driven context.** The end-user need is at the origin of the process, which finally makes the developed solution actually relevant;
- **Iterative process.** The fact of having three distinct partners requires different regular and consecutive exchanges between all of them.

AVALON Team

5. Software

5.1. BitDew

Participants: Gilles Fedak [correspondant], Haiwu He, Bing Tang, José Francisco Saray Villamizar, Mircea Moca, Lu Lu.

BITDEW is an open source middleware implementing a set of distributed services for large scale data management on Desktop Grids and Clouds. BITDEW relies on five abstractions to manage the data : i) replication indicates how many occurrences of a data should be available at the same time on the network, ii) fault-tolerance controls the policy in presence of hardware failures, iii) lifetime is an attribute absolute or relative to the existence of other data, which decides of the life cycle of a data in the system, iv) affinity drives movement of data according to dependency rules, v) protocol gives the runtime environment hints about the protocol to distribute the data (http, ftp, or bittorrent). Programmers define for every data these simple criteria, and let the BITDEW runtime environment manage operations of data creation, deletion, movement, replication, and fault-tolerance operation.

BITDEW is distributed open source under the GPLv3 or Cecill licence at the user's choice. 10 releases were produced over the last two years, and it has been downloaded approximately 6,000 times on the Inria forge. Known users are Université Paris-XI, Université Paris-XIII, University of Florida (USA), Cardiff University (UK) and University of Sfax (Tunisia). In terms of support, the development of BitDew is partly funded by the Inria ADT BitDew and by the ANR MapReduce projects. Thanks to this support, we have developed and released the first prototype of the MapReduce programming model for Desktop Grids on top of BitDew. In 2012, 8 versions of the software have been released, including the version 1.2.0 considered as a stable release of BitDew with many advanced features. Our most current work focuses on providing reliable storage on top of hybrid distributed computing infrastructures.

5.2. SBAM

Participants: Eddy Caron [correspondant], Florent Chuffart.

SBAM (<http://graal.ens-lyon.fr/SBAM>) is the middleware directly coming from results of the ANR project SPADES. SBAM initiates a non-intrusive, but highly dynamic environment able to take advantages of available resources without disturbing their native mechanism. SBAM federates multisite resources in order to schedule, submit and compute users' tasks in a transparent way.

SBAM is, firstly, a decentralized grid middleware. It relies on a P2P approach, i.e., a set of agents able to discover resources and schedule computing tasks over a federation of heterogeneous computing platforms (petascale computers, data centers, clouds, ...). SBAM dynamically acquires and releases resources of computing sites according to users' needs and conditions, to federate them into a global constantly growing or shrinking logical platform, referred to as the overlay.

5.3. DIET

Participants: Daniel Balouek, Eddy Caron [correspondant], Frédéric Desprez, Maurice Djibril Faye, Cristian Klein, Arnaud Lefray, Guillaume Mercier, Adrian Muresan, Jonathan Rouzaud-Cornabas, Lamiel Toch, Huaxi Zhang.

Huge problems can now be processed over the Internet thanks to Grid and Cloud middleware systems. The use of on-the-shelf applications is needed by scientists of other disciplines. Moreover, the computational power and memory needs of such applications may of course not be met by every workstation. Thus, the RPC paradigm seems to be a good candidate to build Problem Solving Environments on the Grid or Cloud. The aim of the DIET project (<http://graal.ens-lyon.fr/DIET>) is to develop a set of tools to build computational servers accessible through a GridRPC API.

Moreover, the aim of a middleware system such as DIET is to provide a transparent access to a pool of computational servers. DIET focuses on offering such a service at a very large scale. A client which has a problem to solve should be able to obtain a reference to the server that is best suited for it. DIET is designed to take into account the data location when scheduling jobs. Data are kept as long as possible on (or near to) the computational servers in order to minimize transfer times. This kind of optimization is mandatory when performing job scheduling on a wide-area network. DIET is built upon *Server Daemons*. The scheduler is scattered across a hierarchy of *Local Agents* and *Master Agents*. Applications targeted for the DIET platform are now able to exert a degree of control over the scheduling subsystem via *plug-in schedulers*. As the applications that are to be deployed on the Grid vary greatly in terms of performance demands, the DIET plug-in scheduler facility permits the application designer to express application needs and features in order that they be taken into account when application tasks are scheduled. These features are invoked at runtime after a user has submitted a service request to the MA, which broadcasts the request to its agent hierarchy.

In 2012, our objective was to extend DIET to benefit from virtualized resources such as ones coming from cloud platforms. We have designed how it can be extended to access virtualized resources. We can easily support new cloud service providers and cloud middleware systems. We have prototyped the new version of DIET which benefits from virtualized resources. As cloud resources are dynamic, we have on-going research in the field of automatic and elastic deployment for middleware systems. DIET will be able to extend and reduce the amount on aggregated resources and adjust itself when resources fail. We have started works to extend our data management software, DAGDA, to take advantage of cloud storage and the new data computing paradigms. Moreover we have upgraded the workflow engine of DIET to take advantage of cloud resources. DIET Cloud will be able to provide a large scale distributed and secured platform that spans on a pool of federated resources that range from dedicated HPC clusters and grid to public and private clouds.

In the context of the Seed4C project, we have studied how secured our platform, authenticated and secured interactions between the different parts of our middleware and between our middleware and its users. We have worked to show how to securely use public cloud storage without taking the risk of losing confidentiality of data stored on them.

5.4. Pilgrim

Participants: Eddy Caron, Matthieu Imbert [correspondant].

Pilgrim (<http://pilgrim.gforge.inria.fr>) is an open metrology and prediction performance framework whose goal is to provide easy and powerful tools for instrumenting computer platforms and predicting their behavior. Those tools are aimed at being used not only by humans but also by programs, in particular by resource managers and schedulers. Pilgrim is designed to be a loosely coupled integration of various custom-developed or off-the-shelf tools.

5.5. SimGrid

Participants: Georges Markomanolis, Jonathan Rouzaud-Cornabas, Frédéric Suter [correspondant].

SimGrid is a toolkit for the simulation of distributed applications in heterogeneous distributed environments. The specific goal of the project is to facilitate research in the area of parallel and distributed large scale systems, such as Grids, P2P systems and clouds. Its use cases encompass heuristic evaluation, application prototyping or even real application development and tuning. SimGrid has an active user community of more than one hundred members, and is available under GPLv3 from <http://simgrid.gforge.inria.fr/>.

5.6. HLCMi & L2C

Participants: Zhengxiong Hou, Cristian Klein, Vincent Lanore, Christian Pérez [correspondant], Vincent Pichon.

HLCMI (<http://hlcm.gforge.inria.fr>) is an implementation of the HLCM component model. HLCM is a generic extensible component model with respect to component implementations and interaction concerns. Moreover, HLCM is abstract; it is its specialization—such as HLCM/L²C—that defines the primitive elements of the model, such as the primitive components and the primitive interactions.

HLCMI is making use of Model-driven Engineering (MDE) methodology to generate a concrete assembly from an high level description. It is based on the Eclipse Modeling Framework (EMF). HLCMI contains 700 Emfatic lines to describe its models and 7000 JAVA lines for utility and model transformation purposes. HLCMI is a general framework that supports several HLCM specializations: HLCM/CCM, HLCM/JAVA, HLCM/L2C and HLCM/Charm++ (known as Gluon++).

L²C (<http://hlcm.gforge.inria.fr>) is a *Low Level Component* model implementation targeting at use-cases where overhead matters such as High-Performance Computing. L²C does not offer network transparency neither language transparency. Instead, L²C lets the user choose between various kinds of interactions between components, some with ultra low overhead and others that support network transport. L²C is extensible as additional interaction kinds can be added quite easily. L²C currently supports C++, MPI and CORBA interactions. FORTRAN will be added in 2013.

L²C and Gluon++ are implemented in the LLCMc++ framework (<http://hlcm.gforge.inria.fr>). It is distributed under a LGPL licence and represents 6400 lines of C++.

AVIZ Project-Team

5. Software

5.1. Graph Cuisine

Participants: Évelyne Lutton [correspondant], Benjamin Bach, André Spritzer, Jean-Daniel Fekete.

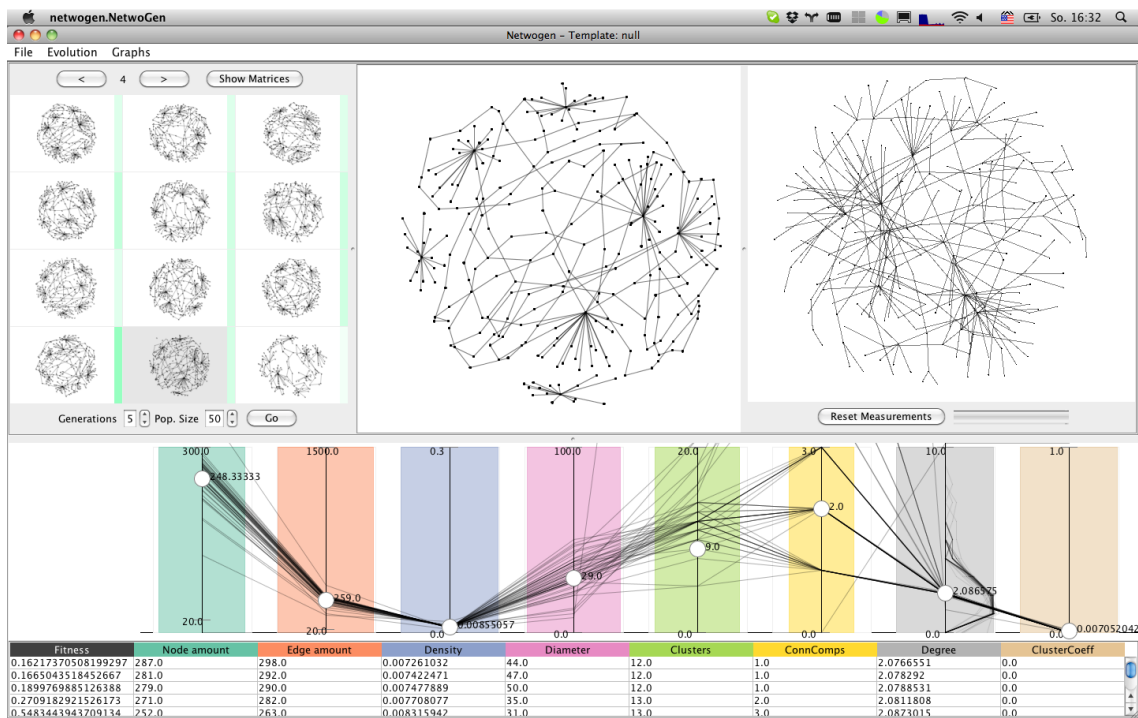


Figure 1. GraphCuisine interface showing one real graph (right), the measures extracted from it (bottom circles), several graphs with similar measures (left) and one of them selected (middle).

GraphCuisine lets users steer an Evolutionary Algorithm (EA) to create random graphs that match user-specified measures. Generating random graphs with particular characteristics is crucial for evaluating graph algorithms, layouts and visualization techniques. Current random graph generators provide limited control of the final characteristics of the graphs they generate. The situation is even harder when one wants to generate random graphs similar to a given one, all-in-all leading to a long iterative process that involves several steps of random graph generation, parameter changes, and visual inspection. Our system follows an approach based on interactive evolutionary computation. Fitting generator parameters to create graphs with pre-defined measures is an optimization problem, while assessing the quality of the resulting graphs often involves human subjective judgment. GraphCuisine has been proved to be able to generate graphs that mimic a given real-world network. <http://www.aviz.fr/Research/Graphcuisine>

5.2. Histomages

Participants: Fanny Chevalier, Pierre Dragicevic [correspondant], Christophe Hurter.

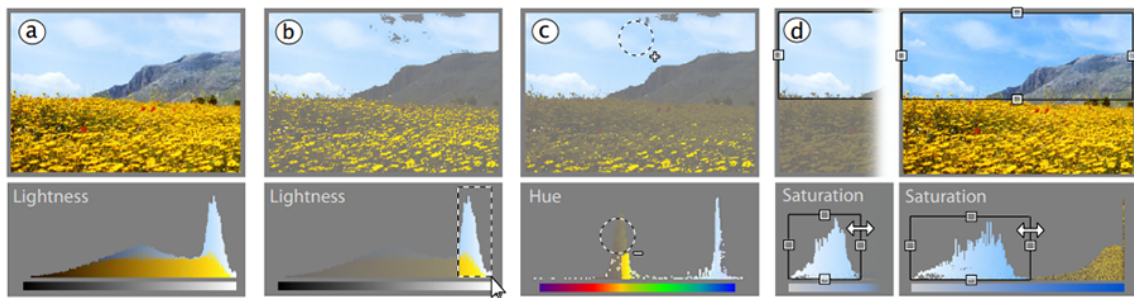


Figure 2. Example of sky enhancement with Histomages: (a) the image is duplicated and its pixels rearranged into a lightness histogram ; (b) bright pixels are selected with the rubber-band selection tool; (c) all pixels are rearranged into a hue histogram and yellow pixels are filtered out with the subtract selection brush (bottom). Missing pixels are added with the add selection brush on the image (top); (d) the sky is enhanced by resizing the selection on the saturation histogram.

Histomages is an image editor based on a new interaction model that considers histogram views as spatial rearrangements of image pixels. Users can select pixels on image histograms as they would select image regions and directly manipulate them to adjust their colors. Histomages are affected by other image tools such as paintbrushes. We explored some possibilities offered by this interaction model, and discussed the four key principles behind it as well as their implications for the design of feature-rich software in general [29]. <http://www.aviz.fr/histomages/>.

5.3. Glimpse

Participants: Pierre Dragicevic [correspondant], Stéphane Huot, Fanny Chevalier.



Figure 3. Glimpse: A detail of the animation between an article and its LaTeX source code.

Glimpse is a quick preview technique that smoothly transitions between document markup code (HTML, LaTeX,...) and its visual rendering. This technique allows users to regularly check the code they are editing in-place, without leaving the text editor. This method can complement classical preview windows by offering rapid overviews of code-to-document mappings and leaving more screen real-estate. A proof-of-concept editor can be downloaded for free at <http://www.aviz.fr/glimpse/>.

5.4. The Obvious Toolkit

Participants: Pierre-Luc Hémary, Jean-Daniel Fekete [correspondant].

Information Visualization, Java, Toolkit

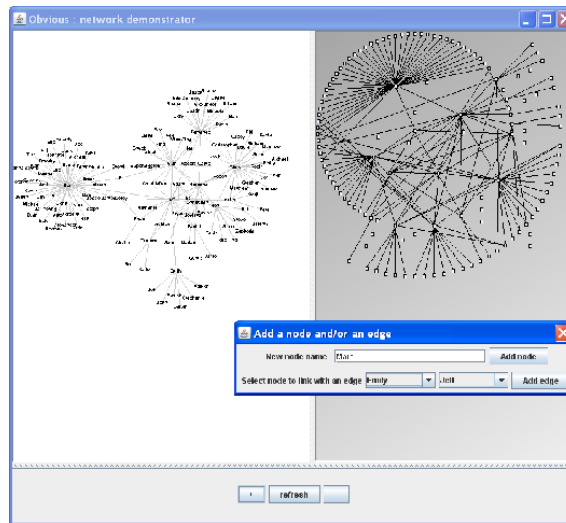


Figure 4. The Obvious toolkit showing the same graph with a Prefuse and an IVTK rendering.

The Obvious Toolkit is a new Interactive Graphics Toolkit written in Java to facilitate the interoperability between Information Visualization toolkits and components (Fig. 4).

The Obvious Toolkit is an abstraction layer above visualization toolkits. Currently, it connects the most popular toolkits in Java: Prefuse, the InfoVis Toolkit, Improvise, JUNG, as well as other libraries such as the Java Database Communication Toolkit (JDBC) and two Machine-Learning toolkits: Weka and RapidMiner.

It is meant to provide an abstraction layer for information visualization application builders so that they can postpone their choice of a concrete toolkit to use. When faced with the final choice, application builders can use one of the toolkits or connect all of them dynamically to Obvious. A paper on Obvious was presented at the IEEE Visual Analytics Science and Technology conference (VAST 2011) [53]. Obvious is available at <http://code.google.com/p/obvious>.

5.5. GeneaQuilts

Participants: Jean-Daniel Fekete [correspondant], Pierre Dragicevic, Anastasia Bezerianos, Julie Bae, Ben Watson.

GeneaQuilts [2] is a new genealogy exploration software that allows genealogists and historians to visualize and navigate in large genealogies of up to several thousand individuals (Fig. 5). The visualization takes the form of a diagonally-filled matrix, where rows are individuals and columns are nuclear families. The GeneaQuilts system includes an overview, a timeline, search and filtering components, and a new interaction technique called Bring & Slide that allows fluid navigation in very large genealogies. The tool has been featured in several InfoVis and genealogy Websites and the website has been visited over 9000 times. It has been integrated in commercial and open-source implementations (4 to date). See also the web page <http://www.aviz.fr/geneaquilts/>.

5.6. Diffamation

Participants: Fanny Chevalier, Pierre Dragicevic [correspondant], Anastasia Bezerianos, Jean-Daniel Fekete.

Animation, Edit histories, Wikipedia, Revision Control

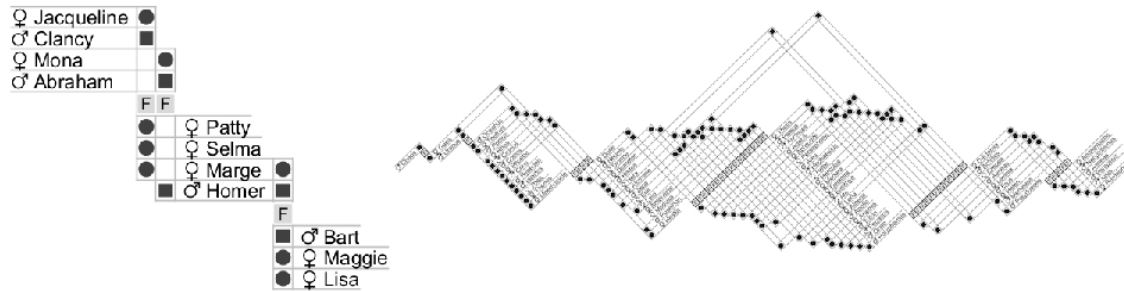


Figure 5. The genealogy of the Simpsons family (left) and of the Greek Pantheon (right), produced by the GeneaQuilts software.

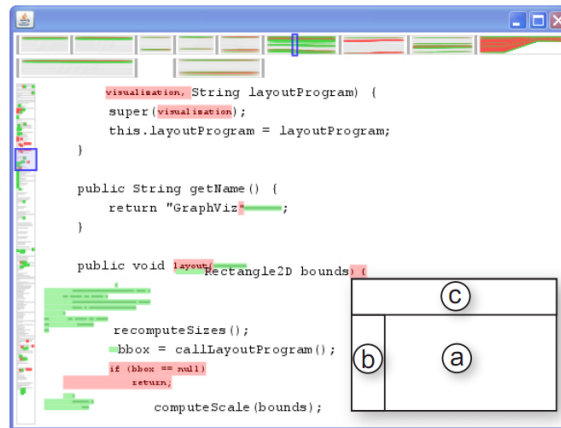


Figure 6. Screenshot the Diffamation system during a transition: (a) the document view, (b) the overview scrollbar and (c) the timeline.

The Diffamation system [3] allows rapid exploration of revision histories such as Wikipedia or subversion repositories by combining text animated transitions with simple navigation and visualization tools. Diffamation can be used for example to get a quick overview of the entire history of a Wikipedia article or to see what has happened to one's contributions. Diffamation complements classical diff visualizations: once moments of interest have been identified, classical diff visualizations can come in useful to compare two given revisions in detail.

The Diffamation revision exploration system has been presented at the plenary session of the Ubuntu Developer Summit. It is available at <http://www.aviz.fr/diffamation/>.

5.7. The InfoVis Toolkit

Participant: Jean-Daniel Fekete [correspondant].

Information Visualization, Java, Toolkit

The InfoVis Toolkit [5] is an Interactive Graphics Toolkit written in Java to facilitate the development of Information Visualization applications and components.

The InfoVis Toolkit implements several visualization techniques, as well as interaction techniques related. It has been used for teaching the Information Visualization course (Masters level, Univ. of Paris-Sud) and is the basis for all AVIZ contracts. It is our main development platform for information visualization; most of our Information Visualization prototypes rely on it. It is available at <http://ivtk.sourceforge.net>.

In the forthcoming years, it will be superseded by extensions of the Obvious Toolkit (see section 5.4).

5.8. GraphDice

Participants: Jean-Daniel FEKETE [correspondant], Pierre Dragicevic, Niklas Elmquist, Anastasia Bezerianos.

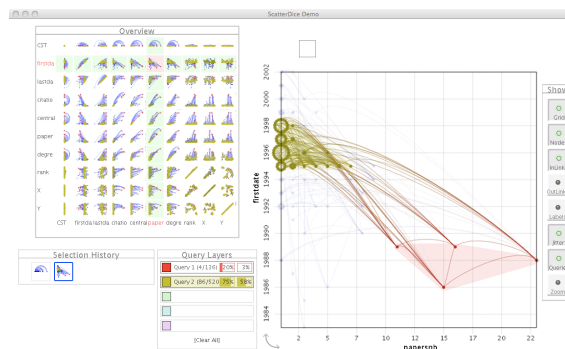


Figure 7. Screenshot the GraphDice system.

GraphDice [1] is a visualization system for exploring multivariate networks (Fig. 7). GraphDice builds upon our previous system ScatterDice (best paper award at the IEEE InfoVis 2008 conference) [4]: it shows a scatter plot of 2 dimensions among the multiple ones available and provides a very simple paradigm of 3D rotation to change the visualized dimensions. The navigation is controlled by a scatter plot matrix that is used as a high-level overview of the dataset as well as a control panel to switch the dimensions.

While ScatterDice works on any tabular dataset (e. g., CSV file), the GraphDice system show networks using a node-link diagram representation as a scatter plot with links drawn between connected nodes. For more information, see the web page at <http://graphdice.gforge.inria.fr>.

AXIS Project-Team

4. Software

4.1. Introduction

From its creation, AxIS has proposed new methods, approaches and **software** validated experimentally on various applications: Data Mining, Web usage Mining, Information Retrieval, Activity Modeling.

Some of our results are under process to be part of the FocusLab platform (CPER Télius 5.6) which is based on a Service oriented Architecture. The development process of the software part has started in 2011, finding ways to fund human resources. Such a platform aims the community of Living Labs domain. In [70], we report the usage of the FocusLab platform (hardware and software components) inside various regional and european projects..

4.2. Data Mining

4.2.1. Classification and Clustering Methods

Participants: Marc Csernel, Yves Lechevallier [co-correspondant], Brigitte Trousse [co-correspondant].

We developed and maintained a collection of clustering and classification software, written in C++ and/or Java:

Supervised methods

- a Java library (Somlib) that provides efficient implementations of several SOM(Self-Organizing Map) variants [77], [76], [101], [100], [104], especially those that can handle dissimilarity data (available on Inria's Gforge server (public access) **Somlib**, developed by AxIS Rocquencourt and Brieuc Conan-Guez from Université de Metz).
- a functional Multi-Layer Perceptron library, called FNET, that implements in C++ supervised classification of functional data [96], [99], [98], [97] (developed by AxIS Rocquencourt).

Unsupervised methods : partitioning methods

- Two partitioning clustering methods on the dissimilarity tables issued from a collaboration between AxIS Rocquencourt team and Recife University, Brazil: CDis and CCclust [84]. Both are written in C++ and use the "Symbolic Object Language" (SOL) developed for SODAS. And one partitioning method on interval data (Div).
- Two standalone versions improved from SODAS modules, SCluster and DIVCLUS-T [74] (AxIS Rocquencourt).

Unsupervised methods : agglomerative methods

- a Java implementation of the 2-3 AHC (developed by AxIS Sophia Antipolis). The software is available as a Java applet which runs the hierarchies visualization toolbox called HCT for Hierarchical Clustering Toolbox (see [75]).

A Web interface developed in C++ and running on our Apache internal Web server .is available for the following methods: SCluster, Div, Cdis, CCclust.

Previous versions of the above software have been integrated in the SODAS 2 Software [95] which was the result of the european project ASSO ⁶ (2001-2004). SODAS 2 supports the analysis of multidimensional complex data (numerical and non numerical) coming from databases mainly in statistical offices and administration using Symbolic Data Analysis [71]. This software is registered at APP (Agence de la Protection des Programmes). The latest executive version of the SODAS 2 software, with its user manual can be downloaded at <http://www.info.fundp.ac.be/asso/sodaslink.htm> [78], [85].

⁶ASSO: Analysis System of Symbolic Official data

As a 2012 result, a release of MND (Dynamic Clustering Method for Multi-Nominal data) algorithm based on previous AxIS research (2003) has been done (cf. section 5.6).

4.2.2. *Extracting Sequential Patterns with Low Support*

Participant: Brigitte Trousse [correspondant].

Two methods for extracting sequential patterns with low support have been developed by D. Tanasa in his thesis (see Chapter 3 in [103] for more details) in collaboration with F. Masseglia and B. Trousse :

- **Cluster & Divide**,
- and **Divide & Discover** [11].

These methods have been successfully applied from 2005 on various Web logs.

4.2.3. *Mining Data Streams*

Participants: Brigitte Trousse [correspondant], Mohamed Gaieb.

In Marascu's thesis (2009) [91], a collection of software have been developed for knowledge discovery and security in data streams. Three **clustering methods for mining sequential patterns (Java) in data streams** method have been developed in Java:

- SMDS compares the sequences to each others with a complexity of $O(n^2)$.
- SCDS is an improvement of SMDS, where the complexity is enhanced from $O(n^2)$ to $O(n.m)$ with n the number of navigations and m the number of clusters.
- ICDS is a modification of SCDS. The principle is to keep the clusters' centroids from one batch to another.

Such methods take batches of data in the format "Client-Date-Item" and provide clusters of sequences and their centroids in the form of an approximate sequential pattern calculated with an alignment technique.

In 2010 the Java code of one method called SCDS has been integrated in the MIDAS demonstrator and a C++ version has been implemented by F. Masseglia for the CRE contract with Orange Labs with the deliverability of a licence) with a visualisation module (in Java).

It has been tested on the following data:

- Orange mobile portal logs (100 million records, 3 months) in the context of Midas project (Java version) and the CRE (Orange C++ version)
- Inria Sophia Antipolis Web logs (4 million records, 1 year, Java version)
- Vehicle trajectories (**Brinkhoff generator**) in the context of MIDAS project (Java version).

In 2011, in the context of the ELLIOT contract [cf. Section 6.3.1.1), SCDS has been integrated as a Web service (Java version) in the first version of FocusLab platform (cf. section 5.6) in the ELLIOT context: a demonstration was made on San Raffaele Hospital media use case at the first ELLIOT review at Brussels.

In 2012 we applied SCDS web service on data issued from co-creation step of two use cases in Logistics (BIBA) and Green Services (ICT Usage Lab). More data are needed to show the relevance of this method, it is planned in 2013 with the experimentation step of Green Services.

The three C++ codes done for the CRE (Orange Labs) have been deposited at APP.

4.3. Web Usage Mining

4.3.1. *AWLH for Pre-processing Web Logs*

Participants: Yves Lechevallier [co-correspondant], Brigitte Trousse [co-correspondant].

AWLH (AxIS Web Log House) for Web Usage Mining (WUM) is issued from AxISlogminersoftware which implements the mult-site log preprocessing methodology and extraction of sequential pattern with low support developed by D. Tanasa in his thesis [15] for Web Usage Mining (WUM). In the context of the Eiffel project (2008-2009), we isolated and redesigned the core of AxISlogMiner preprocessing tool (we called it AWLH) composed of a set of tools for pre-processing web log files. The web log files are cleaned before to be used by data mining methods, as they contain many noisy entries (for example, robots requests). The data are stored within a database whose model has been improved.

So AWLH offers:

- Processing of several log files from several servers,
- Support of several input formats (CLF, ECLF, IIS, custom, ...),
- Incremental pre-processing,
- Java API to help integration of AWLH in external application.

An additional tool has been developed for capturing user actions in real time based on an open source project called "OpenSymphony ClickStream". An extension version of AWLH called **AWLH-Debate** has been developed for recording and structuring data issued from annotated documents inside discussion forums.

4.3.2. *ATWUEDA for Analysing Evolving Web Usage Data*

Participants: Yves Lechevallier [correspondant], Brigitte Trousse, Mohamed Gaieb, Yves Lechevallier [correspondant].

ATWUEDA for Web Usage Evolving Data Analysis [80] was developed by A. Da Silva in her thesis [79] under the supervision of Y. Lechevallier. This tool was developed in Java and uses the JRI library in order to allow the application of **R** which is a programming language and software environment for statistical computing functions in the Java environment.

ATWUEDA is able to read data from a cross table in a MySQL database. It splits the data according to the user specifications (in logical or temporal windows) and then applies the approach proposed in the Da Silva's thesis in order to detect changes in dynamic environment. The proposed approach characterizes the changes undergone by the usage groups (e.g. appearance, disappearance, fusion and split) at each timestamp. Graphics are generated for each analyzed window, exhibiting statistics that characterizes changing points over time.

Version 2. of ATWUEDA (september 2009) is available at Inria's gforce website.

In 2011 we have demonstrated the efficiency of ATWUEDA [82] by applying it on another real case study on condition monitoring data streams of an electric power plant provided by EDF.

ATWUEDA is used by Telecom Paris Tech and EDF [4].

This year we studied how to transform the code of ATWUEDA as a web service for the version 1.2 of FocusLab: in fact we gave up this objective, which would require more resource than we have.

4.4. Information Retrieval

4.4.1. *CBR*Tools for Managing and Reusing Past Experiences based on Historical Data*

Participant: Brigitte Trousse [correspondant].

CBR*Tools [87], [88] is an object-oriented framework [89], [86] for Case-Based Reasoning which is specified with the UMT notation (Rational Rose) and written in Java. It offers a set of abstract classes to model the main concepts necessary to develop applications integrating case-based reasoning techniques: case, case base, index, measurements of similarity, reasoning control. It also offers a set of concrete classes which implements many traditional methods (closest neighbors indexing, Kd-tree indexing, neuronal approach based indexing, standards similarities measurements). CBR*Tools currently contains more than 240 classes divided in two main categories: the core package for basic functionality and the time package for the specific management of the behavioral situations. The programming of a new application is done by specialization of existing classes, objects aggregation or by using the parameters of the existing classes.

CBR*Tools addresses application fields where the re-use of cases indexed by behavioral situations is required. The CBR*Tools framework was evaluated via the design and the implementation of several applications such as Broadway-Web, Educaid, BeCKB, Broadway-Predict, e-behaviour and Be-TRIP.

CBR*Tools is concerned by two past contracts: EPIA and MobiVIP.

CBR*Tools will be available for research, teaching and academic purpose via the FocusLab platform. The user manual can be downloaded at the URL: <http://www-sop.inria.fr/axis/cbrtools/manual/>.

See also the web page <http://www-sop.inria.fr/axis/cbrtools/manual/>.

4.4.2. Broadway*Tools for Building Recommender Systems on the Web

Participant: Brigitte Trousse [correspondant].

Broadway*Tools is a toolbox supporting the creation of adaptive recommendation systems on the Web or in a Internet/Intranet information system. The toolbox offers different servers, including a server that computes recommendations based on the observation of the user sessions and on the re-use of user groups' former sessions. A recommender system created with Broadway*tools observes navigations of various users and gather evaluations and annotations, to draw up a list of relevant recommendations (Web documents, keywords, etc).

Based on Jaczynski's thesis [87], different recommender systems have been developed for supporting Web browsing, but also browsing inside a Web-based information system or for query formulation in the context of a meta search engine.

4.5. Activity Modeling

4.5.1. K-MADe for Describing Human Operator or User Activities

Participant: Dominique Scapin [correspondant].

K-MADe tool (Kernel of Model for Human Activity Description Environment). The K-MADe is intended for people wishing to describe, analyze and formalize the activities of human operators, of users, in environments (computerized or not), in real or simulated situation; in the field, or in the laboratory. Although all kinds of profiles of people are possible, this environment is particularly intended for ergonomics and HCI (Human Computer Interaction) specialists. It has been developed through collaboration between ENSMA (LISI XSLaboratory) and Inria.

This year a new version v1.2 of K-MAD was released in december. Its history, documentation and tool are available at: <http://kmade.sourceforge.net/index.php>. This work follows up the findings from the work of Caffiau and al. [73].

AYIN Team

5. Software

5.1. Software

5.1.1. Transfers

- The software MAD V2.0 was transferred to Galderma R&D in November 2012.
- The software Scombo v1.1 was transferred to Cutis laboratory (Galderma R& D, Sophia Antipolis) in May 2012, and to the French-Singaporean laboratory IPAL (Image and Pervasive Access Lab) in November 2012.

5.1.2. Deposits

- The software MAD (Melasma Automatic Detector) V2.0 was deposited with the APP in November 2012. A patent has also been deposited jointly by Galderma R& D and Inria during the same month. It deals with the melasma severity scoring from multi-spectral imaging.
- The software Scombo (Supervised Classifier of MultiBand Optical images) v1.1 was deposited with the APP in April 2012. It deals with the supervised classification of multiband optical images by using Markov random fields. It was developed with Aurélie Voisin, Vladimir Krylov and Josiane Zerubia.

BACCHUS Team

5. Software

5.1. AeroSol

Participants: Damien Genêt [corresponding member for Bacchus], Maxime Mogé, Dragan Amenga-Mbengoué, François Pellegrini, Vincent Perrier [corresponding member], Mario Ricchiutto, François Rue.

The `Aerosol` software is jointly developed by teams BACCHUS and `Cagire`. It is a high order finite element library written in C++. The code design has been carried for being able to perform efficient computations, with continuous and discontinuous finite elements methods on hybrid and possibly curvilinear meshes. The distribution of the unknowns is made with the software PaMPA, developed within teams BACCHUS and PUMAS. Maxime Mogé has been hired on a young engineer position (IJD) obtained in the ADT OuBa HOP for participating in the parallelization of the library, and arrived on November, 1st 2011. On January 2012, Dragan Amenga-Mbengoué was recruited on the ANR `Realfluids`.

At the end of 2011, `Aerosol` had the following features:

- Development environment: use of `CMake` for compilation, `CTest` for automatic testing and memory checking, `lcov` and `gcov` for code coverage reports.
- In/Out: link with the XML library for handling with parameter files. Reader for `GMSH`, and writer to the VTK-ASCII legacy format.
- Quadrature formula: up to 11th order for Lines, Quadrangles, Hexaedra, Pyramids, Prisms, up to 14th order for tetrahedron, up to 21st order for triangles.
- Finite elements: up to fourth degree for Lagrange finite elements on lines, triangles and quadrangles.
- Geometry: elementary geometrical functions for first order lines, triangles, quadrangles.
- Time iteration: explicit Runge-Kutta up to fourth order, explicit Strong Stability Preserving schemes up to third order.
- Linear Solvers: link with the external linear solver `UMFPack`.
- Memory handling: discontinuous and continuous discretizations based on PaMPA for triangular and quadrangular meshes.
- Numerical schemes: continuous Galerkin method for the Laplace problem (up to fifth order) with non consistent time iteration or with direct matrix inversion. Scalar stabilized residual distribution schemes with explicit Euler time iteration have been implemented for steady problems.

This year, the following features were added:

- Development environment: development of a `CDash` server for collecting the unitary tests and memory checking. Beginning of the development of an interface for functional tests.
- General structure: Parts of the code were abstracted in order to allow for parallel development: Linear solvers (template type abstraction for generic linear solver external library), Generic integrator classes (integrating on elements, on faces with handling neighbor elements, or for working on Lagrange points of a given element), models (template abstraction for generic hyperbolic systems), equations of state (template-based abstraction for a generic equation of state).
- In/Out: Parallel `GMSH` reader, cell and point centered visualization based on VTK-legacy formats. XML paraview files on unstructured meshes (`vtu`), and parallel XML based files (`pvtu`).
- Quadrature formula: Gauss-Lobatto type quadrature formula.
- Finite elements: Hierarchical orthogonal finite element basis on lines, triangles (with Dubiner transform). Finite element basis that are interpolation basis on Gauss-Legendre points for lines, quadrangles, and hexaedra. Lagrange, and Hierarchical orthogonal finite elements basis for hexaedra, prisms and tetrahedra.

- Geometry: elementary geometrical functions for first order three dimensional shapes: hexaedra, prisms, and tetrahedra.
- Time iteration: CFL time stepping, optimized CFL time schemes: SSP(2,3) and SSP (3,4)
- Linear Solvers: Internal solver for diagonal matrices. Link with the external solvers PETSc and MUMPS.
- Memory handling: parallel degrees of freedom handling for continuous and discontinuous approximations
- Numerical schemes: Discontinuous Galerkin methods for hyperbolic systems. SUPG and Residual Distribution schemes.
- Models: Perfect gas Euler system, real gas Euler system, scalar advection, Waves equation in first order formulation, generic interface for defining space-time models from space models.
- Numerical fluxes: centered fluxes, exact Godunov' flux for linear hyperbolic systems, and Lax-Friedrich flux.
- Parallel computing: Mesh redistribution, computation of Overlap with PaMPA. Collective asynchronous communications (PaMPA based). Tests on the cluster Avakas from MCIA, and on Mésocentre de Marseille. The library was also compiled on PlaFRIM.
- C++/Fortran interface: Tests for binding fortran with C++.

5.2. COCA

Participants: Mario Ricchiuto [corresponding member], Gérard Vignoles.

COCA(CodeOxydationCompositesAutocicatrisants) is a Fortran 90 code for the simulation of the oxidation process in self-healing composite materials, developed in collaboration with the Laboratoire des Composites ThermoStructuraux in Bordeaux (UMR-5801 LCTS). This process involves the chemical oxidation of some of the matrix components of the composite, and the production of a liquid oxide that flows and fills material cracks, acting as a diffusion barrier against oxygen and thus protecting the ceramic fibers of the material. COCA simulates this process using a finite element discretization of the model equations. In its current version only transverse cracks are available. COCA makes use of PaStiX to solve the algebraic systems arising from the discretization.

5.3. RealfluidS

Participants: Dante de Santis, Gianluca Geraci, Pietro Marco Congedo, Rémi Abgrall [corresponding member].

RealfluidS is a software dedicated to the simulation of inert or reactive flows. It is also able to simulate multiphase, multimaterial, MHD flows and turbulent flows (using the SA model). There exist 2D and 3D dimensional versions. The 2D version is used to test new ideas that are later implemented in the 3D one. This software implements the more recent residual distribution schemes. The code has been parallelized with and without overlap of the domains. An Uncertainty Quantification library has been added to the software. A partitioning tool exists in the package, which uses Scotch. In the years to come, all the know-how of RealfluidS will be transferred to Aerosol.

5.4. MMG3D

Participants: Cécile Dobrzynski [corresponding member], Algiane Froehly.

MMG3D is a tetrahedral fully automatic remesher. Starting from a tetrahedral mesh, it produces quasi-uniform meshes with respect to a metric tensor field. This tensor prescribes a length and a direction for the edges, so that the resulting meshes will be anisotropic. The software is based on local mesh modifications and an anisotropic version of Delaunay kernel is implemented to insert vertices in the mesh. Moreover, MMG3D allows one to deal with rigid body motion and moving meshes. When a displacement is prescribed on a part of the boundary, a final mesh is generated such that the surface points will be moved according this displacement. MMG3D is used in particular in GAMMA for their mesh adaptation developments, but also at EPFL (maths department), Dassault Aviation, Lemma (a french SME), etc. MMG3D can be used in FreeFem++ (<http://www.freefem.org>), a free software which eases the solving of PDEs and in Gmsh (<http://geuz.org/gmsh/>). More details can be found on <http://www.math.u-bordeaux1.fr/~dobj/logiciels/mmg3d.php>.

A new version of MMG3D is under development. The big novelty of this version is the modification of the surface triangulation. A. Froehly, ingénieur in the FUI Rodin, is working on this new version.

5.5. ORComp

Participants: Pietro Marco Congedo [Corresponding member], Rémi Abgrall, Nassim Razaaly, Dante De Santis, Maria-Giovanna Rodio.

The ORComp platform is a simulation tool permitting to design an ORC cycle. Starting from the solar radiation, this platform computes the cycle providing the best performance with optimal choices of the fluid and the operating conditions. It includes RobUQ, a simulation block of the ORC cycles, the RealFluid code for the simulation of the turbine and of the heat exchanger, the software FluidProp (developed at the University of Delft) for computing the fluid thermodynamic properties.

5.6. PaMPA

Participants: Cédric Lachat, François Pellegrini [Corresponding member], Cécile Dobrzynski, Hervé Guillard [PUMAS], Laurent Hascoët [Tropics].

PaMPA (“Parallel Mesh Partitioning and Adaptation”) is a middleware library dedicated to the management of distributed meshes. Its purpose is to relieve solver writers from the tedious and error prone task of writing again and again service routines for mesh handling, data communication and exchange, remeshing, and data redistribution. It is based on a distributed data structure that represents meshes as a set of *entities* (elements, faces, edges, nodes, etc.), linked by *relations* (that is, computation dependencies).

PaMPA interfaces with Scotch for mesh redistribution, and with MMG3D for parallel remeshing of tetrahedral elements. Other sequential remeshers can be plugged in order to handle other types of elements.

Version 0.2 allows users to declare a distributed mesh, to declare values attached to the entities of the meshes (e.g. temperature attached to elements, pressures to the faces, etc.), to exchange values between overlapping entities located at the boundaries of subdomains assigned to different processors, to iterate over the relations of entities (e.g. iterate over the faces of elements), to remesh the pieces of the mesh that need to, and to redistribute evenly the remeshed mesh across the processors of the parallel architecture.

PaMPA is already used as the data structure manager for two solvers being developed at Inria: Plato and Aerosol.

5.7. PLATO

Participants: Hervé Guillard [PUMAS], Laure Combe [PUMAS,contact], Cédric Lachat, Pierre Ramet [corresponding member].

The development of Plato (“A platform for Tokamak simulation”) (<http://www-sop.inria.fr/pumas/plato.php>) is being supported by an ADT action of the D2T. Plato is a suite of data and software dedicated to the geometry and physics of Tokamaks and its main objective is to provide the Inria large scale initiative FUSION teams working with plasma fluid models with a common development tool. The construction of this platform will integrate the following developments.

1. A (small) database corresponding to axi-symmetrical solutions of the equilibrium plasma equations for realistic geometrical and magnetic configurations (ToreSupra, JET and ITER). The construction of meshes is always an important time consuming task. Plato will provide meshes and solutions corresponding to equilibrium solutions that will be used as initial data for more complex computations.
2. A set of tools for the handling, manipulation and transformation of meshes and solutions using different discretisations (P1, Q1, P3, etc)
3. Numerical templates allowing the use of 3D discretization schemes using finite element schemes in the poloidal plane and spectral Fourier or structured finite volume representations in the toroidal one.
4. Several applications (Ideal MHD and drift approximation) used in the framework of the Inria large scale initiative FUSION.

5.8. RobUQ

Participants: Pietro Marco Congedo [Corresponding member], Rémi Abgrall, Gianluca Geraci, Julie Tryoen, Nassim Razaaly.

The RobUQ platform has been conceived to solve problems in uncertainty quantification and robust design. It includes the optimization code ALGEN, and the uncertainty quantification code NISP. It includes also some methods for the computation of high-order statistics, efficient strategies for robust optimization, the Simplex2 method. Some methods are developed in partnership with the Stanford University (in the framework of the associated team AQUARIUS). Other methods are developed in the context of ANR UFO.

5.9. Scotch

Participants: François Pellegrini [corresponding member], Sébastien Fourestier.

parallel graph partitioning, parallel static mapping, parallel sparse matrix block ordering, graph repartitioning, mesh partitioning.

Scotch (<http://www.labri.fr/~pelegrin/scotch/>) is a software package for parallel and sequential sparse matrix ordering, parallel and sequential graph partitioning, as well as sequential static mapping and remapping, without and with fixed vertices, and mesh and hypergraph partitioning.

The initial purpose of Scotch was to compute high-quality static mappings of valued graphs representing parallel computations onto target architectures of arbitrary topologies. This allows the mapper to take into account the topology and heterogeneity of the target architecture in terms of processor speed and link bandwidth. This feature, which was meant for the NUMA machines of the 1980’s, has not been widely used in the past because machines in the 1990’s became UMA again thanks to hardware advances. Now, architectures become NUMA again, and these features are regaining popularity.

Version 5.0 of Scotch, released on August 2007, was the first version to comprise parallel routines. This extension, called PT-Scotch (for “Parallel Threaded Scotch”), is based on a distributed memory model, and makes use of the MPI and, optionally, Posix thread APIs. Version 5.1, released on September 2008, extended the parallel features of PT-Scotch, which can now compute graph partitions in parallel by means of a parallel recursive bipartitioning framework. Release 5.1.10 had made Scotch the first full 64-bit implementation of a general purpose graph partitioner.

Version 6.0, released on December 2012, corresponding to the 20-year anniversary of Scotch, offers many new features: static mapping with fixed vertices, static remapping, and static remapping with fixed vertices. Several critical algorithms of the formerly sequential Scotch library can now run in a multi-threaded way. All of these features will be available for the parallel PT-Scotch library in the upcoming release 6.1.

Scotch has been integrated in numerous third-party software, which indirectly contribute to its diffusion, e.g. OPENFOAM (fluid mechanics solver, OpenCFD ltd.), the CODE_ASTER LIBRE solver (thermal and mechanical analysis software developed by French state-owned electricity producer EDF), the ZOLTAN module of the TRILINOS software (SANDIA Labs), the parallel linear system solvers MUMPS (ENSEEITH/IRIT, LIP and LaBRI), SuperLUDist (U.C. Berkeley), PaStiX (LaBRI) and HIPS (LaBRI), etc. Scotch is natively available in several Linux and Unix distributions, as well as on some vendors platform (SGI, etc).

5.10. SLOWS

Participant: Mario Ricchiuto [corresponding member].

SLOWS (“*Shallow-water fLOWS*”) is a C-platform allowing the simulation of free surface shallow water flows with friction. Arbitrary bathymetries are allowed, defined either by some complex piecewise analytical expression, or by *xyz* data files, the classical Manning model for friction is used, and an Exner model is implemented for sediment transport. The equations are discretized with a residual based approach which is an adaptation of the schemes developed for aeronautics applications. Due to the inherent unsteadiness of these flows, the time discretization plays an important role. Three different approaches are available, based on conditionally depth-positivity preserving implicit schemes, or on conditionally depth-positivity preserving genuinely explicit discretizations, or on an unconditionally depth-positivity preserving space-time approach.

BAMBOO Project-Team

5. Software

5.1. AcypiCyc

Participants: Hubert Charles [EPI], Patrice Baa Puyoule [Contact, Patrice.Baa-Puyoulet@lyon.inra.fr], Stefano Colella [Contact, stefano.colella@lyon.inra.fr], Ludovic Cottret, Marie-France Sagot [EPI], Augusto Vellozo [Contact, augusto@cycadsys.org], Amélie Véron.

Database of the metabolic network of *Acyrtosiphon pisum*.

<http://acypicyc.cycadsys.org/>

5.2. ALViE

Participants: Pierluigi Crescenzi [Contact, pierluigi.crescenzi@unifi.it, ext. member EPI], Giorgio Gambosi, Roberto Grossi, Carlo Nocentini, Tommaso Papini, Walter Verdese.

ALViE is a post-mortem algorithm visualization Java environment, which is based on the interesting event paradigm. The current distribution of ALViE includes more than forty visualizations. Almost all visualizations include the representation of the corresponding algorithm C-like pseudo-code. The ALViE distribution allows a programmer to develop new algorithms with their corresponding visualization: the included Java class library, indeed, makes the creation of a visualization quite an easy task (once the interesting events have been identified).

<http://piluc.dsi.unifi.it/alvie/>

5.3. Cassis

Participants: Christian Baudet [EPI, Contact, christian.baudet@univ-lyon1.fr], Christian Gautier [EPI], Claire Lemaitre [Contact, claire.lemaitre@inria.fr], Marie-France Sagot [EPI], Eric Tannier.

Algorithm for precisely detecting genomic rearrangement breakpoints.

<http://pbil.univ-lyon1.fr/software/Cassis/>

5.4. Cravela

Participants: Ana Teresa Freitas, Nuno Mendes [Contact, ndm@kdbio.inesc-id.pt], Marie-France Sagot [EPI, Contact, marie-france.sagot@inria.fr].

Framework for the identification and evaluation of miRNA precursors (finished), targets (in development) and regulatory modules (in development).

<http://www.cravela.org/>

5.5. C3P

Participants: Frédéric Boyer, Anne Morgat [EPI, ext. member], Alain Viari [EPI, Contact, alain.viari@inria.fr].

Merging two or more graphs representing biological data (e.g. pathways, ...).

<http://www.inrialpes.fr/helix/people/viari/cccpart>

5.6. CycADS

Participants: Hubert Charles [EPI], Patrice Baa Puyoule [Contact, Patrice.Baa-Puyoulet@lyon.inra.fr], Stefano Colella [Contact, stefano.colella@lyon.inra.fr], Ludovic Cottret, Marie-France Sagot [EPI], Augusto Vellozo [Contact, augusto@cycadsys.org].

Cyc annotation database system.

<http://www.cycadsys.org/>

5.7. Gobbolino

Participants: Vicente Acuña [EPI], Etienne Birmelé [EPI, délégation], Ludovic Cottret, Pierluigi Crescenzi, Fabien Jourdan, Vincent Lacroix, Alberto Marchetti-Spaccamela [EPI, ext. member], Andrea Marino, Paulo Vieira Milreu [EPI, Contact, pvmilreu@gmail.com], Marie-France Sagot [EPI], Leen Stougie [EPI, ext. member].

Algorithm to enumerate all metabolic stories in a metabolic network given a set of metabolites of interest.
Code available on request.

5.8. kisSNP

Participants: Vincent Lacroix [EPI], Pierre Peterlongo [Contact, pierre.peterlongo@inria.fr], Nadia Pisanti, Marie-France Sagot [EPI], Nicolas Schnel.

Algorithm for identifying SNPs without a reference genome by comparing raw reads.

<http://alcovna.genouest.org/kissnp/>

5.9. kisSplice

Participants: Rayan Chikhi, Janice Kielbassa [EPI], Vincent Lacroix [Contact, EPI], Pierre Peterlongo [Contact, pierre.peterlongo@inria.fr], Gustavo Sacomoto [EPI], Marie-France Sagot [EPI], Raluca Uricaru.

Algorithm for de-novo calling alternative splicing events from RNA-seq data.

<http://alcovna.genouest.org/kisssplice/>

5.10. LASAGNE

Participants: Pierluigi Crescenzi [Contact, pierluigi.crescenzi@unifi.it, ext. member EPI], Roberto Grossi, Michel Habib, Claudio Imbrenda, Leonardo LANZI, Andrea Marino.

LASAGNE is a Java application which allows the user to compute distance measures on graphs by making a clever use either of the breadth-first search or of the Dijkstra algorithm. In particular, the current version of LASAGNE can compute the exact value of the diameter of a graph: the graph can be directed or undirected and it can be weighted or unweighted. Moreover, LASAGNE can compute an approximation of the distance distribution of an undirected unweighted graph. These two features are integrated within a graphical user interface along with other features, such as computing the maximum (strongly) connected component of a graph.

<http://amici.dsi.unifi.it/lasagne/>

5.11. MetExplore

Participants: Michael Barrett, Hubert Charles [EPI], Ludovic Cottret [Contact, Ludovic.Cottret@toulouse.inra.fr], Fabien Jourdan, Marie-France Sagot [EPI], Florence Vinson, David Wildridge.

Web server to link metabolomic experiments and genome-scale metabolic networks.

<http://metexplore.toulouse.inra.fr/metexplore/>

5.12. Migal

Participants: Julien Allali [Contact, julien.allali@labri.fr], Marie-France Sagot [EPI, Contact, marie-france.sagot@inria.fr].

RNA, tree comparison

Algorithm for comparing RNA structures.

<http://www-igm.univ-mlv.fr/~allali/logiciels/index.en.php>

5.13. MotusWEB

Participants: Ludovic Cottret, Fabien Jourdan, Vincent Lacroix [EPI, Contact, vincent.lacroix@univ-lyon1.fr], Odile Rogier, Marie-France Sagot [EPI].

Algorithm for searching and inferring coloured motifs in metabolic networks (web-based version - offers different functionalities from the downloadable version).

http://pbil.univ-lyon1.fr/software/motus_web/

5.14. Motus

Participants: Ludovic Cottret, Fabien Jourdan, Vincent Lacroix [EPI, Contact, vincent.lacroix@univ-lyon1.fr], Odile Rogier, Marie-France Sagot [EPI].

Algorithm for searching and inferring coloured motifs in undirected graphs (downloadable version - offers different functionalities from the web-based version).

<http://pbil.univ-lyon1.fr/software/motus/>

5.15. PhEVER

Participants: Christian Gautier [EPI], Vincent Lotteau, Leonor Palmeira [Contact, mlpalmeira@ulg.ac.be], Chantal Rabourdin-Combe, Simon Penel.

Database of homologous gene families built from the complete genomes of all available viruses, prokaryotes and eukaryotes and aimed at the detection of virus/virus and virus/host lateral gene transfers.

<http://pbil.univ-lyon1.fr/databases/phever/>

5.16. PepLine

Participants: Jérôme Garin, Alain Viari [EPI, Contact, alain.viari@inria.fr].

Pipeline for the high-throughput analysis of proteomic data.

<http://www.grenoble.prabi.fr/protehome/software/pepline>

5.17. Pitufu and family

Participants: Vicente Acuña [EPI], Ludovic Cottret [Contact, Ludovic.Cottret@toulouse.inra.fr], Alberto Marchetti-Spaccamela [EPI, ext. member], Paulo Vieira Milreu [EPI, Contact, pvmilreu@gmail.com], Marie-France Sagot [EPI], Leen Stougie [EPI, ext. member], Fabio Viduani-Martinez.

Algorithms to enumerate all minimal sets of precursors of target compounds in a metabolic network.

<http://sites.google.com/site/pitufosoftware/>

5.18. PSbR

Participants: Yoan Diekmann, Marie-France Sagot [EPI, Contact, marie-france.sagot@inria.fr], Eric Tannier.

Algorithm for testing the evolution and conservation of common clusters of genes.

<http://pbil.univ-lyon1.fr/members/sagot/htdocs/team/software/PSbR/>

5.19. Repseek

Participants: Guillaume Achaz [Contact, achaz@abi.snv.jussieu.fr], Eric Coissac, Alain Viari [EPI].

Finding approximate repeats in large DNA sequences.

<http://www.abi.snv.jussieu.fr/public/RepSeek/>

5.20. Smile

Participants: Laurent Marsan, Marie-France Sagot [EPI, Contact, marie-france.sagot@inria.fr].

Motif inference algorithm taking as input a set of biological sequences.

5.21. Tuiuiu

Participants: Alair Pereira do Lago, Pierre Peterlongo [Contact, pierre.peterlongo@inria.fr], Nadia Pisanti, Gustavo Sacomoto [EPI], Marie-France Sagot [EPI].

Multiple repeat search filter with edit distance.

<http://mobyale.genouest.org/cgi-bin/Mobyale/portal.py?form=tuiuiu>

5.22. UniPathway

Participants: Eric Coissac, Anne Morgat [EPI, Contact, anne.morgat@inria.fr], Alain Viari [EPI].

Database of manually curated pathways developed with the Swiss-Prot group.

<http://www.unipathway.org>

BANG Project-Team

5. Software

5.1. Software

5.1.1. Continuation of M3N

A large part of the software currently in use in the project-team was initiated and developed within former projects (MenuSin, M3N).

5.1.2. CellSys

Participants: Dirk Drasdo [correspondent], Stefan Höhme [Research Associate, University of Leipzig], Adrian Friebel [PhD student, University of Leipzig], Tim Johann [Software Engineer, University of Leipzig], Nick Jagiella [PhD student].

Computer simulation software for image analysis of tissue samples at histological scales, as well as individual cell (agent)-based models of tumour and tissue growth solved either by systems of coupled equations of motion for each individual cell or by Kinetic Monte Carlo methods [56].

The software CellSys is currently being completely reorganised to permit easier use by external and internal researchers. The idea is to progressively go open-source and offer consultancy for potential users.

BEAGLE Team

5. Software

5.1. aevol (artificial evolution)

Participants: Guillaume Beslon, Stephan Fischer, Carole Knibbe, David P Parsons, Bérénice Batut.

- Contact: Carole Knibbe (carole.knibbe@inria.fr).
- Aevol is a simulation software dedicated to the study of genome evolution. It allows to carry out *in silico* experimental evolution. Populations of digital organisms reproduce and mutate randomly, with both small mutations and large chromosomal rearrangements, in a steady or varying environment. A curve-fitting task is used to determine the fitness of the organisms and thus their rate of reproduction. The number of genes, their order, their sequences, their intergenic distances are all free to evolve.
- URL: <http://www.aevol.fr>

5.2. FluoBacTracker

Participants: Hugues Berry, David P Parsons, Magali Vangkeosay.

- contact: Hugues Berry (hugues.berry@inria.fr)
- FluoBacTracker is a software for automated quantification of bacterial cells in microscopy movies, developed in collaboration with INSERM U1001 and Paris 5 MAP (Applied Mathematics) Labs. The development (started october 2012) is supported by is a 2-year grant (ADT) funded by Inria's Technological Development Department (Sept 2012- July 2014, project name: "MultiPop"). We hope this software will be useful to all the experimental biology labs that tries to derive single-cell data from bacteria growth microscopy movies. Codeveloppers include Magali Vangkeosay (Beagle), David P Parsons (SED, Inria Grenoble) and Xiaohu Song (INSERM U1001).

5.3. Ancestral Genome Reconstructions

Participant: Eric Tannier.

- Contact: Eric Tannier (eric.tannier@inria.fr).
- We participated in the development of a series of softwares for genome organization analysis:
 - ANGES, for ANcestral GENomeS maps, is a toolkit for ordering ancestral genomic markers in chromosomes. An application note has been published in *Bioinformatics* in 2012 to advertise its first release. It is hosted at SFU in Vancouver, URL: <http://paleogenomics.irmacs.sfu.ca/ANGES/>, under a GNU license, 2012.
 - DeCo, for Detection of Co-evolution, reconstructs neighborhood relationships between genes of ancient genomes, in the presence of gene duplications and losses. It is hosted at the PRABI, the bioinformatics platform in Lyon, URL: <http://pbil.univ-lyon1.fr/software/DeCo/>, under a Cecill license, 2012.
 - DCJ2HP provides bayesian samples of rearrangements scenarios between 2 genomes. It is hosted at the Renyi Institute in Budapest, URL <http://www.renyi.hu/~miklosi/DCJ2HP/>

BIGS Project-Team

5. Software

5.1. Light diffusion into tissues

We are currently considering the possibility to implement our Matlab algorithms concerning light diffusion into tissues into the Matlab toolbox *Contsid*, developed by the System Identification team of the CRAN (<http://www.iris.cran.uhp-nancy.fr/contsid/>).

5.2. Online data analysis

A R package performing most of the methods of factorial analysis in an online way is under development by R. Bar and J-M. Monnez. Starting from a simulated data flow, the main goal of the program is to perform online factorial analyses (Principal Component Analyses, Canonical Correlation Analysis, Canonical Discriminant Analysis, Correspondence Analysis). Data are supposed to be independent and identically distributed observations of a random vector (whose distribution is a priori unknown). Defining stochastic approximation processes, the procedure is adaptative in the sense that the results of the analyses are updated recursively each time that a new data is taken into account.

From a theoretical point of view, the i.i.d case has been recently extended to the case of an expectation and/or covariance matrix of the random vector varying with time. We plan to include these improvements into our software.

5.3. Socio-economic index

A R package called *SesIndexCreaoR* has been written by B. Lalloué and J-M. Monnez in order to implement our socio-economic index for health inequalities. The version 1.0 of this package is currently freely available on the website of the Equit'Area project: http://www.equitarea.org/documents/packages_1.0-0/. It contains the functions needed to run the procedure (either integrally or partially) and obtain the corresponding SES index. The user may also create categories of this index with different methods (hierarchical clustering with or without k-nearest neighbors, quantiles, or intervals) and generate automatic reports of the results. Visualization and plotting functions are provided in the package.

BIOCORE Project-Team

5. Software

5.1. Supervision software

We are developing a software for the supervision of bioreactors: this platform, named ODIN, has been built for the smart management of bioreactors (data acquisition, fault diagnosis, automatic control algorithm,...). This software was developed in C++ and uses a Scilab engine to run the advanced algorithms developed within BIOCORE. It has been implemented and validated with four different applications.

BIPOP Project-Team

5. Software

5.1. Nonsmooth dynamics: Siconos

Participants: Vincent Acary, Maurice Bremond, Olivier Bonnefon.

In the framework of the European project Siconos, Bipop was the leader of the Work Package 2 (WP2), dedicated to the numerical methods and the software design for nonsmooth dynamical systems. The aim of this work is to provide a common platform for the simulation, modeling, analysis and control of abstract nonsmooth dynamical systems. Besides usual quality attributes for scientific computing software, we want to provide a common framework for various scientific fields, to be able to rely on the existing developments (numerical algorithms, description and modeling software), to support exchanges and comparisons of methods, to disseminate the know-how to other fields of research and industry, and to take into account the diversity of users (end-users, algorithm developers, framework builders) in building expert interfaces in Python and end-user front-end through Scilab.

After the requirement elicitation phase, the Siconos Software project has been divided into 5 work packages which are identified to software products:

1. SICONOS/NUMERICS This library contains a set of numerical algorithms, already well identified, to solve non smooth dynamical systems. This library is written in low-level languages (C,F77) in order to ensure numerical efficiency and the use of standard libraries (Blas, Lapack, ...)
2. SICONOS/KERNEL This module is an object-oriented structure (C++) for the modeling and the simulation of abstract dynamical systems. It provides the users with a set of classes to describe their nonsmooth dynamical system (dynamical systems, interconnections, nonsmooth laws, ...) and to perform a numerical time integration and solving.
3. SICONOS/FRONT-END. This module is mainly an auto-generated wrapper in Python which provides a user-friendly interface to the Siconos libraries. A scilab interface is also provided in the Front-End module.
4. SICONOS/CONTROL This part is devoted to the implementation of control strategies of non smooth dynamical systems.
5. SICONOS/MULTIBODY. This part is dedicated to the modeling and the simulation of multi-body systems with 3D contacts, impacts and Coulomb's friction. It uses the Siconos/Kernel as simulation engine but relies on an industrial CAD library (OpenCascade and pythonOCC) to deal with complex body geometries and to compute the contact locations and distances.

Further informations may be found at <http://siconos.gforge.inria.fr/>

5.2. Humanoid motion analysis and simulation

Participant: Pierre-Brice Wieber.

The HuMAnS toolbox offers tools for the modelling, control and analysis of humanoid motion, be it of a robot or a human. It is a C/C++/Scilab/Maple-based set of integrated tools for the generation of dynamical models of articulated bodies with unilateral contact and friction, their simulation with an event-driven integration scheme, their 3D visualization, the computation of stability measures, optimal positions and trajectories, the generation of control laws and observers, the reconstruction of movements from different sensing systems.

5.3. AMELIF

Participants: Pierre-Brice Wieber, François Keith.

The AMELIF framework is an integrative framework that proposes an API for the representation and simulation of virtual scenes including articulated bodies. AMELIF was devised to realize interactive scenario studies with haptic feedback while providing an interface enabling fast and general prototyping of humanoids (avatars or robots). It is entirely developed in C++ and is cross-platform. The framework is articulated around a core library, upon which several modules have been developed for collision detection, dynamic simulation (contact handling in a time stepping scheme), 3D rendering, haptic interaction, posture generation. This framework is developed mostly at the CNRS/AIST UMI JRL, but we started using it in the Bipop team and therefore started contributing actively to its development.

5.4. Optimization

Participant: Claude Lemaréchal.

Essentially two possibilities exist to distribute our optimization software: library programs (say Modulopt codes), communicated either freely or not, depending on what they are used for, and on the other hand specific software, developed for a given application.

The following optimization codes have been developed in the framework of the former Promath project. They are generally available at <http://www-rocq.inria.fr/~gilbert/modulopt/>; M1QN3 is also distributed under GPL.

5.4.1. Code M1QN3

Optimization without constraints for problems with many variables ($n \geq 10^3$, has been used for $n = 10^6$). Technically, uses a limited-memory BFGS algorithm with Wolfe's line-search (see Chap. 4 of [3] for the terminology).

5.4.2. Code M2QN1

Optimization with simple bound-constraints for (small) problems: D is a parallelootope in \mathbb{R}^n . Uses BFGS with Wolfe's line-search and active-set strategy.

5.4.3. Code NICV2

Minimization without constraints of a convex nonsmooth function by a proximal bundle method (Chap. XV of [10], Chap. 9 of [3]).

5.4.4. Modulopt

In addition to codes such as above, the Modulopt library contains application problems, synthetic or from the real world. It is a field for experimentation, functioning both ways: to assess a new algorithm on a set of test-problems, or to select among several codes one best suited to a given problem.

5.5. MECHE: Simulation of fibrous materials

Participants: Florence Bertails-Descoubes, Alexandre Derouet-Jourdan, Romain Casati, Laurence Boissieux.

The software MECHE was essentially developed during the MECHE ADT (2009-2011), for simulating the dynamics of assemblies of thin rods (such as hair), subject to contact and friction. Currently, this software is extensively used by two PhD students (A. Derouet-Jourdan and R. Casati) and continues to be enriched with new rod models and inversion modules. This software combines a panel of well-accepted models for rods (ranging from reduced coordinates to maximal coordinates models, and including models recently developed by some members of the group) with classical as well as innovative schemes for solving the problem of frictional contact (incorporating the most recent results of the group, as well as the new contact solver we published in [8]). The aim of this software is twofold: first, to compare and analyze the performance of nonsmooth schemes for the frictional contact problem, in terms of realism (capture of dry friction, typically), robustness, and computational efficiency. A first study of this kind was conducted in 2010-2011 onto the different rod models that were available in the software. New studies are planned for evaluating further rod models. Second, we believe such a software will help us understand the behavior of a fibrous material (such as hair) through virtual experiments, thanks to which we hope to identify and understand some important

emergent phenomena. A careful validation study against experiments started to be conducted in 2011 in collaboration with physicists from L'Oréal. Once this discrete elements model will be fully validated, our ultimate goal would be to build a continuous macroscopic model for the hair medium relying on nonsmooth laws. The core of this software was transferred to L'Oréal in 2011.

BONSAI Project-Team

5. Software

5.1. YASS – Local homology search

Actively maintained.

Software self-assessment following the mechanisms provided by Inria Evaluation Committee for software evaluation: **A-4, SO-3, SM-2, EM-3, SDL-4, DA-4, CD-4, MS-4, TPM-4**

Software web site : <http://bioinfo.lifl.fr/yass/>

Licence: GPL

YASS is a software devoted to the classical problem of genomic pairwise alignment, and use most of our knowledge to design and implement efficient seeding techniques these last years. It is frequently used, it always receives more than 300 web queries per month (excluding local queries), and is also frequently downloaded and cited.

5.2. Carnac – RNA structure prediction

Actively maintained.

Software self-assessment: **A-4, SO-3, SM-2, EM-3, SDL-4, DA-4, CD-4, MS-4, TPM-4**

Software web site : <http://bioinfo.lifl.fr/carnac/>

Licence: Cecill

The CARNAC program is for RNA structure prediction by comparative analysis. The web interface also offers 2D visualisation tools and alignment functionalities with Gardenia. It has proven to be very fast and very specific compared to its competitors [19].

5.3. TFM-Explorer – Identification and analysis of transcription factor binding sites

Actively maintained.

Software self-assessment: **A-4, SO-3, SM-2, EM-3, SDL-4, DA-4, CD-4, MS-4, TPM-4**

Software web site : <http://bioinfo.lifl.fr/TFM/>

Licence: GPL

The TFM suite is a set of tools for analysis of transcription factor binding sites modeled by Position Weight Matrices. In this suite, the TFM-EXPLORER tool is designed to analyze regulatory regions of eukaryotic genomes using comparative genomics and local over-representation.

5.4. Regliss – RNA locally optimal structures

Actively maintained.

Software self-assessment: **A-2, SO-4, SM-2, EM-2, SDL-4, DA-4, CD-4, MS-4, TPM-4**

Software web site : <http://bioinfo.lifl.fr/RNA/regliss/>

REGLISS is a tool that studies the energy landscape of a given RNA sequence by generating all locally optimal structures, that are maximal thermodynamically stable structures.

5.5. RNAspace – A platform for noncoding RNA annotation

Actively developped.

Software self-assessment: **A-5, SO-3, SM-3-up4, EM-2-up3, SDL-4**, DA-4, CD-4, MS-4, TPM-4

Software web site : <http://www.rnaspaces.org/>

Licence: GPL

RNAspaces is a national collaborative initiative conducted with Genopole Midi-Pyrénées and originally supported by IBISA¹. The goal is to develop an open source platform for structural and functional noncoding RNA annotation in genomes (see Section 6.2): <http://www.rnaspaces.org>. The project will be pursued within France Génomique (see Section 7.2.1).

5.6. CGseq – A toolbox for comparative analysis

Actively maintained.

Software self-assessment: **A-4, SO-3, SM-2, EM-3, SDL-4**, DA-4, CD-4, MS-4, TPM-4

Software web site : <http://bioinfo.lifl.fr/CGseq/>

Licence: GPL

CG-seq is a toolbox for identifying functional regions in a genomic sequence by comparative analysis using multispecies comparison.

5.7. SortMeRNA – Metatranscriptome classification

Actively developed.

Software self-assessment: **A-4, SO-3, SM-2, EM-3, SDL-4**, DA-4, CD-4, MS-4, TPM-4

Software web site: <http://bioinfo.lifl.fr/RNA/sortmerna>

Licence: GPL

SortMeRNA is a software designed to rapidly filter ribosomal RNA fragments from metatranscriptomic data produced by next-generation sequencers. It is capable of handling large RNA databases and sorting out all fragments matching to the database with high accuracy and specificity.

5.8. Biomanycores.org – A community for bioinformatics on manycore processors

Actively developed.

Software self-assessment: **A-3, SO-2, SM-3, EM-3down2, SDL-4up5**, OC-4 (DA-4, CD-4, MS-4, TPM-4)

Software web site : <http://biomanycores.org/>

Manycore architectures are an emerging field of research full of promises for parallel bioinformatics. However the usage of GPUs is not so widespread in the end-user bioinformatics community. The goal of the `biomanycores.org` project is to gather open-source CUDA and OpenCL parallel codes and to provide easy installation, benchmarking, and interoperability. The last point includes interfaces to popular frameworks such as Biopython, BioPerl and BioJava.

The development of Biomanycores was supported by a national ADT² between BONSAI, SYMBIOSE (CRI Rennes) and DOLPHIN (CRI Lille), from October 2010 to October 2012. This ADT led to the hiring of J.-F. Berthelot (IJD) who completely redesigned the existing code and added more applications. Biomanycores has now a comprehensive developer and user documentation, large test suites and continuous integration. In June 2012, the project has been presented during a workshop dedicated to manycore programming (see Section 8.1).

¹IBISA is a French consortium for evaluating and funding national technological platforms in life sciences.

²ADT (Action for Technological Development) is an Inria internal call

5.9. Norine – A resource for nonribosomal peptides

Actively maintained.

Software self-assessment: **A-5, SO-3, SM-3-up4, EM-2-up3, SDL-4**, DA-4, CD-4, MS-4, TPM-4

Software web site : <http://bioinfo.lifl.fr/norine/> Norine is a public computational resource that contains a database of NRPs with a web interface and dedicated tools, such as a 2D graph viewer and editor for peptides or comparison of NRPs. Norine was created and is maintained by members of BONSAI team, in tight collaboration with members of the ProBioGEM lab, a microbial laboratory of Lille1 University. Since its creation in 2006, Norine has gained an international recognition as the unique database dedicated to non-ribosomal peptides because of its high quality and manually curated annotations, and has been selected by wwPDB as a reference database. It is queried from all around the world by biologists or biochemists. It receives more than 3000 queries per month. Norine main users come for 13% from the United States of America, for 12% from the United Kingdom, for 5% from China or for 4% from Germany where renowned biology laboratories work on nonribosomal peptides (NRPs) or on their synthetases.

5.10. GkArrays – Indexing high throughput sequencer reads

Actively maintained.

Software self-assessment: **A-3, SO-3, SM-3, EM-2, SDL-4**, DA-4, CD-4, MS-4, TPM-3

Software web site : <http://crac.gforge.inria.fr/gkarrays/>

Objective : Gk-Arrays is a C++ library specifically dedicated to indexing reads produced by high-throughput sequencers. This index allows to answer queries centred on reads. It also takes benefits from the input specificity to lower space consumption.

This library is the result of a collaboration with N. Philippe and T. Commes (IGH laboratory, Montpellier), M. Léonard and T. Lecroq (LITIS laboratory, Rouen) and É. Rivals (LIRMM laboratory, Montpellier).

BYMOORE Exploratory Action

3. Software

3.1. Software

- **IODC:** Framework for implementing transparent iterative optimization in data centers, see Result [4.5](#).
- **P & R for neuromorphic accelerator:** A place and route software which maps a neural network graph on an analog neural network hardware.
- **Spiking neural network model:** A model for investigating and comparing various unsupervised neural network models.
- **Component programming:** A framework for programming heterogeneous multi-cores. The framework is compatible with a broad range of accelerators, including accelerators with distinct memory address spaces.

We first list the results corresponding to hardware accelerators (especially neural network accelerators), and then results corresponding to finishing, ALCHEMY (former Inria project)-related, research.

CAD Team

5. Software

5.1. Softwares

1. Tsinghua University, Inria. Software: Modeling Software 2010.
2. Inria, Tsinghua University. Prototype: Tool for Curve Projection on Surfaces 2011.
3. Tsinghua University & Inria. Software: Spline Software Tool 2012.
4. Inria, Tsinghua University. Software: Mesh Segmentation Tool 2012.

We have developed various prototype software but, currently, they are not distributed and used only within the project. Regarding software, we know (see the last LIAMA evaluation report) that we should consider disseminating some of its codes more widely. Using established libraries may improve the impact of some of the results. We did not do it, due to the fast turn over of students in the Chinese team and the lack of Manpower and know-how in the French part.

CAGIRE Team

5. Software

5.1. AeroSol

Participants: Dragan Amenga-Mbengoué [Bacchus], Damien Genet [Bacchus], Maxime Mogé, Francois Pellegrini [Bacchus], Vincent Perrier [correspondant], Francois Rué [Bacchus], Mario Ricchiuto [Bacchus].

The software AeroSol is jointly developed in the team Bacchus and the team Cagire. It is a high order finite element library written in C++. The code design has been carried for being able to perform efficient computations, with continuous and discontinuous finite elements methods on hybrid and possibly curvilinear meshes. The distribution of the unknowns is made with the software PaMPA, developed within the team Bacchus and the team Pumas. This year, Dragan Amenga-Mbengoué was recruited on the ANR Realfluids, and François Rué (Service Experimentation et Développement) joined the team Bacchus for working on Aerosol.

At the end of 2011, Aerosol had the following features

- **development environment** use of CMake for compilation, CTest for automatic tests and memory checking, lcov and gcov for code coverage reports.
- **In/Out** link with the XML library for handling with parameter files. Reader for GMSH, and writer on the VTK-ASCII legacy format.
- **Quadrature formula** up to 11th order for Lines, Quadrangles, Hexaedra, Pyramids, Prisms, up to 14th order for tetrahedron, up to 21st order for triangles.
- **Finite elements** up to fourth degree for Lagrange finite elements on lines, triangles and quadrangles.
- **Geometry** elementary geometrical functions for first order lines, triangles, quadrangles.
- **Time iteration** explicit Runge-Kutta up to fourth order, explicit Strong Stability Preserving schemes up to third order.
- **Linear Solvers** link with the external linear solver UMFPack.
- **Memory handling** discontinuous and continuous discretizations based on PaMPA for triangular and quadrangular meshes.
- **Numerical schemes** continuous Galerkin method for the Laplace problem (up to fifth order) with non consistent time iteration or with direct matrix inversion. Scalar stabilized residual distribution schemes with explicit Euler time iteration have been implemented for steady problems.

This year, the following features were added

- **development environment** development of a CDash server for collecting the unitary tests and memory checking. Beginning of the development of an interface for functional tests.
- **General structure** Parts of the code were abstracted in order to allow for parallel development: Linear solvers (template type abstraction for generic linear solver external library), Generic integrator classes (integrating on elements, on faces with handling neighbour elements, or for working on Lagrange points of a given element), models (template abstraction for generic hyperbolic systems), equations of state (template-based abstraction for a generic equation of state).
- **In/Out** Parallel GMSH reader, cell and point centered visualization based on VTK-legacy formats. XML paraview files on unstructured meshes (vtu), and parallel XML based files (pvtu).
- **Quadrature formula** Gauss-Lobatto type quadrature formula.
- **Finite elements** Hierarchical orthogonal finite element basis on lines, triangles (with Dubiner transform). Finite element basis that are interpolation basis on Gauss-Legendre points for lines, quadrangles, and hexaedra. Lagrange, and Hierarchical orthogonal finite elements basis for hexaedra, prisms and tetrahedra.

- **Geometry** elementary geometrical functions for first order three dimensional shapes: hexaedra, prisms, and tetrahedra.
- **Time iteration** CFL time stepping, optimized CFL time schemes: SSP(2,3) and SSP (3,4)
- **Linear Solvers** Internal solver for diagonal matrices. Link with the external solvers PETSc and MUMPS.
- **Memory handling** parallel degrees of freedom handling for continuous and discontinuous approximations
- **Numerical schemes** Discontinuous Galerkin methods for hyperbolic systems. SUPG and Residual Distribution schemes.
- **Models** Perfect gas Euler system, real gas Euler system, scalar advection, Waves equation in first order formulation, generic interface for defining space-time models from space models.
- **Numerical fluxes** centered fluxes, exact Godunov' flux for linear hyperbolic systems, and Lax-Friedrich flux.
- **Parallel computing** Mesh redistribution, computation of Overlap with PaMPA. collective asynchronous communications (PaMPA based). Tests on the cluster Avakas from MCIA, and on Mésocentre de Marseille, and PlaFRIM.
- **C++/Fortran interface** Tests for binding fortran with C++.

CAIRN Project-Team

5. Software

5.1. Panorama

With the ever raising complexity of embedded applications and platforms, the need for efficient and customizable compilation flows is stronger than ever. This need of flexibility is even stronger when it comes to research compiler infrastructures that are necessary to gather quantitative evidence of the performance/energy or cost benefits obtained through the use of reconfigurable platforms. From a compiler point of view, the challenges exposed by these complex reconfigurable platforms are quite significant, since they require the compiler to extract and to expose an important amount of coarse and/or fine grain parallelism, to take complex resource constraints into consideration while providing efficient memory hierarchy and power management.

Because they are geared toward industrial use, production compiler infrastructures do not offer the level of flexibility and productivity that is required for compiler and CAD tool prototyping. To address this issue, we have designed an extensible source-to-source compiler infrastructure that takes advantage of leading edge model-driven object-oriented software engineering principles and technologies.

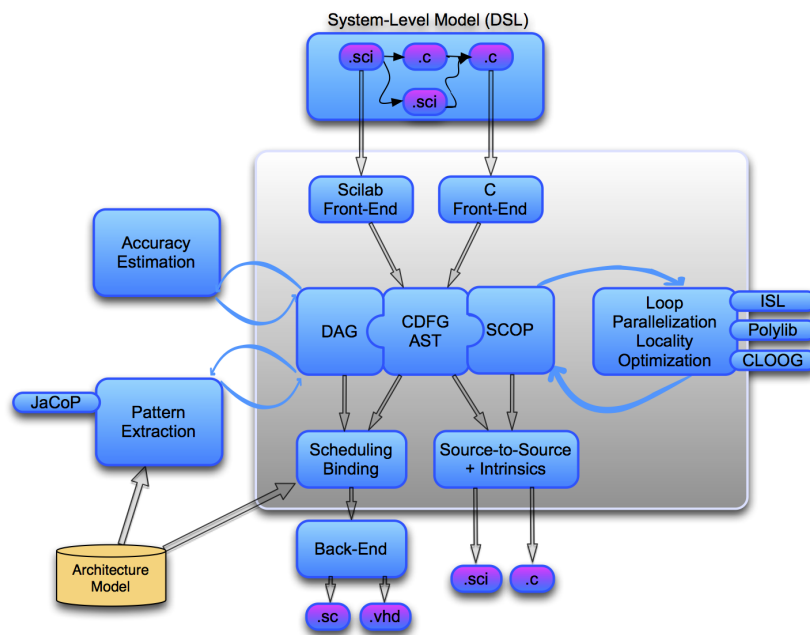


Figure 2. CAIRN's general software development framework.

Figure 2 shows the global framework that is being developed in the group. Our compiler flow mixes several types of intermediate representations. The baseline representation is a simple tree-based model enriched with control flow information. This model is mainly used to support our source-to-source flow, and serves as the backbone for the infrastructure. We use the extensibility of the framework to provide more advanced representations along with their corresponding optimizations and code generation plug-ins. For example,

for our pattern selection and accuracy estimation tools, we use a data dependence graph model in all basic blocks instead of the tree model. Similarly, to enable polyhedral based program transformations and analysis, we introduced a specific representation for affine control loops that we use to derive a Polyhedral Reduced Dependence Graph (PRDG). Our current flow assumes that the application is specified as a system level hierarchy of communicating tasks, where each task is expressed using C (or Scilab in the short future), and where the system level representation and the target platform model are defined using Domain Specific Languages (DSL).

Gecos (Generic Compiler Suite) is the main backbone of CAIRN's flow. It is an open source Eclipse-based flexible compiler infrastructure developed for fast prototyping of complex compiler passes. Gecos is a 100% Java based implementation and is based on modern software engineering practices such as Eclipse plugin or model-driven software engineering with EMF (Eclipse Modeling Framework). As of today, our flow offers the following features:

- An automatic floating-point to fixed-point conversion flow (for HLS and embedded processors). **ID.Fix** is an infrastructure for the automatic transformation of software code aiming at the conversion of floating-point data types into a fixed-point representation. <http://idfix.gforge.inria.fr>.
- A polyhedral-based loop transformation and parallelization engine (mostly targeted at HLS). <http://gecos.gforge.inria.fr>. It was used for source-to-source transformations in the context of Nano2012 projects in collaboration with STMicroelectronics.
- A custom instruction extraction flow (for ASIP and dynamically reconfigurable architectures). **Durase** and **UPaK** are developed for the compilation and the synthesis targeting reconfigurable platforms and the automatic synthesis of application specific processor extensions. They use advanced technologies, such as graph matching and graph merging together with constraint programming methods.
- Several back-ends to enable the generation of VHDL for specialized or reconfigurable IPs, and SystemC for simulation purposes (e.g. fixed-point simulations).

5.2. Gecos

Participants: Steven Derrien [corresponding author], Nicolas Simon, Maxime Naullet, Antoine Floc'h, Antoine Morvan, Clément Guy.

Keywords: source-to-source compiler, model-driven software engineering, retargetable compilation.

The Gecos (Generic Compiler Suite) project is a source-to-source compiler infrastructure developed in the CAIRN group since 2004. It was designed to enable fast prototyping of program analysis and transformation and it aims the hardware synthesis and retargetable compilation domains.

Gecos is 100% Java based and takes advantage of modern model driven software engineering practices. It uses the Eclipse Modeling Framework (EMF) as an underlying infrastructure and takes benefits of its features to make it easily extensible. Gecos is open-source and is hosted on the Inria gforge at <http://gecos.gforge.inria.fr>.

The Gecos infrastructure is still under very active development, and serves as a backbone infrastructure to projects of the group (project S2S4HSL, ID.FIX). Part of the framework is jointly developed with Colorado State University and since 2012 it is used in the context of the ALMA European project.

Development in Gecos in 2012 have mostly focused on the polyhedral loop transformation engine and its use for hardware synthesis. As a part of the ALMA project, significant efforts are also being made to develop a coarse-grain parallelization engine targeting a distributed memory machine model.

5.3. ID.Fix: Infrastructure for the Design of Fixed-point Systems

Participants: Daniel Menard, Olivier Sentieys [corresponding author], Romuald Rocher, Nicolas Simon.

Keywords: fixed-point arithmetic, source-to-source code transformation, accuracy optimization, dynamic range evaluation

The different techniques proposed by the team for fixed-point conversion are implemented on the ID.Fix infrastructure. The application is described with a C code using floating-point data types and different pragmas, used to specify parameters (dynamic, input/output word-length, delay operations) for the fixed-point conversion. This tool determines and optimizes the fixed-point specification and then, generates a C code using fixed-point data types (`ac_fixed`) from Mentor Graphics. The infrastructure is made-up of two main modules corresponding to the fixed-point conversion (ID.Fix-Conv) and the accuracy evaluation (ID.Fix-Eval)

The different developments carried-out in 2012 allowed us to obtain a fixed-point conversion tool handling functions, conditional structures and repetitive structures having a fixed number of iterations during time. New optimization algorithms have been added. A simulator has been created to verify the results from our analytical approach. For the accuracy evaluation (Acc.Eval), conditional structures and correlation between noise sources have been considered. Some optimizations have been implemented to reduce the computing time and the division operator treatment has been integrated. A tutorial has also been created to install and use this tool.

The development of this tool has been achieved thanks to a University of Rennes graduate engineer from November 2011 in the context of DEFIS ANR project and different students during their training period.

5.4. UPaK: Abstract Unified Pattern-Based Synthesis Kernel for Hardware and Software Systems

Participants: Christophe Wolinski [corresponding author], François Charot, Antoine Floc'h.

Keywords: compilation for reconfigurable systems, pattern extraction, constraint-based programming.

We are developing (with strong collaboration of Lund University, Sweden and Queensland University, Australia) UPaK *Abstract Unified Pattern Based Synthesis Kernel for Hardware and Software Systems* [123]. The preliminary experimental results obtained by the UPaK system show that the methods employed in the systems enable a high coverage of application graphs with small quantities of patterns. Moreover, high application execution speed-ups are ensured, both for sequential and parallel application execution with processor extensions implementing the selected patterns. UPaK is one of the basis for our research on compilation and synthesis for reconfigurable platforms. It is based on the HCDG representation of the Polychrony software designed at Inria-Rennes in the project-team Espresso.

5.5. DURASE: Automatic Synthesis of Application-Specific Processor Extensions

Participants: Christophe Wolinski [corresponding author], François Charot, Antoine Floc'h.

Keywords: compilation for reconfigurable systems, instruction-set extension, pattern extraction, graph covering, constraint-based programming.

We are developing a framework enabling the automatic synthesis of application specific processor extensions. It uses advanced technologies, such as algorithms for graph matching and graph merging together with constraints programming methods. The framework is organized around several modules.

- CoSaP: Constraint Satisfaction Problem. The goal of CoSaP is to decouple the statement of a constraint satisfaction problem from the solver used to solve it. The CoSaP model is an Eclipse plugin described using EMF to take advantage of the automatic code generation and of various EMF tools.
- HCDG: Hierarchical Conditional Dependency Graph. HCDG is an intermediate representation mixing control and data flow in a single acyclic representation. The control flow is represented as hierarchical guards specifying the execution or the definition conditions of nodes. It can be used in the Gecos compilation framework via a specific pass which translates a CDFG representation into an HCDG.

- **Patterns:** Flexible tools for identification of computational pattern in a graph and graph covering. These tools model the concept of pattern in a graph and provide generic algorithms for the identification of pattern and the covering of a graph. The following sub-problems are addressed: (sub)-graphs isomorphism, patterns generation under constraints, covering of a graph using a library of patterns. Most of the implemented algorithms use constraints programming and rely on the CoSaP module to solve the optimization problem.

5.6. PowWow: Power Optimized Hardware and Software Framework for Wireless Motes (AP-L-10-01)

Participants: Olivier Sentieys [corresponding author], Olivier Berder, Arnaud Carer, Steven Derrien.

Keywords: Wireless Sensor Networks, Low Power, Preamble Sampling MAC Protocol, Hardware and Software Platform

PowWow is an open-source hardware and software platform designed to handle wireless sensor network (WSN) protocols and related applications. Based on an optimized preamble sampling medium access (MAC) protocol, geographical routing and protothread library, PowWow requires a lighter hardware system than Zigbee [86] to be processed (memory usage including application is less than 10kb). Therefore, network lifetime is increased and price per node is significantly decreased.

CAIRN's hardware platform (see Figure 3) is composed of:

- The motherboard, designed to reduce power consumption of sensor nodes, embeds an MSP430 microcontroller and all needed components to process PowWow protocol except radio chip. JTAG, RS232, and I2C interfaces are available on this board.
- The radio chip daughter board is currently based on a TI CC2420.
- The coprocessing daughter board includes a low-power FPGA which allows for hardware acceleration for some PowWow features and also includes dynamic voltage scaling features to increase power efficiency. The current version of PowWow integrates an Actel IGLOO AGL250 FPGA and a programmable DC-DC converter. We have shown that gains in energy of up to 700 can be obtained by using FPGA acceleration on functions like CRC-32 or error detection with regards to a software implementation on the MSP430.
- Finally, a last daughter board is dedicated to energy harvesting techniques. Based on the energy management component LTC3108 from Linear Technologies, the board can be configured with several types of stored energy (batteries, micro-batteries, super-capacitors) and several types of energy sources (a small solar panel to recover photovoltaic energy, a piezoelectric sensor for mechanical energy and a Peltier thermal energy sensor).

PowWow distribution also includes a generic software architecture using event-driven programming and organized into protocol layers (PHY, MAC, LINK, NET and APP). The software is based on Contiki [102], and more precisely on the Protothread library which provides a sequential control flow without complex state machines or full multi-threading.

To optimize the network regarding a particular application and to define a global strategy to reduce energy, PowWow offers the following extra tools: over-the-air reprogramming (and soon reconfiguration), analytical power estimation based on software profiling and power measurements, a dedicated network analyzer to probe and fix transmissions errors in the network. More information can be found at <http://powwow.gforge.inria.fr>.

5.7. SoCLib: Open Platform for Virtual Prototyping of Multi-Processors System on Chip

Participants: François Charot [corresponding author], Laurent Perraudau.

Keywords: SoC modeling, SystemC simulation model



Figure 3. CAIRN's PowWow motherboard with radio and energy-harvesting boards connected

SoCLib is an open platform for virtual prototyping of multi-processors system on chip (MP-SoC) developed in the framework of the SoCLib ANR project. The core of the platform is a library of SystemC simulation models for virtual components (IP cores), with a guaranteed path to silicon. All simulation models are written in SystemC, and can be simulated with the standard SystemC simulation environment distributed by the OSCI organization. Two types of models are available for each IP-core: CABA (Cycle Accurate / Bit Accurate), and TLM-DT (Transaction Level Modeling with Distributed Time). All simulation models are distributed as free software. We have developed the simulation model of the NIOSII processor, of the Altera Avalon interconnect, and of the TMS320C62 DSP processor from Texas Instruments. Find more information on its dedicated web page: <http://www.soclib.fr>.

CALVI Project-Team

5. Software

5.1. SeLaLib

Participants: Raphaël Blanchard, Edwin Chacon Golcher, Samuel De Santis, Aliou Diouf, Pierre Navaro, Morgane Bergot, Emmanuel Frénod, Philippe Helluy, Sever Hirstoaga, Michel Mehrenberger, Laurent Navoret, Eric Sonnendrücker.

Under the 'Fusion' large scale initiative, we have continued our work in the development of the ADT Selalib (the Semi-Lagrangian Library), now finishing its second year. This library provides building blocks for the development of numerical simulations for the solution of the fundamental equation of plasma physics: the Vlasov equation. In this context we have continued to add new modules improved interfaces and implemented 'continuous integration' software development techniques to improve code robustness and portability. Furthermore, we continue to involve other researchers within France and abroad to aid in the further development of this software product.

One of the aims of the ADT is to provide numerical building blocks for the GYSELA code developed at CEA Cadarache in collaboration with the Calvi project-team. GYSELA is used by physicists for simulating the development of turbulence in magnetic fusion plasmas in particular in view of the ITER project.

5.2. CLAC

Participants: Anaïs Crestetto, Philippe Helluy.

The objective of the three-dimensional parallel software CM2 (Code Multiéchelle Multiphysique) software is to implement a general solver for hyperbolic conservation laws. It is for instance able to solve the MHD model. CLAC is a C++ OpenCL/MPI based library derived from algorithms and ideas developed in CM2. CLAC means "Compute Language Approximation of Conservation laws".

It is clear now that a future supercomputer will be made of a collection of thousands of interconnected multicore processors. Globally it appears as a classical distributed memory MIMD machine. But at a lower level, each of the multicore processors is itself made of a shared memory MIMD unit (a few classical CPU cores) and a SIMD unit (a GPU). When designing new algorithms, it is important to adapt them for this architecture. Our philosophy will be to program our algorithms in such a way that they can be run efficiently on this kind of computers. Practically, we will use the MPI library for managing the high level parallelism, while the OpenCL library will efficiently operate the low level parallelism.

We have invested for several years now into scientific computing on GPU, using the open standard OpenCL (Open Computing Language). With Anaïs Crestetto, who is preparing a PhD in the CALVI project, we were recently awarded a prize in the international AMD OpenCL innovation challenge thanks. We have developed an OpenCL 2D Vlasov-Maxwell solver, coupling a PIC and a DG algorithms, which fully runs on a GPU. OpenCL is a very interesting tool because it is an open standard now available on almost all brands of multicore processors and GPU. The same parallel program can run on a GPU or a multicore processor without modification.

CLAC is written in C++, which is almost mandatory, because we use the OpenCL library. It also uses the MPI paradigm and is thus able to run on a cluster of GPU. CLAC is also inside a collaboration with a Strasbourg SME, AxesSim, which develops software for electromagnetic simulations. Thomas Strub, who is employed in AxesSim with a CIFRE position, is doing his Ph. D. on the conception and the development of CLAC applied to electromagnetic problems.

Because of the envisaged applications of CLAC, which may be either academical or commercial, it is necessary to conceive a modular framework. The heart of the library is made of generic parallel algorithms for solving conservation laws. The parallelism can be both fine grain (oriented towards GPU and multicore processors) and large grain (oriented towards GPU clusters). The separated modules allow managing the meshes and some specific applications. In this way, it is possible to isolate parts that can be protected by trade secret.

CAMUS Team

5. Software

5.1. PolyLib

PolyLib ⁸ is a C library of polyhedral functions, that can manipulate unions of rational polyhedra of any dimension, through the following operations: intersection, difference, union, convex hull, simplify, image and preimage. It was the first to provide an implementation of the computation of parametric vertices of a parametric polyhedron, and the computation of an Ehrhart polynomial (expressing the number of integer points contained in a parametric polytope) based on an interpolation method.

It is used by an important community of researchers (in France and the rest of the world) in the area of compilation and optimization using the polyhedral model. Vincent Loechner is the maintainer of this software. It is distributed under GNU General Public License version 3 or later, and it has a Debian package maintained by Serge Guelton (Symbiose Projet, IRISA).

5.2. ZPolyTrans

ZPolyTrans ⁹ is a C library and a set of executables, that permits to compute the integer transformation of a union of parametric \mathbb{Z} -polyhedra (the intersection between lattices and parametric polyhedra), as a union of parametric \mathbb{Z} -polyhedra. The number of integer points of the result can also be computed. It is build upon PolyLib and Barvinok library. This work is based on some theoretical results obtained by Rachid Seghir and Vincent Loechner [15].

It allows for example to compute the number of solutions of a Presburger formula by eliminating existential integer variables, or to compute the number of different data accessed by some array accesses contained in an affine parametric loop nest.

The authors of this software are Rachid Seghir (Univ. Batna, Algeria) and Vincent Loechner. It is distributed under GNU General Public License version 3 or later.

5.3. NLR

Participant: Alain Ketterlin.

We have developed a program implementing our loop-nest recognition algorithm, detailed in [7]. This standalone, filter-like application takes as input a raw trace and builds a sequence of loop nests that, when executed, reproduce the trace. It is also able to predict forthcoming values at an arbitrary distance in the future. Its simple, text-based input format makes it applicable to all kinds of data. These data can take the form of simple numeric values, or have more elaborate structure, and can include symbols. The program is written in standard ANSI C. The code can also be used as a library.

We have used this code to evaluate the compression potential of loop nest recognition on memory address traces, with very good results. We have also shown that the predictive power of our model is competitive with other models on average.

The software is available upon request to anybody interested in trying to apply loop nest recognition. It has been distributed to a dozen of colleagues around the world. In particular, it has been used by Andres Charif-Rubial for his PhD work (Université de Versailles Saint-Quentin en Yvelines), and is now included in a released tool called MAQAO (<http://www.maqao.org>). Our code is also used by Jean-Thomas ACQUAVIVA, at Commissariat à l'Énergie Atomique, for work on compressing instruction traces. These colleagues have slightly modified the code we gave them. We plan to release a stable version incorporating most of their changes in the near future. We also plan to change the license to avoid such drifts in the future.

⁸<http://icps.u-strasbg.fr/PolyLib>

⁹<http://ZPolyTrans.gforge.inria.fr>

5.4. Binary files decompiler

Participant: Alain Ketterlin.

Our research on efficient memory profiling has led us to develop a sophisticated decompiler. This tool analyzes x86-64 binary programs and libraries, and extracts various structured representations of the code. It works on a routine per routine basis, and first builds a loop hierarchy to characterize the overall structure of the algorithm. It then puts the code into Static Single Assignment (SSA) form to highlight the fine-grain data-flow between registers and memory. Building on these, it performs the following analyzes:

- All memory addresses are expressed as symbolic expressions involving specific versions of register contents, as well as loop counters. Loop counter definitions are recovered by resolving linearly incremented registers and memory cells, i.e., registers that act as induction variables.
- Most conditional branches are also expressed symbolically (with registers, memory contents, and loop counters). This captures the control-flow of the program, but also helps in defining what amounts to loop “trip-counts”, even though our model is slightly more general, because it can represent any kind of iterative structure.

This tool embodies several passes that, as far as we know, do not exist in any existing similar tool. For instance, it is able to track data-flow through stack slots in most cases. It has been specially designed to extract a representation that can be useful in looking for parallel (or parallelizable) loops [45]. It is the basis of several of our studies.

Because binary program decompilation is especially useful to reduce the cost of memory profiling, our current implementation is based on the Pin binary instrumenter. It uses Pin’s API to analyze binary code, and directly interfaces with the upper layers we have developed (e.g., program skeletonization, or minimal profiling). However, we have been careful to clearly decouple the various layers, and to not use any specific mechanism in designing the binary analysis component. Therefore, we believe that it could be ported with minimal effort, by using a binary file format extractor and a suitable binary code parser. It is also designed to abstract away the detailed instruction set, and should be easy to port (even though we have no practical experience in doing so).

We feel that such a tool could be useful to other researchers, because it makes binary code available under abstractions that have been traditionally available for source code only. If sufficient interest emerges, e.g., from the embedded systems community, or from researchers working on WCET, or from teams working on software security, we are willing to distribute and/or to help make it available under other environments.

5.5. Parwiz: a dynamic dependency analyser

Participant: Alain Ketterlin.

We have developed a dynamic dependence analyzer. Such a tool consumes the trace of memory (or object) accesses, and uses the program structure to list all the data dependences appearing during execution. Data dependences in turn are central to the search for parallel sections of code, with the search for parallel loops being only a particular case of the general problem. Most current works of these questions are either specific to a particular analysis (e.g., computing dependence densities to select code portions for thread-level speculation), or restricted to particular forms of parallelism (e.g., typically to fully parallel loops). Our tool tries to generalize existing approaches, and focuses on the program structures to provide helpful feedback either to a user (as some kind of “smart profiler”), or to a compiler (for feedback-directed compilation). For example, the tool is able to produce a dependence schema for a complete loop nest (instead of just a loop). It also targets irregular parallelism, for example analyzing a loop execution to estimate the expected gain of parallelization strategies like inspector-executor.

We have developed this tool in relation to our minimal profiling research project. However, the tool itself has been kept independent of our profiling infrastructure, getting data from it via a well-defined trace format. This intentional design decision has been motivated by our work on distinct execution environments: first on our usual x86-64 benchmark programs, and second on less regular, more often written in Java, real-world applications. The latter type of applications is likely the one that will most benefit from such tools, because their intrinsic execution environment does not offer enough structure to allow effective static analysis techniques. Parallelization efforts in this context will most likely rely on code annotations, or specific programming language constructs. Programmers will therefore need tools to help them choose between various constructs. Our tool has this ambition. We already have a working tool-chain for C/C++/Fortran programs (or any binary program). We are in the process of developing the necessary infrastructure to connect the dynamic dependence profiler to instrumented Java programs. Other managed execution environments could be targeted as well, e.g., Microsoft's .Net architecture, but we have no time and/or workforce to devote to such time-consuming engineering efforts.

5.6. VMAD software and LLVM

Participants: Alexandra Jimborean, Philippe Clauss, Jean-François Dollinger, Aravind Sukumaran-Rajam, Juan Manuel Martínez Caamaño.

For dynamic analysis and optimization of programs, we are developing a virtual machine called VMAD, and specific passes to the LLVM compiler suite, plus a modified Clang frontend. It is fully described in subsection 6.1 .

As the final result of Alexandra Jimborean's PhD thesis, the VMAD framework now handles speculative parallelization of loop nests by applying dynamically polyhedral code transformations. It is currently extended to handle even more advanced code transformations as tiling in particular, and also to handle codes whose memory behavior is not fully linear.

Alexandra Jimborean (PhD student), Matthieu Herrmann (former Master student), Luis Mastrangelo (former Master student), Juan Manuel Martínez Caamaño (Master student), Jean-François Dollinger (PhD student), Aravind Sukumaran-Rajam (PhD student) and Philippe Clauss are the main contributors of this software. It is not yet distributed.

5.7. Polyhedral prover

Participants: Nicolas Magaud, Julien Narboux, Éric Violard [correspondant].

polyhedral transformations, verified compiler

We are currently developing a formal proof of program transformations based on the polyhedral model. We use the CompCert verified compiler [51] as a framework. This tool is written in the specification language of Coq. It is connected to the activity described in section 6.5 .

CAMEL Project-Team

5. Software

5.1. Introduction

A major part of the research done in the CAMEL team is published within software. On the one hand, this enables everyone to check that the algorithms we develop are really efficient in practice; on the other hand, this gives other researchers — and us of course — basic software components on which they — and we — can build other applications.

5.2. GNU MPFR

Participant: Paul Zimmermann [contact].

GNU MPFR is one of the main pieces of software developed by the CAMEL team. Since end 2006, with the departure of Vincent Lefèvre to ENS Lyon, it has become a joint project between CAMEL and the ARÉNAIRE project-team (now AriC, INRIA Grenoble - Rhône-Alpes). GNU MPFR is a library for computing with arbitrary precision floating-point numbers, together with well-defined semantics, and is distributed under the LGPL license. All arithmetic operations are performed according to a rounding mode provided by the user, and all results are guaranteed correct to the last bit, according to the given rounding mode.

Several software systems use GNU MPFR, for example: the GCC and GFORTRAN compilers; the SAGE computer algebra system; the KDE calculator Abakus by Michael Pyne; CGAL (Computational Geometry Algorithms Library) developed by the Geometrica project-team (INRIA Sophia Antipolis - Méditerranée); Gappa, by Guillaume Melquiond; Sollya, by Sylvain Chevillard, Mioara Joldeş and Christoph Lauter; Genius Math Tool and the GEL language, by Jiri Lebl; Giac/Xcas, a free computer algebra system, by Bernard Parisse; the iRRAM exact arithmetic implementation from Norbert Müller (University of Trier, Germany); the Magma computational algebra system; and the Wcalc calculator by Kyle Wheeler.

The main development in 2012 is the release of version 3.1.1 (the “canard à l’orange” release) in July. With respect to version 3.1.0, this new version improves the reference manual, and fixes a few bugs. Also, a workshop was organized in June in Bordeaux, on the development of GNU MPFR and GNU MPC. In particular, the test coverage of GNU MPFR was improved.

5.3. GNU MPC

Participant: Paul Zimmermann [contact].

GNU MPC is a floating-point library for complex numbers, which is developed on top of the GNU MPFR library, and distributed under the LGPL license. It is co-written with Andreas Enge (LFANT project-team, INRIA Bordeaux - Sud-Ouest). A complex floating-point number is represented by $x + iy$, where x and y are real floating-point numbers, represented using the GNU MPFR library. The GNU MPC library provides correct rounding on both the real part x and the imaginary part y of any result. GNU MPC is used in particular in the TRIP celestial mechanics system developed at IMCCE (*Institut de Mécanique Céleste et de Calcul des Éphémérides*), and by the Magma and Sage computational number theory systems.

A new version, GNU MPC 1.0 (*Fagus silvatica*), was released in July 2012. Up from this release, GNU MPC is considered to be a mature library. Due to a security issue in automake, we had to release a bug-fix version 1.0.1 in September 2012.

5.4. GMP-ECM

Participants: Cyril Bouvier, Paul Zimmermann [contact].

GMP-ECM is a program to factor integers using the Elliptic Curve Method. Its efficiency comes both from the use of the GMP library, and from the implementation of state-of-the-art algorithms. GMP-ECM contains a library (LIBECM) in addition to the binary program (ECM). The binary program is distributed under GPL, while the library is distributed under LGPL, to allow its integration into other non-GPL software. The Magma computational number theory software and the SAGE computer algebra system both use LIBECM.

In January 2012, a new version 6.4 was released, followed by 6.4.1 and 6.4.2 in March, and 6.4.3 in June. Apart from bug fixes, and the fact that GMP-ECM is now distributed under the LGPL version 3, those new releases provide a new *-batch* option with faster Stage 1 code, and an improved tuning mechanism.

In February, Paul Leyland found a 43-digit factor using the GPU implementation of Stage 1 written by C. Bouvier, and in August, a new record prime of 79 digits was found by Sam Wagstaff (Purdue University) using GMP-ECM.

5.5. Finite fields

Participants: Pierrick Gaudry, Emmanuel Thomé [contact].

$\text{mp}\mathbb{F}_q$ is (yet another) library for computing in finite fields. The purpose of $\text{mp}\mathbb{F}_q$ is not to provide a software layer for accessing finite fields determined at runtime within a computer algebra system like Magma, but rather to give a very efficient, optimized code for computing in finite fields precisely known at *compile time*. $\text{mp}\mathbb{F}_q$ is not restricted to a finite field in particular, and can adapt to finite fields of any characteristic and any extension degree. However, one of the targets being the use in cryptology, $\text{mp}\mathbb{F}_q$ somehow focuses on prime fields and on fields of characteristic two.

$\text{mp}\mathbb{F}_q$'s ability to generate specialized code for desired finite fields differentiates this library from its competitors. The performance achieved is far superior. For example, $\text{mp}\mathbb{F}_q$ can be readily used to assess the throughput of an efficient software implementation of a given cryptosystem. Such an evaluation is the purpose of the "eBATS" benchmarking tool¹. $\text{mp}\mathbb{F}_q$ entered this trend in 2007, establishing reference marks for fast elliptic curve cryptography: the authors improved over the fastest examples of key-sharing software in genus 1 and 2, both over binary fields and prime fields. These timings are now comparison references for other implementations [18].

The library's purpose being the *generation* of code rather than its execution, the working core of $\text{mp}\mathbb{F}_q$ consists of roughly 18,000 lines of Perl code, which generate most of the C code. $\text{mp}\mathbb{F}_q$ is distributed at <http://mpfq.gforge.inria.fr/>.

In 2012, $\text{mp}\mathbb{F}_q$ evolved somewhat, in order to do the required code generation needed for evolutions of CADO-NFS, notably in relation with linear algebra over prime fields. A new release is planned soon, once hindrances related to the license of some code fragments are dealt with.

5.6. gf2x

Participants: Pierrick Gaudry, Emmanuel Thomé [contact], Paul Zimmermann.

GF2X is a software library for polynomial multiplication over the binary field, developed together with Richard Brent (Australian National University, Canberra, Australia). It holds state-of-the-art implementation of fast algorithms for this task, employing different algorithms in order to achieve efficiency from small to large operand sizes (Karatsuba and Toom-Cook variants, and eventually Schönhage's or Cantor's FFT-like algorithms). GF2X takes advantage of specific processors instruction (SSE, PCLMULQDQ).

The current version of GF2X is 1.1, released in May 2012 under the GNU GPL. Since 2009, GF2X can be used as an auxiliary package for the widespread software library NTL, as of version 5.5.

An LGPL-licensed portion of GF2X is also part of the CADO-NFS software package.

¹<http://www.ecrypt.eu.org/ebats/>

5.7. CADO-NFS

Participants: Cyril Bouvier, Jérémie Detrey, Alain Filbois, Pierrick Gaudry, Alexander Kruppa, Emmanuel Thomé [contact], Paul Zimmermann.

CADO-NFS is a program to factor integers using the Number Field Sieve algorithm (NFS), originally developed in the context of the ANR-CADO project (November 2006 to January 2010).

NFS is a complex algorithm which contains a large number of sub-algorithms. The implementation of all of them is now complete, but still leaves some places to be improved. Compared to existing implementations, the CADO-NFS implementation is already a reasonable player. Several factorizations have been completed using our implementations.

Since 2009, the source repository of CADO-NFS is publicly available for download. No new release was made in 2012, but several improvements have been made in the development version, with the help of Alain Filbois (SED engineer) and of Alexander Kruppa, recruited in October for a 2-year engineer contract.

Alain Filbois improved the *purge* program for filtering, by gaining a factor of about 5 in the input-output routines. Together with P. Zimmermann, he also wrote a special-purpose *clique removal* code for huge factorizations requiring out-of-core computing; this code has been used for a new filtering experiment on the relations collected for RSA-768 (not yet finished at the time of writing).

The *Objectif 1024* ADT started in 2012, with the recruitment of Alexander Kruppa as a engineer for 2 years. The four main objectives of this ADT are: (1) be able to use CADO-NFS routinely on clusters of 20 to 100 nodes, including on Amazon EC2; (2) develop precise tools to optimize parameters in the sieving phase; (3) develop more professional test mechanisms; (4) make two major releases of CADO-NFS, and advertize them on potential users.

Overall, CADO-NFS keeps improving its competitiveness over alternative code bases. Improvements in CADO-NFS and new results obtained with CADO-NFS are described below.

CARMEN Team (section vide)

CARTE Project-Team

5. Software

5.1. Morphus/MMDEX

MMDEX is a virus detector based on morphological analysis. It is composed of our own disassembler tool, on a graph transformer and a specific tree-automaton implementation. The tool is used in the EU-Fiware project and by some other partners (e.g. DAVFI project).

Written in C, 20k lines.

APP License, IDDN.FR.001.300033.000.R.P.2009.000.10000, 2009.

5.2. TraceSurfer

TraceSurfer is a self-modifying code analyzer coming with an IDA add-on. It works as a wave-builder. In the analysis of self-modifying programs, one basic task is indeed to separate parts of the code which are self-modifying into successive layers, called waves. TraceSurfer extracts waves from traces of program executions. Doing so drastically simplifies program verification.

Written in C, 5k lines.

Private licence.

<http://code.google.com/p/tartetatintools/>

5.3. CROCUS

CROCUS is a program interpretation synthetizer. Given a first order program (possibly written in OCAML), it outputs a quasi-interpretation based on max, addition and product. It is based on a random algorithm. The interpretation is actually a certificate for the program's complexity. Users are non academics (some artists).

Written in Java, 5k lines.

Private licence.

CASCADE Project-Team

5. Software

5.1. MitMTool

Participants: Patrick Derbez, Jérémy Jean.

The purpose of MITMTOOL is to look for guess-and-determine and meet-in-the-middle attacks on AES and AES-based constructions. This tool allows us to improve known attacks on round-reduced versions of AES, on the LEX stream-cipher on the PELICAN Message Authentication Code and on fault attack on AES. Basically, it solves the problem to find all the solutions of a linear system of equations on the variables x and $S(x)$ where S is an inert function. The tool allows to compute the complexity of some good attack as well as the C code of the attack. We verify that the complexity estimates are accurate using experiments. We also use it to find one solution of the system for chosen-key differential attacks. There are mainly two tools: the first one only looks for guess-and-determine attack and tries to propagate some knowledge and guesses value when it cannot find automatically the value of some variable. The second tool uses the technique of the first tool and more advanced technique to take into account attacks with memory that use the meet-in-the-middle attack.

CASSIS Project-Team

5. Software

5.1. Protocol Verification Tools

Participants: Pierre-Cyrille Héam, Olga Kouchnarenko, Michaël Rusinowitch, Mathieu Turuani, Laurent Vigneron.

5.1.1. AVISPA

Cassis has been one of the 4 partners involved in the European project AVISPA, which has resulted in the distribution of a tool for automated verification of security protocols, named AVISPA Tool. It is freely available on the web ¹ and it is well supported. The AVISPA Tool compares favourably to related systems in scope, effectiveness, and performance, by (i) providing a modular and expressive formal language for specifying security protocols and properties, and (ii) integrating 4 back-ends that implement automatic analysis techniques ranging from *protocol falsification* (by finding an attack on the input protocol) to *abstraction-based verification* methods for both finite and infinite numbers of sessions.

5.1.2. CL-AtSe

We develop, as a first back-end of AVISPA, *CL-AtSe*, a Constraint Logic based Attack Searcher for cryptographic protocols. The *CL-AtSe* approach to verification consists in a symbolic state exploration of the protocol execution, for a bounded number of sessions. This necessary restriction (for decidability, see [79]) allows *CL-AtSe* to be correct and complete, i.e., any attack found by *CL-AtSe* is a valid attack, and if no attack is found, then the protocol is secure for the given number of sessions. Each protocol step is represented by a constraint on the protocol state. These constraints are checked lazily for satisfiability, where satisfiability means reachability of the protocol state. *CL-AtSe* includes a proper handling of sets (operations and tests), choice points, specification of any attack states through a language for expressing secrecy, authentication, fairness, non-abuse freeness, advanced protocol simplifications and optimizations to reduce the problem complexity, and protocol analysis modulo the algebraic properties of cryptographic operators such as XOR (exclusive or) and Exp (modular exponentiation). The handling of XOR and Exp has required to implement an optimized version of the combination algorithm of Baader & Schulz [68] for solving unification problems in disjoint unions of arbitrary theories.

CL-AtSe has been successfully used [67] to analyse France Telecom R&D, Siemens AG, IETF, or Gemalto protocols in funded projects. It is also employed by external users, e.g., from the AVISPA's community. Moreover, *CL-AtSe* achieves very good analysis times, comparable and sometimes better than state-of-the-art tools in the domain (see [82] for tool details and precise benchmarks).

Recently, *CL-AtSe* has been enhanced in various ways. As an official back-end for the Avantssar European Project, the tool's development followed the project's requirements for semantic and functionalities. In particular, the tool now fully supports the Aslan semantic, including support for Horn Clauses (for intruder-independent deductions, like e.g. management of credentials), improved support for LTL-based security properties, objects management w.r.t. a set semantic (instead of multiset by default), or smarter behavior in presence of ACM communication channels (default and preferred channel mode for *CL-AtSe* is CCM). While unofficial in Avantssar, the tracing option to target some specific traces during analysis has also been renewed w.r.t. the new modeling of transitions within the Aslan syntax. Also, tool support and bug corrections for all Avantssar's tools is now processed through a bugzilla server (see <https://regis.scienze.univr.it/bugzilla/bugzilla-4.0.4/>), and online analysis and orchestration are available on our team server (<https://cassis.loria.fr>). Then again, *CL-AtSe* now supports negative constraints on the intruder's knowledge. This support is correct and complete without algebraic operators (like Xor and Exp.), and implements in practice the assumptions and

¹<http://www.avispa-project.org>

methods from [32]. This important improvement to the analysis algorithm in CI-Atse allows us to find much more adequate orchestrations, and thus to reduce the orchestrator's processing times in a large scale. It was also used to model e.g. separation of duties.

5.2. Testing Tools

Participants: Fabrice Bouquet, Frédéric Dadeau, Philippe Paquelier, Kalou Cabrera.

5.2.1. Hydra

In December 2008, we have started the redevelopment of our original testing tools environment, with two objectives: first, refactoring the existing developments, and, second, providing an open platform aiming at gathering together the various developments, increasing the reusability of components. The resulting platform, named Hydra, is a Eclipse-like platform, based on Plug-ins architecture. Plug-ins can be of five kinds: *parser* is used to analyze source files and build an intermediate format representation of the source; *translator* is used to translate from a format to another or to a specific file; *service* denotes the application itself, i.e. the interface with the user; *library* denotes an internal service that can be used by a service, or by other libraries; *tool* encapsulates an external tool. The following services have been developed so far:

- BZPAnimator: performs the animation of a BZP model (a B-like intermediate format);
- Angluin: makes it possible to perform a machine learning algorithm (à la Angluin) in order to extract an abstraction of a system behavior;
- UML2SMT: aims at extracting first order logic formulas from the UML Diagrams and OCL code of a UML/OCL model to check them with a SMT solver.

These services involve various libraries (sometimes reusing each other), and rely on several *tool* plug-ins that are: SMTProver (encapsulating Z3 solver), PrologTools (encapsulating CLPS-B solver), Grappa (encapsulating a graph library). We are currently working on transferring the existing work on test generation from B abstract machines, JML, and statecharts using constraint solving techniques.

5.2.2. jMuHLPSL

jMuHLPSL [9] is a mutant generator tool that takes as input a verified HLPSL protocol, and computes mutants of this protocol by applying systematic mutation operators on its contents. The mutated protocol then has to be analyzed by a dedicated protocol analysis tool (here, the AVISPA tool-set). Three verdicts may then arise. The protocol can still be *safe*, after the mutation, this means that the protocol is not sensitive to the realistic “fault” represented by the considered mutation. This information can be used to inform the protocol designers of the robustness of the protocol w.r.t. potential implementation choices, etc. The protocol can also become *incoherent*, meaning that the mutation introduced a functional failure that prevents the protocol from being executed entirely (one of the participants remains blocked in a given non-final state). The protocol can finally become *unsafe* when the mutation introduces a security flaw that can be exploited by an attacker. In this case, the AVISPA tool-set is able to compute an attack-trace, that represents a test case for the implementation of the protocol. If the attack can be replayed entirely, then the protocol is not safe. If the attack can not be replayed then the implementation does not contain the error introduced in the original protocol.

The tool is written in Java, and it is freely available at: <http://disc.univ-fcomte.fr/home/~fdadeau/tools/jMuHLPSL.jar>.

5.3. Collaborative Tools

Participants: Abdessamad Imine, Asma Cherif.

The collaborative tools allow us to manage collaborative works on shared documents using flexible access control models. These tools have been developed in order to validate and evaluate our approach on combining collaborative edition with optimistic access control.

- **P2PEdit.** This prototype is implemented in Java and supports the collaborative editing of HTML pages and it is deployed on P2P JXTA platform ². In our prototype, a user can create a HTML page from scratch by opening a new collaboration group. Other users (peers) may join the group to participate in HTML page editing, as they may leave this group at any time. Each user can dynamically add and remove different authorizations for accessing to the shared document according the contribution and the competence of users participating in the group. Using JXTA platform, users exchange their operations in real-time in order to support WYSIWIS (What You See Is What I See) principle. Furthermore, the shared HTML document and its authorization policy are replicated at the local memory of each user. To deal with latency and dynamic access changes, an optimistic access control technique is used where enforcement of authorizations is retroactive.
- **P2PCalendar.** To extend our collaboration and access control models to mobile devices, we implemented a shared calendar on iPhone OS which is decentralized and scalable (i.e. it can be used over both P2P and ad-hoc networks). This application aims to make a collaborative calendar where users can simultaneously modify events (or appointments) and control access on events. The access rights are determined by the owner of an event. The owner decides who is allowed to access the event and what privileges they have. Likewise to our previous tool, the calendar and its authorization policy are replicated at every mobile device.

5.4. Other Tools

Several software tools described in previous sections are using tools that we have developed in the past. For instance BZ-TT uses the set constraints solver CLPS. Note that the development of the SMT prover haRVey has been stopped. The successor of haRVey is called veriT and is developed by David Déharbe (UFRN Natal, Brasil) and Pascal Fontaine (Veridis team). We have also developed, as a second back-end of AVISPA, TA4SP (Tree Automata based on Automatic Approximations for the Analysis of Security Protocols), an automata based tool dedicated to the validation of security protocols for an unbounded number of sessions.

²<http://www.sun.com/software/jxta/>

CASTOR Team

4. Software

4.1. FluidBox

Participants: Boniface Nkonga [contact], Hervé Guillard.

FluidBox is a software dedicated to the simulation of inert or reactive flows. It is also able to simulate multiphase, multi-material and MDH flows. There exist 2D and 3D dimensional versions. The 2D version is used to test new ideas that are later implemented in 3D. Two classes of schemes are available : A classical finite volume scheme and the more recent residual distribution schemes. Several low Mach number preconditioning are also implemented. The code has been parallelized with and without domain overlapping. The linear solver PaStiX is integrated in FluidBox. A partitioning tool exists in the package and uses Scotch. At present the software is only a private project but some parts of FluidBox are expected to be in the public domain by the end of the year.

4.2. PlaTo

Participants: Hervé Guillard [contact], Laure Combe.

The development of PlaTo (A platform for Tokamak simulation) (<http://www-sop.inria.fr/pumas/plato.php>) has been supported by an ADT action of the D2T an by the ANR ESPOIR. PlaTo is a suite of data and softwares dedicated to the geometry and physics of Tokamaks and its main objective is to provide the Inria large scale initiative “FUSION” teams working in plasma fluid models with a common development tool.

4.3. PaMPA

Participants: Cécile Dobrzynski [Bacchus], Hervé Guillard, Laurent Hascoët [Tropics], Cédric Lachat, François Pellegrini [Bacchus].

PaMPA (“Parallel Mesh Partitioning and Adaptation”) is a middleware library dedicated to the management of distributed meshes. Its purpose is to relieve solver writers from the tedious and error prone task of writing again and again service routines for mesh handling, data communication and exchange, remeshing, and data redistribution. An API of the future platform has been devised, and the coding of the mesh handling and redistribution routines is in progress. PaMPA will be used as a base module for the PLATO solvers, to balance dynamically, refine and coarsen its distributed mesh.

4.4. Cedres++

In Tokamaks, at the slow resistive diffusion time scale, the magnetic configuration in the plasma can be described by the MHD equilibrium equations inside the plasma and the Maxwell equations outside. Moreover, the magnetic field is often supposed not to depend on the azimuthal angle.

Under this assumption of axisymmetric configuration, the equilibrium in the whole space reduces to solving a 2D problem in which the magnetic field in the plasma is described by the well known Grad Shafranov equation. The unknown of this problem is the poloidal magnetic flux. The P1 finite element code CEDRES++ solves this free boundary equilibrium problem in direct and inverse mode. The direct problem consists in the computation of the magnetic configuration and of the plasma boundary, given a plasma current density profile and the total current in each poloidal field coils (PF coils). The aim of the inverse problem is to find currents in the PF coils in order to best fit a given plasma shape. An evolutive version of the code has also been recently developed. This version takes into account the circuit equations in the PF coils. These equations give a time dependent relation between the voltages, the total current in the coils and the time derivative of the magnetic flux. Induced currents in passive structures like the vacuum vessel are also considered in this dynamic equilibrium problem. This new version of the code is an important tool for plasma scenario development and Tokamak design studies.

4.5. Equinox

EQUINOX is a code dedicated to the numerical reconstruction of the equilibrium of the plasma in a Tokamak. The problem solved consists in the identification of the plasma current density, a non-linear source in the 2D Grad-Shafranov equation which governs the axisymmetric equilibrium of a plasma in a Tokamak. The experimental measurements that enable this identification are the magnetics on the vacuum vessel, but also polarimetric and interferometric measures on several chords, as well as motional Stark effect measurements. The reconstruction can be obtained in real-time and the numerical method implemented involves a finite element method, a fixed-point algorithm and a least-square optimization procedure.

CELTIQUE Project-Team

4. Software

4.1. Javalib

Participants: Frédéric Besson [correspondant], David Pichardie, Vincent Monfort.

Javalib is an efficient library to parse Java .class files into OCaml data structures, thus enabling the OCaml programmer to extract information from class files, to manipulate and to generate valid .class files.

See also the web page <http://sawja.inria.fr/>.

- Version: 2.2
- Programming language: Ocaml

4.2. SAWJA

Participants: Frédéric Besson [correspondant], David Pichardie, Vincent Monfort.

Sawja is a library written in OCaml, relying on Javalib to provide a high level representation of Java bytecode programs. Its name comes from Static Analysis Workshop for JAva. Whereas Javalib is dedicated to isolated classes, Sawja handles bytecode programs with their class hierarchy and with control flow algorithms.

Moreover, Sawja provides some stackless intermediate representations of code, called JBir and A3Bir. The transformation algorithm, common to these representations, has been formalized and proved to be semantics-preserving.

See also the web page <http://sawja.inria.fr/>.

- Version: 1.2
- Programming language: Ocaml

4.3. Jacal

Participants: Frédéric Besson [correspondant], Thomas Jensen, David Pichardie, Delphine Demange, Vincent Monfort, Pierre Vittet.

Static program analysis, Javacard, Certification, AFSCM

Jaca1 is a JAvacard AnaLyseur developed on top of the SAWJA^{4.2} platform. This proprietary software verifies automatically that Javacard programs conform with the security guidelines issued by the AFSCM (Association Française du Sans Contact Mobile). Jaca1 is based on the theory of abstract interpretation and combines several object-oriented and numeric analyses to automatically infer sophisticated invariants about the program behaviour. The result of the analysis is thereafter harvested to check that it is sufficient to ensure the desired security properties.

4.4. Timbuk

Participant: Thomas Genet [correspondant].

Timbuk is a library of OCAML functions for manipulating tree automata. More precisely, Timbuk deals with finite bottom-up tree automata (deterministic or not). This library provides the classical operations over tree automata (intersection, union, complement, emptiness decision) as well as exact or approximated sets of terms reachable by a given term rewriting system. This last operation can be certified using a checker extracted from a Coq specification. The checker is now part of the Timbuk distribution. Timbuk distribution now also provides a CounterExample Guided Abstraction Refinement (CEGAR) tool for tree automata completion. The CEGAR part is based on the Buddy BDD library.

- Version: 3.1
- Programming language: Ocaml

CEPAGE Project-Team

5. Software

5.1. SimGrid

Participants: Przemyslaw Uznanski, Lionel Eyraud-Dubois [correspondant].

SimGrid (<http://simgrid.gforge.inria.fr/>) SimGrid is a toolkit that provides core functionalities for the simulation of distributed applications in heterogeneous distributed environments. The specific goal of the project is to facilitate research in the area of parallel and distributed large scale systems, such as Grids, P2P systems and clouds. Its use cases encompass heuristic evaluation, application prototyping or even real application development and tuning. It is based on experimentally validated models, and features very high scalability, which allows to perform very large scale simulations. It is used by over a hundred academic users all over the world, and has been used in about one hundred scientific articles.

CEPAGE has contributed to this software by participating in the management of the project and in many design decisions. We also implemented simpler models, based on our works in this area, allowing a better scalability while keeping a reasonable precision.

Software assessment : A-4, SO-4, SM-4, EM-3, SDL-5.

Contribution : DA-2, CD-2, MS-2, TPM-3.

5.2. Hubble

Participants: Ludovic Courtès, Nicolas Bonichon [correspondant].

Hubble is implemented in Scheme, using GNU Guile version 2. Details of the simulation, such as keeping track of processor occupation and network usage, are taken care of by SimGrid, a toolkit for the simulation of distributed applications in heterogeneous distributed environments.

The input to Hubble is an XML description of the DAG of build tasks. For each task, a build duration and the size in bytes of the build output are specified. For our evaluation purposes, we collected this data on a production system, the <http://hydra.nixos.org/> build farm hosted at the Technical University of Delft. The DAG itself is the snapshot of the Nix Package Collection (Nixpkgs) corresponding to this data. Hubble has its own in-memory representation of the DAG in the form of a purely functional data structure.

The Nixpkgs DAG contains fixed-output nodes, i.e., nodes whose output is known in advance and does not require any computation. These nodes are typically downloads of source code from external web sites. The raw data collected on <http://hydra.nixos.org/> specifies a non-zero duration for these nodes, which represents the time it took to perform the download. This duration info is irrelevant in our context, since they don't require any computation, and Hubble views these nodes as instantaneous.

See also the web page <http://hubble.gforge.inria.fr/>.

Software assessment: A-3, SO-3, SM-2, EM-1, SDL-2.

Contribution: DA-4, CD-4, MS-4, TPM-4.

5.3. Gengraph

Participant: Cyril Gavaille [correspondant].

This is a command-line tool for generating graphs. There are several output formats, includes the dot format from GraphViz. It generates also .pdf files for visualization. Several graph algorithms have been implemented (diameter, connectivity, treewidth, etc.) which can be tested on the graphs. The software has been originally designed for teaching purpose so that students can test their project algorithms on many non trivial families like random geometric graphs, graphs of given density, given treewidth. It is also used for research purpose, in particular the exhaustive search results in the Emilie Diot's thesis are based on gengraph. The program can filter a list of graphs based to many criteria, as for instance it can extract all graphs of a given list that are 2-connected, of diameter at least four, and that exclude some minor (or some induced subgraph).

Currently, more than 100 parametrized graph families are implemented, supporting simple operators like complementation, random edge/vertex removal, and others. The source has more than 10,000 lines including a command-line documentation of 2,000 lines. The single source file is available at <http://dept-info.labri.fr/~gavoille/gengraph.c>

Software assessment: A-3, SO-3, SM-2, EM-2, SDL-2.

Contribution: DA-4, CD-4, MS-4, TPM-4.

5.4. Bedibe

Participants: Lionel Eyraud-Dubois [correspondant], Przemyslaw Uznanski.

Bedibe (Benchmarking Distributed Bandwidth Estimation) is a software to compare different models for bandwidth estimation on the Internet, and their associated instantiation algorithms. The goal is to ease the development of new models and algorithms, and the comparison with existing solutions. The development of this software is just starting.

See also the web page <http://bedibe.gforge.inria.fr/>.

Software assessment : A-1-up2, SO-3, SM-1-up2, EM-2, SDL-1-up2.

5.5. MineWithRounds

Participants: Sofian Maabout [correspondant], Nicolas Hanusse.

The software implements a parallel algorithm aiming at computing *Borders* that's sets of maximal/minimal subsets of objects satisfying some anti-monotone condition. It is implemented in C++ together with the openMP library to exploit multi-core machines. In its current status, it outperforms state of the art implementations addressing the Maximal Frequent Itemsets problem.

Software assessment: A-2, SO-4, SM-2, EM-2, SDL-2.

Contribution: DA-4, CD-4, MS-4, TPM-4.

CIDRE Project-Team

5. Software

5.1. Intrusion Detection

Members of Supélec have developed several intrusion detectors.

Blare implements our approach of illegal information flow detection at the OS level. This implementation is a modification of a standard Linux kernel and it monitors information flows between typical OS containers as files, sockets or IPC. System active entities are processes viewed as black-boxes as we only observe their inputs and outputs. Detection at the OS level is in some cases too coarse-grained to avoid the generation of false positives and to detect attacks targeting the application logic. Even if it remains convenient to define the security policy at the OS-level, sound illegal information flow detection implies an additional detection at the language level. This has led us to implement a detector for Java applications, **JBlare**, to complement the detection at the OS level. JBlare extends the OS-level one by refining the observation of information flows at the language level.

GNG is an intrusion detection system that correlates different sources (such as different logs) in order to identify attacks against the system. The attack scenarios are defined using the Attack Description Language (**ADeLe**) proposed by our team, and are internally translated to attack recognition automaton. GNG intends to define time efficient algorithms based on these automaton to recognize complex attack scenarios.

SIDAN (Software Instrumentation for Detecting Attacks on Non-control-data) is a tool that aims to instrument automatically C-language software with assertions whose role is to detect attacks against the software. This tool is implemented as a plugin of the FRAMA-C framework that provides an implementation of static analysis techniques.

5.2. Privacy

GEPETO (GEOPrivacy-Enhancing TOolkit) is an open source software for managing geolocated data (currently in development in cooperation with LAAS). GEPETO can be used to visualize, sanitize, perform inference attacks and measure the utility of a particular geolocated dataset. For each of these actions, a set of different techniques and algorithms can be applied. The global objective of GEPETO is to enable a user to design, tune, experiment and evaluate various sanitization algorithms and inference attacks as well as visualizing the following results and evaluating the resulting trade-off between privacy and utility. An engineer (Izabela Moise) is currently working on the development of a distributed version of GEPETO based on the MapReduce paradigm and the Hadoop framework, in order to make it able to deal with datasets composed of millions of mobility traces.

CLASSIC Project-Team (section vide)

CLIME Project-Team

5. Software

5.1. Polyphemus

Participants: Vivien Mallet, Anne Tilloy.

Polyphemus (see the web site <http://cerea.enpc.fr/polyphemus/>) is a modeling system for air quality. As such, it is designed to yield up-to-date simulations in a reliable framework: data assimilation, ensemble forecast and daily forecasts. Its completeness makes it suitable for use in many applications: photochemistry, aerosols, radionuclides, *etc.* It is able to handle simulations from local to continental scales, with several physical models. It is divided into three main parts:

- libraries that gather data processing tools (SeldonData), physical parameterizations (AtmoData) and postprocessing abilities (AtmoPy);
- programs for physical preprocessing and chemistry-transport models (Polair3D, Castor, two Gaussian models, a Lagrangian model);
- drivers on top of the models in order to implement advanced simulation methods such as data assimilation algorithms.

Figure 1 depicts a typical result produced by Polyphemus. Clime is involved in the overall design of the system and in the development of advanced methods in model coupling, data assimilation and ensemble forecast (through drivers and post-processing).

In 2012, Polyphemus received several physical developments on secondary organic aerosols, modeling of pollution due to traffic emissions and coupling of local and regional scale models. Further integration with the data assimilation library Verdandi has been carried out. A Python interface to the Eulerian model Polair3D has been introduced.

5.2. Data assimilation library: Verdandi

Participants: Kévin Charpentier, Marc Fragu [MACS], Vivien Mallet, Dominique Chapelle [MACS], Philippe Moireau [MACS], Sergiy Zhuk, Anne Tilloy.

The leading idea is to develop a data assimilation library intended to be generic, at least for high-dimensional systems. Data assimilation methods, developed and used by several teams at Inria, are generic enough to be coded independently of the system to which they are applied. Therefore these methods can be put together in a library aiming at:

- making easier the application of methods to a great number of problems,
- making the developments perennial and sharing them,
- improving the broadcast of data assimilation works.

An object-oriented language (C++) has been chosen for the core of the library. A high-level interface to Python is automatically built. The design study raised many questions, related to high dimensional scientific computing, the limits of the object contents and their interfaces. The chosen object-oriented design is mainly based on three class hierarchies: the methods, the observation managers and the models. Several base facilities have also been included, for message exchanges between the objects, output saves, logging capabilities, computing with sparse matrices.

In 2012, versions 1.2, 1.3 and 1.4 were released. The design of the library has been improved for optimal performance and is now stable. It is possible to write models and observation managers in Python. Efficient support for parallel computations has been introduced. The documentation has been improved.

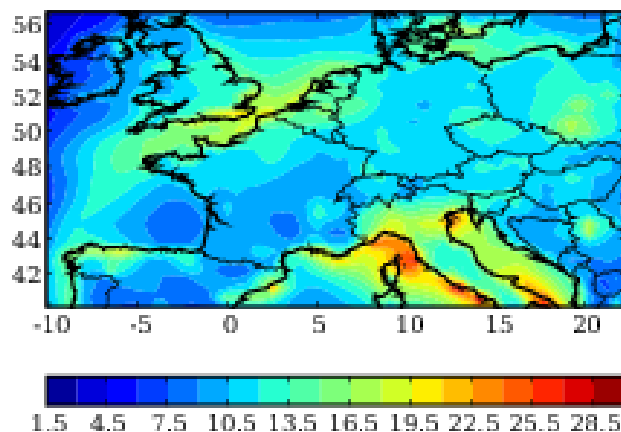


Figure 1. Map of the relative standard deviation (or spread, %) of an ensemble built with Polyphemus (ozone simulations, $\mu\text{g m}^{-3}$). The standard deviations are averaged over the summer of 2001. They provide an estimation of the simulation uncertainties.

5.3. Urban air quality analysis

Participants: Anne Tilloy, Vivien Mallet.

“Urban Air Quality Analysis” carries out data assimilation at urban scale. It merges the outputs of a numerical model (maps of pollutant concentrations) with observations from an air quality monitoring network, in order to produce the so-called analyses, that is, corrected concentration maps. The data assimilation computes the Best Linear Unbiased Estimator (BLUE), with a call to the data assimilation library Verdandi. The error covariance matrices are parameterized for both model simulations and observations. For the model state error covariances, the parameterization primarily relies on the road network. The software handles ADMS Urban output files, for a posteriori analyses or in an operational context.

COFFEE Project-Team

5. Software

5.1. NS2DDV

The code NS2DDV is developed jointly with the team SIMPAF, of the Inria Research Centre Lille Nord Europe. It is devoted to the simulation of non-homogeneous viscous flows, in two-dimensional geometries. The code is based on an original hybrid Finite Volume/Finite Element scheme; it works on unstructured meshes and can include mesh refinements strategies. Further details can be found in the research papers *J. Comput. Phys.*, 227, 4671–4696, 2008 and *J. Comput. Phys.*, 229 (17), 6027–6046, 2010. The code exists in two versions: a Matlab public version, a C++ prototype version allowing more ambitious simulations. Both versions are still subject to developments. The current versions is restricted to incompressible flows but ongoing progress are concerned with the simulation of avalanches. The source code of the public version is downloadable and several benchmarks tests can be reproduced directly.

5.2. FV_PM

We are developing codes based on Finite Volume discretization, for the (2d and 3d) simulations of multiphase flows in porous media. For instance these methods apply to the simulation of problems motivated by CO₂ storage, oil recovery or nuclear waste depository. A preliminary version, the code ComPASS (Computing Parallel Architecture to Speed up Simulations), which includes parallel procedures, has been recently developed, through a successful CEMRACS project.

5.3. SimBiof

We are developing numerical methods, currently by using Finite Differences approaches, for the simulation of biofilms growth. The underlying system of PDEs takes the form of multiphase flows equations with conservation constraints and vanishing phases. The numerical experiments have permitted to bring out the influence of physical parameters on the multidimensional growth dynamics.

5.4. AP_PartFlow

We are developing experimental codes, mainly based on Finite Differences, for the simulation of particulate flows. A particular attention is paid to guaranty the asymptotic properties of the scheme, with respect to relaxation parameters.

COMETE Project-Team

5. Software

5.1. A model checker for the probabilistic asynchronous π -calculus

Participants: Miguel Andrés [correspondant], Catuscia Palamidessi.

In collaborations with Dave Parker and Marta Kwiatkowska, we are developing a model checker for the probabilistic asynchronous π -calculus. Case studies with Fair Exchange and MUTE, an anonymous peer-to-peer file sharing system, are in progress.

Technically we use MMC as a compiler to encode the probabilistic π -calculus into certain PRISM representation, which will then be verified against PCTL using PRISM. The transitional semantics defined in MMC can be reused to derive the symbolic transition graphs of a probabilistic process. The code for derivation will work as an add-on to MMC under XSB and invoke a graph traversal to enumerate all reachable nodes and transitions of the probabilistic process.

In the meanwhile we are also attempting a direct and more flexible approach to the development of a model checker for the probabilistic π -calculus, using OCaml. This should allow to extend the language more easily, to include cryptographic primitives and other features useful for the specification of security protocols. As the result of our preliminary steps in this direction we have developed a rudimentary model checker, available at the following URL: <http://vamp.gforge.inria.fr/>.

5.2. PRISM model generator

Participants: Konstantinos Chatzikokolakis [correspondant], Catuscia Palamidessi.

This software generates PRISM models for the Dining Cryptographers and Crowds protocols. It can also use PRISM to calculate the capacity of the corresponding channels. More information can be found in [39] and in the file README file with instructions at the URL <http://www.lix.polytechnique.fr/comete/software/README-anonmodels.html>.

The software can be download at <http://www.lix.polytechnique.fr/comete/software/anonmodels.tar.gz>. These scripts require Perl to run and have been tested in Linux. The GUI of the corners tool also requires the Perl/TK library. Finally some parts of the model generator tool require PRISM and gnuplot to be installed.

5.3. Calculating the set of corner points of a channel

Participants: Konstantinos Chatzikokolakis [correspondant], Catuscia Palamidessi.

The corner points can be used to compute the maximum probability of error and to improve the Hellman-Raviv and Santhi-Vardy bounds. More information can be found in [40] and in the file README file with instructions at the URL <http://www.lix.polytechnique.fr/comete/software/README-corners.html>.

The software can be download at <http://www.lix.polytechnique.fr/comete/software/corners.tar.gz>. These scripts require Perl to run and have been tested in Linux. The GUI of the corners tool also requires the Perl/TK library. Finally some parts of the model generator tool require PRISM and gnuplot to be installed.

5.4. MMCsp, a compiler for the π -calculus

Participants: Peng Wu [correspondant], Catuscia Palamidessi.

MMCsp is a compiler from a simple probabilistic π -calculus to PRISM models. It is built on XSB, a tabled logic programming system, and generates the symbolic semantic representation of a probabilistic pi-calculus term in text. A separate Java program then translates this semantic representation into a probabilistic model for PRISM.

The tool was developed by Peng Wu during his postdoc period in Comète in 2005-2007, in the context of the collaboration between the teams Comète and PRISM under the Inria/ARC **Project ProNoBis**. It is based on the papers [44] and [42].

The source code is free and can be download from http://www.cs.ucl.ac.uk/staff/p.wu/mmc_sp_manual.html.

COMMANDS Project-Team

5. Software

5.1. Bocop

Participants: Pierre Martinon [corresponding author], Vincent Grélard, Daphné Giorgi, Frédéric Bonnans.

Web page: <http://bocop.org>

The Bocop project aims to develop an open-source toolbox for solving optimal control problems, with collaborations with industrial and academic partners. Optimal control (optimization of dynamical systems governed by differential equations) has numerous applications in transportation, energy, process optimization, and biology. The software reuses some packages from the COIN-OR library, in particular the well-known nonlinear programming solver Ipopt, and also features a user-friendly interface.

The project is supported by Inria with the arrival of Vincent Grelard as developer in October 2010, and then Daphné Giorgi in October 2012. The first prototype was released in 2011, and version 1.4 is scheduled for the end of 2012. Bocop was first successfully tested on several academic problems, see [28] available on <http://bocop.org>. In 2012, several research collaborations were initiated in fields such as bio-reactors for energy production (ref ECC), swimming micro-robots, and quantum control for medical imaging. Bocop was also featured during our participation in the Imatch "Optimisation and Control" in October.

5.2. CollAv

Participants: Hasnaa Zidani [corresponding author], Olivier Bokanowski, Anna Desilles.

This software simulates the evolution of controlled dynamical systems (possibly under uncertainties). The numerical algorithm here is based on HJB or viability approaches, and allows the design of optimal planning strategies (according to a criterion determined by the user: time, energy, ...). It also provides conflict resolution and avoidance of collisions with fixed or moving obstacles. So far, the software is used in collaboration with DGA for avoidance collision of UaVs, and by Volkswagen in some studies related to collision avoidance of cars.

5.3. OCOPHyS

Participants: Hasnaa Zidani [corresponding author], Giovanni Granato.

This is a software for optimisation-based controller design for operating in different regimes or modes of operation. The software can be used, for example, to determine the optimal management for hybrid vehicles or hybrid engines with multiple energy sources. However, the methods used in software are still quite general and can be used in many applications.

5.4. BiNoPe-HJ

Participants: Hasnaa Zidani [corresponding author], Olivier Bokanowski, Anna Desilles, Jun-Yi Zhao.

Web page: <http://www.ensta-paristech.fr/zidani/BiNoPe-HJ>

This project aims at developing sequential and parallel MPI/openMP C++ solvers for the approximation of Hamilton-Jacobi-Bellman (HJB) equations in a d-dimensional space. The main goal is to provide an HJB solvers that can work in dimension d (limited by the machine's capacity). The solver outputs can be visualized with Matlab or Paraview (via VTK files).

The development of the HJB Solver has been initiated under a partnership between COMMANDS and the SME HPC-project in the period between December 2009 to November 2011. Currently, it is still maintained and improved by COMMANDS.

In 2012, two versions were released:

- HJB-SEQUENTIAL-REF: sequential version that can run on any machine
- HJB-PARALLEL-REF: parallel version that can run only on multi-core architectures.

5.5. Shoot

Participant: Pierre Martinon [corresponding author].

Web page: <http://www.cmap.polytechnique.fr/~martinon/codes.html>

Shoot was designed for the resolution of optimal control problems via indirect methods (necessary conditions, Pontryagin's Maximum Principle). Such methods transform the original problem into finding a zero of a certain shooting function. The package offers several choices of integrators and solvers, and can handle control discontinuities. Features also include the use of variational equations to compute the Jacobian of the shooting function, as well as homotopy and grid shooting techniques for easier initialization.

COMPSYS Project-Team

5. Software

5.1. Introduction

This section lists and briefly describes the software developments conducted within Compsys. Most are tools that we extend and maintain over the years. They now concern two activities only: a) the development of tools linked to polyhedra and loop/array transformations, b) the development of algorithms within the back-end compiler of STMicroelectronics.

Many tools based on the polyhedral representation of codes with nested loops are now available. They have been developed and maintained over the years by different teams, after the introduction of Paul Feautrier's Pip, a tool for parametric integer linear programming. This "polytope model" view of codes is now widely accepted: it used by Inria projects-teams Cairn and Alchemy/Parkas, PIPS at École des Mines de Paris, Suif from Stanford University, Compaan at Berkeley and Leiden, PiCo from the HP-Labs (continued as PicoExpress by Synfora and now Synopsis), the DTSE methodology at Imec, Sadayappan's group at Ohio State University, Rajopadhye's group at Colorado State's University, etc. More recently, several compiler groups have shown their interest in polyhedral methods, e.g., the Gcc group, IBM, and Reservoir Labs, a company that develops a compiler fully based on the polytope model and on the techniques that we (the french community) introduced for loop and array transformations. Polyhedra are also used in test and certification projects (Verimag, Lande, Vertecs). Now that these techniques are well-established and disseminated in and by other groups, we prefer to focus on the development of new techniques and tools, which are described here.

The other activity concerns the developments within the compiler of STMicroelectronics. These are not stand-alone tools, which could be used externally, but algorithms and data structures implemented inside the LAO back-end compiler, year after year, with the help of STMicroelectronics colleagues. As these are also important developments, it is worth mentioning them in this section. They are also completed by important efforts for integration and evaluation within the complete STMicroelectronics toolchain. They concern exact (ILP-based) methods, algorithms for aggressive optimizations, techniques for just-in-time compilation, and for improving the design of the compiler.

5.2. Pip

Participants: Cédric Bastoul [MCF, IUT d'Orsay], Paul Feautrier.

Paul Feautrier is the main developer of Pip (Parametric Integer Programming) since its inception in 1988. Basically, Pip is an "all integer" implementation of the Simplex, augmented for solving integer programming problems (the Gomory cuts method), which also accepts parameters in the non-homogeneous term. Pip is freely available under the GPL at <http://www.piplib.org>. It is widely used in the automatic parallelization community for testing dependences, scheduling, several kind of optimizations, code generation, and others. Beside being used in several parallelizing compilers, Pip has found applications in some unconnected domains, as for instance in the search for optimal polynomial approximations of elementary functions (see the Inria project Arénaire).

5.3. Syntol

Participant: Paul Feautrier.

Syntol is a modular process network scheduler. The source language is C augmented with specific constructs for representing communicating regular process (CRP) systems. The present version features a syntax analyzer, a semantic analyzer to identify DO loops in C code, a dependence computer, a modular scheduler, and interfaces for CLoog (loop generator developed by C. Bastoul) and Cl@k (see Sections 5.4 and 5.6). The dependence computer now handles casts, records (structures), and the modulo operator in subscripts and conditional expressions. The latest developments are, firstly, a new code generator, and secondly, several experimental tools for the construction of bounded parallelism programs.

- The new code generator, based on the ideas of Boulet and Feautrier [17], generates a counter automaton that can be presented as a C program, as a rudimentary VHDL program at the RTL level, as an automaton in the Aspic input format, or as a drawing specification for the DOT tool.
- Hardware synthesis can only be applied to bounded parallelism programs. Our present aim is to construct threads with the objective of minimizing communications and simplifying synchronization. The distribution of operations among threads is specified using a placement function, which is found using techniques of linear algebra and combinatorial optimization.

5.4. Cl@k

Participants: Christophe Alias, Fabrice Baray [Mentor, Former post-doc in Compsys], Alain Darté.

Cl@k (Critical Lattice Kernel) is a stand-alone optimization tool useful for the automatic derivation of array mappings that enable memory reuse, based on the notions of admissible lattice and of modular allocation (linear mapping plus modulo operations). It has been developed in 2005-2006 by Fabrice Baray, former post-doc Inria under Alain Darté's supervision. It computes or approximates the critical lattice for a given 0-symmetric polytope. (An admissible lattice is a lattice whose intersection with the polytope is reduced to 0; a critical lattice is an admissible lattice with minimal determinant.)

Its application to array contraction has been implemented by Christophe Alias in a tool called Bee (see Section 5.6). Bee uses Rose as a parser, analyzes the lifetimes of the elements of the arrays to be compressed, and builds the necessary input for Cl@k, i.e., the 0-symmetric polytope of conflicting differences. Then, Bee computes the array contraction mapping from the lattice provided by Cl@k and generates the final program with contracted arrays. More details on the underlying theory are available in previous reports. Cl@k can be viewed as a complement to the Polylib suite, enabling yet another kind of optimizations on polyhedra. Initially, Bee was the complement of Cl@k in terms of its application to memory reuse. Now, Bee is a stand-alone tool that contains more and more features for program analysis and loop transformations.

5.5. PoCo

Participant: Christophe Alias.

PoCo is a polyhedral compilation framework providing many features to quickly prototype program analysis and optimizations in the polyhedral model. Essentially, PoCo provides:

- A C front-end extracting the polyhedral representation of the input program. The parser itself is based on EDG (*via* Rose), an industrial C/C++ parser from Edison group used in Intel compilers.
- An extended language of pragmas to feed the source code with compilation directives (a schedule, for example).
- A symbolic layer on polyhedral libraries Polylib (set operations on polyhedra) and Piplib (parameterized ILP). This feature simplifies drastically the developer task.
- Some dependence analysis (polyhedral dependence graph, array dataflow analysis), array region analysis, array liveness analysis.
- A C and VHDL code generation based on the ideas of P. Boulet and P. Feautrier [17].

The array dataflow analysis (ADA) of PoCo has been extended to a FADA (Fuzzy ADA) by M. Belaoucha, former PhD student at Université de Versailles. FADALib is available at <http://www.prism.uvsq.fr/~bem/fadalib/>.

PoCo has been developed by Christophe Alias. It represents more than 19000 lines of C++ code. The tools Bee, Chuba, and RanK presented thereafter make an extensive use of PoCo abstractions.

5.6. Bee

Participants: Christophe Alias, Alain Darte.

Bee is a source-to-source optimizer that contracts the temporary arrays of a program under scheduling constraints. Bee bridges the gap between the mathematical optimization framework described in [19] and implemented in Cl@k (Section 5.4), and effective source-to-source array contraction. Bee applies a precise lifetime analysis for arrays to build the mathematical input of Cl@k. Then, Bee derives the array allocations from the basis found by Cl@k and generates the C code accordingly. Bee is – to our knowledge – the only complete array contraction tool.

Bee is sensitive to the program schedule. This latter feature enlarges the application field of array contraction to parallel programs. For instance, it is possible to mark a loop to be software-pipelined (with an affine schedule) and to let Bee find an optimized array contraction. But the most important application is the ability to optimize communicating regular processes (CRP). Given a schedule for every process, Bee can compute an optimized size for the channels, together with their access functions (the corresponding allocations). We currently use this feature in source-to-source transformations for high-level synthesis (see Section 3.3).

- Bee was made available to STMicroelectronics as a binary.
- Bee will be transferred to the (incubated) start-up Zettice, initiated by Alexandru Plesco.
- Bee is used as an external tool by the compiler Gecos developed in the Cairn team at Irista.

Bee has been implemented by Christophe Alias, using the compiler infrastructure PoCo. It represents more than 2400 lines of C++ code.

5.7. Chuba

Participants: Christophe Alias, Alain Darte, Alexandru Plesco [Compsys/Zettice].

Chuba is a source-level optimizer that improves a C program in the context of the high-level synthesis (HLS) of hardware. Chuba is an implementation of the work described in the PhD thesis of Alexandru Plesco. The optimized program specifies a system of multiple communicating accelerators, which optimize the data transfers with the external DDR memory. The program is divided into blocks of computations obtained thanks to tiling techniques, and, in each block, data are fetched by block to reduce the penalty due to line changes in the DDR accesses. Four accelerators achieve data transfers in a macro-pipeline fashion so that data transfers and computations (performed by a fifth accelerator) are overlapped.

So far, the back-end of Chuba is specific to the HLS tool C2H but the analysis is quite general and adapting Chuba to other HLS tools should be possible. Besides, it is interesting to mention that the program analysis and optimizations implemented in Chuba address a problem that is also very relevant in the context of GPGPUs.

Chuba has been implemented by Christophe Alias, using the compiler infrastructure PoCo. It represents more than 900 lines of C++. The reduced size of Chuba is mainly due to the high-level abstractions provided by PoCo.

5.8. IceBuilder

Participants: Christophe Alias, Alexandru Plesco [Compsys/Zettice].

IceBuilder is the HLS tool to be transferred in the start-up Zettice. It is a compiler, whose input is a C program annotated with pragmas, and whose output is an equivalent hardware description as synthesizable VHDL. Also, IceBuilder produces a non-synthesizable SystemC description for debugging purpose. As for any compiler, IceBuilder consists into two steps: (i) a front-end, which generates an intermediate representation from the C program, and (ii) a back-end, which translates the intermediate representation into hardware. The intermediate representation of IceBuilder is a data-aware process network (DPN) (see Section 6.3). The front-end does most of the high-level optimizations (communication pipelining, buffer sizing, datapath pipeline scheduling), which are explicitly represented in the DPN. The front-end is implemented as a separate tool, Dcc, so as to be reused with different targets, for instance GPGPUs. Then, the back-end generates the hardware implementation of the DPN. It produces and connects the required buffers, multiplexors, demultiplexors, synchronization channels, finite-state machines, and datapaths.

IceBuilder represents more than 3000 lines of C++ code.

5.9. Dcc

Participants: Christophe Alias, Alexandru Plesco [Compsys/Zettice].

Dcc is the front-end of the IceBuilder tool. Dcc takes as input a C program annotated with pragmas and produces an optimized data-aware process network (DPN). To do so, Dcc reuses most of the analysis implemented in PoCo (dataflow analysis and control generation), Chuba (communication pipelining), Cl@k and Bee (buffer sizing). Dcc and DPNs are very critical parts of IceBuilder and will require a patent before any publication.

Dcc represents more than 2500 lines of C++ code.

5.10. C2fsm

Participant: Paul Feautrier.

C2fsm is a general tool that converts an arbitrary C program into a counter automaton. This tool reuses the parser and pre-processor of Syntol, which has been greatly extended to handle `while` and `do while` loops, `goto`, `break`, and `continue` statements. C2fsm reuses also part of the code generator of Syntol and has several output formats, including FAST (the input format of Aspic), a rudimentary VHDL generator, and a DOT generator which draws the output automaton. C2fsm is also able to do elementary transformations on the automaton, such as eliminating useless states, transitions and variables, simplifying guards, or selecting cut-points, i.e., program points on loops that can be used by RanK to prove program termination.

5.11. RanK

Participants: Christophe Alias, Alain Darte, Paul Feautrier, Laure Gonnord [Compsys/LIFL].

RanK is a software tool that can prove the termination of a program (in some cases) by computing a *ranking function*, i.e., a mapping from the operations of the program to a well-founded set that *decreases* as the computation advances. In case of success, RanK can also provide an upper bound of the worst-case time complexity of the program as a symbolic affine expression involving the input variables of the program (parameters), when it exists. In case of failure, RanK tries to prove the non-termination of the program and then to exhibit a counter-example input. This last feature is of great help for program understanding and debugging, and has already been experimented.

The input of RanK is an integer automaton, computed by C2fsm (see Section 5.10), representing the control structure of the program to be analyzed. RanK uses the Aspic tool, developed by Laure Gonnord during her PhD thesis, to compute automaton invariants. RanK has been used to discover successfully the worst-case time complexity of many benchmarks programs of the community. It uses the libraries Piplib and Polylib.

RanK has been implemented by Christophe Alias, using the compiler infrastructure PoCo. It represents more than 3000 lines of C++.

5.12. SToP

Participants: Christophe Alias, Guillaume Andrieu [LIFL], Laure Gonnord [Compsys/LIFL].

SToP (Scalable Termination of Programs) is the implementation of the modular termination technique presented in Section 6.7. It takes as input a large irregular C program and conservatively checks its termination. To do so, SToP generates a set of small programs whose termination implies the termination of the whole input program. Then, the termination of each small program is checked thanks to RanK. In case of success, SToP infers a ranking (schedule) for the whole program. This schedule can be used in a subsequent analysis to optimize the program.

SToP represents more than 2000 lines of C++.

5.13. Simplifiers

Participant: Paul Feautrier.

The aim of the `simple` library is to simplify Boolean formulas on affine inequalities. It works by detecting redundant inequalities in the representation of the subject formula as an ordered binary decision diagram (OBDD). It uses PIP for testing the feasibility – or unfeasibility – of a conjunction of affine inequalities.

The library is written in Java and is presented as a collection of class files. For experimentation, several front-ends have been written. They differ mainly in their input syntax, among which are a C like syntax, the Mathematica and SMTLib syntaxes, and an ad hoc Quast (quasi-affine syntax tree) syntax.

5.14. LAO Developments in Aggressive Compilation

Participants: Benoit Boissinot, Florent Bouchez, Florian Brandner, Quentin Colombet, Alain Darte, Benoît Dupont de Dinechin [Kalray], Christophe Guillon [STMicroelectronics], Sebastian Hack [Former post-doc in Compsys], Fabrice Rastello, Cédric Vincent [Former student in Compsys].

Our aggressive optimization techniques are all implemented in stand-alone experimental tools (as for example for register coalescing algorithms) or within LAO, the back-end compiler of STMicroelectronics, or both. They concern SSA construction and destruction, instruction-cache optimizations, register allocation. Here, we report only our more recent activities, which concern register allocation.

Our developments on register allocation within the STMicroelectronics compiler started when Cédric Vincent (bachelor degree, under Alain Darte supervision) developed a complete register allocator in LAO, the assembly-code optimizer of STMicroelectronics. This was the first time a complete implementation was done with success, outside the MCDT (now CEC) team, in their optimizer. This continued with developments made during the master internships and PhD theses of Florent Bouchez, Benoit Boissinot, and Quentin Colombet, and post-doctoral works of Sebastian Hack and Florian Brandner. In 2009, Quentin Colombet started to develop and integrate into the main trunk of LAO a full implementation of a two-phases register allocation. This implementation now includes two different decoupled spilling phases, the first one as described in Sebastian Hack's PhD thesis and a second ILP-based solution. It also includes an up-to-date graph-based register coalescing. Finally, since all these optimizations take place under SSA form, it includes also a mechanism for going out of colored-SSA (register-allocated SSA) form that can handle critical edges and does further optimizations.

5.15. LAO Developments in JIT Compilation

Participants: Benoit Boissinot, Florian Brandner, Quentin Colombet, Alain Darte, Benoît Dupont de Dinechin [Kalray], Christophe Guillon [STMicroelectronics], Fabrice Rastello.

The other side of our work in the STMicroelectronics compiler LAO has been to adapt the compiler to make it more suitable for JIT compilation. This means lowering the time and space complexity of several algorithms. In particular we implemented our fast out-of-SSA translation method, and we programmed and tested various ways to compute the liveness information. Recent efforts also focused on developing a tree-scan register allocator for the JIT part of the compiler, in particular a JIT conservative coalescing. The technique is to bias the tree-scan coalescing, taking into account register constraints, with the result of a JIT aggressive coalescing.

5.16. Low-Level Exchange Format (TireX) and Minimalist Intermediate Representation (MinIR)

Participants: Christophe Guillon [STMicroelectronics], Fabrice Rastello, Benoît Dupont de Dinechin [Kalray].

Most compilers define their own intermediate representation (IR) to be able to work on a program. Sometimes, they even use a different representation for each representation level, from source code parsing to the final object code generation. MinIR (Minimalist Intermediate Representation) is a new intermediate representation, designed to ease the interconnection of compilers, static analyzers, code generators, and other tools. In addition to the specification of MinIR, generic core tools have been developed to offer a basic toolkit and to help the connection of client tools. MinIR generators exist for several compilers, and different analyzers are developed as a testbed to rapidly prototype different static analyses over SSA code. This new common format enables the comparison of the code generator of several production compilers, and simplifies the connection of external tools to existing compilers.

MinIR has been extended into TireX, a Textual Intermediate Representation for EXchanging target-level information between compiler optimizers and whole or parts of code generators (aka compiler back-end). The first motivation for this intermediate representation is to factor target-specific compiler optimizations into a single component, in case several compilers need to be maintained for a particular target (e.g., operating system compiler and application code compiler). Another motivation is to reduce the run-time cost of JIT compilation and of mixed mode execution, since the program to compile is already in a representation lowered to the level of the target processor. Beside the lowering at the target level, the extensions of MinIR include the program data stream and loop scoped information. TireX is currently produced by the Open64/Path64 and the LLVM compilers, with a GCC producer under work. It is used by the LAO code generator.

Detailed information, generic core tools, and LLVM IR based generator for MinIR are available at <http://www.assembla.com/spaces/minir-dev/wiki>. Open64/Path64 emitter for TireX and its LAO back-end are available at <https://compilation.ens-lyon.fr/>. MinIR was presented at WIR'11 [28].

CONCHA Project-Team

5. Software

5.1. C++ library Concha

Participants: Roland Becker, Daniela Capatina, Robert Luce, David Trujillo.

The objectives of our library CONCHA are to offer a flexible and extensible software with respect to:

- Numerical methods and
- Physical models.

The aim is to have a flexible code which could easily switch between the different discretizations, in order to provide a toolbox for rapid testing of new ideas.

The software architecture is designed in such a way that a group of core developers can contribute in an efficient manner, and that independent development of different physical applications is possible. Further, in order to accelerate the integration of new members and in order to provide a basis for our educational purposes (see Section 8.1), the software proposes different entrance levels. The basic structure consists of a common block, and several special libraries which correspond to the different fields of applications described in Sections 4.1–4.4 Hyperbolic solvers, Low-Mach number flow solvers, DNS, and viscoelastic flows. A more detailed description of each special library may be found below. In order to coordinate the cooperative development of the library, Concha is based on the Inria-Gforge.

5.2. User interface and python interface

Participants: Roland Becker, David Trujillo.

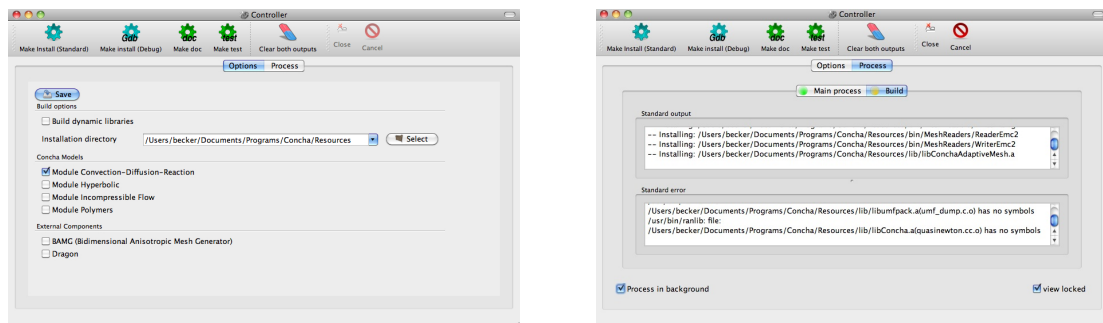


Figure 3. Graphical user interface: option panel (left) and process panel (right) of the install tool.

We are confronted with heterogenous backgrounds and levels of implication of the developers and users. It seems therefore crucial to be able to respond to the different needs. Our aim is to facilitate the development of the library, and at the same time, to make it possible that our colleagues involved in physical modeling can have access to the functionality of the software with a reasonable investment of time. Two graphical user interfaces have been developed: one for the installation of the library and another one for the building and execution of projects. They are based on common database and scripts written in python. The scripts can also be launched in a shell. In Figure 3 the user interface of the install tool is shown. The option panel allows to choose the components for conditional compilation and the compilation type (debug and release).

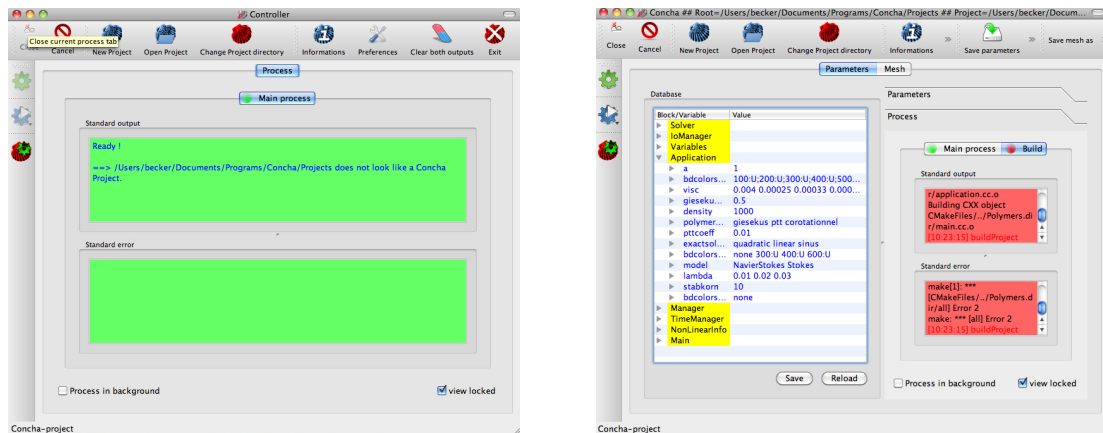


Figure 4. Graphical user interface: project build panel (left) and parameter panel (right) of the project tool.

In Figure 4 the user interface of the project tool is shown. A project consists of a number of sources files and a parameter file used by the C++-executable. The sources define classes derived from the library, which are used to specify certain data such as boundary conditions and employed finite element spaces. The parameter file contains algorithmic information and physical parameters. It is generated from a database by the python utilities.

The tools offered by this development platform are based on a python interface for the library, called pyConcha. It offers a common interface, based on a pluggin-system, which allows the developpement of command line tools in parallel. This year the consolidation of the interface part of pyConcha has been an important task. The pyConcha library is now a framework rather than a simple interface to Concha C++ library. It allows now creation of plugins, so that each user-programmer can customize pyConcha to his own goals. Previously, two main programs were working: concha-install.py to install library, and concha-project.py for (semi-)end-users. Both are now plugins of pyConcha, and can be launched by pyConcha at startup. A plugin visualization could now be developed in an independant way, and launched by pyConcha on demand.

The structure of pyConcha framework is clearly splitted in various modules(layers): Command Line Interface module, Graphical User Interface module and Handlers modules, see Figure 5 . A great effort has been made for internationalization of pyConcha.

5.3. Euler equations

Participants: Roland Becker, Kossivi Gokpi, Robert Luce, Eric Schall, David Trujillo.

Based on the library CONCHA we have developed a solver for hyperbolic PDE's based on DGFEM. So far different standard solvers for the Euler equations such as Lax-Friedrichs, Steger-Warming, and HLL have been implemented for test problems. A typical example is the scram jet test case shown in Figure 6 .

5.4. Incompressible flow solvers

Participants: Roland Becker, Daniela Capatina, Robert Luce, David Trujillo.

We have started the validation of the implementation of different finite element methods for incompressible flows at hand of standard benchmark problems as the Stokes flow around a symmetric cylinder [65] and the stationary flow around a slightly non symmetric cylinder [70], see Figure 7 .

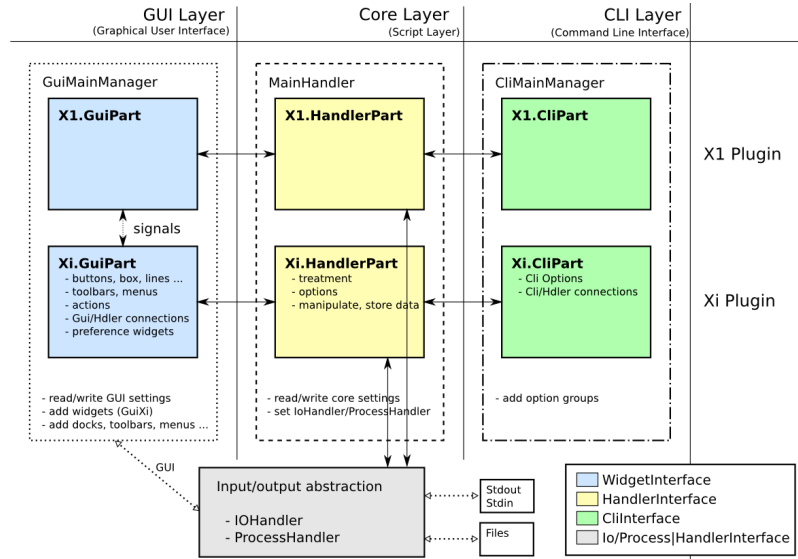


Figure 5. Structure of the pyConcha framework.

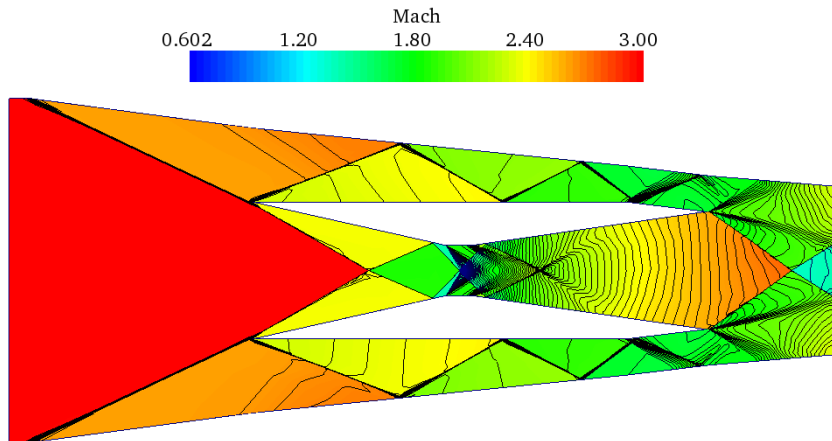


Figure 6. Computed Mach-number distribution for the Scramjet test problem.

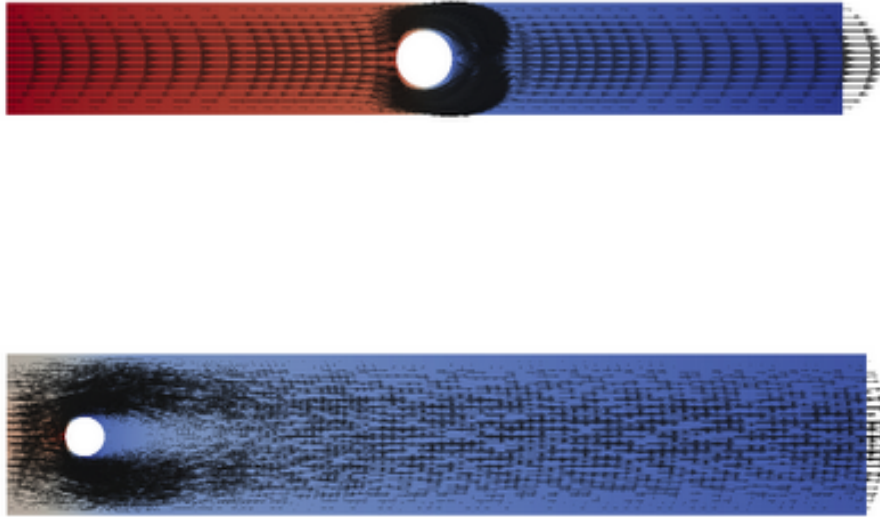


Figure 7. Flow fields for the Stokes (above) and Navier-Stokes (below) benchmark.

5.5. DNS

Participants: Roland Becker, David Trujillo.

For the direct numerical simulation of incompressible turbulent flows, we have started to develop a special solver based on structured meshes with a fast multigrid algorithm incorporating projection-like schemes. The main idea is to use non-conforming finite elements for the velocities with piecewise constant pressures, leading to a special structure of the discrete Schur complement, when an explicit treatment of the convection and diffusion term is used.

5.6. Validation and comparison with other CFD-software

Participants: Roland Becker, Didier Graebbling, Eric Schall, David Trujillo.

Validation and comparison with other CFD-software is crucial in order to evaluate the potential of our numerical schemes concerning accuracy, computing time and other practical aspects.

We have compared the Concha library for incompressible and compressible flows. For incompressible flows, we have used a test case proposed by Hulsén and the well-known Schäfer-Turek cylinder benchmark in order to validate the accuracy of the Stokes and Navier-Stokes solvers. The viscoelastic code has been compared with PolyFlow for different test configurations.

The compressible Euler code has been compared to the ELSA software developed by ONERA.

For further comparison and validation, it would be important to consider other commercial and research tools such as: *Aéro3* (Inria-Smash), AVBP (CERFACS), Fluent (ANSYS), and OpenFOAM (OpenCfd).

For this purpose we have proposed the ADT-project VALSE in collaboration with a small company involved in aerodynamics (EPSILON Toulouse), which unfortunately has been rejected by Inria.

CONTRAINTE Project-Team

5. Software

5.1. BIOCHAM, biochemical abstract machine

Participants: François Fages, Steven Gay, Sylvain Soliman.

The Biochemical Abstract Machine **BIOCHAM** [18] is a modeling environment for systems biology distributed as open-source since 2003. Current version is v3.4, released in October. **BIOCHAM** uses a compositional rule-based language for modeling biochemical systems, allowing patterns for expressing set of rules in a compact form. This rule-based language is compatible with the Systems Biology Markup Language (**SBML**) and is interpreted with three semantics corresponding to three abstraction levels:

1. the boolean semantics (presence or absence of molecules),
2. the stochastic semantics (discrete numbers of molecules),
3. the differential semantics (concentrations of molecules).

Based on this formal framework, **BIOCHAM** features:

- Boolean and numerical simulators (Rosenbrock's method for the differential semantics, Gillespie's algorithm with tau lipping for the stochastic semantics);
- a temporal logic language (CTL for qualitative models and $LTL(R_{lin})$ with numerical constraints for quantitative models) for formalizing biological properties such as reachability, checkpoints, oscillations or stability, and checking them automatically with model-checking techniques;
- automatic search procedures to infer parameter values, initial conditions and even reaction rules from temporal logic properties;
- automatic detection of invariants, through constraint-based analysis of the underlying Petri net;
- an SBGN-compatible reaction graph editor;
- an event handler allowing the encoding of hybrid models and formalisms.

BIOCHAM is implemented in GNU-Prolog and interfaced to the symbolic model checker **NuSMV** and to the continuous optimization tool **CMAES** developed by the EPI TAO.

5.2. Nicotine

Participant: Sylvain Soliman.

Nicotine is a GNU Prolog framework dedicated to the analysis of Petri nets. It was originally built for the computation of invariants using GNU Prolog's CLP(FD) solver [5] but has been further extended to allow import/export of various Petri nets formats. It provides as independent modules different features that can sometimes also be integrated in **BIOCHAM**, like SEPI computation, or left aside, like **unambiguous ODE to Petri net conversion**, since a more general heuristic conversion has been developed for **BIOCHAM** [8], [19].

5.3. STSE, spatio-temporal simulation environment

Participant: Szymon Stoma.

The overall goal of this project is to provide a software platform gathering a set of open-source tools and workflows facilitating spatio-temporal simulations (preferably of biological systems) based on microscopy data. The framework currently contains modules to digitize, represent, analyze, and model spatial distributions of molecules in static and dynamic structures (e.g. growing). A strong accent is put on the experimental verification of biological models by actual, spatio-temporal data acquired using microscopy techniques. Project was initially started at Humboldt University Berlin and moved to Inria with its founder. Project webpage is: <http://stse-software.org>.

5.4. YeastImageToolkit

Participant: Szymon Stoma.

YeastImageToolkit is an extension of YeastTracker software started originally by Jannis Uhlendorf. It allows following single cells in movies and quantifying fluorescent images based on this tracking as well as creating cell lineages. The software is currently under development and is designed to be a CellProfiler plugin facilitating yeast cell tracking, lineage and fluorescent signal quantification. Project webpage is: <http://yeast-sectrack.weebly.com/>.

5.5. SBMC, systems biology model-checker

Participant: Szymon Stoma.

Systems Biology Model Checker (SBMC) is a webservice allowing to verify biological models (e.g. signaling pathways stored in SBML files) against their specifications given in Signal Temporal Logics (STL). This project aims at providing to a large audience the methods described in [21] and used to analyse extensic apoptosis pathway. Project webpage is: [SBMC](#).

5.6. FO-CTL(R_{lin}), first-order computation tree logic over the reals

Participants: François Fages, Thierry Martinez.

FO-CTL(R_{lin}) is a solver for full First-Order Computation Tree Logic with linear arithmetic over the reals in constrained transition systems (CTS). CTS are transition systems where both states and transitions are described with constraints. FO-CTL(R_{lin}) generalizes the implementation done in Biocham of LTL(R_{lin}) for linear traces to branching Kripke structure.

5.7. Rules2CP

Participants: François Fages, Raphaël Martin, Thierry Martinez.

Rules2CP is a rule-based modeling language for constraint programming. It is distributed since 2009 as open-source. Unlike other modeling languages for constraint programming, Rules2CP adopts a single knowledge representation paradigm based on rules without recursion, and a restricted set of data structures based on records and enumerated lists given with iterators. This allows us to model complex constraint satisfaction problems together with search strategies, where search trees are expressed by logical formulae and heuristic choice criteria are defined with preference orderings by pattern-matching on the rules' left-hand sides.

The expressiveness of Rules2CP has been illustrated in the FP6 Strep project **Net-WMS** by a complete library for packing problems, called PKML (Packing Knowledge Modeling Library), which, in addition to pure bin packing and bin design problems, can deal with common sense rules about weights, stability, as well as specific packing business rules.

5.8. SiLCC, linear concurrent constraint programming

Participant: Thierry Martinez.

SiLCC is an extensible modular concurrent constraint programming language relying upon linear logic. It is a complete implementation of the Linear logic Concurrent Constraint programming paradigm of Saraswat and Lincoln using the formal semantics of Fages, Ruet and Soliman. It is a single-paradigm logical language, enjoying concurrency, imperative traits, and a clean module system allowing to develop hierarchies of constraint systems within the language.

This software prototype is used to study the design of hierarchies of extensible libraries of constraint solvers. SiLCC is also considered as a possible implementation language for restructuring the code of **BIOCHAM**.

5.9. EMoP, existential modules for Prolog

Participant: Thierry Martinez.

EMoP is an extension of Prolog with first-class modules. These modules have the formal semantics of the LCC modules and provide Prolog with notions of namespaces, closures and objects within a simple programming model. Modules are also the support for user-definition of macros and modular syntax extensions. EMoP is bootstrapped and uses the GNU Prolog compilation chain as back-end.

5.10. CHRat, CHR with ask and tell

Participant: Thierry Martinez.

CHRat is a modular version of the well known Constraint Handling Rules language CHR, called for CHRat for CHR with *ask* and *tell*. Inspired by the LCC framework, this extension of CHR makes it possible to reuse CHRat components both in rules and guards in other CHRat components, and define hierarchies of constraint solvers. CHRat is a bootstrapped preprocessor for CHR which generates code for SWI/Prolog.

5.11. CLPGUI, constraint logic programming graphical user interface

Participant: François Fages.

CLPGUI is a generic graphical user interface written in Java for constraint logic programming. It is available for GNU-Prolog and SICStus Prolog. CLPGUI has been developed both for teaching purposes and for debugging complex programs. The graphical user interface is composed of several windows: one main console and several dynamic 2D and 3D viewers of the search tree and of finite domain variables. With CLPGUI it is possible to execute incrementally any goal, backtrack or recompute any state represented as a node in the search tree. The level of granularity for displaying the search tree is defined by annotations in the CLP program.

CLPGUI has been mainly developed in 2001 and is distributed as third-party software on GNU-Prolog and SICStus Prolog web sites. In 2009, CLPGUI has been interfaced to Rules2CP/PKML and used in the FP6 Strep **Net-WMS** with a non-released version.

CONVECS Team

5. Software

5.1. The CADP Toolbox

Participants: Hubert Garavel [correspondent], Frédéric Lang, Radu Mateescu, Wendelin Serwe.

We maintain and enhance CADP (*Construction and Analysis of Distributed Processes* – formerly known as *CAESAR/ALDEBARAN Development Package*) [4], a toolbox for protocols and distributed systems engineering (see <http://cadp.inria.fr>). In this toolbox, we develop and maintain the following tools:

- CAESAR.ADT [41] is a compiler that translates LOTOS abstract data types into C types and C functions. The translation involves pattern-matching compiling techniques and automatic recognition of usual types (integers, enumerations, tuples, etc.), which are implemented optimally.
- CAESAR [48], [47] is a compiler that translates LOTOS processes into either C code (for rapid prototyping and testing purposes) or finite graphs (for verification purposes). The translation is done using several intermediate steps, among which the construction of a Petri net extended with typed variables, data handling features, and atomic transitions.
- OPEN/CAESAR [42] is a generic software environment for developing tools that explore graphs on the fly (for instance, simulation, verification, and test generation tools). Such tools can be developed independently of any particular high level language. In this respect, OPEN/CAESAR plays a central role in CADP by connecting language-oriented tools with model-oriented tools. OPEN/CAESAR consists of a set of 16 code libraries with their programming interfaces, such as:
 - CAESAR_GRAPH, which provides the programming interface for graph exploration,
 - CAESAR_HASH, which contains several hash functions,
 - CAESAR_SOLVE, which resolves Boolean equation systems on the fly,
 - CAESAR_STACK, which implements stacks for depth-first search exploration, and
 - CAESAR_TABLE, which handles tables of states, transitions, labels, etc.

A number of tools have been developed within the OPEN/CAESAR environment, among which:

- BISIMULATOR, which checks bisimulation equivalences and preorders,
- CUNCTATOR, which performs on-the-fly steady-state simulation of continuous-time Markov chains,
- DETERMINATOR, which eliminates stochastic nondeterminism in normal, probabilistic, or stochastic systems,
- DISTRIBUTOR, which generates the graph of reachable states using several machines,
- EVALUATOR, which evaluates regular alternation-free μ -calculus formulas,
- EXECUTOR, which performs random execution,
- EXHIBITOR, which searches for execution sequences matching a given regular expression,
- GENERATOR, which constructs the graph of reachable states,
- PROJECTOR, which computes abstractions of communicating systems,
- REDUCTOR, which constructs and minimizes the graph of reachable states modulo various equivalence relations,
- SIMULATOR, XSIMULATOR, and OCIS, which enable interactive simulation, and
- TERMINATOR, which searches for deadlock states.

- BCG (*Binary Coded Graphs*) is both a file format for storing very large graphs on disk (using efficient compression techniques) and a software environment for handling this format. BCG also plays a key role in CADP as many tools rely on this format for their inputs/outputs. The BCG environment consists of various libraries with their programming interfaces, and of several tools, such as:
 - BCG_DRAW, which builds a two-dimensional view of a graph,
 - BCG_EDIT, which allows the graph layout produced by BCG_DRAW to be modified interactively,
 - BCG_GRAPH, which generates various forms of practically useful graphs,
 - BCG_INFO, which displays various statistical information about a graph,
 - BCG_IO, which performs conversions between BCG and many other graph formats,
 - BCG_LABELS, which hides and/or renames (using regular expressions) the transition labels of a graph,
 - BCG_MIN, which minimizes a graph modulo strong or branching equivalences (and can also deal with probabilistic and stochastic systems),
 - BCG_STEADY, which performs steady-state numerical analysis of (extended) continuous-time Markov chains,
 - BCG_TRANSIENT, which performs transient numerical analysis of (extended) continuous-time Markov chains, and
 - XTL (*eXecutable Temporal Language*), which is a high level, functional language for programming exploration algorithms on BCG graphs. XTL provides primitives to handle states, transitions, labels, *successor* and *predecessor* functions, etc.

For instance, one can define recursive functions on sets of states, which allow evaluation and diagnostic generation fixed point algorithms for usual temporal logics (such as HML [50], CTL [37], ACTL [38], etc.) to be defined in XTL.
- PBG (*Partitioned BCG Graph*) is a file format implementing the theoretical concept of *Partitioned LTS* [46] and providing a unified access to a graph partitioned in fragments distributed over a set of remote machines, possibly located in different countries. The PBG format is supported by several tools, such as:
 - PBG_CP, PBG_MV, and PBG_RM, which facilitate standard operations (copying, moving, and removing) on PBG files, maintaining consistency during these operations,
 - PBG_MERGE (formerly known as BCG_MERGE), which transforms a distributed graph into a monolithic one represented in BCG format,
 - PBG_INFO, which displays various statistical information about a distributed graph.
- The connection between explicit models (such as BCG graphs) and implicit models (explored on the fly) is ensured by OPEN/CAESAR-compliant compilers, e.g.:
 - BCG_OPEN, for models represented as BCG graphs,
 - CAESAR.OPEN, for models expressed as LOTOS descriptions,
 - EXP.OPEN, for models expressed as communicating automata,
 - FSP.OPEN, for models expressed as FSP [55] descriptions,
 - LNT.OPEN, for models expressed as LNT descriptions, and
 - SEQ.OPEN, for models represented as sets of execution traces.

The CADP toolbox also includes TGV (*Test Generation based on Verification*), which has been developed by the VERIMAG laboratory (Grenoble) and the VERTECS project team at Inria Rennes – Bretagne-Atlantique.

The CADP tools are well-integrated and can be accessed easily using either the EUCALYPTUS graphical interface or the SVL [43] scripting language. Both EUCALYPTUS and SVL provide users with an easy and uniform access to the CADP tools by performing file format conversions automatically whenever needed and by supplying appropriate command-line options as the tools are invoked.

5.2. The TRAIAN Compiler

Participants: Hubert Garavel [correspondent], Frédéric Lang.

We develop a compiler named TRAIAN for translating LOTOS NT descriptions into C programs, which will be used for simulation, rapid prototyping, verification, and testing.

The current version of TRAIAN, which handles LOTOS NT types and functions only, has useful applications in compiler construction [44], being used in all recent compilers developed by the CONVECS team.

The TRAIAN compiler can be freely downloaded from the CONVECS Web site (see <http://convecs.inria.fr/software/traian>).

5.3. The PIC2LNT Translator

Participants: Radu Mateescu, Gwen Salaün [correspondent].

We develop a translator named PIC2LNT from an applied π -calculus (see Section 6.1) to LNT, which enables the analysis of concurrent value-passing mobile systems using CADP.

PIC2LNT is developed by using the SYNTAX tool (developed at Inria Paris-Rocquencourt) for lexical and syntactic analysis together with LOTOS NT for semantical aspects, in particular the definition, construction, and traversal of abstract trees.

The PIC2LNT translator can be freely downloaded from the CONVECS Web site (see <http://convecs.inria.fr/software/pic2lnt>).

COPRIN Project-Team

5. Software

5.1. Introduction

Software development is an essential part of the research done by COPRIN since a large part of our methods can only be validated experimentally (both for our numerical experiments and in robotics). Software developments follow various directions:

1. interval arithmetic: although we do not plan to work in this very specialized area (we generally rely on existing packages) interval arithmetic is an important part of our interval analysis algorithms and we may have to modify the existing packages so as to deal, in particular, with multi-precision and arithmetic extensions
2. interval analysis libraries: we daily use the ALIAS library that has been designed in the project and is still under development. A long term work is to develop a generic programming framework that allows for modularity and flexibility, with the objectives of testing new functionalities easily and building specific solvers by a simple juxtaposition of existing modules
3. interface to interval analysis: in our opinion interval analysis software must be available within general purpose scientific software (such as Maple, Mathematica, Scilab) and not only as a stand-alone tool. Indeed most end-users are reluctant to learn a new programming language just to solve problems that are only small elements of a more general problem. Furthermore interval analysis efficiency may benefit from the functionalities available in the general purpose scientific software.

5.2. Interval analysis libraries

5.2.1. ALIAS

Participants: David Daney, Jean-Pierre Merlet [correspondant], Odile Pourtallier.

The ALIAS library (*Algorithms Library of Interval Analysis for Systems*), whose development started in 1998, is a collection of procedures based on interval analysis for systems solving and optimization.

ALIAS is made of two parts:

- ALIAS-C++: the C++ library (87 000 code lines) which is the core of the algorithms
- ALIAS-Maple: the Maple interface for ALIAS-C++ (55 000 code lines). This interface allows one to specify a solving problem within Maple and get the results within the same Maple session. The role of this interface is not only to generate the C++ code automatically, but also to perform an analysis of the problem in order to improve the efficiency of the solver. Furthermore, a distributed implementation of the algorithms is available directly within the interface.

Although these libraries are intended to be used within the project-team they can be freely downloaded as a library file (but the user may introduce its own code in several part of the package) and has been used for example at LIRMM and IRCCyN.

5.2.2. Int4Sci : a Scilab interface for interval analysis

Participants: David Daney, Gilles Trombettoni, Bertrand Neveu.

In 2006, we have started the development of a Scilab interface to C++ Bias/Profil interval arithmetic package and to the library ALIAS. The first version of Int4Sci has been released in 2008 – see <http://www-sop.inria.fr/coprin/logiciels/Int4Sci/> for linux, MacOS and Windows. A second version, compatible with Scilab 5.3 is in preparation . This interface provides an interval arithmetic, basic interval manipulation tools as well as the solving of linear interval systems. All functions are documented and a tutorial is available. Int4Sci is used in several universities for teaching the basis of interval analysis in place of using Rump's INTLAB for Matlab. We however lack the manpower to further enhance this software.

5.2.3. Mathematica Interface to Interval Analysis

Participants: Yves Papegay [correspondant], Jean-Pierre Merlet.

Since 2006, we have been implementing in Mathematica a high-level modular interface to the ALIAS library. Lack of manpower has slowed down this development.

CORIDA Project-Team

5. Software

5.1. Simulation of viscous fluid-structure interactions

Participants: Takeo Takahashi [correspondant], Jean-François Scheid, Jérôme Lohéac.

A number of numerical codes for the simulation for fluids and fluid-structure problems has been developed by the team. These codes are mainly written in MATLAB Software with the use of C++ functions in order to improve the sparse array process of MATLAB. We have focused our attention on 3D simulations which require large CPU time resources as well as large memory storage. In order to solve the 3D Navier-Stokes equations which model the viscous fluid, we have implemented an efficient 3D Stokes sparse solver for MATLAB and a 3D characteristics method to deal with the nonlinearity of Navier-Stokes equations. This year, we have also started to unify our 2D fluid-structure codes (fluid alone, fluid with rigid bodies and fluid with fishes).

Another code has been developed in the case of self-propelled deformable object moving into viscous fluid. Our aim is to build a deformable ball which could swim in a viscous fluid. In order to do this we have started a collaboration with a team from the CRAN (Research Centre for Automatic Control). This software solves numerically 3D Stokes equations using finite elements methods. The source code is written for use with MATLAB thanks to a C++ library developed by ALICE.

- Version: v0.5
- Programming language: MATLABc++

5.2. Fish locomotion in perfect fluids with potential flow

Participants: Alexandre Munnier [correspondant], Marc Fuentes, Bruno Pinçon.

SOLEIL is a Matlab suite to simulate the self-propelled swimming motion of a single 3D swimmer immersed in a potential flow. The swimmer is modeled as a shape-changing body whose deformations can be either prescribed as a function of time (simulation of the direct swimming problem) or computed in such a way that the swimmer reaches a prescribed location (control problem). For given deformations, the hydrodynamical forces exerted by the fluid on the swimmer are expressed as solutions of 2D integral equations on the swimmer's surface, numerically solved by means of a collocation method.

SOLEIL is free, distributed under licence GPL v3. More details are available on the project web page <http://soleil.gforge.inria.fr/>.

The next step of SOLEIL (under progress) is to take into account a fluid whose flow is governed by Stokes equations.

- Version: 0.1
- Programming language: Matlab/C++

CORTEX Project-Team

5. Software

5.1. Spiking neural networks simulation

Participants: Dominique Martinez, Yann Boniface.

A spiking neuron is usually modeled as a differential equation describing the evolution over time of its membrane potential. Each time the voltage reaches a given threshold, a spike is sent to other neurons depending on the connectivity. A spiking neural network is then described as a system of coupled differential equations. For the simulation of such a network we have written two simulation engines : (i) Mvaspike based on an event-driven approach and (ii) sirene based on a time-driven approach.

- Mvaspike : The event-driven simulation engine was developed in C++ and is available on <http://mvaspike.gforge.inria.fr>. Mvaspike is a general event-driven purpose tool aimed at modeling and simulating large, complex networks of biological neural networks. It allows to achieve good performance in the simulation phase while maintaining a high level of flexibility and programmability in the modeling phase. A large class of spiking neurons can be used ranging from standard leaky integrate-and-fire neurons to more abstract neurons, e.g. defined as complex finite state machines.
- Sirene : The time-driven simulator engine was written in C and is available on <http://sirene.gforge.inria.fr>. It has been developed for the simulation of biologically detailed models of neurons —such as conductance-based neurons— and synapses. Its high flexibility allows the user to implement easily any type of neuronal or synaptic model and use the appropriate numerical integration routine (e.g. Runge-Kutta at given order).

5.2. DANA: Implementation of computational neuroscience mechanisms

Participants: Nicolas Rougier, Mathieu Lefort, Wahiba Taouali.

Computational neuroscience is a vast domain of research going from the very precise modeling of a single spiking neuron, taking into account ion channels and/or dendrites spatial geometry up to the modeling of very large assemblies of simplified neurons that are able to give account of complex cognitive functions. DANA attempts to address this latter modeling activity by offering a Python computing framework for the design of very large assemblies of neurons using numerical and distributed computations. However, there does not exist something as a unified model of neuron: if the formal neuron has been established some sixty years ago, there exists today a myriad of different neuron models that can be used within an architecture. Some of them are very close to the original definition while some others tend to refine it by providing extra parameters or variables to the model in order to take into account the great variability of biological neurons. DANA makes the assumption that a neuron is essentially a set of numerical values that can vary over time due to the influence of other neurons and learning. DANA aims at providing a constrained and consistent Python framework that guarantee this definition to be enforced anywhere in the model, i.e., no symbol, no homonculus, no central executive.

5.3. ENAS: Event Neural Assembly Simulation

Participants: Frédéric Alexandre, Axel Hutt, Nicolas Rougier, Thierry Viéville.

EnaS (that stands for “Event Neural Assembly Simulation”) is a middleware implementing our last numerical and theoretical developments, allowing to simulate and analyze so called "event neural assemblies". The recent achievements include (in collaboration with the Neuromathcomp EPI): spike trains statistical analysis via Gibbs distributions, spiking network programing for exact event’s sequence restitution, discrete neural field parameters algorithmic adjustments and time-constrained event-based network simulation reconciling clock and event based simulation methods. It has been designed as plug-in for our simulators (e.g. DANA or Mvaspike) as other existing simulators (via the NeuralEnsemble meta-simulation platform) and additional modules for computations with neural unit assembly on standard platforms (e.g. Python or the Scilab platform).

5.4. OpenViBE

Participants: Laurent Bougrain, Octave Boussaton.

OpenViBE is a C++ open-source software devoted to the design, test and use of Brain-Computer Interfaces. The OpenViBE platform consists of a set of software modules that can be integrated easily and efficiently to design BCI applications. Key features of the platform are its modularity, its high-performance, its portability, its multiple-users facilities and its connection with high-end/Virtual Reality displays. The “designer” of the platform enables to build complete scenarios based on existing software modules using a dedicated graphical language and a simple Graphical User Interface (GUI). This software is available on the Inria Forge under the terms of the LGPL-V2 license. The development of OpenVibe is done in association with the Inria research team BUNRAKU for the national Inria project: ADT LOIC (cf. § 7.2).

5.5. CLONES: Closed-Loop Neural Simulations

Participant: Thomas Voegtlin.

The goal of this work is to provide an easy-to-use framework for closed-loop simulations, where interactions between the brain and body of an agent are simulated.

We developed an interface between the Sofa physics engine, (<http://www.sofa-framework.org>) and the Brian neural simulator (<http://www.briansimulator.org>). The interface consists in a Sofa plugin and a Python module for Brian. Sofa and Brian use different system processes, and communicate via shared memory. Synchronization between processes is achieved through semaphores.

As a demonstration of this interface, a physical model of undulatory locomotion in the nematode *c. elegans* was implemented, based on the PhD work of Jordan H. Boyle.

5.6. GINNet-DynNet: Decision-making platform

Participant: Marie Tonnelier.

GINNet (Graphical Interface for Neural Networks) is a decision-aid platform written in Java, intended to make neural network teaching, use and evaluation easier, by offering various parametrizations and several data pre-treatments. GINNet is based upon a local library for dynamic neural network developments called DynNet. DynNet (Dynamic Networks) is an object-oriented library, written in Java and containing base elements to build neural networks with dynamic architecture such as Optimal Cell Damage and Growing Neural Gas. Classical models are also already available (multi-layer Perceptron, Kohonen self-organizing maps, ...). Variable selection methods and aggregation methods (bagging, boosting, arcing) are implemented too.

The characteristics of GINNet are the following: Portable (100% Java), accessible (model creation in few clicks), complete platform (data importation and pre-treatments, parametrization of every models, result and performance visualization). The characteristics of DynNet are the following: Portable (100% Java), extensible (generic), independent from GINNet, persistent (results are saved in HML), rich (several models are already implemented), documented.

This platform is composed of several parts:

1. Data manipulation: Selection (variables, patterns), descriptive analysis (stat., PCA..), detection of missing, redundant data.
2. Corpus manipulation: Variable recoding, permutation, splitting (learning, validation, test sets).
3. Supervised networks: Simple and multi-layer perceptron.
4. Competitive networks: Kohonen maps, Neural Gas, Growing Neural Gas.
5. Metalearning: Arcing, bagging, boosting.
6. Results: Error curves, confusion matrix, confidence interval.

DynNet and GINNet are free softwares, registered to the APP and distributed under CeCILL license, Java 1.4 compatible (<http://ginnet.gforge.inria.fr>). GINNet is available as an applet. For further information, see <http://gforge.inria.fr/projects/ginnet> (news, documentations, forums, bug tracking, feature requests, new releases...)

CQFD Project-Team (section vide)

DAHU Project-Team (section vide)

DANTE Team

5. Software

5.1. Sensor Network Tools: drivers, OS and more

Participants: Éric Fleury [correspondant], Sandrine Avakian.

As a outcomes of the ANR SensLAB project and the Inria ADT SensTOOLS and SensAS, several softwares (from low level drivers to OSes) were delivered and made available to the research community. The main goal is to lower the cost of developing/deploying a large scale wireless sensor network application. All software are gathered under the SensLAB web site: <http://www.senslab.info/> web page where one can find:

- low C-level drivers to all hardware components;
- ports of the main OS, mainly TinyOS, FreeRTOS and Contiki;
- ports and development of higher level library like routing, localization.

5.2. <http://queueing-systems.ens-lyon.fr>

Participant: Thomas Begin [correspondant].

Queueing models, steady-state solution, online tool, web interface

This tool aims at providing a simple web based interface to promote the use of our proposed solutions to numerically solve classical queueing systems. In 2011, the tools merely implemented the solution to get the distribution for the number of customers along with customary performance parameters for a queue with multiple servers, general arrivals, exponential services and a possibly finite buffer, (i.e., $Ph/M/c/N$ -like queue). The steady-state solution to this queue is based on a simple and stable recurrence [2] and was performed in collaboration with Pr. Brandwajn (UCSC). In 2012 we extended our tool so as to include the solution for a queue with a single server, Poisson arrivals, general services and a possibly finite buffer, (i.e., $M/Ph/1/N$ -like queue). Our tool was presented at the conference [43] and attracts hundreds of visitors each month. Associated URL is: <http://queueing-systems.ens-lyon.fr>

DART Project-Team

5. Software

5.1. Gaspard 2

Participants: Jean-Luc Dekeyser [correspondant], All DaRT team.

Gaspard2 is an Integrated Development Environment (IDE) for SoC visual co-modeling. It allows or will allow modeling, simulation, testing and code generation of SoC applications and hardware architectures. Its purpose is to provide a single environment for all the SoC development processes:

- High level modeling of applications and hardware architectures
- Application and hardware architecture association (mapping and scheduling)
- Application refactoring
- Deployment specification
- Model to model transformation (to automatically produce models for several target platforms)
- Code generation
- Simulation
- Reification of any stages of the development

The Gaspard2 tool is based on the Eclipse [35] IDE. A set of plugins provides the different functionalities. Gaspard2 provides an internal engine to execute transformation chains. This engine is able to run either QVT (OMG standard) or Java transformations. It is also able to run model-to-text transformations based on Accelleo [37]. The Gaspard2 engine is defined to execute models conform to an internal transformation chains meta-model. A GUI has been developed to specify transformation chain models by drawing them. For the final user, application, hardware architecture, association, deployment and technology models are specified and manipulated by the developer through UML diagrams, and saved by the UML tool in an XMI file format. Gaspard2 manipulates these models through repositories (Java interfaces and implementations) automatically generated thanks to the Ecore specification. Several transformation chains are provided with Gaspard2 to target, from UML models, several execution or simulation platforms (OpenMP, OpenCL, Pthread, SystemC, VHDL, ...). This input language is based on the MARTE UML profile. A tool to generate SIMD configurations derived from the mppSoC model was developed. It allows to automatically generate the VHDL code from a high specification modeled at a high abstraction level (UML model using MARTE profile) based on the IP mppSoC library. The developed tool facilitates to the user to choose a SIMD configuration adapted to his application needs. It has been integrated in the Gaspard environment. **Gaspard2 as an educational resource.** The Gaspard2 platform was one of the topics taught in the context of the courses on embedded systems in Telecom Lille and in a Master 2 (TNSI) lecture " Design tools for embedded systems" at the University of Valenciennes. These lectures focused on the potentiality to generate several targets from a subset of the Marte profile and the ability to target system on chip architectures at the TLM level respectively. Furthermore, the model driven engineering characteristics of Gaspard2 are largely detailed in the lecture of Software engineering at Polytech Lille and in the Master of research at university of Lille too.

- See also the web page <http://www.gaspard2.org/>
- Inria software evaluation: A-2, SO-4, SM-2, EM-1, SDL-2, DA-4, CD-4, MS-4, TPM4
- Version: 2.1.0

DEDUCTEAM Team

4. Software

4.1. Dedukti

Dedukti is a proof-checker for the $\lambda\Pi$ -calculus modulo. As it can be parametrized by an arbitrary set of rewrite rules, defining an equivalence relation, this calculus can express many different theories. Dedukti has been created for this purpose: to allow the interoperability of different theories.

Dedukti is designed to be versatile: it must be efficient on proofs that contain many computations—such as proofs by reflection—as well as proofs that do not contain any—such as proofs coming from HOL. These constraints has led us to adopt a Just-In-Time compilation architecture. And instead of designing our own JIT compiler, we have chosen to reuse the cutting-edge LuaJIT compiler. This technological choice, namely devolving the type-checking to Lua, makes Dedukti a proof-checker generator.

This has allowed the introduction of many optimizations: a normalization by evaluation strategy, a higher-order abstract syntax representation of terms and a context-free, bidirectional type-checking algorithm [22].

Dedukti has been developed by Mathieu Boespflug, Olivier Hermant, Quentin Carbonneaux, and Ronan Saillard.

4.2. CoqInE and HOLiDe

Dedukti comes with two companion tools: HOLiDe, an embedding of HOL proofs through the OpenTheory format [51], and CoqInE, an embedding of Coq proofs. Almost all the standard library of HOL and a significant part of that of Coq are checked by Dedukti.

CoqInE now supports the following features of Coq: the raw Calculus of Constructions, inductive types, and fixpoint definitions. It is now able to translate more than 80% of the standard library of Coq [21]. Ongoing work focuses on modules and functors, and on universes.

CoqInE has been developed by Mathieu Boespflug, Guillaume Burel, and Ali Assaf.

HOLiDe supports all the features of HOL, including polymorphism, constant definitions, and type definitions. It is able to translate all of the OpenTheory standard theory library.

HOLiDe has been developed by Ali Assaf.

4.3. iProver Modulo

iProver Modulo is an extension of the automated theorem prover iProver originally developed by Konstantin Korovin at the University of Manchester. It implements Ordered polarized resolution modulo, a refinement of the Resolution method based on Deduction modulo. It takes as input a proposition in predicate logic and a clausal rewriting system defining the theory in which the formula has to be proved. Normalization with respect to the term rewriting rules is performed very efficiently through translation into OCaml code, compilation and dynamic linking. Experiments have shown that Ordered polarized resolution modulo dramatically improves proof search compared to using raw axioms. iProver modulo is also able to produce proofs that can be checked by Dedukti, therefore improving confidence. iProver modulo is written in OCaml, it consists of 1,200 lines of code added to the original iProver.

It is developed by Guillaume Burel.

These four systems are available on the website of the team.

DEFI Project-Team

5. Software

5.1. RODIN

Participant: Grégoire Allaire [correspondant].

RODIN project: developing an industrial software for geometry and topology optimization of solid structures, based on the level set method.

5.2. FreeFem++ Toolboxes

5.2.1. Shape optimization toolbox in FreeFem++

Participants: Grégoire Allaire, Olivier Pantz.

We propose several FreeFem++ routines which allow the users to optimize the thickness, the geometry or the topology of elastic structures. All examples are programmed in two space dimensions. These routines have been written by G. Allaire, B. Boutin, C. Dousset, O. Pantz. A web page of this toolbox is available at http://www.cmap.polytechnique.fr/~allaire/freefem_en.html.

We also have written a C++ code to solve the Hamilton Jacoby equation used in the Level-set shape optimization method. This code has been linked with FreeFem++ routines.

5.2.2. Inverse shape and medium problem for thin coatings

Participant: Nicolas Chaulet.

We developed a FreeFem++ toolbox which retrieve an obstacle and two coefficients that define a generalized impedance boundary condition from a few far field data in dimension 2. The reconstruction algorithm relies on regularized non linear optimization technique. The toolbox also contains a forward solver for the scattering of acoustic waves by obstacle on which a generalized impedance boundary condition is applied using an approximate Dirichlet-to-Neuman map to bound the computational domain.

5.2.3. Inverse shape problems for axisymmetric eddy current problems

Participant: Zixian Jiang.

This FreeFem++ toolbox solves inverse problems for an axisymmetric eddy current model using shape optimization techniques. The underlying problem is to find inclusions in a tubular and unbounded domain. The direct scattering problems are solved using an adaptive finite element method, and Dirichlet-to-Neumann operators are used to implement the transparent boundary conditions. Based on the shape derivative of an inclusion with respect to the domain, the toolbox offers regularized iterative algorithms to solve the inverse problem.

5.2.4. Contact managements

Participant: Olivier Pantz.

We have developed a toolbox running under Freefem++ in order to take into account the non-intersection constraints between several deformable bodies. This code has been used to treat contacts between red blood cells in our simulations, but also between genuine non linear elastic structure. It can handle both contacts and self-contacts.

Moreover, a toolbox based on the Penalization method has also been developed.

5.2.5. De-Homogenization

Participant: Olivier Pantz.

We have developed a code under Freefem++ that implements our De-Homogenization method. It has been used to solve the compliance minimization problem of the compliance of an elastic shape. In particular, it enables us to recover well known optimal Michell's trusses for shapes of low density.

5.3. Scilab and Matlab Toolboxes

5.3.1. Shape optimization toolbox in Scilab

Participant: Grégoire Allaire [correspondant].

Together with Georgios Michailidis, we improved a Scilab toolbox for 2-d shape and topology optimization by the level set method which was originally produced by Anton Karrman and myself. The routines, a short user's manual and several examples are available on the web page: http://www.cmap.polytechnique.fr/~allaire/levelset_en.html

5.3.2. Conformal mapping method

Participant: Housseem Haddar [correspondant].

This Scilab toolbox is dedicated to the resolution of inverse 2-D electrostatic problems using the conformal mapping method introduced by Akdumann, Kress and Haddar. The toolbox treats the cases of a simply connected obstacle with Dirichlet, Neumann or impedance boundary conditions or a simply connected inclusion with a constant conductivity. The latest development includes the extension of the method to the inverse scattering problem at low frequencies as introduced by Haddar-Kress (2012).

5.3.3. Direct and inverse problems in waveguides

Participants: Armin Lechleiter [correspondant], Dinh Liem Nguyen.

This Matlab toolbox includes fast solvers for direct and inverse scattering problems in planar 3D waveguides for inhomogeneous media. The direct scattering problems are solved using an spectral integral equation approach relying on the Lippmann-Schwinger integral equation, discretized as a Galerkin method via the fast Fourier transform. The toolbox includes preconditioning by a two-grid scheme and multipole expansions coupled to the spectral solver to allow for multiple scattering objects. The inverse problem to find the shape of the scattering object from near-field measurements is solved using a Factorization method.

5.4. Sampling methods for inverse problems

5.4.1. Samplings-2d

Participant: Housseem Haddar [correspondant].

This software is written in Fortran 90 and is related to forward and inverse problems for the Helmholtz equation in 2-D. It includes three independent components. The first one solves to scattering problem using integral equation approach and supports piecewise-constant dielectrics and obstacles with impedance boundary conditions. The second one contains various samplings methods to solve the inverse scattering problem (LSM, RGLSM(s), Factorization, MuSiC) for near-field or far-field setting. The third component is a set of post processing functionalities to visualize the results

See also the web page <http://sourceforge.net/projects/samplings-2d/>.

- License: GPL
- Type of human computer interaction: sourceforge
- OS/Middleware: Linux
- Programming language: Fortran
- Documentation: fichier

5.4.2. Samplings-3d

Participant: Housseem Haddar [correspondant].

This software is written in Fortran 90 and is related to forward and inverse problems for the Helmholtz equation in 3-D. It contains equivalent functionalities to samplings-2d in a 3-D setting.

5.4.3. *Time domain samplings-2d*

Participants: Housseem Haddar [correspondant], Armin Lechleiter.

This software is written in Fortran 90 and is related to forward and inverse problems for the time dependent wave equation in 2-D. The forward solver is based on a FDTD method with PMLs. The inverse part is an implementation of the linear sampling method in a near field setting and the factorization method in a far field setting.

5.4.4. *Factorization Method for EIT*

Participant: Giovanni Migliorati.

We developed a numerical code that implements the Factorization Method applied to the Continuous Model, in the framework of Electrical Impedance Tomography featuring an inhomogeneous background. The numerical scheme relies on the approximation by the finite element method of the solution to the dipole-like Neumann boundary-value problem. Two regularization techniques are implemented, i.e. the Tikhonov regularization embedding Morozov principle, and the classical Picard Criterion. The code now supports the case of piecewise-wise constant by unknown background.

5.5. BlochTorreyPDESolver

Participants: Jing-Rebecca Li [correspondant], Dang Van Nguyen.

We developed numerical codes to solve the multiple compartment Bloch Torrey partial differential equation in 2D and 3D to simulate the bulk magnetization of a sample under the influence of a diffusion-encoding gradient magnetic field. We coupled a mass-conserving spatial discretization with a stable time discretization using an explicit Runge-Kutta-Chebyshev method and we are able to solve the Bloch-Torrey PDE in multiple compartments for an arbitrary diffusion sequence with reasonable accuracy for moderately complicated geometries in computational time that is on the order of tens of minutes per bvalue on a laptop computer.

This code has been implemented in Fortran90, C++, as well as Matlab. A Matlab Toolbox with graphical user interface for the simulation of DMRI signals in 2D and 3D cellular geometries using this numerical method is being developed.

The version of the code using Finite Volume discretization on a Cartesian grid is complete (written by Jing-Rebecca Li). The version of the code using linear Finite Elements discretization is in the final testing phase (written by Dang Van Nguyen)

See the web page <http://www.cmap.polytechnique.fr/~jingrebecali/> for more details.

DEMAR Project-Team

5. Software

5.1. Software

5.1.1. *RdP to VHDL tool*

Participants: Gregory Angles, David Andreu, Thierry Gil.

Our SENIS (Stimulation Electrique Neurale dIStribuee) based FES architecture relies on distributed stimulation units (DSU) which are interconnected by means of a 2-wire based network. A DSU is a complex digital system since it embeds among others a dedicated processor (micro-machine with a specific reduced instruction set), a monitoring module and a 3-layer protocol stack. To face the complexity of the unit's digital part and to ease its prototyping on programmable digital devices (e.g. FPGA), we developed an approach for high level hardware component programming (HILECOP). To support the modularity and the reusability of sub-parts of complex hardware systems, the HILECOP methodology is based on components. An HILECOP component has: a Petri Net (PN) based behavior, a set of functions whose execution is controlled by the PN, and a set of variables and signals. Its interface contains places and transitions from which its PN model can be inter-connected as well as signals it exports or imports. The interconnection of those components, from a behavioral point of view, consists in the interconnection of places and/or transitions according to well-defined mechanisms: interconnection by means of oriented arcs or by means of the "merging" operator (existing for both places and transitions).

GALS (Globally Asynchronous Locally Synchronous) systems can be specified, connecting different clocks to HILECOP components, and interconnecting them by means of asynchronous signals.

Undergoing work includes the modification of the formalism in order to allow behavior aggregation as well as exception handling.

The Eclipse-based version of HILECOP is regularly updated. The last version of HILECOP (registered at the french Agence de Protection des Programmes (APP)) is accessible to the academic community (<http://www.lirmm.fr/~gil/Temp/>).

5.1.2. *SENISManager*

Participants: Robin Passama, David Andreu.

We developed a specific software environment called SENISManager allowing to remotely manage and control a network of DSUs, i.e. the distributed FES architecture. SENISManager performs self-detection of the architecture being deployed. This environment allows the manipulation of micro-programs from their edition to their remote control. It also allows the programming of control sequences executed by an external controller in charge of automatically piloting a stimulator.

A new version of SENISManager is under development according to an Eclipse-based design. This new version should be available in 2013.

DIONYSOS Project-Team

4. Software

4.1. T3devKit testing toolkit and IPv6 test suites

Participants: Anthony Baire, César Viho.

We have built a toolkit for easing executing tests written in the standardized TTCN-3 test specification language. This toolkit is made of a C++ library together with a highly customizable CoDec generator that allows fast development of external components (that are required to execute a test suite) such as CoDec (for message Coding/Decoding), System and Platform Adapters. It also provides a framework for representing and manipulating TTCN-3 events so as to ease the production of test reports. The toolkit addresses issues that are not yet covered by ETSI standards while being fully compatible with the existing standard interfaces: TRI (Test Runtime Interfaces) and TCI (Test Control Interfaces), it has been tested with four TTCN-3 environments (IBM, Elvior, Danet and Go4IT) and on three different platforms (Linux, Windows and Cygwin). It is publicly released under the CeCILL-C License.

All these tools with associated test suites (for RIPng, DHCPv6 and examples for DNS) are freely available at <http://www.irisa.fr/tipi>.

4.2. Interoperability Assessment

Participants: Anthony Baire, Nanxing Chen, Arulnambi Nandagoban, César Viho.

Our experience in interoperability assessment (since 1996) and in using the TTCN-3 standard allowed us to develop a tool (called `ttproto`) that helps in: (i) experimenting new concepts for long term evolution of the TTCN-3 standard [37] and (ii) facilitating new approaches and methods for interoperability assessment. For instance, new passive approaches (see [45], [46], [47]) that we developed have been implemented and validated using `ttproto`. This tool `ttproto` has been used to develop test suites for 6LoWPAN-ND (IPv6 for Low Power Networks) and CoAP (Constrained Application Protocol). The CoAP test suites have been successfully used for two Plugtest interoperability events organized by ETSI, IPSO Alliance and the FP7 PROBE-IT project, respectively 28-29 March in Paris (see [44]) and 28-30 November in Sophia-Antipolis. The tool `ttproto` and the test suites indicated above are freely available at <http://www.irisa.fr/tipi>.

4.3. Performance and dependability evaluation

Participants: Gerardo Rubino, Bruno Sericola, Bruno Tuffin.

We develop software tools for the evaluation of two classes of models: Markov models and reliability networks. The main objective is to quantify dependability aspects of the behaviors of the modeled systems, but other aspects of the systems can be handled (performance, performability, vulnerability). The tools are specialized libraries implementing numerical, Monte Carlo and Quasi-Monte Carlo algorithms.

One of these libraries has been developed for the Celar (DGA), and its goal is the evaluation of dependability and vulnerability metrics of wide area communication networks (WANs). The algorithms in this library can also evaluate the sensitivities of the implemented dependability measures with respect to the parameters characterizing the behavior of the components of the networks (nodes, lines).

We are also developing tools with the objective of building Markovian models and to compute bounds of asymptotic metrics such as the asymptotic availability of standard metrics of models in equilibrium, loss probabilities, blocking probabilities, mean backlogs,...). A set of functions designed for dependability analysis is being built under the name `DependLib`.

DISCO Project-Team

5. Software

5.1. OreModules

Participants: Alban Quadrat [correspondent], Daniel Robertz [Univ. Aachen], Frédéric Chyzak [Inria Rocquencourt, Algorithms Project].

The OREMODULES package [88], based on the commercial Maple package Ore_algebra [89], is dedicated to the study of linear multidimensional systems defined over certain Ore algebras of functional operators (e.g., ordinary or partial differential systems, time-delay systems, discrete systems) and their applications in mathematical systems theory, control theory and mathematical physics. OREMODULES is original because it combines the recent developments of the Gröbner bases over some noncommutative polynomial rings [97], [99] and new algorithms of algebraic analysis in order to effectively check classical properties of module theory (e.g., existence of a non-trivial torsion submodule, torsion-freeness, reflexiveness, projectiveness, stably freeness, freeness), it gives their system-theoretical interpretations (existence of autonomous elements or successive parametrizations, existence of minimal/injective parametrizations or Bézout equations) [102], [101], [87] and it computes important tools of homological algebra (e.g., (minimal) free resolutions, split exact sequences, extension functors, projective or Krull dimensions, Hilbert power series). The abstract language of homological algebra used in the algebraic analysis approach carries over to the implementations in OREMODULES: up to the choice of the domain of functional operators which occurs in a given system, all algorithms are stated and implemented in sufficient generality such that linear systems defined over the Ore algebras developed in the Ore_algebra package are covered at the same time. Applications of the OREMODULES package to mathematical systems theory, control theory and mathematical physics are illustrated in a large library of examples. The binary of the package is freely available at <http://wwwb.math.rwth-aachen.de/OreModules/>.

5.2. Stafford

Participants: Alban Quadrat [correspondent], Daniel Robertz [Univ. Aachen].

The STAFFORD package of OREMODULES [88] contains an implementation of two constructive versions of Stafford's famous but difficult theorem [114] stating that every ideal over the Weyl algebra $A_n(k)$ (resp., $B_n(k)$) of partial differential operators with polynomial (resp., rational) coefficients over a field k of characteristic 0 (e.g., $k = \mathbb{Q}, \mathbb{R}$) can be generated by two generators. Based on this implementation and algorithmic results developed in [109] by the authors of the package, two algorithms which compute bases of free modules over the Weyl algebras $A_n(\mathbb{Q})$ and $B_n(\mathbb{Q})$ have been implemented. The rest of Stafford's results developed in [114] have recently been made constructive in [112] (e.g., computation of unimodular elements, decomposition of modules, Serre's splitting-off theorem, Stafford's reduction, Bass' cancellation theorem, minimal number of generators) and implemented in the STAFFORD package. The development of the STAFFORD package was motivated by applications to linear systems of partial differential equations with polynomial or rational coefficients (e.g., computation of injective parametrization, Monge problem, differential flatness, the reduction and decomposition problems and Serre's reduction problem). To our knowledge, the STAFFORD package is the only implementation of Stafford's theorems nowadays available. The binary of the package is freely available at <http://wwwb.math.rwth-aachen.de/OreModules/>.

5.3. QuillenSuslin

Participants: Alban Quadrat [correspondent], Anna Fabiańska [Univ. Aachen].

The QUILLEN-SUSLIN package [93] contains an implementation of the famous Quillen-Suslin theorem [113], [115]. In particular, this implementation allows us to compute bases of free modules over a commutative polynomial ring with coefficients in a field (mainly \mathbb{Q}) and in a principal ideal domain (mainly \mathbb{Z}). The development of the QUILLEN-SUSLIN package was motivated by different constructive applications of the Quillen-Suslin theorem in multidimensional systems theory [93] (e.g., the Lin-Bose conjectures, the computation of (weakly) left/right/doubly coprime factorizations of rational transfer matrices, the computation of injective parametrizations of flat linear multidimensional systems with constant coefficients, the reduction and decomposition problems, Serre's reduction problem). To our knowledge, the QUILLEN-SUSLIN package is the only implementation of the Quillen-Suslin theorem nowadays available. The binary of the package is freely available at <http://wwwb.math.rwth-aachen.de/QuillenSuslin>.

5.4. OreMorphisms

Participants: Alban Quadrat [correspondent], Thomas Cluzeau [ENSIL, Univ. Limoges].

The ORE MORPHISMS package [91] of ORE MODULES [87] is dedicated to the implementation of homological algebraic tools such as the computations of homomorphisms between two finitely presented modules over certain noncommutative polynomial algebras (Ore algebras), of kernel, coimage, image and cokernel of homomorphisms, Galois transformations of linear multidimensional systems and idempotents of endomorphism rings. Using the packages STAFFORD and QUILLEN-SUSLIN, the factorization, reduction and decomposition problems can be constructively studied for different classes of linear multidimensional systems. Many linear systems studied in engineering sciences, mathematical physics and control theory have been factorized, reduced and decomposed by means of the ORE MORPHISMS package. The binary of the package is freely available at <http://www-sop.inria.fr/members/Alban.Quadrat/OreMorphisms/index.html>.

5.5. JanetMorphisms

Participants: Alban Quadrat [correspondent], Thomas Cluzeau [ENSIL, Univ. Limoges], Daniel Robertz [Univ. Aachen].

The JANET MORPHISMS package is dedicated to a new mathematic approach to quasilinear systems of partial differential equations (e.g., Burger's equation, shallow water equations, Euler equations of a compressible fluid) based on algebraic analysis and differential algebra techniques [86]. This package computes symmetries, first integrals of motion, conservation laws, study Riemann invariants... The JANET MORPHISMS package is based on the Janet package (<http://wwwb.math.rwth-aachen.de/Janet/>).

5.6. PurityFiltration

Participant: Alban Quadrat [correspondent].

The PURITY FILTRATION package, built upon the ORE MODULES package, is an implementation of a new effective algorithm obtained in [108] which computes the purity/grade filtration [82], [83] of linear functional systems (e.g., partial differential systems, differential time-delay systems, difference systems) and equivalent block-triangular matrices. See Section 6.1. This package is used to compute closed form solutions of over/underdetermined linear partial differential systems which cannot be integrated by the standard computer algebra systems such as Maple and Mathematica. This package will soon be available.

5.7. AbelianSystems

Participants: Alban Quadrat [correspondent], Mohamed Barakat [Univ. Kaiserslautern].

The ABELIANSYSTEMS package is an implementation of an algorithm developed in [34] for the computation of the purity/grade filtration [82], [83] in the powerful homalg package of GAP 4 dedicated to constructive homological algebra methods, and developed by Barakat (University of Kaiserslautern) and his collaborators (<http://homalg.math.rwth-aachen.de/>). This package both supersedes the existing PURITYFILTRATION package which uses the non-efficient Maple Gröbner basis computation (see Section 5.6), and the original homalg procedure which computes purity filtration by means of time-consuming spectral sequences. Using the homalg package philosophy, the ABELIANSYSTEMS package can be used for the computation of the purity filtration of objects in different constructive abelian categories such as coherent sheaves over projective schemes as demonstrated in the homalg package called Sheaves (see <http://homalg.math.rwth-aachen.de/>).

5.8. SystemTheory

Participants: Alban Quadrat [correspondent], Thomas Cluzeau [ENSIL, Univ. Limoges], Markus Lange-Hegermann [Univ. Aachen], Mohamed Barakat [Univ. Kaiserslautern].

The SYSTEMTHEORY package is a homalg based package dedicated to mathematical systems. This package, still in development, will include the algorithms developed in the OREMODULES and OREMORPHISMS packages. It currently contains an implementation of the OREMORPHISMS procedures which handle the decomposition problem aiming at decomposing a module/system into direct sums of submodules/subsystems, and Serre's reduction problem aiming at finding an equivalent system defined by fewer unknowns and fewer equations.

5.9. YALTA

Participants: David Avanesoff [correspondent], Catherine Bonnet, André Fioravanti [UNICAMP].

The YALTA package is dedicated to the study of classical and fractional systems with delay in the frequency-domain. Its objective is to provide basic but important information such as, for instance, the position of the neutral chains of poles and unstable poles, as well as the root locus with respect to the delay of the system. The corresponding algorithms are based on recent theoretical results (see, for instance, [84] and [95]) and on classical continuation methods exploiting the particularities of the problem [96], [14]. We have included this year a Pade2 approximation scheme as well as H_∞ -stability properties. The package is freely available at <http://team.inria.fr/disco/fr/software/>.

DISTRIBCOM Project-Team

5. Software

5.1. SOFAT

Participants: Loïc Hélouët [correspondant], Rouwaida Abdallah.

SOFAT is the acronym for Scenario Oracle and Formal Analysis Toolbox. As this name suggests it is a formal analysis toolbox for scenarios. Scenarios are informal descriptions of behaviors of distributed systems. SOFAT allows the edition and analysis of distributed systems specifications described using Message Sequence Charts, a scenario language standardized by the ITU [Z.120]. The main functionalities proposed by SOFAT are the textual edition of Message Sequence Charts, their graphical visualization, the analysis of their formal properties, and their simulation. The analysis of the formal properties of a Message Sequence Chart specification determines if a description is regular, local choice, or globally cooperative. Satisfaction of these properties allow respectively for model-checking of logical formulae in temporal logic, implementation, or comparison of specifications. All these applications are either undecidable problems or unfeasible if the Message Sequence Chart description does not satisfy the corresponding property. The SOFAT toolbox implements most of the theoretical results obtained on Message Sequence Charts this last decade. It is regularly updated and re-distributed. The purpose of this is twofold:

- Provide a scenario based specification tool for developers of distributed applications
- Serve as a platform for theoretical results on scenarios and partial orders

SOFAT provides several functionalities, that are: syntactical analysis of scenario descriptions, Formal analysis of scenario properties, Interactive Simulation of scenarios when possible, and diagnosis. This year, SOFAT was extended with code synthesis functionalities, allowing to generate communicating automata, promela code, or rest based web services from HMSCs. A new release of the software is expected before the end of the year.

See also the web page <http://www.irisa.fr/distribcom/Prototypes/SOFAT/index.html>.

- AMS: Order; lattices; ordered algebraic structures
- APP: IDDN.FR.001.080027.000.S.P.2003.00.10600
- Programming language: Java

5.2. PLASMA

Participants: Sean Sedwards, Benoit Boyer, Kevin Corre, Axel Legay [correspondant].

PLASMA is our implementation of Statistical Model Checking. PLASMA adopts a modular architecture to facilitate the extension of its features. Models can currently be specified using the PRISM reactive modules syntax or a biochemical syntax, while properties are specified in a discrete bounded temporal logic. Our goal is to allow the implementation of other modeling languages and logics by means of self-contained *drop-in* modules. PLASMA facilitates this by providing an intermediate language to generate transition systems based on the notion of the construct (guard, rate, actions), where guard, rate and actions are functions over the current state of the system and control whether and how fast the system may perform certain actions in each state. New modeling languages may be thus added to PLASMA's repertoire by constructing parsers that translate such languages into the intermediate language.

Web site: <https://project.inria.fr/plasma-lab/>

5.3. LotrecScheme

Participant: François Schwarzentruher [correspondant].

LotrecScheme is the implementation of a generic tableau method prover based on LoTREC (<http://www.irit.fr/Lotrec/>). LotrecScheme is more expressive than LoTREC. Both LoTREC and LotrecScheme provides tableau methods for standard modal logic K, KT, S4, etc. Contrary to LoTREC, LotrecScheme is expressive enough to capture some satisfiability problem for Dynamic Epistemic Logic.

The prover inside LotrecScheme is written in Scheme and embedded in a JAVA application.

See also the web page <http://www.irisa.fr/distribcom/Prototypes/LotrecScheme/index.html>.

DOLPHIN Project-Team

5. Software

5.1. ParadisEO

Participants: Clive Canape, Laetitia Jourdan, Arnaud Liefvooghe, Nouredine Melab, Alexandre Quemy, El-Ghazali Talbi [correspondent], Sébastien Verel.

ParadisEO (PARallel and DIStributed Evolving Objects) is a C++ white-box object-oriented framework dedicated to the flexible design of metaheuristics. See web site <http://paradisEO.gforge.inria.fr>. Based on EO, a template-based ANSI-C++ compliant evolutionary computation library, it is composed of five modules:

- ParadisEO-EO provides tools for the development of population-based metaheuristic (evolutionary and genetic algorithm, genetic programming, particle swarm optimization, etc.)
- ParadisEO-MO provides tools for the development of single solution-based metaheuristics (hill-climbing, tabu search, simulated annealing, iterative local search, variable neighborhood search, incremental evaluation, partial neighborhood, etc.)
- ParadisEO-MOEO provides tools for the design of multi-objective metaheuristics (MO fitness assignment, MO diversity preservation, elitism, performance indicators, easy-to-use state-of-the-art algorithms, etc)
- ParadisEO-PEO provides tools for the design of parallel and distributed metaheuristics (parallel evaluation, parallel evaluation function, island model)
- ParadisEO-SMP provides tools for the design of shared memory parallel metaheuristics (parallel evaluation, island model)

Furthermore, ParadisEO also introduces tools for the design of distributed, hybrid and cooperative models:

- High level hybrid metaheuristics: coevolutionary and relay models.
- Low level hybrid metaheuristics: coevolutionary and relay models.

The ParadisEO framework has been especially designed to best suit to the following objectives:

- Maximum design and code reuse: ParadisEO is based on a clear conceptual separation of the solution methods from the problems they are intended to solve. This separation confers to the user a maximum code and design reuse.
- Flexibility and adaptability: The fine-grained nature of the classes provided by the framework allows a higher flexibility compared to other frameworks.
- Utility: ParadisEO allows the user to cover a broad range of metaheuristics, problems, parallel distributed models, hybridization mechanisms, etc.
- Transparent and easy access to performance and robustness: As the optimization applications are often time-consuming the performance issue is crucial. Parallelism and distribution are two important ways to achieve high performance execution. ParadisEO is one of the rare frameworks that provide the most common parallel and distributed models. These models can be exploited in a transparent way, one has just to instantiate their associated provided classes.
- Portability: The implemented models are portable on distributed-memory machines as well as on shared-memory multiprocessors, as they use standard libraries such as MPI and `std::threads`. ParadisEO supports the most recent version of standard of the C++ programming, `c++11`.

This year a new module, ParadisEO-SMP, has been released. All the new features is managed via the Inria Gforge project <http://paradisEO.gforge.inria.fr>. The version 2.0 has been released in Septembre 2012.

5.1.1. *Paradiseo-SMP: a new module for shared memory parallel*

This year, we released a new module dedicated to shared memory parallel. This module improves the technical mechanisms of Paradiseo-PEO thanks a new software architecture and the new c++11 features.

Paradiseo-SMP implements parallel evaluation, dynamic heterogeneous island model, and their hybridization. The main features are:

- Dynamic Island Model: topology can be changed during the execution.
- Heterogeneous Islands: different kinds of population-based metaheuristics can communicate (evolutionary and genetic algorithm, particle swarm optimization, etc.).
- Island Model and master/slave model can be hybridized.

All these new features are developed in c++11.

5.1.2. *New technical features*

Regarding the technical aspects, the compatibility with dependencies taken into account is:

- c++11 features supporting.
- Checked compatibility with different operating systems.
- Reviewed and checked compatibility with new versions of the tools used (CMake, g++, clang, MinGW...).
- Unit and integration test of all additional components, and experiments on classical applications.

5.1.3. *Contributions and documentations*

Many investigations were made in this context in order to help users to manipulate the framework.

- New quick start guide is available.
- New tutorials:
 - Tutorials SMP.
 - Tutorials GPU.
- Updated implementation for classical problems.

Self-assessment of the team effort (software criteria: <http://www.inria.fr/institut/organisation/instances/commission-d-evaluation>)

(A-4-up5) Audience: 4 - Used in many universities for teaching and several companies.

(SO-4) Software Originality: 4 - ParadisEO aggregates the last results of the Dolphin team.

(SM-4) Software Maturity: 4 - Extensive documentation, strong software engineering and testing, regression testing, user feedback ...

(EM-2-up3) Evolution and Maintenance: 2 - Basic maintenance with persistent attention to users.

(SDL-4) Software Distribution and Licensing: 4 - CeCILL license, public source, Windows and Mac installer, Linux packages.

(OC) Own Contribution: (Design/Architecture) DA-4, (Coding/Debugging) CD-4, (Maintenance/Support)

MS-4, (Team/Project Management) TPM-4

DRACULA Project-Team

5. Software

5.1. CelDyn

Participants: Laurent Pujo-Menjouet, Alen Tosenberger, Vitaly Volpert [correspondant].

Software "Celdyn" is developed in order to model cell population dynamics for biological applications. Cells are represented either as soft spheres or they can have more complex structure. Cells can divide, move, interact with each other or with the surrounding medium. Different cell types can be introduced. When cells divide, the types of daughter cells are specified. A user interface is developed.

DREAM Project-Team

5. Software

5.1. Introduction

The pieces of software described in this section are prototypes implemented by members of the project. Any interested person should contact relevant members of the project.

5.2. QTempIntMiner: quantitative temporal sequence mining

QTEMPINTMINER (Quantitative Temporal Interval Miner) is a data mining (cf. 3.2.2) software that implements several algorithms presented in [48] and [3].

The software is mainly implemented in Matlab. A standalone application is now available. It uses the Mixmod toolbox [35] to compute multi-dimensional Gaussian distributions. The main features of QTEMPINTMINER are:

- a tool for generating synthetic noisy sequences of temporal events,
- an implementation of the QTEMPINTMINER, QTIAPRIORI and QTIPREFIXSPAN algorithms,
- a graphical interface that enables the user to generate or import data set and to define the parameters of the algorithm and that displays the extracted temporal patterns.
- a sequence transformer to process long sequences of temporal events. Long sequences are transformed into a database of short temporal sequences that are used as input instances for the available algorithms.

The following website gives many details about the algorithms and provides the latest stable implementation of QTEMPINTMINER: <http://www.irisa.fr/dream/QTempIntMiner/>.

5.3. Sacadeau: qualitative modeling and decision-aid to preserve the water quality from pollutants as herbicides

SACADEAU is an environmental decision software (cf. 4.3) that implements the SACADEAU transfer model presented in section 7.2.1 . The SACADEAU simulation model couples two qualitative models, a transfer model describing the pesticide transfer through the catchment and a management model describing the farmer decisions. Giving as inputs a climate file, a topological description of a catchment, and a cadastral repartition of the plots, the SACADEAU model simulates the application of herbicides by the farmers on the maize plots, and the transfer of these pollutants through the catchment until the river. The two main simulated processes are the runoff and the leaching. The output of the model simulation is the quantity of herbicides arriving daily to the stream and its concentration at the outlets. The originality of the model is the representation of water and pesticide runoffs with tree structures where leaves and roots are respectively up-streams and down-streams of the catchment.

The software allows the user to see the relationships between these tree structures and the rules learnt from simulations (cf. 3.2.1). A more elaborated version allows to launch simulations and to learn rules on-line. This year, we have developed this new version by enabling access to two recommendation action algorithms (see section 6.3.1). The user can choose different parameters (set of classification rules from which actions will be built, parameters concerning action feasibility, etc) before asking for action recommending process, and then easily visualize the characteristics of situations to improve (polluted ones) compared with the different recommended actions. The software is mainly in Java.

The following website is devoted to the presentation of the SACADEAU: <http://www.irisa.fr/dream/SACADEAU/>.

5.4. Ecomata

EcoMata is a tool-box for qualitative modeling and exploring ecosystems and for aiding to design environmental guidelines. We have proposed a new qualitative approach for ecosystem modeling (cf. 4.3) based on timed automata (TA) formalism combined to a high-level query language for exploring scenarios.

To date, EcoMata is dedicated to ecosystems that can be modeled as a collection of species (prey-predator systems) under various human pressures and submitted to environmental disturbances. It has two main parts: the Network Editor and the Query Launcher. The Network Editor let a stakeholder describe the trophic food web in a graphical way (the species icons and interactions between them). Only few ecological parameters are required and the user can save species in a library. The number of qualitative biomass levels is set as desired. An efficient algorithm generates automatically the network of timed automata. EcoMata provides also a dedicated window to help the user to define different fishing pressures, a nice way being by using chronograms. In the Query Launcher, the user selects the kind of query and the needed parameters (for example the species biomass levels to define a situation). Results are provided in a control panel or in files that can be exploited later. Several additional features are proposed in EcoMata: building a species library, import/export of ecosystem model, batch processing for long queries, etc. EcoMata is developed in Java (Swing for the GUI) and the model-checker called for the timed properties verification is UPPAAL.

The following website is devoted to the presentation of ECOMATA: <http://oban.agrocampus-ouest.fr:8080/ecomata>.

5.5. Manage Yourself

ManageYourself is a collaborative project between Dream and the Telelogos company aiming at monitoring smartphones from a stream of observations made on the smartphone state (cf. 3.2.3).

Today's smartphones are able to perform calls, as well as to realize much more complex activities. They are small computers. But as in computers, the set of applications embedded on the smartphone can lead to problems. The aim of the project ManagerYourself is to monitor smartphones in order to avoid problems or to detect problems and to repair them.

The ManageYourself application includes three parts :

- A monitoring part which triggers preventive rules at regular time to insure that the system is working correctly, e.g. *if the memory is full then delete the tmp directory*. This part is always running on the smartphone.
- A reporting part which records regularly the state of the smartphone (the memory state - free vs allocated -, the connection state, which applications are running, etc.). This part also is always running on the smartphone. The current state is stored in a report at regular period and is labeled *normal*. When an application or the system bugs, the current buggy state is stored in a report and is labeled *abnormal*. At regular timestamps, all the reports are sent to a server where the learning process is executed.
- A learning part which learns new bug rules from the report dataset. This part is executed offline on the server. Once the bug rules are learnt, human experts translates them into preventive rules which are downloaded and integrated in the monitoring part of the smartphones.

The following website is devoted to the presentation of MANAGEYOURSELF: <http://www.irisa.fr/dream/ManageYourself/Site/ManageYourself.html>.

DYLISS Team

5. Software

5.1. Data integration: actors involved in the response of a living system

The goal is to offer a toolbox for the reconstruction of networks from genome, literature and large-scale observation data (expression data, metabolomics...) in order to elucidate the main regulators of an observed phenotype.

- **Mobyle@GenOuest network portal** We are developing a web service ² to use several tools to confront knowledge and data towards the correction of large-scale networks, based either on decision diagrams or on answer set programming. BioQuali ³⁴ allows one to confront model and data, localize errors and, when model and data are consistent, to predict the variation of non observed nodes [6]. BioASP ⁵ was developed in Potsdam and allows one to perform prediction even if model and observations are contradictory, by considering all possible repairs of data and models and computing the common predictions of all repaired models [5]. The portal also include tools for the completion of metabolic networks [32].
- **Combined set of key actors in reaction-based networks: Cadbiom**⁶. This tool is based on state-chart like graphical language. It allows investigating synchronization events in biological networks. It is applied to cancer signaling networks [10].

5.2. Dynamics: actor/parameter combination controlling the response of a system

We wish to develop tools predicting some characteristics of a biological system behavior from incomplete sets of parameters or observations.

- **caspo: Cell ASP Optimizer**. We have implemented a Python package which combines BioASP ⁷ and CellNOpt⁸ to provide an easy to use software for learning Boolean logic models using ASP [19]. The software is available for download⁹ and also as a web service through the Mobyle framework.
- **Event network and quantitative time-series data: POGG**¹⁰. POGG is a tool developed in collaboration with the LINA lab (Nantes) that uses mean dynamics to score the respective relevance of regulatory pathways in a higher-scale phenotype. It was applied to the quantitative prediction of protein quantities under exponential growth [2]. It predicts the main features of a Markov chain model derived from a reaction-based model when confronted to a single time-series quantitative observation.

²<http://mobyle.genouest.org/cgi-bin/Mobyle/portal.py>

³<http://www.irisa.fr/symbiose/bioquali/>

⁴<http://www.irisa.fr/symbiose/projects/bioqualiCytoscapePlugin/>

⁵<http://www.cs.uni-potsdam.de/bioasp/>

⁶<http://cadbiom.genouest.org/>

⁷<http://www.cellnopt.org/>

⁸

⁹[sthiele/bioasp/http://www.cs.uni-potsdam.de/~sthiele/bioasp/](http://www.cs.uni-potsdam.de/~sthiele/bioasp/)

⁹<http://pypi.python.org/pypi/caspo>

¹⁰<http://pogg.genouest.org/wiki.php/Home>

5.3. Sequence annotation

We develop tools for discovery and search of complex pattern signatures within biological sequences, with a focus on protein sequences. An integrated environment, Dr Motif ¹¹ is available on the GenOuest Platform that gathers state-of-the-art tools for pattern discovery and pattern matching including our own developments.

- **Complex pattern discovery: Protomata learner**¹² is a grammatical inference framework suitable for the inference of accurate protein signatures [3], [4]. It was completely redesigned in 2010-2011 thanks to a specific Inria action (ADT support). It is currently applied to the recognition of olfactory receptor genes.
- **Complex pattern matching: Logol**¹³. We have completely redesigned Stan (suffix-tree analyser), a former tool to search for nucleotidic and peptidic patterns within whole chromosomes [7]. The result is Logol, a software suite accepting a syntax based on String Variable Grammars, which allows the description of realistic complex patterns including ambiguities, insertions/ deletions, gaps, repeats and palindromes. It has been presented for the first time in [21]. Logol has been applied to the detection of -1 frameshifts, a structure including pseudo knots, on a reference benchmark (Recode2).

¹¹<http://www.drmotifs.org/>

¹²<http://protomata-learner.genouest.org/>

¹³<http://webapps.genouest.org/LogolDesigner/>

E-MOTION Project-Team

4. Software

4.1. PROTEUS Software

Participants: Amaury Nègre, Juan Lahera-Perez.

This toolkit offers a automatic mobile robot driver, some sensors drivers (sensors as Sick laser, GPS, motion tracker, mono or stereo camera), and a 3D Simulator.

The latest developments have been focuses on the robotics simulator. This simulator is based on the simulation and 3D rendering engine “mgEngine“ (<http://mgengine.sourceforge.net/>) embedded with the physics engine “bullets physics” (<http://bulletphysics.org>) for realistic robot dynamic simulation.

We also worked on the interface with the robotics middleware “ROS“ (<http://www.ros.org>) in order to offer interoperability with many robotics applications.

The simulator is now fully integrated with the robotics middleware "ROS" (<http://www.ros.org>) which allow interoperability with a large set of robotics applications and visualization tools.

This software is developed in C++ and the simulator operates with the Lua scripting language.

The simulation software is used in the ANR Proteus (<http://www.anr-proteus.fr>), as a simulation engine for the PROTEUS Toolkit.

- Version: 2.0
- APP:IDDN.FR.001.510040.000.S.P.2005.000.10000
- Programming language: C/C++, Lua

4.2. AROSDYN

Participants: Igor Paromtchik, Mathias Perrollaz, Alexandros Makris, Amaury Nègre, Christian Laugier.

ArosDyn (<http://arosdyn.gforge.inria.fr/>) is a system which integrates our recently developped techniques to provide a real-time collision risk estimation in a dynamic environment. The main features of this software are:

1. The design provides high maintainability, scalability and reuseness of the models and algorithms.
2. The software has a user interface (UI) which is user-friendly.
3. The software facilitates the parameter tuning of the models.
4. It uses the GPU to accelerate the computation.
5. Working together with the Hugn middleware (<http://gforge.inria.fr/projects/cycabtk>), it can run on our experimental vehicle in real-time.

Another important property of this software is a large part of the computation task executed on GPU. As the processing of stereo image and the computaion in the BOF can be highly parallelized, we run these tasks on the GPU to improve the time performance. The GPU calculation is based on CUDA library and is carried out in an independent thread.

Furthermore, thanks to the design of the software, we can easily add new models to it and let them work together. The fast detection and tracking algorithm (FCTA) and the Gaussian process based collision assessment algorithm are added into this framework. The software is implemented on the Lexus car. In 2012, a demand for deposing the GPU BOF software to the APP is in progress.

4.3. Bayesian Occupancy Filter

People involved: Kamel Mekhnacha, Tay Meng Keat Christopher, C. Laugier, M. Yguel, Pierre Bessière.

The BOF toolbox is a C++ library that implements the Bayesian Occupancy Filter. It is often used for modelling dynamic environments. It contains the relevant functions for performing bayesian filtering in grid spaces. The output from the BOF toolbox are the estimated probability distributions of each cell's occupancy and velocity. Some basic sensor models such as the laser scanner sensor model or Gaussian sensor model for gridded spaces are also included in the BOF toolbox. The sensor models and BOF mechanism in the BOF toolbox provides the necessary tools for modelling dynamic environments in most robotic applications. This toolbox is patented under two patents : "Procédé d'assistance à la conduite d'un véhicule et dispositif associé" n. 0552735 (9 september 2005) and "Procédé d'assistance à la conduite d'un véhicule et dispositif associé amélioré" n. 0552736 (9 september 2005) and commercialized by ProBayes.

- Version: 1
- Patent: 0552736 (2005), 0552735 (2005)
- Programming language: C/C++

4.4. PROBT

People involved: Juan-Manuel Ahuactzin, Kamel Mekhnacha, Pierre Bessière, Emmanuel Mazer, Manuel Yguel, Christian Laugier.

ProBT is both available as a commercial product (ProBAYES.com) and as a free library for public research and academic purposes (<http://emotion.inrialpes.fr/BP/spip.php?rubrique6>). Formerly known as *OPL*, *ProBT* is a C++ library for developing efficient Bayesian software. It is available for Linux, Unix, PC Windows (Visual C++), MacOS9, MacOSX and Irix systems. The ProBT library (<http://www.probayes.com/>) has two main components: (i) a friendly Application Program Interface (API) for building Bayesian models, and (ii) a high-performance Bayesian Inference Engine (BIE) allowing to execute all the probability calculus in exact or approximate way. *ProBT* is now commercialized by our start-up *Probayes*; it represents the main Bayesian programming tool of the *e-Motion* project-team, and it is currently used in a variety of external projects both in the academic and industrial field (e.g., for the European project BACS and for some industrial applications such as Toyota or Denso future driving assistance systems).

ESPRESSO Project-Team

5. Software

5.1. The Polychrony toolset and its hypertext source documentation

Participants: Loïc Besnard, Thierry Gautier, Paul Le Guernic.

The Polychrony toolset is an Open Source development environment for critical/embedded systems. It is based on Signal, a real-time polychronous data-flow language. It provides a unified model-driven environment to perform design exploration by using top-down and bottom-up design methodologies formally supported by design model transformations from specification to implementation and from synchrony to asynchrony. It can be included in heterogeneous design systems with various input formalisms and output languages.

The Polychrony toolset provides a formal framework:

- to validate a design at different levels, by the way of formal verification and/or simulation,
- to refine descriptions in a top-down approach,
- to abstract properties needed for black-box composition,
- to assemble heterogeneous predefined components (bottom-up with COTS),
- to generate executable code for various architectures.

The Polychrony toolset contains three main components and an experimental interface to GNU Compiler Collection (GCC):

- The Signal toolbox, a batch compiler for the Signal language, and a structured API that provides a set of program transformations. The Signal toolbox can be installed without other components. The Signal toolbox is distributed under GPL V2 license.
- The Signal GUI, a Graphical User Interface to the Signal toolbox (editor + interactive access to compiling functionalities). The Signal GUI is distributed under GPL V2 license.
- The SME/SSME platform, a front-end to the Signal toolbox in the Eclipse environment. The SME/SSME platform is distributed under EPL license.
- GCCst, a back-end to GCC that generates Signal programs (not yet available for download).

The Polychrony toolset also provides:

- libraries of Signal programs,
- a set of Signal program examples,
- user oriented and implementation documentations,
- facilities to generate new versions.

The Polychrony toolset can be freely downloaded on the following web sites:

- The Polychrony toolset public web site: <http://www.irisa.fr/espresso/Polychrony>. This site, intended for users and for developers, contains downloadable executable and source versions of the software for different platforms, user documentation, examples, libraries, scientific publications and implementation documentation. In particular, this is the site for the new open-source distribution of Polychrony.
- The Inria GForge: <https://gforge.inria.fr>. This site, intended for internal developers, contains the whole sources of the environment and their documentation.
- The TOPCASED distribution site: <http://www.topcased.org>. This site provides the current reference version of the SSME platform, including the executable of the Signal toolbox.

The Polychrony toolset currently runs on Linux, MacOS and Windows systems.

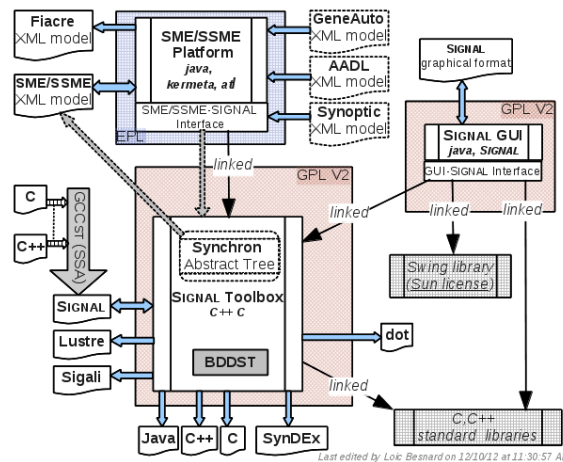


Figure 7. The Polychrony toolset high-level architecture

The Geensoft company, now part of Dassault Systèmes, supplies a commercial implementation of Polychrony, called RT-Builder, used for industrial scale projects (see <http://www.geensoft.com>).

As part of its open-source release, the Polychrony toolset not only comprises source code libraries but also an important corpus of structured documentation, whose aim is not only to document each functionality and service, but also to help a potential developer to package a subset of these functionalities and services, and adapt them to developing a new application-specific tool: a new language front-end, a new back-end compiler. This multi-scale, multi-purpose documentation aims to provide different views of the software, from a high-level structural view to low-level descriptions of basic modules. It supports a distribution of the software “by apartment” (a functionality or a set of functionalities) intended for developers who would only be interested by part of the services of the toolset.

A high-level architectural view of the Polychrony toolset is given in Figure 7 .

5.2. The Eclipse interface

Participants: Loïc Besnard, Yue Ma, Huafeng Yu.

Meta-modeling, Eclipse, Ecore, Signal, Model transformation

We have developed a meta-model and interactive editor of Polychrony in Eclipse. Signal-Meta is the meta-model of the Signal language implemented with Eclipse/Ecore. It describes all syntactic elements specified in [35]: all Signal operators (e.g. arithmetic, clock synchronization), model (e.g. process frame, module), and construction (e.g. iteration, type declaration).

The meta-model primarily aims at making the language and services of the Polychrony environment available to inter-operation and composition with other components (e.g. AADL, Simulink, GeneAuto) within an Eclipse-based development toolchain. Polychrony now comprises the capability to directly import and export Ecore models instead of textual Signal programs, in order to facilitate interaction between components within such a toolchain.

It also provides a graphical modeling framework allowing to design applications using a component-based approach. Application architectures can be easily described by just selecting components via drag and drop, creating some connections between them and specifying their parameters as component attributes. Using the modeling facilities provided with the Topcased framework, we have created a graphical environment for

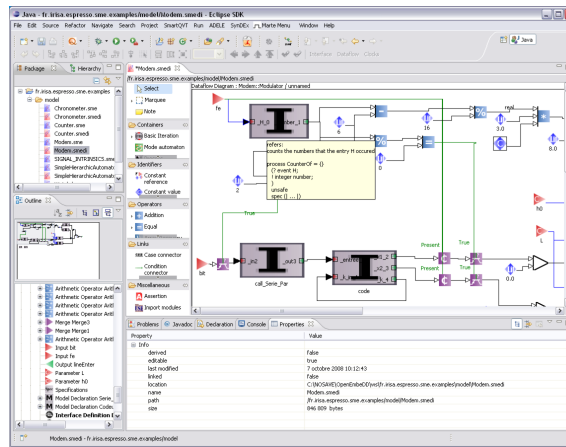


Figure 8. Eclipse SME Environment.

Polychrony (see figure 8) called SME (Signal-Meta under Eclipse). To highlight the different parts of the modeling in Signal, we split the modeling of a Signal process in three diagrams: one to model the interface of the process, one to model the computation (or data-flow) part, and one to model all explicit clock relations and dependences. The SME environment is available through the ESPRESSO update site [23], in the current OpenEmbed distribution [22], or in the TopCased distribution [25]. Note that a new meta-model of Signal, called SSME (Syntactic Signal-Meta under Eclipse), closer to the Signal abstract syntax, has been defined and integrated in the Polychrony toolset.

5.3. Integrated Modular Avionics design using Polychrony

Participants: Loïc Besnard, Thierry Gautier, Paul Le Guernic, Jean-Pierre Talpin.

The Apex interface, defined in the ARINC standard [26], provides an avionics application software with the set of basic services to access the operating-system and other system-specific resources. Its definition relies on the Integrated Modular Avionics approach (IMA [27]). A main feature in an IMA architecture is that several avionics applications (possibly with different critical levels) can be hosted on a single, shared computer system. Of course, a critical issue is to ensure safe allocation of shared computer resources in order to prevent fault propagations from one hosted application to another. This is addressed through a functional partitioning of the applications with respect to available time and memory resources. The allocation unit that results from this decomposition is the *partition*.

A partition is composed of *processes* which represent the executive units (an ARINC partition/process is akin to a Unix process/task). When a partition is activated, its owned processes run concurrently to perform the functions associated with the partition. The process scheduling policy is priority preemptive. Each partition is allocated to a processor for a fixed time window within a major time frame maintained by the operating system. Suitable mechanisms and devices are provided for communication and synchronization between processes (e.g. *buffer*, *event*, *semaphore*) and partitions (e.g. *ports* and *channels*). The specification of the ARINC 651-653 services in Signal [5] is now part of the Polychrony distribution and offers a complete implementation of the Apex communication, synchronization, process management and partitioning services. Its Signal implementation consists of a library of generic, parameterizable Signal modules.

EXMO Project-Team

5. Software

5.1. Alignment API

We have designed a format for expressing alignments in a uniform way [1]. The goal of this format is to be able to share available alignments on the web. It should help systems using alignments, e.g., mediators, translators, to take advantage of any alignment algorithm and it will help alignment algorithms to be used in many different tasks. This format is expressed in RDF, so it is freely extensible, and has been defined by a DTD (Document Type Description for RDF/XML), an OWL ontology and an RDF Schema.

The API itself [1] is a JAVA description of tools for accessing the common format. It defines five main interfaces (OntologyNetwork, Alignment, Cell, Relation and Evaluator) and proposes the following services:

- Storing, finding, and sharing alignments;
- Piping matching algorithms (improving an existing alignment);
- Manipulating alignments (thresholding and hardening);
- Generating processing output (transformations, axioms, rules);
- Comparing alignments.

We provide an implementation for this API which can be used for producing transformations, rules or bridge axioms independently from the algorithm which produced the alignment. The proposed implementation features:

- a base implementation of the interfaces with all useful facilities;
- a library of sample matchers;
- a library of renderers (XSLT, SWRL, OWL, C-OWL, SEKT mapping language, SPARQL);
- a library of evaluators (various generalisation of precision/recall, precision/recall graphs);
- a flexible test generation framework which allows for generating evaluation datasets;
- a library of wrapper for several ontology API;
- a parser for the format.

To instantiate the API, it is sufficient to refine the base implementation by implementing the `align()` method. Doing so, the new implementation will benefit from all the services already implemented in the base implementation.

We have developed on top of the Alignment API an Alignment server that can be used by remote clients for matching ontologies and for storing and sharing alignments. It is developed as an extensible platform which allows to plug-in new interfaces. The Alignment server can be accessed through HTML, web service (SOAP and REST) and agent communication interfaces.

The Alignment API is used in the Ontology Alignment Evaluation Initiative data and result processing (§6.1.1). It is also used by more than 30 other teams worldwide.

The Alignment API is freely available since december 2003, under the LGPL licence, at <http://alignapi.gforge.inria.fr>.

5.2. The OntoSim library

OntoSim is a library offering similarity and distance measures between ontology entities as well as between ontologies themselves. It materialises our work towards better ontology proximity measures.

There are many reasons for measuring a distance between ontologies. For example, in semantic social networks, when a peer looks for a particular information, it could be more appropriate to send queries to peers having closer ontologies because it will be easier to translate them and it is more likely that such a peer has the information of interest [12]. OntoSim provides a framework for designing various kinds of similarities. In particular, we differentiate similarities in the ontology space from those in the alignment space. The latter ones make use of available alignments in an ontology network while the former only rely on ontology data. OntoSim is provided with 4 entity measures which can be combined using various aggregation schemes (average linkage, Hausdorff, maximum weight coupling, etc.), 2 kinds of vector space measures (boolean and TF.IDF), and 4 alignment space measures. It also features original comparison methods such as agreement/disagreement measures. In addition, the framework embeds external similarity libraries which can be combined to our own.

OntoSim is based on an ontology interface allowing for using ontology parsed with different APIs.

OntoSim is written in Java and is available, under the LGPL licence, at <http://ontosim.gforge.inria.fr>.

In the continuation of our previous work, in 2012, we developed our work on evaluation of ontology matching and especially in running new experiments and generating new tests (§6.1.1). We introduced a new semantics for weighted correspondences (§6.1.2). We also continued our work on ontology matching for linking data (§6.2) and the use of the μ -calculus for evaluating RDF path queries (§6.3.1).

FLOWERS Project-Team

5. Software

5.1. Perception Tools

Participants: David Filliat [correspondant], Natalia Lyubova, Louis-Charles Caron, Alexander Geppeth.

5.1.1. Perception Abstraction Engine

Participants: David Filliat [correspondant], Natalia Lyubova.

PAE (Perception Abstraction Engine) is a C++ library developed to provide a uniform interface to existing visual feature detector such as SIFT, SURF, MSER, superpixels, etc... Its main goal is to be able to use these various feature detectors in a "bag of feature" approach for applications such as robot localisation and object recognition. Several approach are also implemented for the visual vocabularies, in particular the fast incremental vocabularies developed in the team.

The library provide common C++ interfaces to feature detectors, visual features and visual vocabularies. A factory approach make it possible to change the feature detectors and visual vocabularies types and parameters through configuration strings, without the need to recompile. Some applications are also included in the library, in particular topological robot localization (room recognition) and visual object recognition. An Urbi interface is also provided for these modules.

5.1.2. Incremental object discovery

Participants: Natalia Lyubova [correspondant], David Filliat.

This software makes it possible to detect, model and recognize objects in a scenario of interaction between a humanoid robot and a human teacher. It is based either on standard images, or on the kinect camera to take advantage of the depth information. The software is written in C++ and relies mainly on PAE and OpenCV.

The software implements several modules: candidate object segmentation based on motion information, keypoint-based object tracking, incremental object model construction integrating multiple features (keypoints + superpixels) and object categorisation based on mutual information with robot motors (making it possible to segment robot parts, objects and humans).

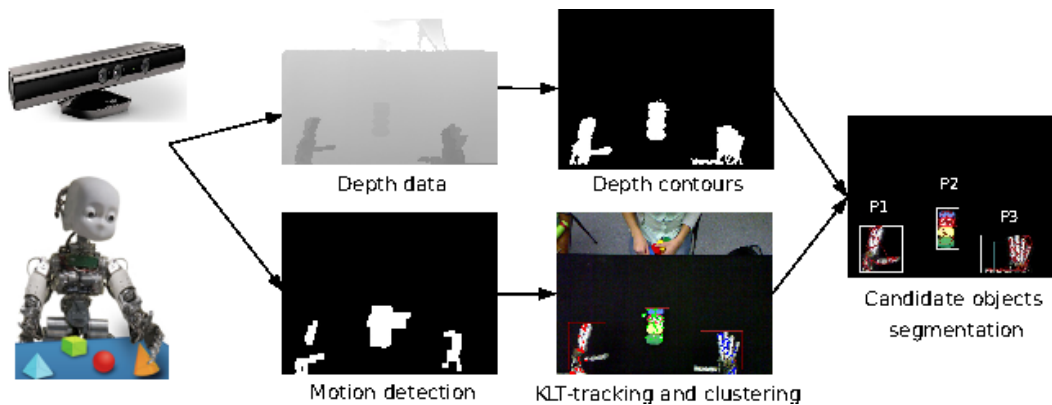


Figure 1. System Overview of the Incremental object discovery Software.

5.1.3. Object recognition from a 3-D point cloud

Participants: Louis-Charles Caron [correspondant], Alexander Gepperth, David Filliat.

This software scans the 3-D point cloud of a scene to find objects and match them against a database of known objects. The process consists in 3 stages. The segmentation step finds the objects in the point cloud, the feature extraction computes discriminating properties to be used in the classification stage for object recognition.

The segmentation is based on simple assumptions about the geometry of an indoor scene. Successive RANSACs are used to find large planes, which correspond to the floor, ceiling and walls. The cloud is stripped from the points belonging to these planes. The remaining points are clustered, meaning that close-by points are considered to form a single object.

Objects are characterized by their shape and color. Color histograms and SIFT features are computed, using the PAE library, to capture the visual appearance of the objects. Their shape is encoded by computing thousands of randomly chosen SURFLET features to construct a relative frequency histogram.

An early classification is done using each of the 3 features separately. For the color features a bag of words approach (from PAE) is used. For the shape feature, the minimum squared distance between the object's histogram and that of all objects in the database is calculated. Classification scores are then fused by a feed-forward neural network to get the final result [39].

5.1.4. PEDDETECT: GPU-accelerated person detection demo

Participant: Alexander Gepperth [correspondant].

PEDDETECT implements real-time person detection in indoor or outdoor environments. It can grab image data directly from one or several USB cameras, as well as from pre-recorded video streams. It detects multiple persons in 800x600 color images at frame rates of >15Hz, depending on available GPU power. In addition, it also classifies the pose of detected persons in one of the four categories "seen from the front", "seen from the back", "facing left" and "facing right". The software makes use of advanced feature computation and nonlinear SVM techniques which are accelerated using the CUDA interface to GPU programming to achieve high frame rates. It was developed in the context of an ongoing collaboration with Honda Research Institute USA, Inc.

5.2. Datasets

5.2.1. Choreography dataset

Participants: Olivier Mangin [correspondant], Haylee Fogg.

This database contains choreography motions recorded through a kinect device. These motions have a combinatorial structure: from a given set of primitive dance motions, choreographies are constructed as simultaneous execution of some of these primitive motions. Primitive dance motions are chosen from a total set of 48 motions and are spanned over one or two limbs, either the legs (e.g. walk, squat), left or right arm (e.g. wave hand, punch) or both arms (e.g. clap in hands, paddle). Complex choreographies are produced as the simultaneous demonstration of two or three of these primitive motion: either one for legs and one for both arm, or one for legs and one for each arm. The dataset has been used in the experiments from [52] for studying learning techniques allowing to identify dictionaries of motion primitives, and is publicly available at https://flowers.inria.fr/choreography_database.html.

5.3. Learning algorithms

5.3.1. RLPark - Reinforcement Learning Algorithms in JAVA

Participant: Thomas Degris [correspondant].

RLPark is a reinforcement learning framework in Java. RLPark includes learning algorithms, state representations, reinforcement learning architectures, standard benchmark problems, communication interfaces for three robots, a framework for running experiments on clusters, and real-time visualization using Zephyr. More precisely, RLPark includes:

- Online Learning Algorithms: Sarsa, Expected Sarsa, Q-Learning, On-policy and off-policy Actor-Critic with normal distribution (continuous actions) and Boltzmann distribution (discrete action), average reward actor-critic, TD, $TD(\lambda)$, $GTD(\lambda)$, $GQ(\lambda)$, TDC
- State Representations: tile coding (with no hashing, hashing and hashing with mumur2), Linear Threshold Unit, observation history, feature normalization, radial basis functions
- Interface with Robots: the Critterbot, iRobot Create, Nao, Puppy, Dynamixel motors
- Benchmark Problems: mountain car, swing-up pendulum, random walk, continuous grid world

An example of RLPark running an online learning experiment on a reinforcement learning benchmark problem is shown in Figure 2 .

RLPark was started in spring 2009 in the RLAI group at the university of Alberta (Canada) when Thomas Degris was a postdoc in this group. RLPark is still actively used by RLAI. Collaborators and users include Adam White, Joseph Modayil and Patrick Pilarski (testing) from the University of Alberta.

RLPark has been used by Richard Sutton, a professor and iCORE chair in the department of computing science at the University of Alberta, for a demo in his invited talk *Learning About Sensorimotor Data* at the Neural Information Processing Systems (NIPS) 2011 ¹. Patrick Pilarski used RLPark for live demos on television (Breakfast Television Edmonton, CityTV, June 5th, 2012) and at TEDx Edmonton on Intelligent Artificial Limbs². So far, RLPark has been used in more than a dozens of publications (see <http://rlpark.github.com/publications.html> for a list).

RLPark has been ported to C++ by Saminda Abeyruwan, a student of the University of Miami (United States of America). The Horde architecture in RLPark has been optimized for GPU by Clément Gehring, a student of the McGill University in Montreal (Canada).

Future developments include the implementation of additional algorithms (the Dyna architecture, back propagation in neural networks, ...). A paper is under review for the JMLR Machine Learning Open Source Software. Documentation and tutorials are included on the RLPark web site ³. RLPark is licensed under the open source Eclipse Public License.

5.3.2. DMP-BBO Matlab library

Participant: Freek Stulp [correspondant].

The `dmp_bbo` (Black-Box Optimization for Dynamic Movement Primitives) Matlab library is a direct consequence of the insight that black-box optimization outperforms reinforcement learning when using policies represented as Dynamic Movement Primitives. It implements several variants of the PI^{BB} algorithm for direct policy search. It is currently being used and extended by several FLOWERS members (Manuel Lopes, Clément Moulin-Frier) and external collaborators (Jonas Buchli, Hwangbo Jemin of ETH Zurich). This code was used for the following publications: [63], [60], [62].

5.3.3. PROPRES: simulation of developmental concept formation using PYTHON

Participant: Alexander Gepperth [correspondant].

This simulation software implements the algorithms described in [24], [40]. It is available online under the URL www.gepperth.net/downloads.html. The simulation is implemented in PYTHON for easy use, yet the time-critical core functions are written in C.

¹ <http://webdocs.cs.ualberta.ca/~sutton/Talks/Talks.html#sensorimotor>

² <http://www.youtube.com/watch?v=YPC-Ae7zqSo>

³ <http://rlpark.github.com>

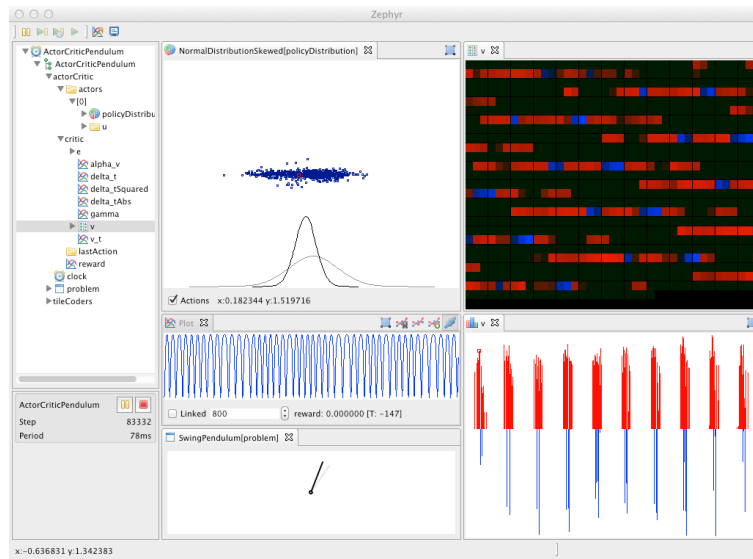


Figure 2. An example of an experiment in RLPark. Zephyr displays two views of a learned weight vector, an animation of the problem, the current policy distribution learned by the algorithm and the reward obtained by the algorithm. Videos are available at: <http://rlpark.github.com>.

5.3.4. pyStreamPlayer: synchronized replay of multiple sensor recordings and supplementary data

Participant: Alexander Geppert [correspondant].

This Python software is intended to facilitate the application of machine learning algorithms by avoiding to work directly with an embodied agent but instead with data recorded in such an agent. Assuming that non-synchronous data from multiple sensors (e.g., camera, Kinect, laser etc.) have been recorded according to a flexible format defined by the pyStreamPlayer architecture, pyStreamPlayer can replay these data while retaining the exact temporal relations between different sensor measurements. As long as the current task does not involve the generation of actions, this software allows to process sensor data as if it was coming from an agent which is usually considerably easier. At the same time, pyStreamPlayer allows to replay arbitrary supplementary information such as, e.g., object information, as if it was coming from a sensor. In this way, supervision information can be stored and accessed together with sensory measurements using an unified interface. pyStreamPlayer has been used to facilitate real-world object recognition tasks, and several of the major databases in this field (CalTech Pedestrian database, HRI RoadTraffic traffic objects database, CVC person database, KITTI traffic objects database) have been converted to the pyStreamPlayer format and now serve as a source of training and test data for learning algorithms.

pyStreamPlayer has been integrated into a ROS node as well, allowing the replay and transmission across networks of distributed processes.

5.4. Software Platforms

5.4.1. Robust robotics manipulation - Object detection and tracking

Participants: Antoine Hoarau [ADT Engineer Since Nov. 2012], Freek Stulp [Supervisor], David Filliat [Supervisor].

Autonomous human-centered robots, for instance robots that assist people with disabilities, must be able to physically manipulate their environment. There is therefore a strong interest within the FLOWERS team to apply the developmental approach to robotics in particular to the acquisition of sophisticated skills for manipulation and perception. ENSTA-ParisTech has recently acquired a Meka (cf. 3) humanoid robot dedicated to human-robot interaction, and which is perfectly fitted to this research. The goal of this project is to install state-of-the-art software architecture and libraries for perception and control on the Meka robot, so that this robot can be jointly used by FLOWERS and ENSTA. In particular, we want to provide the robot with an initial set of manipulation skills.

The goal is to develop a set of demos, which demonstrate the capabilities of the Meka, and provide a basis on which researchers can base their experiments. As the robot is not yet available at ENSTA, initial work focused on the robot's environment, meaning ROS and the M3 software (provided by Meka Robotics, based on both C++ and Python scripts) and on trying to implement a simple ball-catching demo : the idea is to throw a ball toward the robot which catch it (basic human-robot interaction, combining both perception and control). Different tracking algorithms are being tried for the ball, such as Camshift, Hough Circles + Kalman Filter, or more complex LineMod (all included in OpenCV) to finally estimate its trajectory for the robot to catch it. The M3 software provided by Meka Robotics contains a simulation environment that allows us to work without the robot hardware (cf. 4 .



Figure 3. The Meka robot platform acquired by ENSTA ParisTech

5.4.2. ErgoRobot/Flowers Field Software

Participants: Jérôme Béchu [correspondant], Pierre-Yves Oudeyer, Pierre Rouanet, Olivier Mangin, Fabien Benureau, Mathieu Lapeyre.

In the context of its participation to the exhibition “Mathematics: A Beautiful Elsewhere” at Fondation Cartier pour l’Art Contemporain in Paris (19th October 2011 to 18th March 2012), the team has elaborated and experimented a robotic experimental set-up called “Ergo-Robots/FLOWERS Fields”. This set-up is not only a way to share our scientific research on curiosity-driven learning, human-robot interaction and language acquisition with the general public, but, as described in the Results and Highlights section, attacks a very important technological challenge impacting the science of developmental robotics: How to design a robot learning experiment that can run continuously and autonomously for several months?

The global scenario for the robots in the installation/experiment is the following. In a big egg that has just opened, a tribe of young robotic creatures evolves and explores its environment, wreathed by a large zero that symbolizes the origin. Beyond their innate capabilities, they are outfitted with mechanisms that allow them to learn new skills and invent their own language. Endowed with artificial curiosity, they explore objects

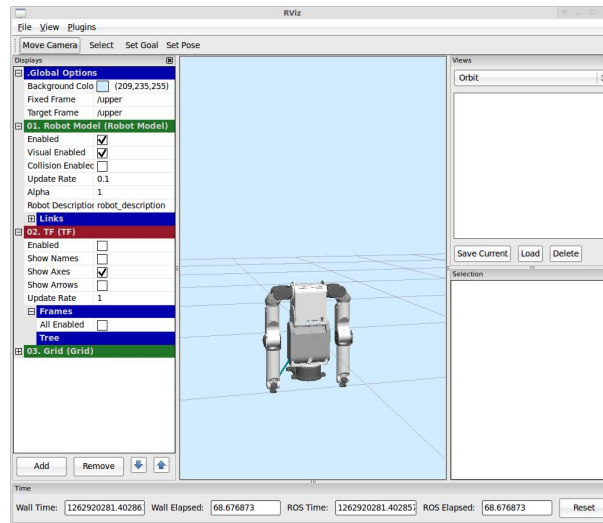


Figure 4. M3 simulation through ROS and Rviz

around them, as well as the effect their vocalizations produce on humans. Human, also curious to see what these creatures can do, react with their own gestures, creating a loop of interaction which progressively self-organizes into a new communication system established between man and ergo-robots.

We now outline the main elements of the software architectures underlying this experimental setup.

5.4.2.1. System components

The software architecture is organized to control the experiment at several levels, and in particular:

- **Scenes:** The organization of behavioural scenes, managing the behaviours that are allowed to each robot at particular times and in particular contexts;
- **Behaviours:** The individual behaviours of robots, also called stems, which are outlined in the next section;
- **stems:** The low-level actions and perceptin of robots while executing their behaviours, including motors control on the five physical stems, color and intensity of lights inside the stem head, production of sounds through speakers. Sensors are the kinect used to interact with visitors, and motor feedback capabilities.

In addition to that a video projector is used to display some artistic view of stem agents internal state.

5.4.2.2. Behaviours

A number of innate behaviours were designed and are used by the robots as elementary behaviours of more complex behaviours, including the three following learning behaviours.

The Naming Game is a behaviour played by stems two-by-two and based on computational models of how communities of language users can self-organize shared lexicons. In the naming game, stems interact with each other in a stylised interaction. Repeated interactions lead to the development of a common repertoire of words for naming objects. More precisely, object belong to meaning spaces. Two such spaces have been implemented for the exhibition. The first one is related to object spatial categorization and the second one is related to movement categorization. The object space contains stems, some hole in walls and the interaction zone. The movement space contains representations of small dances that stem can produce and reproduce.

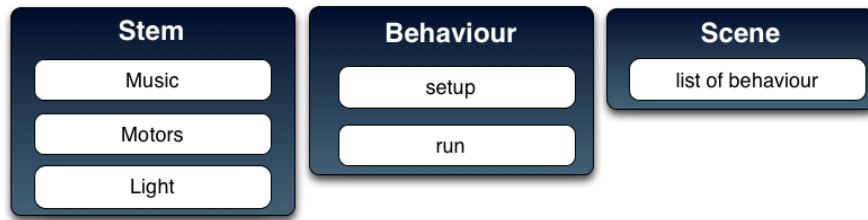


Figure 5. Three important concepts in ErgoRobots

Object Curiosity is a behaviour in controlling intrinsically motivated exploration of the physical environment by the stems. A small wood object is present in the reachable physical environment of the stem, attached on the top of a spring so that it is guaranteed that it comes back to its original position. The stem uses a motor primitive to act on the object and motor feedback to detect movements of the object. The robot learns through active exploration what kind of parameters motor primitive will result in touching the object.

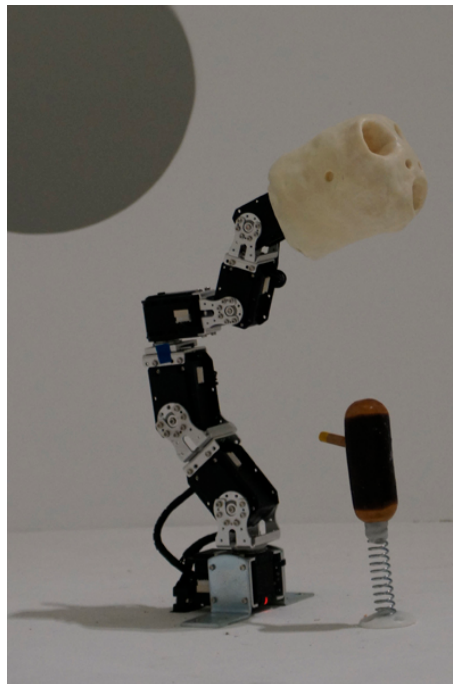


Figure 6. A Stem with the head designed by David Lynch and an Object

Birds Curiosity is a behaviour that drives robots to explore, through curiosity-driven learning, interaction with humans. One stem, generally the stem in the center, plays a sound, predicts the visitor reaction, look the interaction zone and wait the gesture of the visitor. To produce a sound the visitor have to make a gesture in space. In the next iterations, the robot chooses to produce sounds to human which produce most surprising

responses from the human (i.e. the robot is “interested” to explore sound interactions which are not easily predictable by itself).. As describe in the picture, the space is split in four. Each zone corresponding with a sound.

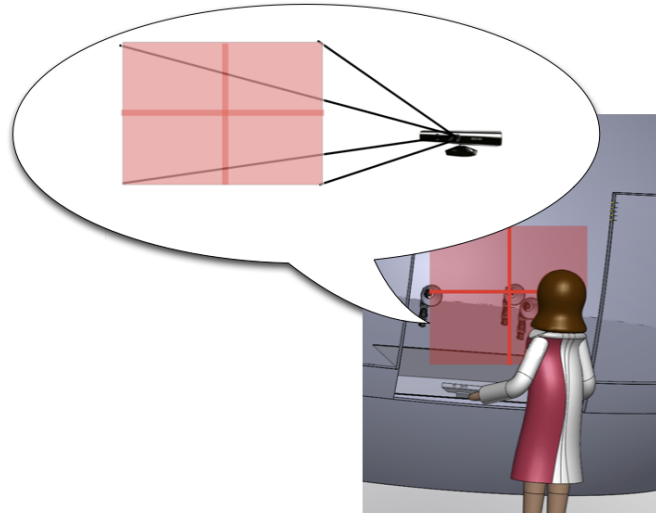


Figure 7. A virtual visitor interact with a virtual grid

5.4.2.3. Programming tools

The system is based on URBI and used some UObjects from UFlow. The most important part of the system is written in URBI script. Python and freenect⁴ are used too.

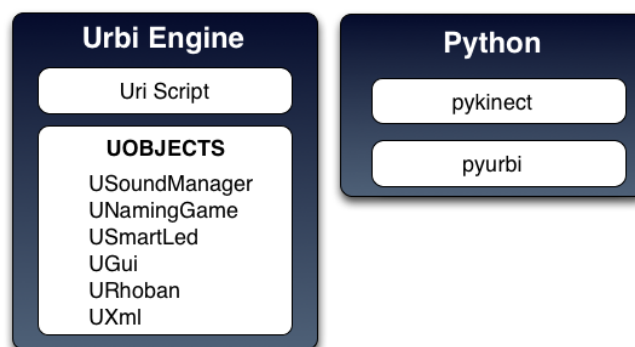


Figure 8. List of software used in ErgoRobots

⁴Kinect library

The system at the startup detects motors and lights. It create dynamically a list of Stem. A Stem is one robot with 6 motors as described in hardware part.

To interact with people, we used the freenect library to interface with the kinect, with a binding to python where detection and following of gestures is made.

For the display, we display an abstract rendering of the structure inside each ErgoRobot, using a python parser to read and parse log file from the ErgoRobot system, and the Bloom/Processing software to create and display the rendering. Currently, the system has three displays, one for the naming game, another one for birds curiosity and the last one for objects curiosity.

The sound system used the UObject USoundManager. It plays sounds when required by a behaviour, it also plays word sounds in Naming Game behaviour.

The Light system used Linkm technologies. In the head of each ErgoRobot we put two lights devices. Each light device is a RGB Light. We can control the intensity of each primary color through I2C control. To control lights we used LinkM USB Device. And finally we used an UObject dedicated to communicate with the USB Device.

5.4.2.4. Maintenance

A dedicate maintenance software is used to switch off, switch on the system. This software is written in Python (and Qt). The status of ErgoRobots is display on the graphical interface. Buttons are present too : Start, Stop, Reset and Take a video.

Recently we added a video system to have a visual feedback of motors usage and also to detect eventual problems. This is a screenshot of the application :

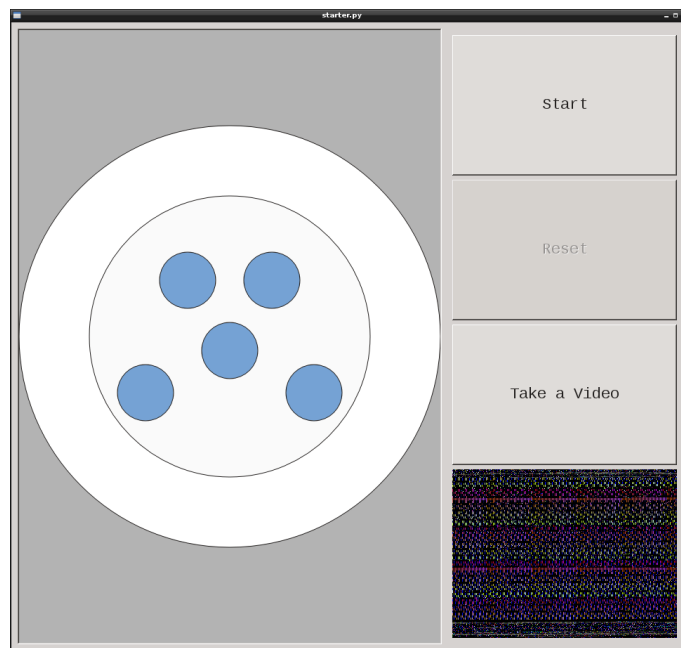


Figure 9. Maintenance Software for the ErgoRobots.

5.4.3. MonitorBoard - Complete solution for monitoring Rhoban Project robots

Participants: Paul Fudal [correspondant], Olivier Ly, Hugo Gimbert.

In collaboration with Rhoban Project/LaBRI/CNRS/Univ. Bordeaux I, the Flowers team took part in a project to exhibit robots at the International Exhibition in Yeosu - 2012 - South Korea (8 millions of visitors expected, from more than 100 countries). The installation consisted in three humanoids (one dancing, two playing on a spring) and five musicians (arms only) playing musical instruments (electric guitar, electric bass guitar, keytar, drums, DJ turntables). In order to increase the robustness of the robotic platform, a complete solution of software and hardware was build. The software solution aims to allow all robots to run safely during the whole exhibition (12 hours per days) and to provide an easy way to diagnose and identify potential electronic and mechanical failures. This software is able to monitor all robots at the same time, verify the health of each motors and each embedded systems. It is able to shutdown or reboot a robot if necessary using PowerSwitches (electric plugs controlled over network) and notify maintenance personal by email explaining the failure. All information is also logged for statistical use. This solution allows to monitor the whole platform without being present, and provides warning signs enabling preventive actions to be taken before an actual failures. It was entirely written in C# using Microsoft Visual Studio 2010 with .NET API and combined with the existing Rhoban Project API, extended and modified for this purpose. It also involved electric plugs controlled over a network connection.

5.4.4. Motor tracking system

Participants: Jérôme Béchu, Olivier Mangin [correspondant].

We developed a website interface to a database of motors used to build robots in the team. This system is designed for internal use in the team and was developed using the django web framework (<https://www.djangoproject.com/>).

5.5. Visualization Tools

5.5.1. Zephyr - Realtime Visualization in JAVA

Participant: Thomas Degris [correspondant].

Zephyr is a software to visualize numeric variables and data structure in real time and at different time scale. Zephyr is practical because it requires only minimal changes in the code: it uses Java reflexivity to automatically detect variables in the code to monitor and data structure with an associated dedicated view. Zephyr can easily be extended with new plugins because it is based on the popular Eclipse Rich Client Platform. Consequently, Zephyr takes advantage of an already existing and fully operational Eclipse plugins for many of its functionalities. Finally, Zephyr is distributed with a Java python virtual machine named Jython and a lisp implementation named Clojure. An example of a Zephyr screen is shown in Figure 10.

Zephyr was started in fall 2009 in the RLAI group at the university of Alberta (Canada) when Thomas Degris was a postdoc in this group. Zephyr is still actively used by RLAI. Users include Adam White, Joseph Modayil and Patrick Pilarski from the University of Alberta. Zephyr has been registered on the Eclipse marketplace since October 2011. Documentation about Zephyr is included on its website: <http://zephyrplugins.github.com>. Zephyr is licensed under the open source Eclipse Public License.

5.6. Hardware

5.6.1. Poppy Platform

Participant: Matthieu Lapeyre [correspondant].

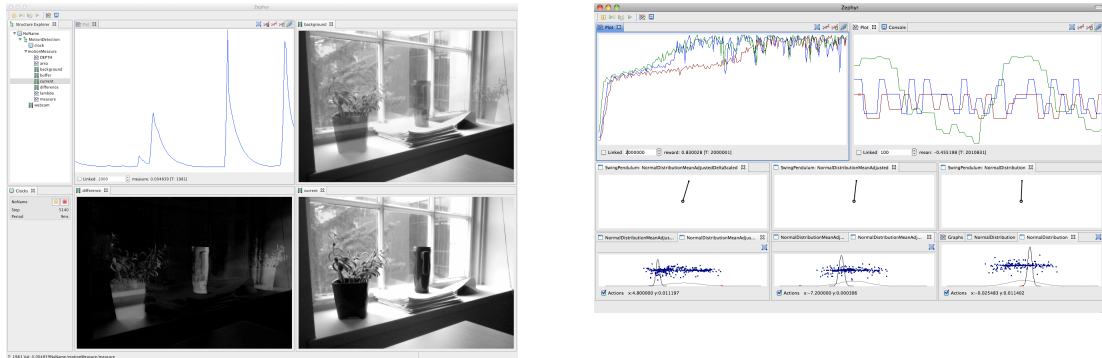


Figure 10. Left: Zephyr showing the different steps of a video processing pipeline in real-time. Right: Zephyr showing different data structure and variables of a reinforcement learning agent at different time scale. A video is available at: <http://zephyrplugins.github.com>.

5.6.1.1. Main goals :

No current platform (Nao [86], Darwin Op [87], Nimbro Op [117], HRP-2, ...) does offer both a adapted morphology in the sense of allowing physical interaction (safe, compliant, playful) and optimized for walking. So to explore these challenges we have decided to build a new bio-inspired humanoid robotic platform, called Poppy, which provides some of the software and hardware features needed to explore both social interaction and biped locomotion for personal robot. It presents the following main features to make it an interesting platform to study how the combination of morphology and social interaction can help the learning:

- Design inspired from the study of the anatomy of the human body and its bio-mechanic
- Dynamic and reactive: we try to keep the weight of the robot as low as possible (geometry of the pieces and smaller motors)
- Social interaction: screen for communication and permits physical interaction thanks to compliance
- Study of the morphology of the leg to improve the biped walking
- Practical platform: low cost, ease of use and easy to reproduce

5.6.1.2. Overview :

Poppy platform (Figure 11) is a humanoid, it is 84cm tall for 3 kg. It has a large sensor motors space including 25 dynamical motors (MX-28 and AX-12), force sensors under its feet and some extra sensors in the head: 2 HD-wide angle-cameras, stereo-micros and an inertial central unit (IMU 9DoF) plus a large LCD Screen (4 inch) for visual communication (e.g. emotions, instructions or debug). The mechanical parts were designed and optimized to be as light as possible while maintaining the necessary strength. For this, the choice of a lattice beam structure manufactured with 3Dprinting polyamide was used.

The poppy morphology is designed based on the actual human body. We have deeply studied the biomechanics of the human body and have extracted some interesting features for humanoid robotics. This inspiration is expressed in the whole structure (e.g. the limb proportions) and in particular in the trunk and legs.

Poppy uses the bio-inspired trunk system introduced by Acroban [98]. These five motors allow it to reproduce the main changes brought by the human spine. This feature allows the integration of more natural and fluid motion while improving the user experience during physical interactions. In addition, the spine plays a fundamental role in bipedal walking and postural balance by actively participating in the balancing of the robot.

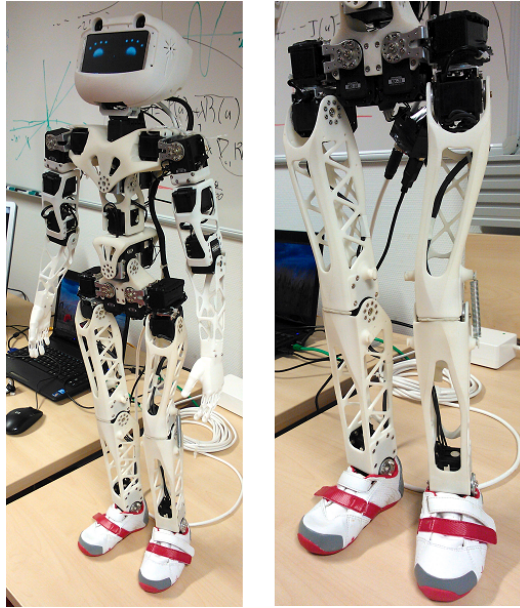


Figure 11. a. Global view of the Poppy platform. b. Zoom on legs design

The legs were designed to increase the stability and agility of the robot during the biped walking by combining bio-inspired, semi-passive, lightweight and mechanical-computation features. We will now describe two examples of this approach:

The architecture of the hips and thighs of Poppy uses biomechanical principles existing in humans. The human femur is actually slightly bent at an angle of about 6 degrees. In addition, the implantation of the femoral head in the hip is on the side. This results in a reduction of the lateral hip movement needed to move the center of gravity from one foot to another and a decrease in the lateral falling speed. In the case of Poppy, the inclination of its thighs by an angle of 6 degrees causes a gain of performance of more than 30% for the two above mentioned points.

Another example is Poppy's feet. Poppy has the particularity of having small feet compared to standard humanoids. It has humanly proportioned feet (ie about 15% of its total size). It is also equipped with compliant toes joints (see Figure 12 .a). We believe that this feet involve two keys features to obtain a human-like and efficient walking gait. However, that raises problems regarding balance because the support polygon is reduced. We decided to add pressure sensors under each foot in order to get accurate feedback of the current state of the robot (see Figure 12 .b).

5.6.1.3. Future works :

In our work, we explore the combination of both a bio-inspired body and bio-inspired learning algorithms. We are currently working on experiments involving Poppy to perform skill learning. First we would like to succeed in achieving an effective postural balance using the articulated spine, the feet pressure sensors and the IMU. Then, we would like to perform experiments on the learning of biped walking using algorithms such as the ones described in [95] or [83]. We are expecting to clearly reduce the learning time needed and increase the quality of the learned tasks thanks to the bio-inspired morphology of Poppy.

We are also interested in social interactions with non-expert users. We would like to conduct user study to evaluate how playful physical interactions and emotions could improve learning in robotics. We think that the poppy platform could be very suitable for such studies.

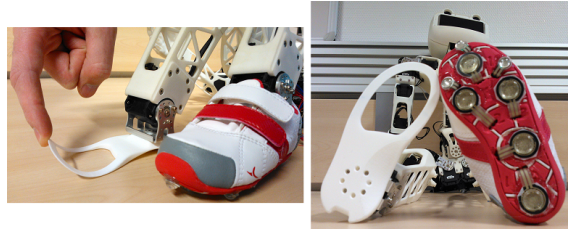


Figure 12. Poppy feet use actual children shoes combine with a compliant feet, toes (a.) and pressure sensors (b.)

5.6.2. Ergo-Robots/FLOWERS Fields: Towards Large-Scale Robot Learning Experiments in the Real World

Participants: Jerome Bechu, Fabien Benureau, Haylee Fogg, Paul Fudal, Hugo Gimbert, Matthieu Lapeyre, Olivier Ly, Olivier Mangin, Pierre Rouanet, Pierre-Yves Oudeyer.

In the context of its participation to the exhibition “Mathematics: A Beautiful Elsewhere” at Fondation Cartier pour l’Art Contemporain in Paris, starting from 19th October 2011 and to be held until 18th March 2012, the team, in collaboration with Labri/Univ. Bordeaux I, has elaborated and experimented a robotic experimental set-up called “Ergo-Robots/FLOWERS Fields” ¹³. This set-up is not only a way to share our scientific investigations with the general public, but attacks a very important technological challenge impacting the science of developmental robotics: How to design a robot learning experiment that can run continuously and autonomously for several months? Indeed, developmental robotics takes life-long learning and development as one of its central objective and object of study, and thus shall require experimental setups that allow robots to run, learn and develop for extended periods of time. Yet, in practice, this has not been possible so far due to the unavailability of platforms adapted at the same time to learning, exploration, easy and versatile reconfiguration, and extended time of experimentation. Most experiments so far in the field have a duration ranging from a few minutes to a few hours. This is an important obstacle for the progress of developmental robotics, which would need experimental set-ups capable of running for several months. This is exactly the challenge explored by the Ergo-Robots installation, which we have approached by using new generations of affordable yet sophisticated and powerful off-the-shelf servomotors (RX Series from Robotis) combined with an adequately designed software and hardware architecture, as well as processes for streamlined maintenance. The experiment is now running for five months, six days a week, in a public exhibition which has strong constraints over periods of functioning and no continual presence of dedicated technicians/engineers on site. The experiment involved five robots, each with 6 degrees of freedoms, which are endowed with curiosity-driven learning mechanisms allowing them to explore and learn how to manipulate physical objects around them as well as to discover and explore vocal interactions with humans/the visitors. The robots are also playing language games allowing them to invent their own linguistic conventions. A battery of measures has been set up in order to study the evolution of the platform, with the aim of using the results (to be described in an article) as a reference for building future robot learning experiments on extended periods of time, both within the team and in the developmental robotics community. The system has been running during 5 months, 8 hours a day, with no major problems. During the two first months, the platform worked during 390h21mn, and was only stopped during 24h59mn (6 percent of time). After retuning the system based on what we learnt in the two first months, this performance was increased in the three last months: the platform worked for 618h23mn and was only stopped during 17h56mn (2.9 percent of time).

More information available at: <http://flowers.inria.fr/ergo-robots.php> and <http://fondation.cartier.com/>.

5.6.2.1. The Ergo-Robots Hardware Platform

Participants: Jerome Bechu [correspondant], Fabien Benureau, Haylee Fogg, Hugo Gimbert, Matthieu Lapeyre, Olivier Ly, Olivier Mangin, Pierre-Yves Oudeyer, Pierre Rouanet.



Figure 13. The Ergo-Robot experiment: robot learning experiment running continuously for 5 months at Fondation Cartier pour l'Art Contemporain, exhibition "Mathématiques: Un Dépaysement Soudain".

ErgoRobots [13](#) is a hardware platform for showcasing a number of curiosity and learning behaviours for the public to interact with. It was designed by the Flowers team in collaboration with Labri/Univ. Bordeaux I. The platform can also have future uses inside the lab for experiments that require more than one robot to complete. Although this system is entirely new this year, a very different previous version existed with the name FLOWERSField. It consists of five ErgoRobots, a control system, an interaction system, a display system, a sound system and a light system. There is an external system which monitors the ErgoRobots which contains a control system, a power system, a surveillance system and a metric capture system. The system has been running during 5 months, 8 hours a day, with no major problems. During the two first months, the platform worked during 390h21mn, and was only stopped during 24h59mn (6 percent of time). After retuning the system based on what we learnt in the two first months, this performance was increased in the three last months: the platform worked for 618h23mn and was only stopped during 17h56mn (2.9 percent of time).

The Ergo-Robot system: The robots themselves are each composed of six motors (see figure). Currently, the heads of the robots have been created in wax by David Lynch and the entire system is displayed at Fondation Cartier inside a large egg shaped orb as shown in the following diagram. The control system module contains both an MMNET1002 control board with an UART-RS485 breakout board which communicates with a ubuntu Linux PC via an ethernet cable. The mment board communicates with the motors, but all other ErgoRobot systems communicate with the PC directly. The sound system is currently externally provided and communicates with the PC. The light system is a series of two or three BlinkM RGB leds placed inside each ErgoRobot head that are controlled through two LinkM USB devices directly with the computer. A kinect placed in front of the system operates as the means for the public to interact with the platform and communicates directly through USB to the PC. The display system is currently an externally provided projector that projects visualisations of the field's current state behind the ErgoRobots.

The external system: This system allows anyone that is monitoring the system to externally control the ErgoRobots system. The PC with which the software control takes place is a Ubuntu Linux system which communicates with the ErgoRobot control system via an ethernet cable. The ErgoRobot hardware system can be managed by an external power system which includes a 15.5V bench top power supply for the ErgoRobot motors, an external 12V plug in adapter for the mment board, an external 5V plug in adapter for the LED lights which are all controlled via an emergency stop button. The maintenance system can be located out of direct

view of the ErgoRobot field as it has a surveillance system: a kinect that can display the current state of the field. More surveillance is conducted through a metric capture system that communicates with the ErgoRobots to obtain various state values of the ErgoRobots through the motor sensors and other data. This surveillance is not entirely in place as of 2011 and will be implemented in early 2012.

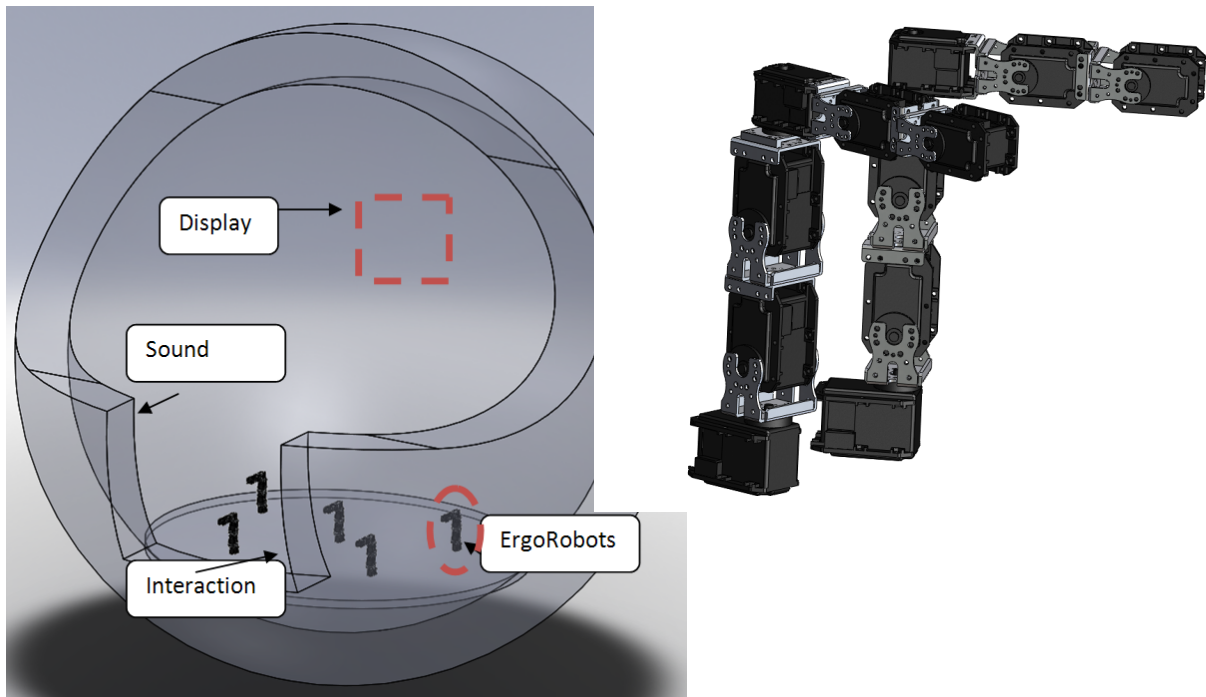


Figure 14. Ergo-Robots

5.6.2.2. Stem Platform for Affordances

Participant: Fabien Benureau [correspondant].

The Stem Platform for Affordances (figure 15) is a hardware platform that is intended for use in the lab for experiments. It features a 6 DOFs arm robot identical to the other robot stems present in the lab, and a physical platform intended for the interaction with objects. Our affordance experiments involves a lot of trials; there was the need for a platform that could reset itself after the robot interacted, as it is an assumption underlying our current algorithms. The stem platform provides exactly that, with the object position and orientation being reset by the platform autonomously and in less than 10 seconds. This provides the potential to do more than 2000 independent interactions with an object over the course of 12 hours.

The platform also provides sensory capabilities, being able to track the position and orientation of the object at all time. On the hardware side, a camera is used. We investigated both a standard PSEye, that provides a high framerate (120Hz) with noise, and a high quality, firewire camera with professional optics, providing higher resolution, low noise at the expense of a low framerate (15Hz). The latter was kindly provided by the Potioc team. On the software side, computing the position is done by the open-source augmented reality library ARToolkitPlus. On the objects themselves, AR tags are placed.

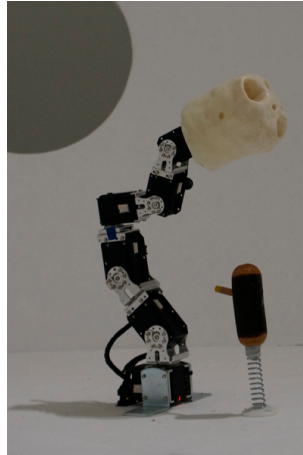


Figure 15. The Stem Platform

The platform is supported by a simulation that reproduce the setup in V-Rep. In order to be able to use the same algorithms for both the hardware and the simulation, a low-level interface was written for Pypot and V-Rep, using the work done by Paul Fudal on V-Rep Bridge.

The complete platform took roughly 3 weeks to make, with 3 additional weeks for the software. The team recently acquired material that would make possible to build a similar platform faster and in a more robust material (wood is used in the first platform). This platform, backed up by its simulation, will allow us to perform planned experiments in a reliable and statistically significant manner.

5.6.2.3. Humanoid Robot Torso

Participant: Haylee Fogg [correspondant].

The Humanoid Robot Torso is a hardware platform that is intended for use in the lab for either experiments or demonstrations [16](#) . It consists of a humanoid robot that contains just a torso, arms with shoulders and grippers, and head. It is entirely new this year, as a new design has been made, and a skeleton built with 3D printing technologies. The arms with the claws contain seven degrees of freedom (including 'grip'). The head consists of a smartphone for the face and an associated camera for the 'eyes' with the ability to move in two degrees (pitch and roll). The hardware is both robotis Dynamixel RX-28 and R-64 motors attached together with standard robotis frames and 3D printed limbs. A wiki has been built, documenting both the hardware and software platform.

5.6.2.4. NoFish platform

Participants: Mai Nguyen [correspondant], Paul Fudal [correspondant], Jérôme Béchu.

The NoFish platform is a hardware platform that is intended for use in the lab for experiments. It consists of an ErgoRobot with an attached fishing rod. The robot is fixed on a table and has in front of him a delimited area where to throw the fishing cap. This area is covered by a camera in order to track the fishing cap and to give its coordinates. The robot is managed by a software written using the Urbi framework. This program controls the robot using pre-programmed moves and also gives a way to uses the robot joint by joint. A second software written in C++ using OpenCV framework tracks the position of the fishing cap and sends the coordinates to the Urbi software controlling the robot. Finally, at the upper layer of the software architecture, MatLab is used to implement different learning algorithms. All MatLab code is able to receive informations from the Urbi part of the software (fishing cap coordinates, joints informations, etc) and also to send order to the robot (position joint by joint, preprogrammed moves, etc). To finish, and because the platform can run a learning algorithms

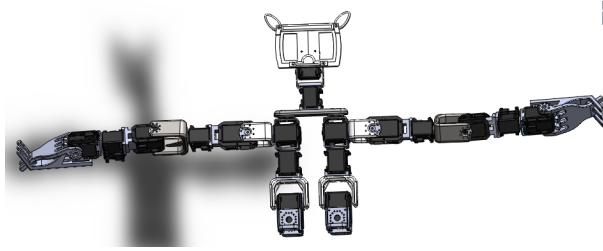


Figure 16. The Humanoid Robot Torso Platform

during a long time, an electric plug managed by the Urbi part of the software is added to the platform to shutdown the power if the robot is blocked or does not respond anymore.

FLUMINANCE Project-Team

5. Software

5.1. DenseMotion software - Estimation of 2D dense motion fields

Participants: Thomas Corpetti, Patrick Héas, Etienne Mémin.

This code allows the computation from two consecutive images of a dense motion field. The estimator is expressed as a global energy function minimization. The code enables the choice of different data model and different regularization functional depending on the targeted application. Generic motion estimator for video sequences or dedicated motion estimator for fluid flows can be specified. This estimator allows in addition the users to specify additional correlation based matching measurements. It enables also the inclusion of a temporal smoothing prior relying on a velocity vorticity formulation of the Navier-Stoke equation for Fluid motion analysis applications. The different variants of this code correspond to research studies that have been published in IEEE transaction on Pattern Analysis and machine Intelligence, Experiments in Fluids, IEEE transaction on Image Processing, IEEE transaction on Geo-Science end Remote Sensing. The binary of this code can be freely downloaded on the FLUID web site <http://fluid.irisa.fr>.

5.2. 2DLayeredMotion software - Estimation of 2D independent mesoscale layered atmospheric motion fields

Participants: Patrick Héas, Etienne Mémin.

This software enables to estimate a stack of 2D horizontal wind fields corresponding to a mesoscale dynamics of atmospheric pressure layers. This estimator is formulated as the minimization of a global energy function. It relies on a vertical decomposition of the atmosphere into pressure layers. This estimator uses pressure data and classification clouds maps and top of clouds pressure maps (or infra-red images). All these images are routinely supplied by the EUMETSAT consortium which handles the Meteosat and MSG satellite data distribution. The energy function relies on a data model built from the integration of the mass conservation on each layer. The estimator also includes a simplified and filtered shallow water dynamical model as temporal smoother and second-order div-curl spatial regularizer. The estimator may also incorporate correlation-based vector fields as additional observations. These correlation vectors are also routinely provided by the Eumetsat consortium. This code corresponds to research studies published in IEEE transaction on Geo-Science and Remote Sensing. It can be freely downloaded on the FLUID web site <http://fluid.irisa.fr>.

5.3. 3DLayeredMotion software - Estimation of 3D interconnected layered atmospheric motion fields

Participants: Patrick Héas, Etienne Mémin.

This software extends the previous 2D version. It allows (for the first time to our knowledge) the recovery of 3D wind fields from satellite image sequences. As with the previous techniques, the atmosphere is decomposed into a stack of pressure layers. The estimation relies also on pressure data and classification clouds maps and top of clouds pressure maps. In order to recover the 3D missing velocity information, physical knowledge on 3D mass exchanges between layers has been introduced in the data model. The corresponding data model appears to be a generalization of the previous data model constructed from a vertical integration of the continuity equation. This research study has been recently accepted for publication in IEEE trans. on Geo-Science and Remote Sensing. A detailed description of the technique can be found in an Inria research report. The binary of this code can be freely downloaded on the FLUID web site <http://fluid.irisa.fr>.

5.4. Low-Order-Motion - Estimation of low order representation of fluid motion

Participants: Anne Cuzol, Etienne Mémin.

This code enables the estimation of a low order representation of a fluid motion field from two consecutive images. The fluid motion representation is obtained using a discretization of the vorticity and divergence maps through regularized Dirac measure. The irrotational and solenoidal components of the motion fields are expressed as linear combinations of basis functions obtained through the Biot-Savart law. The coefficient values and the basis function parameters are obtained as the minimizer of a functional relying on an intensity variation model obtained from an integrated version of the mass conservation principle of fluid mechanics. Different versions of this estimation are available. The code which includes a Matlab user interface can be downloaded on the FLUID web site <http://fluid.irisa.fr>. This program corresponds to a research study that has been published in the International Journal on computer Vision.

FOCUS Project-Team

5. Software

5.1. Jolie

Members of Focus have developed Jolie [8] (Java Orchestration Language Interpreter Engine, see <http://www.jolie-lang.org/>). Jolie is a service-oriented programming language. Jolie can be used to program services that interact over the Internet using different communication protocols. Differently from other Web Services programming languages such as WS-BPEL, Jolie is based on a user-friendly C/Java-like syntax (more readable than the verbose XML syntax of WS-BPEL) and, moreover, the language is equipped with a formal operational semantics. This language is used for the *proof of concepts* developed around Focus activities. For instance, contract theories can be exploited for checking the conformance of a Jolie program with respect to a given contract. A spin-off, called “Italiana Software”, has been launched around Jolie, its general aim is to transfer the expertise in formal methods for Web Services matured in the last few years onto Service Oriented Business Applications. The spin-off is a software producer and consulting company that offers service-oriented solutions (for instance, a “single sign-on” application) based on the Jolie language.

In 2012 the development of Jolie has continued. The main activities have been:

- We have enhanced the correlation mechanism in Jolie to handle multiparty sessions with concurrent interactions with multiple participants.
- We have developed a compiler that projects choreography-based programs in Chor (<http://www.chor-lang.org>) to Jolie.
- We have developed a new website for Jolie (using Jolie itself), significantly updating its documentation.
- We have improved Jolie’s compatibility with the Java RMI technology.
- We have developed a first experimental implementation of a monitoring layer for Jolie services.

As last year, so in 2012 Jolie has been used for teaching, in a master course at the IT University of Copenhagen (ITU, Denmark) and in a master course at the Technical University of Denmark (DTU, Denmark).

5.2. Others

Below we list some software that has been developed, or is under development, in Focus.

- *IntML* is a functional programming language guaranteeing sublinear space bounds for all programs [53]. See the Activity Reports of previous years (in particular 2010) for more details. During 2012 no substantial modifications have been made.
- *Lideal* (<http://lideal.cs.unibo.it/>) is an experimental tool implementing type inference for dependently linear type systems. The tool reduces the problem of evaluating the complexity of PCF (i.e. functional programs with primitive integers and recursive definitions) to checking a set of first-order inequalities for validity. The latter can then be handled through SMT solvers or put in a form suitable for managing them with tools such as CoQ.
- We have implemented a technique for the deadlock analysis of a concurrent object oriented language (ABS, designed within the European project HATS). The technique consists of
 - an inference system for contracts to be associated to methods. Contracts are terms that retain information about resource dependencies;
 - a fixpoint algorithm for solving contract definitions, which are recursive and may introduce new resource names.

The release of the software is planned for early 2013.

- *Croll-pi Interpreter* (<http://proton.inrialpes.fr/~mlienhar/croll-pi/implem/>). We have developed an interpreter for croll-pi using Maude. Croll-pi is a concurrent reversible language featuring a rollback operator to undo a past action (together with all the actions depending on it), and a compensation mechanism to avoid cycling by redoing the same action again and again.

We used the interpreter to test the expressive power of croll-pi on various problems, including the 8-queen problem, error handling in an automotive scenario from the EU project Sensoria, and constructs for distributed error handling such as stabilizers.

For other software, such as PiDuce, see the activity reports for Focus of previous years.

FORMES Team

5. Software

5.1. aCiNO

Participants: Fei He [correspondant], Min Zhou.

aCiNO is an SMT (Satisfiability Modulo Theory) solver based on a Nelson-Oppen [62] architecture, and written in C++. Currently, two popular theories are considered: linear real arithmetic (LRA) and uninterpreted functions (UF). A lazy approach is used for solving SMT problem. For efficiency consideration, the solver is implemented in an incremental way. It also invokes an online SAT solver, which is now a modified MiniSAT, so that recovery from conflict is possible.

5.2. CoLoR

Participants: Frédéric Blanqui [correspondant], Kim-Quyen Ly.

CoLoR is a Coq [42] library on rewriting theory and termination of more than 72,000 lines of code [4]. It provides definitions and theorems for:

- Mathematical structures: relations, (ordered) semi-rings.
- Data structures: lists, vectors, polynomials with multiple variables, finite multisets, matrices.
- Term structures: strings, algebraic terms with symbols of fixed arity, algebraic terms with varyadic symbols, simply typed lambda-terms.
- Transformation techniques: conversion from strings to algebraic terms, conversion from algebraic to varyadic terms, arguments filtering, rule elimination, dependency pairs, dependency graph decomposition, semantic labelling.
- Termination criteria: polynomial interpretations, multiset ordering, lexicographic ordering, first and higher order recursive path ordering, matrix interpretations.

CoLoR is distributed under the CeCILL license on <http://color.inria.fr/>. Various people participated to its development (see the website for more information).

5.3. CoqMT

Participants: Qian Wang [correspondant], Jean-Pierre Jouannaud.

The proof-assistant Coq is based on a complex type theory, which resulted from various extensions of the Calculus of Constructions studied independently from each other. With the collaboration of Bruno Barras, we decided to address the challenge of proving the real type theory underlying Coq, and even, indeed, of its recent extension CoqMT developed in FORMES by Pierre-Yves Strub. To this end, we have studied formally the theory CoqMTU, which extends the pure Calculus of Constructions by inductive types, a predicative hierarchy of universes, and a decidable theory T for some first-order inductive types [1]. Recently, we were able to announce the complete certification of CoqMTU in Coq augmented with appropriate intuitionistic set-theoretic axioms in order to fight Gödel's incompleteness theorem, a work which has not been published yet. As a consequence, Coq and CoqMTU are the first proof assistants which consistency (relative to intuitionistic set theory IZF augmented with the afore-mentioned axioms) is formally entirely proved (in Coq). While previous formal proofs for Coq and other proof assistants all assumed strong normalization, the present one *proves* strong normalization thanks to the new notion of *strongly-normalizing* model introduced by Bruno Barras. While consistency is done already, decidability of type-checking remains to be done. This is a straightforward consequence for Coq, but a non-trivial task for CoqMTU because of the interaction between inductive types and the first-order theory T. It should however be announced around the turn of the year. We consider this work as a major scientific achievement of the team.

5.4. EDOLA

Participants: Hehua Zhang [correspondant], Ming Gu, Hui Kong.

Joint work with Jiaguang Sun (Tsinghua University, China).

EDOLA [72] is an integrated tool for domain-specific modeling and verification of PLC applications [70]. It is based on a domain-specific modeling language to describe system models. It supports both model checking and automatic theorem proving techniques for verification. The goal of this tool is to possess both the usability in domain modeling, the reusability in its architecture and the capability of automatic verification.

For the moment, we have developed a prototype of the EDOLA language, which can easily describe the features of PLC applications like the scan cycle mechanism, the pattern of environment model, time constraints and five property patterns. TLA+ [56] was chosen as the intermediate language to implement the automatic verification of EDOLA models. A prototype of EDOLA has also been developed, which comes along with an editor to help writing EDOLA models. To automatically verify properties on EDOLA models, it provides the interface for both a model checker TLC [56] and a first-order theorem prover SPASS [71].

5.5. HOT

Participant: Frédéric Blanqui [correspondant].

HOT is an automated termination prover for higher-order rewrite systems based on the notion of computability closure and size annotation [13]. It won the 2012 **competition** in the category “higher-order rewriting union beta”. The sources are not public.

5.6. Moca

Participant: Frédéric Blanqui [correspondant].

Joint work with Pierre Weis (Inria Rocquencourt) and Richard Bonichon (CEA).

Moca is a construction functions generator for OCaml [57] data types with invariants.

It allows the high-level definition and automatic management of complex invariants for data types. In addition, it provides the automatic generation of maximally shared values, independently or in conjunction with the declared invariants.

A relational data type is a concrete data type that declares invariants or relations that are verified by its constructors. For each relational data type definition, Moca compiles a set of construction functions that implements the declared relations.

Moca supports two kinds of relations:

- predefined algebraic relations (such as associativity or commutativity of a binary constructor),
- user-defined rewrite rules that map some pattern of constructors and variables to some arbitrary user’s define expression.

The properties that user-defined rules should satisfy (completeness, termination, and confluence of the resulting term rewriting system) must be verified by a programmer’s proof before compilation. For the predefined relations, Moca generates construction functions that allow each equivalence class to be uniquely represented by their canonical value.

Moca is distributed under QPL on <http://moca.inria.fr/>.

5.7. Rainbow

Participants: Frédéric Blanqui [correspondant], Kim-Quyen Ly.

Rainbow is a tool for verifying the correctness of termination certificates expressed in the CPF XML format as used in the termination **competition**. Termination certificates are currently translated and checked in Coq by using the CoLoR library. But a new standalone version is under development using Coq extraction mechanism.

Rainbow is distributed under the CeCILL license on <http://color.inria.fr/rainbow.html>. See the website for more information.

5.8. SimSoC

Participant: Vania Joloboff [correspondant].

SimSoC is an infrastructure to run simulation models which comes along with a library of simulation models. SimSoC allows its users to experiment various system architectures, study hardware/software partition, and develop embedded software in a co-design environment before the hardware is ready to be used. SimSoC aims at providing high performance, yet accurate simulation, and provide tools to evaluate performance and functional or non functional properties of the simulated system.

SimSoC is based on SystemC standard and uses Transaction Level Modeling for interactions between the simulation models. The current version of SimSoC is based on the open source libraries from the OSCI Consortium: SystemC version 2.2 and TLM 2.0.1 [52], [25]. Hardware components are modeled as TLM models, and since TLM is itself based on SystemC, the simulation is driven by the SystemC kernel. We use standard, unmodified, SystemC (version 2.2), hence the simulator has a single simulation loop.

The second open source version of SimSoC, SimSoC v0.7.1, has been released in November 2010. It contains a full simulator for ARM V5 and PowerPC both running at an average speed of about 80 Millions instructions per second in, and a simulator for the MIPS architecture with an average speed of 20 Mips in mode DT1. It represents about 70,000 lines of source code and includes:

SimSoC is distributed under LGPL on <https://gforge.inria.fr/projects/simsoc>.

5.9. SimSoC-Cert

Participants: Frédéric Blanqui, Vania Joloboff, Jean-François Monin [correspondant], Xiaomu Shi.

SimSoC-Cert is a set of tools that can automatically generate in various target languages (Coq and C) the decoding functions and the state transition functions of each instruction and addressing mode of the ARMv6 architecture manual [22] (implemented by the ARM11 processor family) but the Thumb and coprocessor instructions. The input of SimSoC-Cert is the ARMv6 architecture manual itself.

Based on this, we first developed *simlight* (8000 generated lines of C, plus 1500 hand-written lines of C), a simulator for ARMv6 programs using no peripheral and no coprocessor. Next, we developed *simlight2*, a fast ARMv6 simulator integrated inside a SystemC/TLM module, now part of SimSoC v0.7.

We can also generate similar programs for SH4 [24] but this is still experimental (work done by Frédéric Tuong in 2011).

Finally, we started to prove that the C code for simulating ARM instructions in Simlight is correct with respect to the Coq model.

FUN Team

4. Software

4.1. Distributed ONS

Participants: Nathalie Mitton, Roberto Quilez [correspondant].

This module implements a DHT-based Distributed EPC Global ONS issued from the ANR WINGS project and published in [30]. APP number: IDDN.FR.001.180033.000.S.P.2012.000.10000.

- Version: version 1

4.2. GOLIATH 1.0

Participants: Fadila Khadar [correspondant], Nathalie Mitton.

GOLIATH (Generic Optimized LIghtweight communication stack for Ambient TecHnologies) is a full protocol stack for wireless sensor networks.

See also the web page <https://gforge.inria.fr/projects/goliath/>.

4.3. Linear variable energy module for WSNET.

Participants: Tony Ducrocq [correspondant], Nathalie Mitton.

This module is to be integrated in the WSNET event-based simulator for wireless networks. It implements a Linear transmission variable energy module for WSNET.

- Version: 1.0

GALAAD Project-Team

5. Software

5.1. Mathemagix, a free computer algebra environment

Participant: Bernard Mourrain.

<http://www.mathemagix.org/>

algebra, univariate polynomial, multivariate polynomial, matrices, series, fast algorithm, interpreter, compiler, hybrid software.

MATHEMAGIX is a free computer algebra system which consists of a general purpose interpreter, which can be used for non-mathematical tasks as well, and efficient modules on algebraic objects. It includes the development of standard libraries for basic arithmetic on dense and sparse objects (numbers, univariate and multivariate polynomials, power series, matrices, etc., based on FFT and other fast algorithms). These developments, based on C++, offer generic programming without losing effectiveness, via the parameterization of the code (*template*) and the control of their instantiations.

The language of the interpreter is imperative, strongly typed and high level. A compiler of this language is available. A special effort has been put on the embedding of existing libraries written in other languages like C or C++. An interesting feature is that this extension mechanism supports template types, which automatically induce generic types inside Mathemagix. Connections with GMP, MPFR for extended arithmetic, LAPACK for numerical linear algebra are currently available in this framework.

The project aims at building a bridge between symbolic computation and numerical analysis. It is structured by collaborative software developments of different groups in the domain of algebraic and symbolic-numeric computation.

In this framework, we are working more specifically on the following components:

- REALROOT: a set of solvers using subdivision methods to isolate the roots of polynomial equations in one or several variables; continued fraction expansion of roots of univariate polynomials; Bernstein basis representation of univariate and multivariate polynomials and related algorithms; exact computation with real algebraic numbers, sign evaluation, comparison, certified numerical approximation.
- SHAPE: tools to manipulate curves and surfaces of different types including parameterised, implicit with different type of coefficients; algorithms to compute their topology, intersection points or curves, self-intersection locus, singularities, ...

These packages are integrated from the former library SYNAPS (SYmbolic Numeric APplicationS) dedicated to symbolic and numerical computations. There are also used in the algebraic-geometric modeler AXEL.

Collaborators: Grégoire Lecerf, Joris van der Hoeven and Philippe Trébuchet.

5.2. Axel, a geometric modeler for algebraic objects

Participants: Anaïs Ducoffe, Bernard Mourrain, Meriadeg Perrinel.

<http://axel.inria.fr>.

computational algebraic geometry, curve, implicit equation, intersection, parameterisation, resolution, surface, singularity, topology

We are developing a software called AXEL (Algebraic Software-Components for gEometric modeLing) dedicated to algebraic methods for curves and surfaces. Many algorithms in geometric modeling require a combination of geometric and algebraic tools. Aiming at the development of reliable and efficient implementations, AXEL provides a framework for such combination of tools, involving symbolic and numeric computations.

The software contains data structures and functionalities related to algebraic models used in geometric modeling, such as polynomial parameterisation, B-Spline, implicit curves and surfaces. It provides algorithms for the treatment of such geometric objects, such as tools for computing intersection points of curves or surfaces, detecting and computing self-intersection points of parameterized surfaces, implicitization, for computing the topology of implicit curves, for meshing implicit (singular) surfaces, etc.

The developments related to isogeometric analysis have been integrated as dedicated plugins. Optimisation techniques and solvers for partial differential equations developed by R. Duvigneau (OPALE) have been connected.

A new version of the algebraic-geometric modelers is developed by Meriadeg Perinnet to connect it to the platform Dtk in order to provide a better modularity and a better interface to existing computation facilities and geometric rendering interface.

The package is distributed as binary packages for Linux as well as for MacOSX. It is hosted at Inria's gforge (<http://gforge.inria.fr>) and referenced by many leading software websites such as <http://apple.com>. The first version of the software has been downloaded more than 15000 times, since it is available.

Collaboration with Gang Xu (Hangzhou Dianzi University, China), Julien Wintz (Dream).

GALEN Team

5. Software

5.1. Deformable Registration Software

Participant: Nikos Paragios [Correspondant].

deformable image and volume registration, is a deformable registration platform in C++ for the medical imaging community (publicly available at <http://www.mrf-registration.net>) developed mainly at Ecole Centrale, Technical University of Munich and University of Crete. This is the first publicly available platform which contains most of the existing metrics to perform registration under the same concept. The platform is used for clinical research from approximately 3,000 users worldwide.

5.2. Dense image and surface descriptors

Participant: Iasonas Kokkinos [Correspondant].

Scale-Invariant Descriptor, Scale-Invariant Heat Kernel Signatures DISD (publicly available at <http://vision.mas.ecp.fr/Personnel/iasonas/descriptors.html>) implements the SID, SI-HKS and ISC descriptors. SID (Scale-Invariant Descriptor) is a densely computable, scale- and rotation- invariant descriptor. We use a log-polar grid around every point to turn rotation/scalings into translation, and then use the Fourier Transform Modulus (FTM) to achieve invariance. SI-HKS (Scale-Invariant Heat Kernel Signatures) extract scale-invariant shape signatures by exploiting the fact that surface scaling amounts to multiplication and scaling of a properly sampled HKS descriptor. We apply the FTM trick on HKS to achieve invariance to scale changes. ISC (Intrinsic Shape Context) constructs a net-like grid around every surface point by shooting outwards and tracking geodesics. This allows us to build a meta-descriptor on top of HKS/SI-HKS that takes neighborhood into account, while being invariant to surface isometries.

5.3. Dissimilarity Coefficient learning

Participant: Pawan Kumar [Correspondant].

weakly supervised learning, dissimilarity coefficient, structured prediction DISC (publicly available at <http://www.centrale-ponts.fr/personnel/pawan/code/DISCAPI.zip>) software provides a convenient API for dissimilarity coefficient (DISC) based learning. DISC allows the use of weakly supervised datasets (with missing information) by jointly learning a structured prediction classifier and a conditional probability distribution of the missing information. The parameters of the classifier and the distribution are learned by minimizing a user-specified dissimilarity coefficient between them.

5.4. Efficient bounding-based object detection

Participant: Iasonas Kokkinos [Correspondant].

branch-and-bound, parts detection, segmentation, DPMS implements branch-and-bound object detection, cutting down the complexity of detection from linear in the number of pixels to logarithmic (publicly available at <http://vision.mas.ecp.fr/Personnel/iasonas/dpms.html>). The results delivered are identical to those of the standard deformable part model detector, but are available in 5 to 20 times less time. This website has been visited 1500 times in 10 months.

5.5. Fast Primal Dual Strategies for Optimization of Markov Random Fields

Participant: Nikos Komodakis [Correspondant].

discrete optimization, Markov random field, duality, graph cuts, FASTPD is an optimization platform in C++ for the computer vision and medical imaging community (publicly available at <http://www.csd.uoc.gr/~komod/FastPD/>) developed mainly at Ecole Centrale and University of Crete. This is the most efficient publicly available platform in terms of a compromise of computational efficiency and ability to converge to a good minimum for the optimization of generic MRFs. The platform is used from approximately 1,500 users worldwide.

5.6. imaGe-based Procedural Modeling Using Shape Grammars

Participant: Iasonas Kokkinos [Correspondant].

procedural modeling, image-based building reconstruction, shape grammars GRAPES is a generic image parsing library based on re-inforcement learning (publicly available at <http://cvc.centrale-ponts.fr/>). It can handle grammars (binary-split, four-color, Hausmannian) and image-based rewards (Gaussian mixtures, Randomized Forests) of varying complexity while being modular and computationally efficient both in terms of grammar and image rewards. The platform is used from approximately 500 users worldwide.

5.7. Learning-based symmetry detection

Participant: Stavros Tsogkas [Correspondant].

Scale-Invariant Descriptor, Scale-Invariant Heat Kernel Signatures LBSD (publicly available at <http://www.centrale-ponts.fr/personnel/tsogkas/code.html>) implements the learning-based approach to symmetry detection published in [32]. It includes the code for running a detector, alongside with the ground-truth symmetry annotations that we have introduced for the Berkeley Segmentation Dataset (BSD) benchmark.

5.8. Texture Analysis Using Modulation Features and Generative Models

Participant: Iasonas Kokkinos [Correspondant].

Texture, modulation, generative models, segmentation, TEXMEG is a front-end for texture analysis and edge detection platform in Matlab that relies on Gabor filtering and image demodulation (publicly available at <http://cvsp.cs.ntua.gr/software/texture/>). Includes frequency- and time- based definition of Gabor- and other Quadrature-pair filterbanks, demodulation with the Regularized Energy Separation Algorithm and Texture/Edge/Smooth classification based on MDL criterion. The platform is used from approximately 250 users worldwide.

5.9. Sparse Prediction

Participant: Andreas Argyriou [Correspondant].

Sparse prediction, K-support norm, SPARSE_K is a sparse prediction code (publicly available at http://www.centrale-ponts.fr/personnel/andreas/code/sparse_k/sparse_k.tar) using regularization with the k -support norm, which we have introduced [36]. The algorithm uses an accelerated first-order method similar to Nesterov's method.

GALLIUM Project-Team

5. Software

5.1. OCaml

Participants: Damien Doligez [correspondant], Xavier Clerc [team SED], Alain Frisch [LexiFi], Jacques Garrigue [Nagoya University], Thomas Gazagnaire [OCamlPro], Fabrice Le Fessant [Inria Saclay and OCaml-Pro], Xavier Leroy, Luc Maranget [EPI Moscova].

OCaml, formerly known as Objective Caml, is our flagship implementation of the Caml language. From a language standpoint, it extends the core Caml language with a fully-fledged object and class layer, as well as a powerful module system, all joined together by a sound, polymorphic type system featuring type inference. The OCaml system is an industrial-strength implementation of this language, featuring a high-performance native-code compiler for several processor architectures (IA32, AMD64, PowerPC, ARM, etc) as well as a bytecode compiler and interactive loop for quick development and portability. The OCaml distribution includes a standard library and a number of programming tools: replay debugger, lexer and parser generators, documentation generator, compilation manager, and the Camlp4 source pre-processor.

Web site: <http://caml.inria.fr/>

5.2. CompCert C

Participants: Xavier Leroy [correspondant], Sandrine Blazy [EPI Celtique], Jacques-Henri Jourdan, Valentin Robert.

The CompCert C verified compiler is a compiler for a large subset of the C programming language that generates code for the PowerPC, ARM and x86 processors. The distinguishing feature of CompCert is that it has been formally verified using the Coq proof assistant: the generated assembly code is formally guaranteed to behave as prescribed by the semantics of the source C code. The subset of C supported is quite large, including all C types except `long long` and `long double`, all C operators, almost all control structures (the only exception is unstructured `switch`), and the full power of functions (including function pointers and recursive functions but not variadic functions). The generated PowerPC code runs 2–3 times faster than that generated by GCC without optimizations, and only 7% (resp. 12%) slower than GCC at optimization level 1 (resp. 2).

Web site: <http://compcert.inria.fr/>

5.3. Zenon

Participant: Damien Doligez.

Zenon is an automatic theorem prover based on the tableaux method. Given a first-order statement as input, it outputs a fully formal proof in the form of a Coq proof script. It has special rules for efficient handling of equality and arbitrary transitive relations. Although still in the prototype stage, it already gives satisfying results on standard automatic-proving benchmarks.

Zenon is designed to be easy to interface with front-end tools (for example integration in an interactive proof assistant), and also to be easily retargeted to output scripts for different frameworks (for example, Isabelle).

Web site: <http://zenon-prover.org/>

GAMMA3 Project-Team (section vide)

GANG Project-Team (section vide)

GECO Team

5. Software

5.1. IRHD

We developed a first version of a software for reconstruction of corrupted and damaged images, named IRHD (for Image Reconstruction via Hypoelliptic Diffusion). One of the main features of the algorithm on which the software is based is that we don't use any information about the location and character of the corrupted places; this allows us to work with real non-academic images. Another important advantage is that this method is massively parallelizable; this allows to work with sufficiently large images. Theoretical background of the presented method is based on the model of geometry of vision due to Petitot, Citti and Sarti. The main step is numerical solution of the equation of 3D hypoelliptic diffusion. IRHD is based on Fortran. Alexey Remizov is in charge with the development of the software, in collaboration with Ugo Boscain and Jean-Paul Gauthier.

GENSCALE Team

5. Software

5.1. Next Generation Sequencing

Participants: Alexan Andrieux, Rayan Chikhi, Dominique Lavenier, Claire Lemaitre, Nicolas Maillet, Pierre Peterlongo, Raluca Uricaru.

- **Genome assembly** [contact: P. Peterlongo]
 - **Minia : ultra low memory assembly** Minia is a short-read assembler based on a de Bruijn graph, capable of assembling a human genome on a desktop computer in a day. The output of Minia is a set of contigs. Minia produces results of similar contiguity and accuracy to other de Bruijn assemblers (e.g. Velvet). <http://minia.genouest.org/>
 - **Mapsembler: targeted assembly software.** Mapsembler is a targeted assembly software. From sets of NGS raw reads and a set of input sequences (starters), it determines if each starter could be constructed from the reads. Then for each "read-coherent" starter, Mapsembler outputs its sequence neighborhood as a linear sequence or as a graph, depending on the user choice. <http://alcovna.genouest.org/mapsembler/>
- **Variant detection** [contact: C. Lemaitre]
 - **kisSnp and kisSplice : variant identification without the use of a reference genome.** kisSnp is a tool to find single nucleotide polymorphisms (SNP) by comparing two sets of raw NGS reads. <http://alcovna.genouest.org/kissnp/> KisSplice finds alternative splicings but also short insertions, deletions and duplications, SNPs and sequencing errors in one or two RNA-seq sets, without assembly nor mapping on a reference genome. <http://alcovna.genouest.org/kissplice/>
 - **Kissreads: quantification of variants** Kissreads considers sets of NGS raw reads and a set of input sequences (starters). Mapping reads to each starter, it provides quantitative (coverage depth) and qualitative (mapped read quality) information about each starter.
- **Read mapping** [contact: D. Lavenier]
 - **GASSST: short reads mapper** The GASSST software (Global Alignment Short Sequence Search Tool) is a general purpose mapper. GASSST finds global alignments of short DNA sequences against large DNA banks. One main characteristic of GASSST is its ability to perform fast gapped alignments and to process long reads compared to other current similar tools. <http://www.irisa.fr/symbiose/projects/gassst/>

5.2. High throughput sequence analysis

Participants: Rayan Chikhi, Erwan Drezén, Dominique Lavenier, Claire Lemaitre, Nicolas Maillet, Pierre Peterlongo.

- **PLAST : efficient bank-to-bank alignments** PLAST (Parallel Local Alignment Search Tool) is a parallel alignment search tool for comparing large protein banks. PLAST runs 3 to 5 times faster than the NCBI-BLAST software. An improved version is commercialized by the Korilog Company, including the DNA bank-to-bank option. [contact: D. Lavenier] <http://www.irisa.fr/symbiose/projects/plast/>
- **Compareads : efficient comparison of large metagenomics NGS datasets** This software extracts similar DNA sequences (reads) between two metagenomic datasets. It requires a small and fixed amount of memory and can thus be used on huge datasets. [contact: P. Peterlongo] <http://alcovna.genouest.org/compareads/>

5.3. 3D Protein structures

Participants: Rumen Andonov, Guillaume Chapuis, Mathilde Le Boudic-Jamin, Antonio Mucherino.

- **CSA and DALIX** CSA (Comparative Structural Alignment) is a webserver for computing and comparing protein structure alignments. CSA is able to compute score-optimal alignments with respect to various inter-residue distance-based scoring schemes. [contact: R. Andonov] <http://csa.project.cwi.nl/>
- **A_purva** A_purva is a Contact Map Overlap maximization (CMO) solver. Given two protein structures represented by two contact maps, A_purva computes the amino-acid alignment which maximize the number of common contacts. [contact: R. Andonov] http://mobylye.genouest.org/cgi-bin/Mobylye/portal.py?forms::A_Purva
- **MD-Jeep** MD-jeep is a software tool for solving distance geometry problems. It is able to solve a subclass of instances of the problem for which a discrete reformulation can be supplied. We refer to this subclass of instances as the Discretizable Molecular Distance Geometry Problem (DMDGP). We employ a Branch & Prune (BP) algorithm for the solution of DMDGPs. [contact: A. Mucherino] <http://www.antoniomucherino.it/en/mdjeep.php>

5.4. HPC and Parallelism

Participants: Guillaume Chapuis, Dominique Lavenier, François Moreews.

- **QTLmap** QTLMap is a tool dedicated to the detection of Quantitative Trait Loci (QTL) from experimental designs in outbred population. QTLMap was recently ported to GPU and offers reduced run times. [contact: D. Lavenier] <http://www.inra.fr/qtlmap/>
- **SLICEE** (Service Layer for Intensive Computation Execution Environment) is part of the BioWIC project. This software proposes (1) to abstract the calls to the cluster scheduler by handling command submission; (2) to take care of exploiting the data parallelism with data specific methods; (3) to manage data using a cache references mechanism and route data between tasks. [contact: F. Moreews] <http://vapor.gforge.inria.fr/>

GEOMETRICA Project-Team

5. Software

5.1. CGAL, the Computational Geometry Algorithms Library

Participants: Pierre Alliez, Jean-Daniel Boissonnat, Olivier Devillers, Monique Teillaud, Mariette Yvinec.

With the collaboration of Hervé Brönnimann, Manuel Caroli, Pedro Machado Manhães de Castro, Frédéric Cazals, Frank Da, Christophe Delage, Andreas Fabri, Julia Flötotto, Philippe Guigue, Michael Hemmer, Samuel Hornus, Menelaos Karavelas, Sébastien Lorient, Abdelkrim Mebarki, Naceur Meskini, Andreas Meyer, Sylvain Pion, Marc Pouget, François Rebufat, Laurent Rineau, Laurent Saboret, Stéphane Tayeb, Jane Tournois, Radu Ursu, and Camille Wormser. <http://www.cgal.org>

CGAL is a C++ library of geometric algorithms and data structures. Its development has been initially funded and further supported by several European projects (CGAL, GALIA, ECG, ACS, AIM@SHAPE) since 1996. The long term partners of the project are research teams from the following institutes: Inria Sophia Antipolis - Méditerranée, Max-Planck Institut Saarbrücken, ETH Zürich, Tel Aviv University, together with several others. In 2003, CGAL became an Open Source project (under the LGPL and QPL licenses), and it also became commercialized by GEOMETRY FACTORY, a company *Born of Inria* founded by Andreas Fabri.

The aim of the CGAL project is to create a platform for geometric computing supporting usage in both industry and academia. The main design goals are genericity, numerical robustness, efficiency and ease of use. These goals are enforced by a review of all submissions managed by an editorial board. As the focus is on fundamental geometric algorithms and data structures, the target application domains are numerous: from geological modeling to medical images, from antenna placement to geographic information systems, etc.

The CGAL library consists of a kernel, a list of algorithmic packages, and a support library. The kernel is made of classes that represent elementary geometric objects (points, vectors, lines, segments, planes, simplices, isothetic boxes, circles, spheres, circular arcs...), as well as affine transformations and a number of predicates and geometric constructions over these objects. These classes exist in dimensions 2 and 3 (static dimension) and d (dynamic dimension). Using the template mechanism, each class can be instantiated following several representation modes: one can choose between Cartesian or homogeneous coordinates, use different types to store the coordinates, and use reference counting or not. The kernel also provides some robustness features using some specifically-devised arithmetic (interval arithmetic, multi-precision arithmetic, static filters...).

A number of packages provide geometric data structures as well as algorithms. The data structures are polygons, polyhedra, triangulations, planar maps, arrangements and various search structures (segment trees, d -dimensional trees...). Algorithms are provided to compute convex hulls, Voronoi diagrams, Boolean operations on polygons, solve certain optimization problems (linear, quadratic, generalized of linear type). Through class and function templates, these algorithms can be used either with the kernel objects or with user-defined geometric classes provided they match a documented interface.

Finally, the support library provides random generators, and interfacing code with other libraries, tools, or file formats (ASCII files, QT or LEDA Windows, OpenGL, Open Inventor, Postscript, Geomview...). Partial interfaces with Python, SCILAB and the Ipe drawing editor are now also available.

GEOMETRICA is particularly involved in general maintenance, in the arithmetic issues that arise in the treatment of robustness issues, in the kernel, in triangulation packages and their close applications such as alpha shapes, in meshes... Three researchers of GEOMETRICA are members of the CGAL Editorial Board, whose main responsibilities are the control of the quality of CGAL, making decisions about technical matters, coordinating communication and promotion of CGAL.

CGAL is about 700,000 lines of code and supports various platforms: GCC (Linux, Mac OS X, Cygwin...), Visual C++ (Windows), Intel C++... A new version of CGAL is released twice a year, and it is downloaded about 10000 times a year. Moreover, CGAL is directly available as packages for the Debian, Ubuntu and Fedora Linux distributions.

More numbers about CGAL: there are now 14 editors in the editorial board, with approximately 20 additional developers. The user discussion mailing-list has more than 1000 subscribers with a relatively high traffic of 5-10 mails a day. The announcement mailing-list has more than 3000 subscribers.

GEOSTAT Project-Team

5. Software

5.1. FluidExponents

Participants: Denis Arrivault [correspondant], Hussein Yahia, Joel Sudre.

Denis Arrivault has joined the team for a complete refoundation, rewriting, generalization and diffusion of the FluidExponents software. FluidExponents is a software implementation of the MMF, presently written in Java, in a cooperative development mode on the Inria GForge, deposited at APP in 2010. The new software is presently in the phase of specification, and will be rewritten in C++, using existing libraries for data containers, mathematical computation and user interface. Denis Arrivault is recruited for a 24 month period on FluidExponents ADT.

During the new development, researchers still make use of the current version of the FluidExponents software written in Java, version number 0.8. Contact: denis.arrivault@inria.fr.

GRACE Team

4. Software

4.1. ECPP

F. Morain has been continually improving his primality proving algorithm called ECPP, originally developed in the early 1990s. Binaries for version 6.4.5 have been available since 2001 on his web page. Proving the primality of a 512 bit number requires less than a second on an average PC. His personal record is around 25,000 decimal digits, using the fast version that he started developing in 2003. All of the code is written in C, and based on publicly available packages (GMP, mpfr, mpc, mpfrx).

4.2. SEA

Together with E. Schost and L. DeFeo, F. Morain has developed a new implementation of the SEA algorithm that computes the cardinality of elliptic curves over finite fields (large prime case, case $p = 2$). It uses NTL and includes the most recent algorithms for solving all subtasks. The large prime case is relevant to cryptographic needs. The $p = 2$ case, though not directly useful, is a good testbed for the FFAST program of Luca De Feo. This program forms a gforge project.

4.3. TIFA

The TIFA library (short for Tools for Integer FActorization), initially developed in 2006, has been continuously improved during the last few years. TIFA is made up of a base library written in C99 using the GMP library, together with stand-alone factorization programs and a basic benchmarking framework to assess the performance of each algorithm.

It is now available online at <http://www.lix.polytechnique.fr/Labo/Jerome.Milan/tifa/tifa.xhtml>; it is distributed under the Lesser General Public License (version 2.1 or later).

4.4. Quintix

The Quintix library is a Mathemagix package, available at <http://www.mathemagix.org/www/main/index.en.html>. Quintix is a very efficient library for Galois rings, extensions of Galois rings and root-finding in Galois rings, developed in C++, within the Mathemagix computer algebra system. It implements basic arithmetic for Galois rings and their unramified extensions, basic functions for the manipulation of Reed–Solomon codes, and the complete Sudan list-decoding algorithm. It also implements the root-finding algorithms presented in [23]. The source code is distributed under the General Public License (version 2 or higher).

4.5. finitefieldz

G. Quintin wrote the finitefieldz package which provides arithmetic for finite fields (of any characteristic) and towers of finite fields. He wrote this package with the help of Grégoire Lecerf during the first year of his PhD thesis. The package uses univariate polynomials and multiprecision integers, and also provides univariate polynomial root finding and factorization over finite fields.

4.6. Decoding

Decoding is a standalone C library licensed under the GPLv2. Its primary goal is to implement Guruswami–Sudan list decoding-related algorithms, as efficiently as possible. Its secondary goal is to give an efficient tool for the implementation of decoding algorithms (not necessarily list decoding algorithms) and their benchmarking.

For now (2012/12/13) you can use the library and have a working list decoding algorithm, but there is no unique decoding algorithm (though you can tell decoding to list decode up to half the minimum distance). The library is being further developed and more algorithms will be added.

The library was presented at the 2012 International Symposium on Symbolic and Algebraic Computation.

GRAND-LARGE Project-Team

4. Software

4.1. APMC-CA

Participants: Sylvain Peyronnet [correspondant], Joel Falcou, Pierre Esterie, Khaled Hamidouche, Alexandre Borghi.

The APMC model checker implements the state-of-the-art approximate probabilistic model checking methods. Last year we develop a version of the tool dedicated to the CELL architecture. Clearly, it was very pedagogic, but the conclusion is that the CELL is not adapted to sampling based verification methods.

This year we develop, thanks to the BSP++ framework, a version compatible with SPM/multicores machines, clusters and hybrid architectures. This version outperforms all previous ones, thus showing the interest of both these new architectures and of the BSP++ framework.

4.2. YML

Participants: Serge Petiton [correspondant], Nahid Emad, Maxime Hugues.

Scientific end-users face difficulties to program P2P large scale applications using low level languages and middleware. We provide a high level language and a set of tools designed to develop and execute large coarse grain applications on peer-to-peer systems. Thus, we introduced, developed and experimented the YML for parallel programming on P2P architectures. This work was done in collaboration with the PRiSM laboratory (team of Nahid Emad).

The main contribution of YML is its high level language for scientific end-users to develop parallel programs for P2P platforms. This language integrates two different aspects. The first aspect is a component description language. The second aspect allows to link components together. A coordination language called YvetteML can express graphs of components which represent applications for peer-to-peer systems.

Moreover, we designed a framework to take advantage of the YML language. It is based on two component catalogues and an YML engine. The first one concerns end-user's components and the second one is related to middleware criteria. This separation enhances portability of applications and permits real time optimizations. Currently we provide support for the XtremWeb Peer-to-Peer middleware and the OmniRPC grid system. The support for Condor is currently under development and a beta-release will be delivered soon (in this release, we plan to propagate semantic data from the end-users to the middleware). The next development of YML concerns the implementation of a multi-backend scheduler. Therefore, YML will be able to schedule at runtime computing tasks to any global computing platform using any of the targeted middleware.

We experimented YML with basic linear algebra methods on a XtremWeb P2P platform deployed between France and Japan. Recently, we have implemented complex iterative restarted Krylov methods, such as Lanczos-Bisection, GMRES and MERAM methods, using YML with the OmniRPC back-end. The experiments are performed either on the Grid5000 testbed or on a Network of Workstations deployed between Lille, Versailles and Tsukuba in Japan. Demos was proposed on these testbeds from conferences in USA. We recently finished evaluations of the overhead generated using YML, without smart schedulers and with extrapolations due to the lack of smart scheduling strategies inside targeted middleware.

In the context of the FP3C project funded by ANR-JST, we have recently extended YML to support a directive distributed parallel language, XcalableMP <http://www.xcalablemp.org/>. This extension is based on the support of the XcalableMP language inside YML components. This allows to develop parallel programs with two programming paradigm and thus two parallelism levels. This work is a part of the project that targets post-Petascale supercomputer that would be composed of heterogeneous and massively parallel hardware.

The software is available at <http://yml.prism.uvsq.fr/>

4.3. The Scientific Programming InterNet (SPIN)

Participant: Serge Petiton [correspondant].

SPIN (Scientific Programming on the InterNet), is a scalable, integrated and interactive set of tools for scientific computations on distributed and heterogeneous environments. These tools create a collaborative environment allowing the access to remote resources.

The goal of SPIN is to provide the following advantages: Platform independence, Flexible parameterization, Incremental capacity growth, Portability and interoperability, and Web integration. The need to develop a tool such as SPIN was recognized by the GRID community of the researchers in scientific domains, such as linear algebra. Since the P2P arrives as a new programming paradigm, the end-users need to have such tools. It becomes a real need for the scientific community to make possible the development of scientific applications assembling basic components hiding the architecture and the middleware. Another use of SPIN consists in allowing to build an application from predefined components ("building blocks") existing in the system or developed by the developer. The SPIN users community can collaborate in order to make more and more predefined components available to be shared via the Internet in order to develop new more specialized components or new applications combining existing and new components thanks to the SPIN user interface.

SPIN was launched at ASCI CNRS lab in 1998 and is now developed in collaboration with the University of Versailles, PRiSM lab. SPIN is currently under adaptation to incorporate YML, cf. above. Nevertheless, we study another solution based on the Linear Algebra KErnel (LAKE), developed by the Nahid Emad team at the University of Versailles, which would be an alternative to SPIN as a component oriented integration with YML.

4.4. V-DS

Participant: Franck Cappello [correspondant].

This project started officially in September 2004, under the name V-Grid. V-DS stands for Virtualization environment for large-scale Distributed Systems. It is a virtualization software for large scale distributed system emulation. This software allows folding a distributed systems 100 or 1000 times larger than the experimental testbed. V-DS virtualizes distributed systems nodes on PC clusters, providing every virtual node its proper and confined operating system and execution environment. Thus compared to large scale distributed system simulators or emulators (like MicroGrid), V-DS virtualizes and schedules a full software environment for every distributed system node. V-DS research concerns emulation realism and performance.

A first work concerns the definition and implementation of metrics and methodologies to compare the merits of distributed system virtualization tools. Since there is no previous work in this domain, it is important to define what and how to measure in order to qualify a virtualization system relatively to realism and performance. We defined a set of metrics and methodologies in order to evaluate and compared virtualization tools for sequential system. For example a key parameter for the realism is the event timing: in the emulated environment, events should occur with a time consistent with a real environment. An example of key parameter for the performance is the linearity. The performance degradation for every virtual machine should evolve linearly with the increase of the number of virtual machines. We conducted a large set of experiments, comparing several virtualization tools including Vserver, VMware, User Mode Linux, Xen, etc. The result demonstrates that none of them provides both enough isolation and performance. As a consequence, we are currently studying approaches to cope with these limits.

We have made a virtual platform on the GDX cluster with the Vserver virtualization tool. On this platform, we have launched more than 20K virtual machines (VM) with a folding of 100 (100 VM on each physical machine). However, some recent experiments have shown that a too high folding factor may cause a too long execution time because of some problems like swapping. Currently, we are conducting experiments on another platform based on the virtualization tool named Xen which has been strongly improved since 2 years. We expect to get better result with Xen than with Vserver. Recently, we have been using the V-DS version based on Xen to evaluate at large scales three P2P middleware [83].

This software is available at <http://v-ds.lri.fr/>

4.5. PVC: Private Virtual Cluster

Participant: Franck Cappello [correspondant].

Current complexity of Grid technologies, the lack of security of Peer-to-Peer systems and the rigidity of VPN technologies make sharing resources belonging to different institutions still technically difficult.

We propose a new approach called "Instant Grid" (IG), which combines various Grid, P2P and VPN approaches, allowing simple deployment of applications over different administration domains. Three main requirements should be fulfilled to make Instant Grids realistic: simple networking configuration (Firewall and NAT), no degradation of resource security, no need to re-implement existing distributed applications.

Private Virtual Cluster, is a low-level middle-ware that meets Instant Grid requirements. PVC turns dynamically a set of resources belonging to different administration domains into a virtual cluster where existing cluster runtime environments and applications can be run. The major objective of PVC is to establish direct connections between distributed peers. To connect firewall protected nodes in the current implementation, we have integrated three techniques: UPnP, TCP/UDP Hole Punching and a novel technique Traversing-TCP.

One of the major application of PVC is the third generation desktop Grid middleware. Unlike BOINC and XtremWeb (which belong to the second generation of desktop Grid middleware), PVC allows the users to build their Desktop Grid environment and run their favorite batch scheduler, distributed file system, resource monitoring and parallel programming library and runtime software. PVC ensures the connectivity layer and provide a virtual IP network where the user can install and run existing cluster software.

By offering only the connectivity layer, PVC allows to deploy P2P systems with specific applications, like file sharing, distributed computing, distributed storage and archive, video broadcasting, etc.

4.6. OpenWP

Participant: Franck Cappello [correspondant].

Distributed applications can be programmed on the Grid using workflow languages, object oriented approaches (Proactive, IBIS, etc), RPC programming environments (Grid-RPC, DIET), component based environments (generally based on Corba) and parallel programming libraries like MPI.

For high performance computing applications, most of the existing codes are programmed in C, Fortran and Java. These codes have 100,000 to millions of lines. Programmers are not inclined to rewrite them in a "non standard" programming language, like UPC, CoArray Fortran or Global Array. Thus environments like MPI and OpenMPI remain popular even if they require hybrid approaches for programming hierarchical computing infrastructures like cluster of multi-processors equipped with multi-core processors.

Programming applications on the Grid add a novel level in the hierarchy by clustering the cluster of multi-processors. The programmer will face strong difficulties in adapting or programming a new application for these runtime infrastructures featuring a deep hierarchy. Directive based parallel and distributed computing is appealing to reduce the programming difficulty by allowing incremental parallelization and distribution. The programmer add directives on a sequential or parallel code and may check for every inserted directive its correction and performance improvement.

We believe that directive based parallel and distributed computing may play a significant role in the next years for programming High performance parallel computers and Grids. We have started the development of OpenWP. OpenWP is a directive based programming environment and runtime allowing expressing workflows to be executed on Grids. OpenWP is compliant with OpenMP and can be used in conjunction with OpenMP or hybrid parallel programs using MPI + OpenMP.

The OpenWP environment consists in a source to source compiler and a runtime. The OpenWP parser, interprets the user directives and extracts functional blocks from the code. These blocks are inserted in a library distributed on all computing nodes. In the original program, the functional blocks are replaced by RPC calls and calls to synchronization. During the execution, the main program launches non blocking RPC calls to functions on remote nodes and synchronize the execution of remote functions based on the synchronization directives inserted by the programmer in the main code. Compared to OpenMP, OpenWP does not consider a shared memory programming approach. Instead, the source to source compiler insert data movements calls in the main code. Since the data set can be large in Grid application, the OpenWP runtime organize the storage of data sets in a distributed way. Moreover, the parameters and results of RPC calls are passed by reference, using a DHT. Thus, during the execution, parameter and result references are stored in the DHT along with the current position of the datasets. When a remote function is called, the DHT is consulted to obtain the position of the parameter data sets in the system. When a remote function terminates its execution, it stores the result data sets and store a reference to the data set in the DHT.

We are evaluating OpenWP from an industrial application (Amibe), used by the European aerospace company EADS. Amibe is the mesher module of jCAE ¹. Amibe generates a mesh from a CAD geometry in three steps. It first creates edges between every patch of the CAD (mesh in one dimension), then generates a surface mesh for every unfolded patch (mesh in two dimensions) and finally adds the third dimension to the mesh by projecting the 2D mesh into the original CAD surfaces. The first and third operation cannot be distributed. However the second step can easily be distributed following a master/worker approach, transferring the meshId results to every computing node and launching the distributed execution of the patches.

4.7. Parallel solvers for solving linear systems of equations

Participant: Laura Grigori.

In the last several years, there has been significant research effort in the development of fully parallel direct solvers for computing the solution of large unsymmetric sparse linear systems of equations. In this context, we have designed and implemented a parallel symbolic factorization algorithm, which is suitable for general sparse unsymmetric matrices. The symbolic factorization is one of the steps that is sequential and represents a memory bottleneck. The code is intended to be used with very large matrices when because of the memory usage, the sequential algorithm is not suitable. This code is available in the SuperLU_DIST, a widely used software, developed at UC Berkeley and LBNL by Professor James W. Demmel and Dr. Xiaoye S. Li. The algorithm is presented in [72]. The SuperLU_DIST is available at <http://crd.lbl.gov/~xiaoye/SuperLU/>.

4.8. OpenScop

Participant: Cédric Bastoul.

OpenScop is an open specification which defines a file format and a set of data structures to represent a *static control part* (SCoP for short), i.e., a program part that can be represented in the *polyhedral model*, an algebraic representation of programs used for automatic parallelization and optimization (used, e.g., in GNU GCC, LLVM, IBM XL or Reservoir Labs R-Stream compilers). The goal of OpenScop is to provide a common interface to various polyhedral compilation tools in order to simplify their interaction.

OpenScop provides a single format for tools that may have different purposes (e.g., as different as code generation and data dependence analysis). We could observe that most available polyhedral compilation tools during the last decade were manipulating the same kind of data (polyhedra, affine functions...) and were actually sharing a part of their input (e.g., iteration domains and context concepts are nearly everywhere). We could also observe that those tools may rely on different internal representations, mostly based on one of the major polyhedral libraries (e.g., Polylib, PPL or isl), and this representation may change over time (e.g., when switching to a more convenient polyhedral library). OpenScop aims at providing a stable, unified format that offers a durable guarantee that a tool can use an output or provide an input to another tool without breaking a compilation chain because of some internal changes in one element of this chain. The other promise of

¹project page: <http://jcae.sourceforge.net>

OpenScop is the ability to assemble or replace the basic blocks of a polyhedral compilation framework at no, or at least low engineering cost. The OpenScop Library (licensed under the 3-clause BSD license) has been developed as an example, yet powerful, implementation of the OpenScop specification.

4.9. Clay

Participant: Cédric Bastoul.

Clay is a free software and library devoted to semi-automatic optimization using the polyhedral model. It can input a high-level program or its polyhedral representation and transform it according to a transformation script. Classic loop transformations primitives are provided. Clay is able to check for the legality of the complete sequence of transformation and to suggest corrections to the user if the original semantics is not preserved (experimental at this document redaction time). Main authors include Joël Poudroux and Cédric Bastoul.

4.10. CALU for multicore architectures

Participant: Laura GRIGORI [correspondant].

The communication avoiding algorithms are implemented in the form of a portable library. In its current form, this library is designed for multicore architectures and uses a hybrid scheduling technique that exploits well the data locality and can adapt to dynamic changes in the machine. The library will be publicly available since February 2012.

See also the web page <http://www-rocq.inria.fr/who/Laura.Grigori/COALA2010/coala.html>.

- Version: 1.0

4.11. MIDAPACK for CMB data analysis

Participants: Laura GRIGORI [correspondant], Mikolaj SZYDLARSKI [correspondant].

Midapack is a library aiming at the crucial stages down the CMB data analysis pipeline. In its current form, the library provides tools for computing spherical harmonic transforms on heterogeneous architectures, and algorithms for finding the solution to a generalized least squares problem. The algorithms are described in [36] and [44]. See also the web page <http://pages.saclay.inria.fr/laura.grigori/soft.html>.

- Version: 1.0

4.12. Fast linear system solvers in public domain libraries

Participant: Marc Baboulin [correspondant].

Hybrid multicore+GPU architectures are becoming commonly used systems in high performance computing simulations. In this research, we develop linear algebra solvers where we split the computation over multicore and graphics processors, and use particular techniques to reduce the amount of pivoting and communication between the hybrid components. This results in efficient algorithms that take advantage of each computational unit [14]. Our research in randomized algorithms yields to several contributions to propose public domain libraries PLASMA and MAGMA in the area of fast linear system solvers for general and symmetric indefinite systems. These solvers minimize communication by removing the overhead due to pivoting in LU and $LDLT$ factorization. Different approaches to reduce communication are compared in [26].

See also the web page <http://icl.cs.utk.edu/magma/>.

4.13. cTuning: Repository and Tools for Collective Characterization and Optimization of Computing Systems

Participant: Grigori Fursin [correspondant].

Designing, porting and optimizing applications for rapidly evolving computing systems is often complex, ad-hoc, repetitive, costly and error prone process due to an enormous number of available design and optimization choices combined with the complex interactions between all components. We attempt to solve this fundamental problem based on collective participation of users combined with empirical tuning and machine learning.

We developed cTuning framework that allows to continuously collect various knowledge about application characterization and optimization in the public repository at cTuning.org. With continuously increasing and systematized knowledge about behavior of computer systems, users should be able to obtain scientifically motivated advices about anomalies in the behavior of their applications and possible solutions to effectively balance performance and power consumption or other important characteristics.

Currently, we use cTuning repository to analyze and learn profitable optimizations for various programs, datasets and architectures using machine learning enabled compiler (MILEPOST GCC). Using collected knowledge, we can quickly suggest better optimizations for a previously unseen programs based on their semantic or dynamic features [6].

We believe that such approach will be vital for developing efficient Exascale computing systems. We are currently developing the new extensible cTuning2 framework for automatic performance and power tuning of HPC applications.

For more information, see the web page <http://cTuning.org>.

GRAPHIK Project-Team

5. Software

5.1. Cogui

Participants: Alain Gutierrez, Michel Leclère, Marie-Laure Mugnier, Michel Chein, Madalina Croitoru.

Cogui (<http://www.lirmm.fr/cogui>) is a tool for building and verifying knowledge bases. It is a freeware written in Java (version 1.2, 2005–2010 GPL Licence). Currently, it supports Conceptual Graphs and import/export in RDFS.

Here are the major evolutions of the version delivered this year:

- *Cogui* now allows import/export in the Datalog+/- language thanks to a new Datalog+/- parser (see Sect. 5.4).
- Scripted rules were introduced. It is a new type of object that allows users to attach a script to a traditional rule in order to modify or control its behavior.
- A new interface ensures connectivity to a NoSQLdatabase (MongoDB).
- Large graphs can now be stored. In a near future, we will be able to perform queries on data too big to fit in central memory (see Sect. 5.2).

5.2. Alaska

Participants: Bruno Paiva Lima Da Silva, Jean-François Baget, Madalina Croitoru.

Alaska (<http://alaska.bplsilva.com/>) is a java library dedicated to the storage and querying of large knowledge bases. It intends to be the foundation layer of our OBDA (Ontology Based Data Access) software developments. It has been built, first as part of a Master's thesis, and now of the PhD of Bruno Paiva Lima da Silva [34].

In *Alaska*, facts and queries are defined via a generic interface that favors a logical view of these objects. Implementations of this interface allow for the storage of facts w.r.t. different storage paradigms and systems (e.g., relational databases *MySQL* and *Sqlite*; triple stores *Sesame* and graph databases *Neo4J*, *DEX*, *HyperGraphDB* and *OrientDB*). For the time being, we can store 10^7 to 10^8 atoms. In the same way, logical queries can be evaluated through different methods, be it the native querying mechanism of the database used (e.g. *SPARQL* or *SQL*), or specifically designed algorithms (from a simple backtrack to a full constraint solver based upon *Choco* for hard problem instances). Note that all these methods provide the same answer set to queries.

This library already allows for testing our OBDA algorithms on large instances (it is already used by other PhD students for their experiments), and will soon be ready to be distributed to a broader audience. Our generic approach will ease this dissemination to different research domains.

5.3. Kiabora

Participants: Swan Rocher [first year master internship], Michel Leclère, Marie-Laure Mugnier.

<http://www2.lirmm.fr/~mugnier/graphik/kiabora/index.html>

Kiabora is a tool dedicated to the analysis of a set of existential rules. It can check if this set belongs to a known *decidable* class of rules, either directly or by means of its *Graph of Rule Dependencies (GRD)*. *Kiabora* analyzes the properties of the strongly connected components in the GRD, which allows to determine properties of the rule set with respect to decidability as well as the kind of paradigm (forward or backward chaining) ensuring decidability.

Besides, *Kiabora* also provides format conversion and rule decomposition services. It is written in Java.

5.4. DLGP

Participants: Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, Alain Gutierrez, Swan Rocher [first year master internship], Clément Sipieter [first year master internship].

http://www2.lirmm.fr/~mugnier/graphik/kiabora/downloads/datalog-plus_en.pdf

DLGP (for Datalog Plus) is a textual exchange format at once human-friendly, concise and easy to parse. This format can be seen as an extension of the commonly used format for plain Datalog. A file may contain four kinds of knowledge elements: facts, existential rules, negative constraints and conjunctive queries. This format will allow us to easily exchange data and ontologies with groups working on the equivalent Datalog+/- formalism, developed in Oxford.

A DLGP parser is now available.

For this section, participants are listed in alphabetical order.

HIEPACS Project-Team

5. Software

5.1. Introduction

We describe in this section the software that we are developing. The first two (MaPHyS and ScalFMM) will be the main milestones of our project. The other software developments will be conducted in collaboration with academic partners or in collaboration with some industrial partners in the context of their private R&D or production activities. For all these software developments, we will use first the various (very) large parallel platforms available through CERFACS and GENCI in France (CCRT, CINES and IDRIS Computational Centers), and next the high-end parallel platforms that will be available via European and US initiatives or projects such that PRACE.

5.2. MaPHyS

MaPHyS (Massively Parallel Hybrid Solver) is a software package whose proptotype was initially developed in the framework of the PhD thesis of Azzam Haidar (CERFACS) and futher consolidated thanks to the ANR-CIS Solstice funding. This parallel linear solver couples direct and iterative approaches. The underlying idea is to apply to general unstructured linear systems domain decomposition ideas developed for the solution of linear systems arising from PDEs. The interface problem, associated with the so called Schur complement system, is solved using a block preconditioner with overlap between the blocks that is referred to as Algebraic Additive Schwarz.

The MaPHyS package is very much a first outcome of the research activity described in Section 3.3 . Finally, MaPHyS is a preconditioner that can be used to speed-up the convergence of any Krylov subspace method. We forsee to either embed in MaPHyS some Krylov solvers or to release them as standalone packages, in particular for the block variants that will be some outcome of the studies discussed in Section 3.3 .

5.3. EPSN

EPSN (Environment for Computational Steering) is a software environment for the steering of legacy parallel-distributed simulations with simple GUI or more complex (possibly parallel) visualization programs (see Figure 1). In order to make a legacy simulation steerable, the user annotates the sourcecode with the EPSN API. These annotations provide the EPSN environment with two kinds of information: the description of the program structure according to a Hierarchical Task Model (HTM) and the description of the distributed data that will be remotely accessible. EPSN provides a distributed data model, that handles common scientific objects such as parameters, structured grids, particles/atoms and unstructured meshes. It is then possible to dynamically connect EPSN with a client program, that provides a GUI with some visualization & interaction features, as for instance SIMONE (SIMulation MONitoring for Epsn). Once a client is connected, it interacts with the simulation via EPSN API. It is possible : 1) to control the execution flow of the remote simulation; 2) to access/modify its data onthefly; and 3) finally to invoke advanced user-defined routines in the simulation. The current version of EPSN is fully based on CORBA for communication on heterogeneous system and VTK/Paraview for visualization. A new release of EPSN, that will be fully based on MPI to handle efficient communication, is currently under development. A prototype is already working.

EPSN has been supported by the ACI-GRID program (grant number PPL02-03), the ARC RedGRID, the ANR MASSIM (grant number ANR-05-MMSA-0008-03) and the ANR CIS NOSSI (2007). More informations are available on our web site: <http://www.labri.fr/projet/epsn>. This software is publicly available at Inria Gforge (<http://epsn.gforge.inria.fr>).

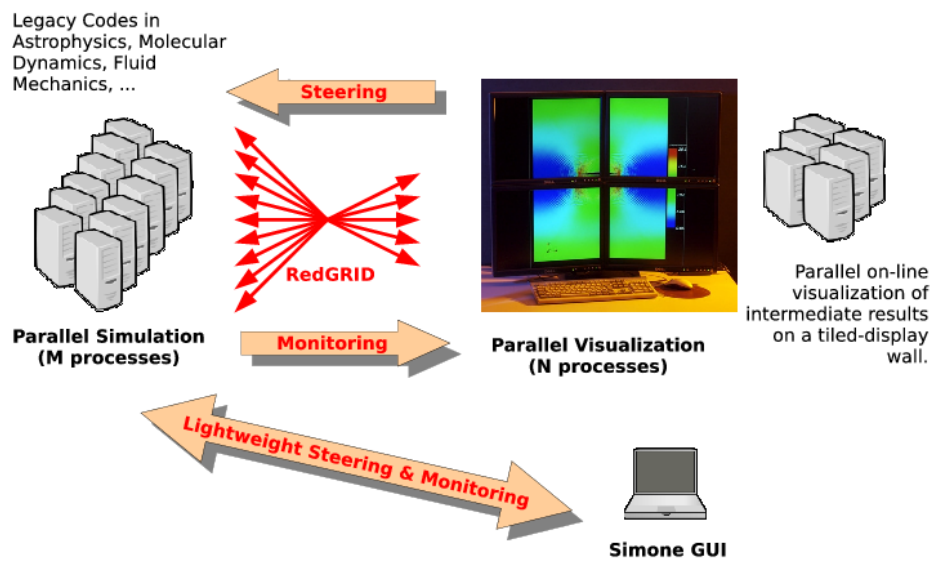


Figure 1. EPSN: software environment for $M \times N$ computational steering.

5.4. MPICPL

MPICPL (MPI CouPLing) is a software library dedicated to the coupling of parallel legacy codes, that are based on the well-known MPI standard. It proposes a lightweight and comprehensive programming interface that simplifies the coupling of several MPI codes (2, 3 or more). MPICPL facilitates the deployment of these codes thanks to the *mpicplrun* tool and it interconnects them automatically through standard MPI inter-communicators. Moreover, it generates the universe communicator, that merges the world communicators of all coupled-codes. The coupling infrastructure is described by a simple XML file, that is just loaded by the *mpicplrun* tool. Future releases will incorporate new features for checkpoint/restart and dynamic parallel code connection.

MPICPL was developed by the Inria HiePACS project-team for the purpose of the ANR CIS NOSSI. It uses advanced features of MPI2 standard. The framework is publicly available at Inria Gforge: <http://mpicpl.gforge.inria.fr>.

5.5. MONIQA

MONIQA (MONitoring graphic user Interface for QM/MM Applications) is a GUI specially designed for the monitoring & steering of the QM/MM application in the ANR CIS NOSSI project. It is based on Tulip, a graph visualization software <http://tulip.labri.fr>), used to display atoms and molecules. It proposes two working modes : offline or online. The offline mode is mainly used to load input files of DL_POLY & Siesta, and to prepare the quantum region for the QM/MM coupling. In online mode, the end-user can monitor & interact with the running QM/MM application thanks to EPSN. It is thus possible to visualize molecular and physical data (distances, angles, charges, energies), and to change simulation parameters on-the-fly, such as the target temperature of the system, thermo or barostat parameters, verbosity of output, ... MONIQA is based on QT4. It was developed specifically for the ANR NOSSI project and is available (restricted access) at Inria Gforge: <http://nossi.gforge.inria.fr>.

5.6. Sca1FMM

Sca1FMM intends to offer all the functionalities needed to perform large parallel simulations while enabling an easy customization of the simulation components: kernels, particles and cells. It works in parallel in a shared/distributed memory model using OpenMP and MPI. The software architecture has been designed with two major objectives: being easy to maintain and easy to understand. The code is extremely documented and the naming convention fully respected. Driven by its user-oriented philosophy, Sca1FMM is using CMAKE as a compiler/installer tool. Even if Sca1FMM is written in C++ it will support a C and fortran API soon.

Sca1FMM (Parallel Fast Multipole Library for Large Scale Simulations) is a software library to simulate N-body interactions using the Fast Multipole Method.

The library offers two methods to compute interactions between bodies when the potential decays like $1/r$. The first method is the classical FMM based on spherical harmonic expansions and the second is the Black-Box method which is an independent kernel formulation (introduced by E. Darve @ Stanford). With this method, we can now easily add new non oscillatory kernels in our library. For the classical method, two approaches are used to decrease the complexity of the operators. We consider either matrix formulation that allows us to use BLAS routines or rotation matrix to speed up the M2L operator.

Sca1FMM intends to offer all the functionalities needed to perform large parallel simulations while enabling an easy customization of the simulation components: kernels, particles and cells. It works in parallel in a shared/distributed memory model using OpenMP and MPI. The software architecture has been designed with two major objectives: being easy to maintain and easy to understand. There is two main parts: 1) the management of the octree and the parallelization of the method ; 2) the kernels. This new architecture allow us to easily add new FMM algorithm or kernels and new paradigm of parallelization.

The Sca1FMM package is available at scalffmm.gforge.inria.fr.

5.7. Other software

These software packages are or will be developed in collaboration with some academic partners (LIP6, LaBRI, CPMOH, IPREM, EPFL) or in collaboration with industrial partners (CEA, TOTAL, EDF) in the context of their private R&D or production activities.

- For the materials physics applications, a lot of development will be done in the context of ANR projects (NOSSI and OPTIDIS, see Section 4.2) in collaboration with LaBRI, CPMOH, IPREM, EPFL and with CEA Saclay and Bruyère-le-Châtel.

HIPERCOM Project-Team

5. Software

5.1. RPL P2P

Participants: Emmanuel Baccelli [correspondant], Oliver Hahm, Matthias Philipp.

P2P-RPL is an implementation of draft-ietf-roll-p2p-rpl, providing reactive discovery of point-to-point routes in low power and lossy networks such as wireless sensor networks. The implementation is based on the Contiki operating system. See also the web page <http://contiki-p2p-rpl.gforge.inria.fr/>.

- Version: 0.4

5.2. MPR-OSPF

Participants: Emmanuel Baccelli [correspondant], Juan-Antonio Cordero.

MPR-OSPF is an implementation of RFC5449, providing OSPF-compatible routing in hybrid networks composed of both mobile ad hoc routers and fixed wired networks. The implementation is based on Quagga/Zebra. See also the web page <http://ospfmanet.gforge.inria.fr>.

- Version: 0.4

5.3. OPERA infrastructure

Participants: Cédric Adjih [correspondant], Ichrak Amdouni, Pascale Minet, Ridha Soua.

OPERA-infrastructure is the system support code of OPERA, the Optimized Protocol for Energy efficient Routing with node Activity scheduling.

5.4. OPERA perf simul

Participants: Cédric Adjih [correspondant], Ichrak Amdouni.

OPERA-perf-simul is a set of tools for simulation and performance evaluation as well as large scale tests of OPERA, the Optimized Protocol for Energy efficient Routing with node Activity scheduling.

5.5. OPERA protocol

Participants: Cédric Adjih [correspondant], Ichrak Amdouni, Pascale Minet, Saoucene Mahfoudh Ridene.

OPERA-protocol is the heart of OPERA, the Optimized Protocol for Energy efficient Routing with node Activity scheduling. It includes EOND a neighborhood discovery protocol, EOSTC a protocol byuiding and maintaining a n energy efficient routing tree and SERENA a node coloring algorithm.

5.6. OPERA validation and tools

Participant: Cédric Adjih [correspondant].

OPERA-validation and tools is a set of tools for validation, debugging, analysis and visualization of OPERA protocol, the Optimized Protocol for Energy efficient Routing with node Activity scheduling. It operates either in a real embedded system or in simulation.

I4S Team

5. Software

5.1. COSMAD

With the help of former engineers, I4S team has developed and maintained a Scilab toolbox devoted to modal analysis and vibration monitoring of structures or machines subjected to known or ambient (unknown) excitation. This software (COSMAD 3.64) has been registered at the APP under the number

IDDN.FR.001.210011.002.S.A.2003.000.20700

A list of test-cases (simulators, laboratory test-beds, real structures) for which COSMAD has been used is available on I4S website. The problem is to identify the eigenstructure (eigenvalues and observed components of the associated eigenvectors) of the state transition matrix of a linear dynamical system, using only the observation of some measured outputs summarized into a sequence of covariance matrices corresponding to successive time shifts. Other services are

- *Output-only and Input/Ouptut subspace-based identification,*
- *Automated on-line identification package,*
- *Subspace-based identification through moving sensors data fusion,*
- *Damage detection and monitoring,*
- *Damage localization,*

The modules have been tested by different partners, especially the French industrial partners, EADS, Dassault and Sopemea, within the FLITE2 project, by partners from the past CONSTRUCTIF project, and within the framework of bilateral contracts with SNECMA and SVS.

Based on intensive internal evaluation of the toolbox, on both simulated and real data sets, EADS Space Transportation and CNES have been investigating how to use the toolbox for the exploitation of the Ariane 5 flight data sets.

This Scilabtoolbox continues to play the role of a programming and development environment for all our newly designed algorithms. Moreover, offering a *maintained* Scilab platform turns out to be a crucial factor in convincing industrial partners to undertake joint investigations with us. Just recently, SNECMA funded development for the Cosmad toolbox in 2010.

5.2. Prototypes

Three software have been deposited to the Agency of Program Protection, i.e.

- 1/ VIBRA-PARTICULAIRE : APP IDDN.FR.001.420016.000.S.P.2012.000.20700

They will be transferred to partners and industrial contracts when possible.

- + A new version of the COSMAD toolbox has been deposited at APP and concerns the transfer to SVS action.

IBIS Project-Team

4. Software

4.1. Genetic Network Analyzer (GNA)

Participants: Hidde de Jong [Correspondent], Michel Page, François Rechenmann, Delphine Ropers.

Gene regulatory networks, qualitative simulation, model checking

GENETIC NETWORK ANALYZER (GNA) is the implementation of a method for the qualitative modeling and simulation of gene regulatory networks developed in the IBIS project. The input of GNA consists of a model of the regulatory network in the form of a system of piecewise-linear differential equations, supplemented by inequality constraints on the parameters and initial conditions. From this information, GNA generates a state transition graph summarizing the qualitative dynamics of the system. In order to analyze large graphs, GNA allows the user to specify properties of the qualitative dynamics of a network in temporal logic, using high-level query templates, and to verify these properties on the state transition graph by means of standard model-checking tools, either locally installed or accessible through a remote web server. GNA is currently distributed by the company Genostar, but remains freely available for academic research purposes. The current version is GNA 8.3. In comparison with the previously distributed versions, GNA 8.3 has the following additional functionalities. First, it supports the editing and visualization of regulatory networks, in an SBGN-compatible format, and second it semi-automatically generates a prototype model from the network structure, thus accelerating the modeling process. For more information, see <http://www-helix.inrialpes.fr/gna>.

4.2. WellReader

Participants: Guillaume Baptist, Johannes Geiselmann, Jérôme Izard, Hidde de Jong [Correspondent], Delphine Ropers.

Gene expression, reporter gene data

WELLREADER is a program for the analysis of gene expression data obtained by means of fluorescent and luminescent reporter genes. WELLREADER reads data files in an XML format or in a format produced by microplate readers, and allows the user to detect outliers, perform background corrections and spline fits, compute promoter activities and protein concentrations, and compare expression profiles across different conditions. WELLREADER has been written in MATLAB and is available under an LGPL licence, both as source code (M files) and compiled code (platform-specific binary files). For more information, see: <http://ibis.inrialpes.fr/article957.html>.

IMAGINE Team

5. Software

5.1. MyCorporisFabrica

Participants: Ali-Hamadi Dicko, François Faure, Olivier Palombi.

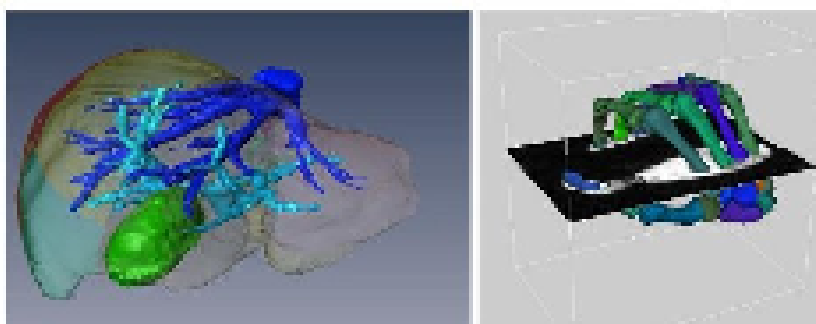


Figure 1. My Corporis Fabrica is an anatomical knowledge database developed in our team.

My Corporis Fabrica (MyCF) is an anatomical knowledge database (see fig. 1). During 2011, we have added new anatomical entities and improved some parts of FMA (Foundational Model of Anatomy). The FMA's license is now under Creative Commons licenses (CC-by : Licensees may copy, distribute, display and perform the work and make derivative works based on it only if they give the author or licensor the credits in the manner specified by these). The license of MyCF is not yet defined. Our new contribution this year, is the creation of a brand new ontology about human functions. Based on the International Classification of Functioning, Disability and Health, also known as ICF, we have organized human functions through a tree of 4330 items. A original journal paper must be submitted soon. MyCF browser is now available on line: <http://www.mycorporisfabrica.org/>. The MyCf's generic programming framework can be used for other domains. The link with semantic and 3D models matches research activities of IMAGINE towards interactive digital creation media. Anatomy can be seen as a study case.

5.2. SOFA

Participants: Guillaume Bousquet, Ali Hamadi Dicko, François Faure, François Jourdes.

SOFA is a C++ library primarily targeted at medical simulation research. Based on an advanced software architecture, it allows to (1) create complex and evolving simulations by combining new algorithms with algorithms already included in SOFA; (2) modify most parameters of the simulation – deformable behavior, surface representation, solver, constraints, collision algorithm, etc. – by simply editing an XML file; (3) build complex models from simpler ones using a scene-graph description; (4) efficiently simulate the dynamics of interacting objects using abstract equation solvers; and (5) reuse and easily compare a variety of available methods.

5.3. Convol

Participants: Marie-Paule Cani, Amaury Jung, Galel Koraa, Maxime Quiblier, Cédric Zanni.

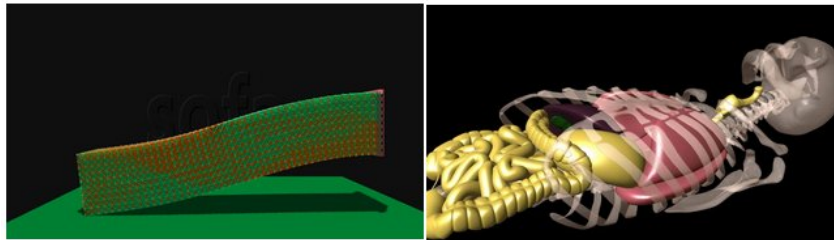


Figure 2. SOFA is an open source simulator for physically based modeling.

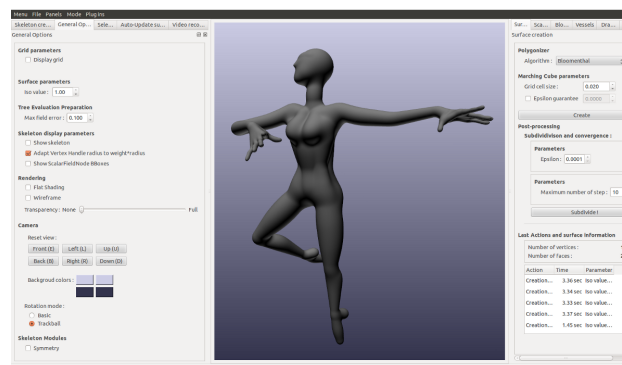


Figure 3. Example of implicit surface and the GUI proposed in the Convul software.

Convol is a new C++ library we develop for easing our work on implicit surfaces – and more particularly on the sub-class of convolution surfaces. It enables us to make our latest research results soon available to the rest of the group and easily usable in our industrial partnerships. Convol incorporates all the necessary material for constructive implicit modeling: skeleton-based distance and convolution primitives, with closed form solution for the field values and gradient whenever possible; a variety of blending operators; and several methods for tessellating an implicit surface into a mesh, and for refining the later in highly curved regions. The creation of new geometry can be performed by direct manipulation of skeleton or through sketch based modeling. This development is funded by Inria as support to our research group.

IMARA Project-Team

5. Software

5.1. MELOSYM

Participants: Fawzi Nashashibi [correspondant], Benjamin Lefaudeux, Paulo Lopes Resende.

MELOSYM is the acronym for “**M**odélisation de l’**E**nvironnement et **L**ocalisation en temps réel pour un **S**ystème **M**obile autonome ou pas, fondé sur des données du capteur laser”. This is a SLAM based algorithm for the environment mapping and vehicle localization in real time using laser data. The particularity of the algorithm is its hierarchical approach that improves the accuracy of the system and speeds up the computations.

- Version: V2

5.2. Stereoloc-3D

Participants: Benjamin Lefaudeux, Fawzi Nashashibi [correspondant].

This software is a stereovision based system capable of performing a vehicle or robot ego-localization and 3D environment mapping in real-time. It has also the capability to ensure mobile objects detection and tracking.

- Version: V1

IMEDIA2 Team

5. Software

5.1. IKONA/MAESTRO Software

Participants: Vera Bakic, Laurent Joyeux, Souheil Selmi.

IKONA is a generalist software dedicated to content-based visual information indexing and retrieval. It has been designed and implemented in our team during the last years [22]. Its main functionalities are the extraction, the management and the indexing of many state-of-the-art global and local visual features. It offers a wide range of interactive search and navigation methods including query-by-example, query-by-window, matching, relevance feedback, search results clustering or automatic annotation. It can manage several types of input data including images, videos and 3D models.

Based on a client/server architecture, it is easily deployable in any multimedia search engine or service. The communication between the two components is achieved through a proprietary network protocol. It is a set of commands the server understands and a set of answers it returns to the client. The communication protocol is extensible, i.e. it is easy to add new functionalities without disturbing the overall architecture. can be replaced by any new or existing protocol dealing with multimedia information retrieval.

The main processes are on the server side. They can be separated in two main categories:

- off-line processes: data analysis, features extraction and structuring
- on-line processes: answer the client requests

Several clients can communicate with the server. A good starting point for exploring the possibilities offered by IKONA is our web demo, available at https://www.rocq.inria.fr/cgi-bin/imedia/circario.cgi/bio_diversity?select_db=1. This CGI client is connected to a running server with several generalist and specific image databases, including more than 23,000 images. It features query by example searches, switch database functionality and relevance feedback for image category searches. The second client is a desktop application. It offers more functionalities. More screen-shots describing the visual searching capabilities of IKONA are available at <https://www.rocq.inria.fr/cgi-bin/imedia/circario.cgi/demos>.

IKONA is a pre-industrial prototype, with exploitation as a final objective. Currently, there does not exist a licensed competitor with the same range of functionalities. It exists several commercial softwares or systems exploiting technologies similar to some functionalities of IKONA but usually not the most advanced ones. We can for example cite the SDK developed by LTU company, the service proposed by AdVestigo company, etc. Many prototypes and demonstrators, industrial or academic, share some functionalities of IKONA but here again not the most advanced (e.g. Google Image Similarity Search Beta, IBM Muffin, etc.).

The main originality of IKONA is its **genericity** (in terms of visual features, metrics, input data, storage format, etc.), its **adaptivity** (to new visual features, new indexing structures or new search algorithms), its innovative **interactive** search functionalities (Local and Global Relevance Feedback, Local Search with Query Expansion, Search results clustering, etc.) and its **scalability** thanks to a generic indexing structure module than can support the integration of any new advances.

Current Users of IKONA include European and National Projects Participants through its integration in prototype multimedia systems, commercial companies through user trials (EXALEAD, INA, BELGA, AFP), General or Specific Public through Web demos (PI@ntNet leaf identification demo).

IKONA software provides a high degree of visibility to IMEDIA2 scientific works through demos in commercial, scientific and general public events (notably in most Inria national showrooms). It is also the mainstay of several Multimedia Systems developed at the European level, in conjunction with many Leader European Companies and Research Centers.

IN-SITU Project-Team

5. Software

5.1. jBricks

Participants: Stéphane Huot, Emmanuel Pietriga [correspondant], Mathieu Nancel, Romain Primet.

jBricks (Figure 1) is a Java toolkit that integrates a high-quality 2D graphics rendering engine based on ZVTM (section 5.2) and a versatile input configuration module (based on ICon [45] and FlowStates 5.4) into a coherent framework, enabling the exploratory prototyping of interaction techniques and rapid development of post-WIMP applications running on cluster-driven interactive visualization platforms such as wall-sized displays. The goal of this framework is to ease the development, testing and debugging of interactive visualization applications. It also offers an environment for the rapid prototyping of novel interaction techniques and their evaluation through controlled experiments.



Figure 1. jBricks applications running on the WILD platform (32 tiles for a total resolution of $20\,480 \times 6\,400$ pixels). (a) Zoomed-in visualization of the North-American part of the world-wide air traffic network (1 200 airports, 5 700 connections) overlaid on NASA's Blue Marble Next Generation images ($86\,400 \times 43\,200$ pixels) augmented with country borders ESRI shapefiles. (b) Panning and zooming in Spitzer's Infrared Milky Way ($396\,032 \times 12\,000$ pixels). (c) Controlled laboratory experiment for the evaluation of mid-air multi-scale navigation techniques.

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- Software benefit: See E. Pietriga, S. Huot, M. Nancel, R. Primet, Rapid Development of User Interfaces on Cluster-Driven Wall Displays with jBricks, EICS '11: Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems, pages 185-190, June 2011
- OS/Middleware: Java (Linux, Mac OS X, Windows)
- Required library or software: several, managed through Maven
- Programming language: Java

5.2. The Zoomable Visual Transformation Machine

Participants: Caroline Appert, Rodrigo de Almeida, Olivier Chapuis, Arjit Gupta, Emmanuel Pietriga [correspondant], Mathieu Nancel, Romain Primet.

ZVTM provides application programmers with building blocks for implementing complex multi-scale interface components that cannot be handled by traditional WIMP widgets. Featuring off-the-shelf visualisation and navigation components that are easy to combine, ZVTM provides a simple yet powerful API and handles low-level operations such as multi-threading, clipping, repaint requests and animation management. The toolkit is based on the metaphor of universes that can be observed through smart movable/zoomable cameras. The graphical object model permits management of a large number of complex geometrical shapes. It emphasizes perceptual continuity via an advanced animation module that can animate virtually any on-screen modification. This ranges from camera movements and activation of distortion lenses to modification of the visual variables of graphical objects. Various temporal pacing functions are available to control the execution of these animations. ZVTM is now one of the core components of our jBricks toolkit for wall-sized displays (Section 5.1), and current development activities around the toolkit focus on making applications run transparently on cluster-driven ultra-high-resolution wall-sized displays such as that of the WILD visualization platform. The toolkit is also used to develop advanced visualization components for the ALMA observatory's operations monitoring and control software [26].

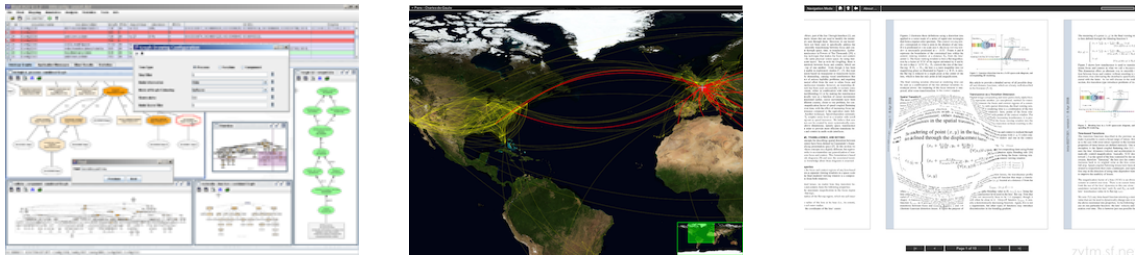


Figure 2. ZVTM used in various applications

Initially developed by Xerox Research Centre Europe and the World Wide Web Consortium (W3C) team at MIT, ZVTM has been available as open-source software under the GNU Lesser General Public License (LGPL) since early 2002. It is used in both academic and industrial projects such as IsaViz (<http://www.w3.org/2001/11/IsaViz/>), W3C's visual browser/editor for RDF, Blast2GO (Figure 2 - left) (<http://www.blast2go.org/>), or ZGRViewer (<http://zvtm.sourceforge.net/zgrviewer.html>) for viewing large graphs generated by AT&T GraphViz¹ (Figure 2 - right). The development of the toolkit is now supported by Inria. More information can be found at <http://zvtm.sourceforge.net> and [52] and [51].

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- Software benefit: See Pietriga, A Toolkit for Addressing HCI Issues in Visual Language Environments, IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC '05), pages 145-152, September 2005
- License: LGPL
- Type of human computer interaction: Graphique
- OS/Middleware: Java (Linux, Mac OS X, Windows)
- Required library or software: several, managed through Maven
- Programming language: Java

5.3. The SwingStates Toolkit

Participants: Caroline Appert [correspondant], Michel Beaudouin-Lafon.

¹<http://www.graphviz.org>

SwingStates [37] is a library that adds state machines and a graphical canvas to the Java Swing user interface toolkit. It was motivated by the lack of widely disseminated toolkits that support advanced interaction techniques and the observation that HCI research toolkits are little used outside the lab. By extending the popular Java Swing toolkit rather than starting from scratch, the goal is to facilitate the dissemination and adoption of SwingStates by practitioners.

SwingStates uses *state machines* to specify interaction. It provides programmers with a natural syntax to specify state machines and reduces the potential for an explosion of the number of states by allowing multiple state machines to work together or separately. SwingStates can be used to add new interaction techniques to existing Swing widgets, e.g. to select buttons and checkboxes by crossing rather than clicking. It can also be used with the SwingStates canvas (see below) and to control high-level dialogues.

SwingStates also provides a powerful *canvas widget*. The canvas can contain any Java2D shape, including geometric shapes, images, text strings and even Swing widgets. Shapes can be manipulated individually or collectively, through *tags*. An intensive use of polymorphism allows to apply almost any command to a tag: the command is then applied to all objects with this tag. Tags are also used in conjunction with state machines, to specify transitions that occur only on objects with a given tag. For example, pie menus can be implemented by creating a canvas in the overlay layer of any Swing application (Figure 3).



Figure 3. A numeric text field whose value can be set by a joystick-like interaction (left) and a semi-transparent menu to change the background color of Swing widgets (right)

SwingStates tightly integrates state machines, the Java language and the Swing toolkit to provide programmers with a natural and powerful extension to their natural programming environment. SwingStates is available at <http://swingstates.sf.net> under the GNU Lesser General Public License (LGPL).

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- Software benefit: See C. Appert and M. Beaudouin-Lafon (2008) SwingStates: Adding State Machines to Java and the Swing Toolkit. *Software: Practice and Experience*, 38(11):1149 - 1182.
- OS/Middleware: Mac OS X, Linux, Windows
- Required library or software: Java virtual machine
- Programming language: Java

5.4. The FlowStates Toolkit

Participants: Caroline Appert [correspondant], Michel Beaudouin-Lafon, Stéphane Huot.

FlowStates [38], is a new toolkit to program advanced interaction techniques which require non standard input (e.g., two different mice that act independently, a joystick, a tablet, etc.). It is built on top of two existing toolkits: SwingStates [37] and ICon [45].

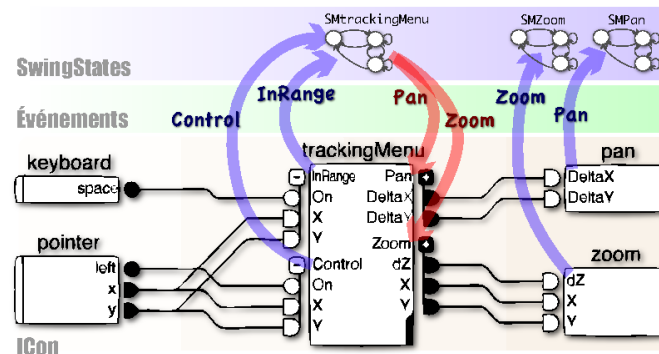


Figure 4. State machines and data flow in FlowStates

With FlowStates the developer can program interaction logic using state machines like SwingStates does but does not restrict the set of possible input channels to Java AWT standard input (a single couple <mouse, keyboard>). The state machines just have to define the virtual input events that are required to trigger their transitions so that FlowStates turns these machines into ICon devices which can be plugged to any physical input channels (Figure 4). An ICon device is a data flow building block that has input and output slots in order to be connected to other devices in the simple graphical environment provided by ICon. State machines can also send out events which appear as output slots in the data flow model.

With FlowStates we showed how two models for programming interaction (state machines and data flow) can be fully integrated to offer a huge power of expression. The explicit decision to not set strict limits between the roles of each model makes this hybrid approach highly flexible, the developer setting himself the limit between the two according to his needs and habits.

FlowStates is available at <http://www.lri.fr/~appert/FlowStates/>.

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- Software benefit: See C. Appert, S. Huot, P. Dragicevic and M. Beaudouin-Lafon (2009) FlowStates: Prototypage d'applications interactives avec des flots de données et des machines à états. In Proceedings of IHM 2009. ACM, pages 119-128.
- OS/Middleware: Mac OS X, Linux, Windows
- Required library or software: ICon, Java virtual machine
- Programming language: Java

5.5. TouchStone

Participants: Caroline Appert [correspondant], Michel Beaudouin-Lafon, Wendy Mackay.

TouchStone [8] is a platform for designing, running and analyzing the results of controlled experiments (Figure 5). While it focuses on experiments comparing interaction techniques, it can be used in a wide variety of contexts.

With the *Touchstone design platform*, a user specifies the factors and the measures of the experiment, the blocking and counterbalancing of trials, and assess the time it will take to run the experiment. Multiple designs can be explored in parallel to assess the various trade-offs. The output of the design platform is an XML file that can be used as input for the run platform.

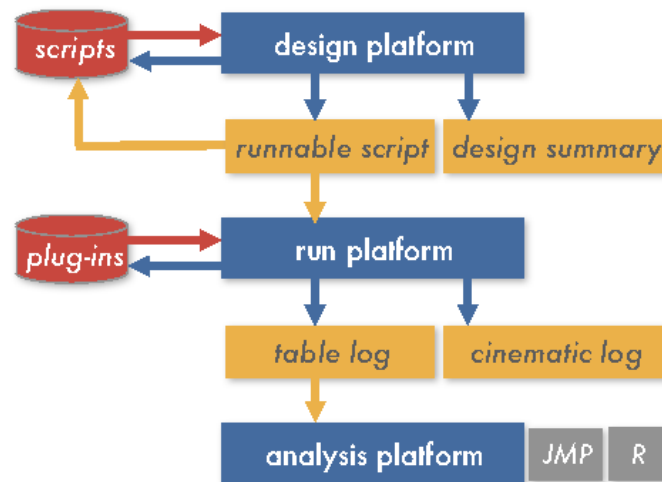


Figure 5. The architecture of the Touchstone platform

The *Touchstone run platform* provides a framework to implement and run an experiment and to collect experimental data. It uses a flexible plug-in architecture to manage a variety of input devices and interaction techniques. The runs of the experiment are controlled by an XML script that can be produced by the design platform.

The analysis platform currently consists of data analysis tools such as JMP, R or Excel. Log data produced by the run platform can be directly loaded into any of these tools. In a future version, analysis sketches will be derived from the experimental design to assist with the analysis.

Touchstone has been used heavily at INSITU over the past three years for the many experiments that we design and run. It has also been used for teaching for the first time in 2011. Students used it to design various experiments during tutorial classes in Master 2 Interaction (“Introduction to HCI” module).

Touchstone is available at <http://code.google.com/p/touchstone-platforms/> under a BSD License.

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- Software benefit: See W. Mackay, C. Appert, M. Beaudouin-Lafon, O. Chapuis, Y. Du, JD. Fekete and Y. Guiard (2007) TouchStone: Exploratory Design of Experiments. In Proceedings of ACM CHI 2007 Conference on Human Factors and Computing Systems. ACM, pages 1425-1434.
- OS/Middleware: Mac OS X, Linux, Windows
- Required library or software: Java virtual machine
- Programming language: Java

5.6. Metisse

Participant: Olivier Chapuis [correspondant].

Metisse [43] is a window system that facilitates the design, implementation and evaluation of innovative window management techniques. The system is based on a compositing approach, making a clear distinction between the rendering and the interactive compositing processes. The Metisse server is a modified X server that supports both input and output redirection. The default compositor is a combination of a slightly modified version of FVWM, a standard window manager, with an interactive viewer application called *FvwmCompositor*.

FvwmCompositor uses OpenGL to display windows, which offers a rich graphics model well adapted to the exploration of new window management techniques. Texture mapping, for example, makes it possible to transform the window shapes in real-time (Figure 6 , left). Alpha blending makes it easy to create translucent objects and shadows. Scaling, rotation and translation can also be used to position windows in $2D\frac{1}{2}$ or 3D (Figure 6 , middle and right). Input redirection makes it still possible to interact with applications no matter the visual transformations applied to the windows. It also makes it possible to adapt, reconfigure or re-combine existing graphical interfaces [54]. This year we used again Metisse to implement novel desktop interaction techniques [4].

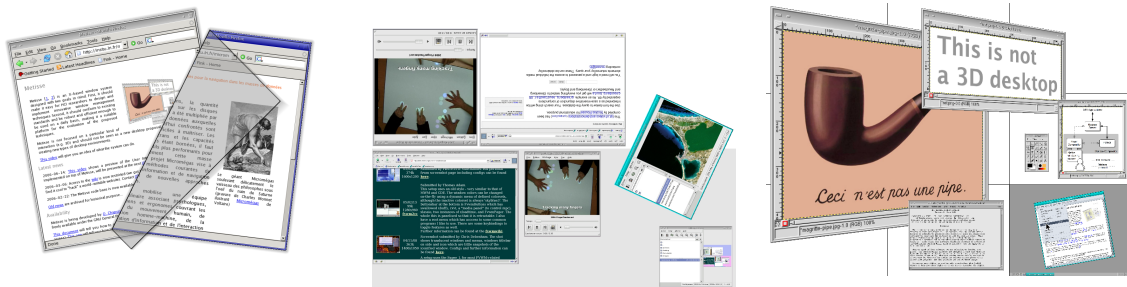


Figure 6. Sample window management techniques implemented with Metisse: extended paper metaphor (left), interactive table configuration that allows to duplicate and rotate windows (middle) and zoomable 3D desktop (right).

- Web: <http://insitu.lri.fr/metisse/>
- ACM: H.5.2 [User Interfaces]: Windowing systems
- Software benefit: see [43], [54], [44], [47] and [4].
- License: GPL
- Type of human computer interaction: Graphique
- OS/Middleware: X Window et Mac OS X
- Required library or software: OpenGL via nucleo² and some usual C/C++ libraries
- Programming language: * C/C++

5.7. Wmtrace

Participant: Olivier Chapuis [correspondant].

Wmtrace [42] includes two tools that help us study an individual user's window management activity. The first tool runs in the background of an X Window session and continuously logs information about windows and how they are being manipulated. The second uses a VCR-like interface (Figure 7) to replay the resulting logs and analyze the entire session. This tool provides several ways to filter the logs and extract high-level information, including interactive move events and mouse speed. Both tools allow HCI researchers to perform qualitative and quantitative statistical analyses of window management activity.

- Web: <http://insitu.lri.fr/~chapuis/software/wmtrace/>.
- ACM: H.5.2 [User Interfaces]: Windowing systems
- Software benefit: see [42], [47], [41].
- License: GPL

²<http://interaction.lille.inria.fr/~rousseau/projects/nucleo/index.html>

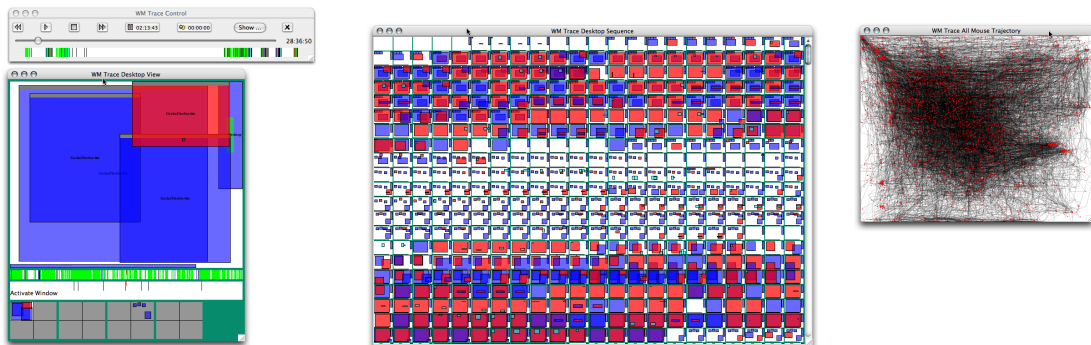


Figure 7. VCR-like interface, session overview and sample plots of mouse trajectories (black) and mouse clicks (red)

- Type of human computer interaction: Daemon and Graphique
- OS/Middleware: X Window (daemon) and Java (VCR interface)
- Required library or software: all X libraries (daemon) and Java (VCR interface)
- Programming language: * C and Java

5.8. The Substance Middleware

Participants: Michel Beaudouin-Lafon [correspondant], Clemens Klokrose, Tony Gjerlufsen, James Eagan, Clement Pillias.

Substance is a middleware based on a novel programming paradigm called *data-oriented programming* and was designed to facilitate the development of multi-surface interactive applications [48]. Such applications are distributed by nature as they involve a varying number of display and interaction surfaces that are controlled by different computers. For example, our WILD room includes a 32-monitor display wall driven by 16 computers plus a front-end, a multi-touch table, various mobile devices such as iPodTouch and iPads, and the laptops that the users of the room may bring with them. We want to support seamless interaction techniques across these surfaces, such as the pick-and-drop technique pioneered by Rekimoto [53].

Data-oriented programming consists of attaching functionality to a tree data structure through *facets* attached to the individual nodes of the tree. Facets can be added and removed dynamically, and notified of changes in the tree. Substance supports two powerful ways to share nodes and facets: mounting, where access to the shared tree is managed through remotely, and replication, where the shared tree is replicated at each site and synchronized.

Substance has been used to create two full-scale applications (Figure 8): a generalized Canvas that can display and manage graphics, PDF files, image files and other content (through an extensible content manager) across surfaces spanning multiple displays and computers; SubstanceGrise, which uses multiple instances of the Anatomist/BrainVISA application to display coordinated 3D imagery of many brains in parallel on the WILD wall and control from a physical model of the brain.

Substance is available at <http://substance-env.sourceforge.net/> under a GNU GPL 3.0 licence.

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)



Figure 8. The Canvas (left) and SubstanceGrise (right) applications developed with Substance.
(©CNRS-Phototheque - Cyril FRESILLON for SubstanceGrise).

- Software benefit: See T. Gjerlufsen, C. Klokmoose, J. Eagan, C. Pillias and M. Beaudouin-Lafon (2011) Shared Substance: Developing Flexible Multi-Surface Applications. In CHI '11: Proceedings of the 29th international conference on Human factors in computing systems. ACM, pages 3383-3392.
- OS/Middleware: Mac OS X, Linux
- Required library or software: several, managed by Python install
- Programming language: Python

5.9. Scotty

Participants: Michel Beaudouin-Lafon [correspondant], James Eagan, Wendy Mackay.

The goal of Scotty is to support *malleable interfaces*, i.e. interfaces that can be modified at run-time in ways not anticipated by the designers [46]. Scotty is a toolkit that allows a programmer to extend an existing Mac OS X application without access to its source code. Scotty provides the following abstractions: hooks to alter the appearance of windows and widgets, event funnels to alter their behavior, glass sheets to overlay graphics and add new interaction methods, dynamic code loading and object proxies to redefine and extend existing objects. Scotty also provides a higher-level interface based on instrumental interaction [39]. Scotty currently runs on Mac OS X for applications written with the Cocoa user interface framework.

Scotty has been used to create a number of extensions (Figure 9). *Scribbler* is a generic extension that uses glass sheets to allow handwritten annotations of any Cocoa window. *Teleportation* is another generic extension that can teleport and resize the content of any Cocoa window onto another computer, including an iPhone or iPad. The user can interact with the teleported content as if it was on the original computer. It was used to create a content provider for the Substance Canvas (see above), making it possible to display any application running on a laptop onto the WILD wall display and/or table. When vector-based content is available, e.g., for text, Scotty provides smooth rescaling without the typical pixelation apparent when enlarging bitmap images. Finally *Stylesheet* is an extension to the Pages word processor that provides a semi-transparent toolglass for specifying the styles of paragraphs.

Scotty is available at <http://insitu.lri.fr/Projects/Scotty> under a GNU GPL 3.0 licence.

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- Software benefit: See J. Eagan, W. Mackay and M. Beaudouin-Lafon (2011) Cracking the Cocoa Nut: User Interface Programming at Runtime. In UIST 2011: Proceedings of the 24th ACM Symposium on User Interface Software and Technology. ACM, pages 225-234.

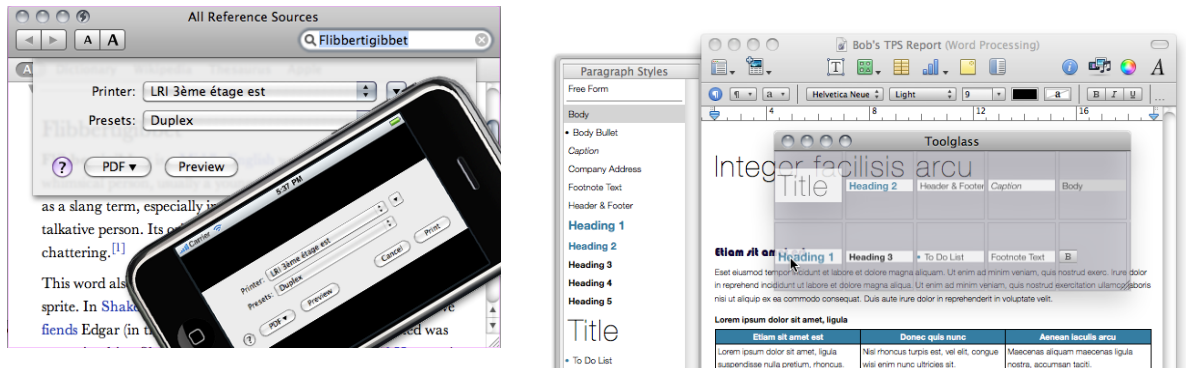


Figure 9. Using Scotty to teleport a window of a Mac OS X application onto an iPhone (left) and to create a toolglass in the Pages word processor (right).

- OS/Middleware: Mac OS X
- Required library or software: none
- Programming language: Objective-C, Python

INDES Project-Team

5. Software

5.1. Introduction

Most INDES software packages, even the older stable ones that are not described in the following sections are freely available on the Web. In particular, some are available directly from the Inria Web site:

<http://www.inria.fr/valorisation/logiciels/langages.fr.html>

Most other software packages can be downloaded from the INDES Web site:

<http://www-sop.inria.fr/teams/indes>

5.2. Functional programming

Participants: Frédéric Boussinot [Inria], Cyprien Nicolas [Inria], Bernard Serpette [Inria], Manuel Serrano [correspondant].

5.2.1. The Bigloo compiler

The programming environment for the Bigloo compiler [5] is available on the Inria Web site at the following URL: <http://www-sop.inria.fr/teams/indes/fp/Bigloo>. The distribution contains an optimizing compiler that delivers native code, JVM bytecode, and .NET CLR bytecode. It contains a debugger, a profiler, and various Bigloo development tools. The distribution also contains several user libraries that enable the implementation of realistic applications.

BIGLOO was initially designed for implementing compact stand-alone applications under Unix. Nowadays, it runs harmoniously under Linux and MacOSX. The effort initiated in 2002 for porting it to Microsoft Windows is pursued by external contributors. In addition to the native back-ends, the BIGLOO JVM back-end has enabled a new set of applications: Web services, Web browser plug-ins, cross platform development, etc. The new BIGLOO .NET CLR back-end that is fully operational since release 2.6e enables a smooth integration of Bigloo programs under the Microsoft .NET environment.

5.2.2. The FunLoft language

FunLoft (described in <http://www-sop.inria.fr/teams/indes/rp/FunLoft>) is a programming language in which the focus is put on safety and multicore.

FunLoft is built on the model of FairThreads which makes concurrent programming simpler than usual preemptive-based techniques by providing a framework with a clear and sound semantics. FunLoft is designed with the following objectives:

- provide a safe language, in which, for example, data-races are impossible.
- control the use of resources (CPU and memory), for example, memory leaks cannot occur in FunLoft programs, which always react in finite time.
- have an efficient implementation which can deal with large numbers of concurrent components.
- benefit from the real parallelism offered by multicore machines.

A first experimental version of the compiler is available on the Reactive Programming site <http://www-sop.inria.fr/teams/indes/rp>. Several benchmarks are given, including cellular automata and simulation of colliding particles.

5.3. Web programming

Participants: Gérard Berry [Inria], Cyprien Nicolas [Inria], Manuel Serrano [correspondant].

5.3.1. The HOP web programming environment

HOP is a higher-order language designed for programming interactive web applications such as web agendas, web galleries, music players, etc. It exposes a programming model based on two computation levels. The first one is in charge of executing the logic of an application while the second one is in charge of executing the graphical user interface. HOP separates the logic and the graphical user interface but it packages them together and it supports strong collaboration between the two engines. The two execution flows communicate through function calls and event loops. Both ends can initiate communications.

The HOP programming environment consists in a web *broker* that intuitively combines in a single architecture a web server and a web proxy. The broker embeds a HOP interpreter for executing server-side code and a HOP client-side compiler for generating the code that will get executed by the client.

An important effort is devoted to providing HOP with a realistic and efficient implementation. The HOP implementation is *validated* against web applications that are used on a daily-basis. In particular, we have developed HOP applications for authoring and projecting slides, editing calendars, reading RSS streams, or managing blogs.

HOP has won the software *open source contest* organized by the ACM Multimedia Conference 2007 <http://mmc36.informatik.uni-augsburg.de/acmmm2007/>. It is released under the GPL license. It is available at <http://hop.inria.fr>.

5.4. Language-based security

Participants: Zhengqin Luo [Inria], Tamara Rezk [correspondant].

5.4.1. CFlow

The prototype compiler “CFlow” takes as input code annotated with information flow security labels for integrity and confidentiality and compiles to F# code that implements cryptography and protocols that satisfy the given security specification.

Cflow has been coded in F#, developed mainly on Linux using mono (as a substitute to .NET), and partially tested under Windows (relying on .NET and Cygwin). The code is distributed under the terms of the CeCILL-B license <http://www.msr-inria.inria.fr/projects/sec/cflow/index.html>.

5.4.2. FHE type-checker

We have developed a type checker for programs that feature modern cryptographic primitives such as fully homomorphic encryption. The type checker is thought as an extension of the “CFlow” compiler developed last year on the same project. It is implemented in F#. The code is distributed under the terms of the CeCILL-B license <http://www.msr-inria.inria.fr/projects/sec/cflow/index.html>.

5.4.3. Mashic compiler

The Mashic compiler is applied to mashups with untrusted scripts. The compiler generates mashups with sandboxed scripts, secured by the same origin policy of the browsers. The compiler is written in Bigloo and can be found at <http://www-sop.inria.fr/indes/mashic/>.

5.5. Old software

5.5.1. Camloo

Camloo is a caml-light to bigloo compiler, which was developed a few years ago to target bigloo 1.6c. New major releases 0.4.x of camloo have been done to support bigloo 3.4 and bigloo 3.5. Camloo make it possible for the user to develop seamlessly a multi-language project, where some files are written in caml-light, in C, and in bigloo. Unlike the previous versions of camloo, 0.4.x versions do not need a modified bigloo compiler to obtain good performance. Currently, the only supported backend for camloo is bigloo/C. We are currently rewriting the runtime of camloo in bigloo to get more portability and to be able to use HOP and camloo together.

5.5.2. *Skribe*

SKRIBE is a functional programming language designed for authoring documents, such as Web pages or technical reports. It is built on top of the SCHEME programming language. Its concrete syntax is simple and looks familiar to anyone used to markup languages. Authoring a document with SKRIBE is as simple as with HTML or LaTeX. It is even possible to use it without noticing that it is a programming language because of the conciseness of its original syntax: the ratio *tag/text* is smaller than with the other markup systems we have tested.

Executing a SKRIBE program with a SKRIBE evaluator produces a target document. It can be HTML files for Web browsers, a LaTeX file for high-quality printed documents, or a set of *info* pages for on-line documentation.

5.5.3. *Scheme2JS*

Scm2JS is a Scheme to JavaScript compiler distributed under the GPL license. Even though much effort has been spent on being as close as possible to R5RS, we concentrated mainly on efficiency and interoperability. Usually Scm2JS produces JavaScript code that is comparable (in speed) to hand-written code. In order to achieve this performance, Scm2JS is not completely R5RS compliant. In particular it lacks exact numbers.

Interoperability with existing JavaScript code is ensured by a JavaScript-like dot-notation to access JavaScript objects and by a flexible symbol-resolution implementation.

Scm2JS is used on a daily basis within HOP, where it generates the code which is sent to the clients (web-browsers). Scm2JS can be found at <http://www-sop.inria.fr/indes/scheme2js>.

IPSO Project-Team (section vide)

KERDATA Project-Team

5. Software

5.1. BlobSeer

Participants: Viet-Trung Tran, Zhe Li, Alexandru Costan, Gabriel Antoniu, Luc Bougé.

Contact: Gabriel Antoniu.

Presentation: BlobSeer is the core software platform for most current projects of the KerData team. It is a data storage service specifically designed to deal with the requirements of large-scale data-intensive distributed applications that abstract data as huge sequences of bytes, called BLOBs (Binary Large Objects). It provides a versatile versioning interface for manipulating BLOBs that enables reading, writing and appending to them.

BlobSeer offers both scalability and performance with respect to a series of issues typically associated with the data-intensive context: *scalable aggregation of storage space* from the participating nodes with minimal overhead, ability to store *huge data objects*, *efficient fine-grain access* to data subsets, *high throughput in spite of heavy access concurrency*, as well as *fault-tolerance*.

Users: Work is currently in progress in several formalized projects (see previous section) to integrate and leverage BlobSeer as a data storage back-end in the reference cloud environments: a) Microsoft Azure; b) the Nimbus cloud toolkit developed at Argonne National Lab (USA); and c) in the OpenNebula IaaS cloud environment developed at UCM (Madrid).

URL: <http://blobseer.gforge.inria.fr/>

License: GNU Lesser General Public License (LGPL) version 3.

Status: This software is available on Inria's forge. Version 1.0 (released late 2010) registered with APP: IDDN.FR.001.310009.000.S.P.000.10700.

A new *Technology Research Action* (ADT, *Action de recherche technologique*) has been launched in Septembre 2012 for one year, with a possible 1-year renewal, to robustify the BlobSeer software and and make it a safely distributable product. This project is funded by Inria *Technological Development Office* (D2T, *Direction du Développement Technologique*). Zhe Li has been hired as a senior (PhD) engineer for this task.

5.2. Damaris

Participants: Matthieu Dorier, Gabriel Antoniu.

Contact: Gabriel Antoniu.

Presentation: Damaris is a middleware for multicore SMP nodes enabling them to efficiently handle data transfers for storage and visualization. The key idea is to dedicate one or a few cores of each SMP node to the application I/O. It is developed within the framework of a collaboration between KerData and the Joint Laboratory for Petascale Computing (JLPC). The current version enables efficient asynchronous I/O, hiding all I/O related overheads such as data compression and post-processing. On-going work is targeting fast direct access to the data from running simulations, and efficient I/O scheduling.

Users: Damaris has been preliminarily evaluated at NCSA (Urbana-Champaign) with the CM1 tornado simulation code. CM1 is one of the target applications of the Blue Waters supercomputer developed by at NCSA/UIUC (USA), in the framework of the Inria-UIUC-ANL Joint Lab (JLPC). Damaris now has external users, including (to our knowledge) visualization specialists from NCSA and researchers from the France/Brazil Associated research team on Parallel Computing (joint team between Inria/LIG Grenoble and the UFRGS in Brazil). Damaris has been successfully integrated into three large-scale simulations (CM1, OLAM, Nek5000). Works are in progress to evaluate it in the context of several other simulations including HACC (cosmology code) and GTC (fusion).

URL: <http://damaris.gforge.inria.fr/>

License: GNU Lesser General Public License (LGPL) version 3.

Status: This software is available on Inria's forge. Registration with APP is in progress.

5.3. Derived software

Derived from BlobSeer, two additional platforms are currently being developed within KerData: 1) Pyramid, a software service for array-oriented active storage developed within the framework of the PhD thesis of Viet-Trung Tran; and 2) BlobSeer-WAN, a data management service specifically optimized for geographically distributed environments. It is also developed within the framework of the PhD thesis of Viet-Trung Tran in relation to the FP3C project. These platforms have not been publicly released yet.

LAGADIC Project-Team

5. Software

5.1. ViSP: a visual servoing platform

Participants: Fabien Spindler [correspondant], Filip Novotny, Aurélien Yol, Eric Marchand, François Chaumette.

Since 2005, we develop and release under the terms of the GPLv2 licence, ViSP, an open source library that allows fast prototyping of visual tracking and visual servoing tasks. ViSP was designed to be independent with the hardware, to be simple to use, expandable and cross-platform.

ViSP allows to design vision-based tasks for eye-in-hand and eye-to-hand visual servoing that contains the most classical visual features that are used in practice. It involves a large set of elementary positioning tasks with respect to various visual features (points, segments, straight lines, circles, spheres, cylinders, image moments, pose,...) that can be combined together, and image processing algorithms that allows tracking of visual cues (dots, segments, ellipses,...) or 3D model-based tracking of known objects. Simulation capabilities are also available. ViSP and its full functionalities are presented in Fig. 1 and described in [6].

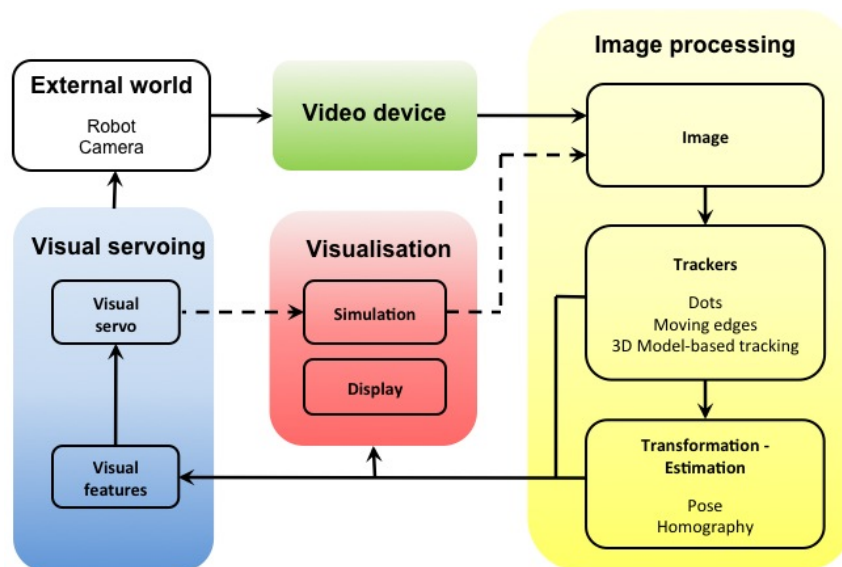


Figure 1. ViSP software architecture.

This year, we continued our efforts to improve the software and documentation quality. A new version available at <http://www.irisa.fr/lagadic/visp/visp.html> was released in July 2012. To ease ViSP installation, we provide also precompiled ViSP SDK including pre-built ViSP library and headers.

This last release under deposit to the APP (“Agence de Protection des Programmes”) has been downloaded 887 times since its availability. It is used in research labs in France, USA, Japan, Korea, India, China, Lebanon, Italy, Spain, Portugal, Hungary, Canada. For instance, it is used as a support in a graduate course delivered at MIT, at IFMA Clermont-Ferrand and ESIR Rennes engineer schools. ViSP is now also part of “vision_visp” ROS stack (see http://www.ros.org/wiki/vision_visp) and ViSP 3D model-based tracker has been proposed by colleagues from Laas in Toulouse as a ROS package. This encouraged us to enhance “vision_visp” stack by proposing new ROS packages to calibrate intrinsic and extrinsic camera parameters, and a new 3D model-based tracker with automatic initialisation and reinitialisation after tracking loss (with help of specific textured patterns on the object).

5.2. DESlam

Participants: Patrick Rives [correspondant], Maxime Meilland.

The DESlam (Dense Egocentric Slam) software developed in collaboration with Andrew Comport from I3S in Sophia Antipolis was deposited to the APP (“Agence de Protection des Programmes”) (IDDN.FR.001.320001.000.S.P.2012.000.21000). This software proposes a full and self content solution to the dense Slam problem. Based on a generic RGB-D representation valid for various type of sensors (stereovision, multi-cameras, RGB-D sensors...), it provides a 3D textured representation of complex large indoors or outdoors environments and it allows to localize in real time (45Hz) a robot or a person carrying out a mobile camera.

5.3. Development work: Robot vision platforms

Participant: Fabien Spindler [correspondant].

We exploit two industrial robotic systems built by Afma Robots in the nineties to validate our researches in visual servoing and active vision. The first one is a Gantry robot with six degrees of freedom, the other one is a cylindrical robot with four degrees of freedom (see Fig. 2). These robots are equipped with cameras. The Gantry robot allows also to embed grippers on its end-effector.

Two papers published by Lagadic in 2012 enclose results validated on this platform. Note that it is also opened to researcher from other labs. For example, this year an associate professor from LSIIT in Strasbourg did experiments on the Gantry robot.

5.4. Development work: Medical robotics platforms

Participants: Fabien Spindler [correspondant], Alexandre Krupa.

This tesbed is of primary interest for researches and experiments concerning ultrasound visual servoing applied to positioning or tracking tasks described in Section 6.4 .

This platform is composed by a six degrees of freedom Adept Viper S850 arm (see Fig. 3). This year we bought a new Adept Viper S650 arm to replace our eight year old Hippocrate medical arm designed by the Sinters company. Ultrasound probes connected either to a SonoSite 180 Plus or an Ultrasonix SonixTouch imaging system can be mounted on a force torque sensor attached to each robot end-effector.

We plan to exploit the two Viper robots for demonstrating needle insersion under ultrasound imaging to precisely guide the needle toward a target while optimizing its visibility (see Section 6.4.4).

Note that four papers published by Lagadic in 2012 enclose experimental results obtained with this platform.

5.5. Development work: Mobile robotics platforms

Participants: Fabien Spindler [correspondant], Marie Babel, Patrick Rives.

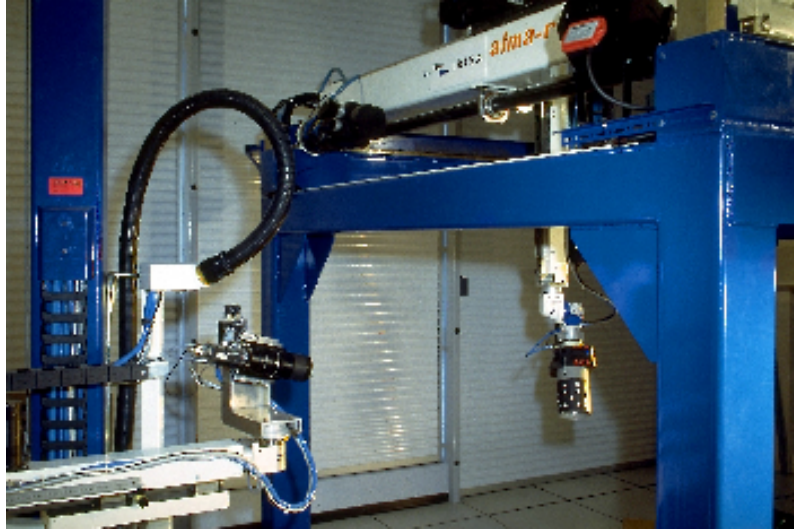


Figure 2. Lagadic robotics platforms for vision-based manipulation



Figure 3. Lagadic medical robotics platforms. On the right Viper S850 robot arm equipped with a SonixTouch 3D ultrasound probe. On the left Viper S650 equipped with a tool changer that allows to attach a classical camera.

5.5.1. Indoors mobile robots

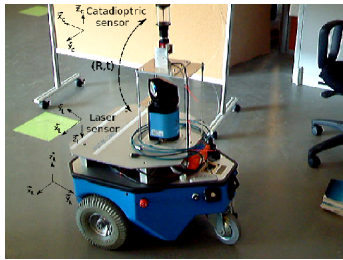
For fast prototyping of algorithms in perception, control and autonomous navigation, the team uses Hannibal in Sophia Antipolis, a cart-like platform built by Neobotix (see Fig. 4 .a), and a Pioneer 3DX from Adept in Rennes (see Fig. 4 .b) as well as a Robotino from Festo. These platforms are equipped with various sensors needed for Slam purposes, autonomous navigation and sensor-based control.

Moreover, to validate the researches in personally assisted living topic (see 6.3.6), we bought in Rennes a six wheel electric wheelchair from Penny and Giles Drives Technology (see Fig. 4 .c). The control of the wheelchair is performed using a plug and play system between the joystick and the low level control of the wheelchair. Such a system let us acquire the user intention through the joystick position and control the wheelchair by applying corrections to its motion. The wheelchair has been fitted with three cameras to perform the required servoing for assisting handicapped people. Moreover, to ensure the direct security of the user, seven infrared proximity sensors have been installed all around the wheelchair.

5.5.2. Outdoors mobile robots

The team exploit also Cycab urban electrical cars (see Figs. 4 .d and 4 .e). Two vehicles in Sophia Antipolis and one in Rennes are instrumented with cameras and range finders to validate researches in the domain of intelligent urban vehicle. Cycabs were used as experimental testbeds in several national projects.

Note that 5 papers published by Lagadic in 2012 enclose experimental results obtained with these mobile robotics platforms.



(a)



(b)



(c)



(d)



(e)

Figure 4. a) Hannibal platform, b) Pioneer P3-DX robot, c) six wheel electric wheelchair, d) Cycab available in Rennes, e) one of the Cycabs available in Sophia Antipolis.

LEAR Project-Team

5. Software

5.1. Face recognition

Participants: Guillaume Fortier [correspondant], Jakob Verbeek.

In a collaboration with Technosens (a start-up based in Grenoble) we are developing an efficient face recognition library. During 18 months Guillaume Fortier, financed by Inria's technology transfer program, had streamlined code developed by different former team members on various platforms. This encompasses detection of characteristic points on the face (eyes, nose, mouth), computing appearance features on these points, and learning metrics on the face descriptors that are useful for face verification (faces of the same person are close, faces of different people are far away). See <http://lear.inrialpes.fr/~fortier/software.php>.

5.2. Large-scale image classification

Participants: Matthijs Douze [correspondant], Zaid Harchaoui, Florent Perronnin [XRCE], Cordelia Schmid.

JSGD is the implementation of a Stochastic Gradient Descent algorithm used to train linear multiclass classifiers. It is biased towards large classification problems (many classes, many examples, high dimensional data). It can be used to reproduce the results from [19] on the ImageNet large scale classification challenge. It uses several optimization techniques, both algorithmic (scale factors to spare vector multiplications, vector compression with product quantizers) and technical (vector operations, multithreading, improved cache locality). It has Python and Matlab interfaces. It is distributed under a Cecill licence. Project page: <http://lear.inrialpes.fr/src/jsgd>.

5.3. Fisher vector image representation

Participants: Matthijs Douze [correspondant], Hervé Jégou [TEXMEX Team Inria Rennes], Cordelia Schmid.

We developed a package that computes Fisher vectors on sparse or dense local SIFT features. The dense feature extraction was optimized, so that they can be computed in real time on video data. The implementation was used for several publications [6], [16] and in our submission to the Trecvid 2012 MED task [31]. We provide a binary version of the local descriptor implementation, and the Fisher implementation is integrated in the Yael library, with Python and Matlab interface, see http://lear.inrialpes.fr/src/inria_fisher.

5.4. Video descriptors

Participants: Dan Oneata, Cordelia Schmid [correspondant], Heng Wang.

We have developed and made on-line available software for video description based on dense trajectories and motion boundary histograms [28]. The trajectories capture the local motion information of the video. A state-of-the-art optical flow algorithm enables a robust and efficient extraction of the dense trajectories. Descriptors are aligned with the trajectories and based on motion boundary histograms (MBH) which are robust to camera motion. This year we have further developed this software to increase its scalability to large datasets. On the one hand we explored the effect of sub-sampling the video input both spatially and temporally, and evaluated the impact on the quality of the descriptors. On the other hand we avoid writing the raw MBH descriptors to disk, but rather aggregate them directly into a signature for the complete video using Fisher vectors, or bag-of-word descriptors. This allowed us to use these descriptors on the 4,000 hour video dataset of the TrecVid 2012 MED task.

LFANT Project-Team

5. Software

5.1. Pari/Gp

Participants: Karim Belabas [correspondant], Bill Allombert, Henri Cohen, Andreas Enge.

<http://pari.math.u-bordeaux.fr/>

PARI/GP is a widely used computer algebra system designed for fast computations in number theory (factorisation, algebraic number theory, elliptic curves, ...), but it also contains a large number of other useful functions to compute with mathematical entities such as matrices, polynomials, power series, algebraic numbers, etc., and many transcendental functions.

- PARI is a C library, allowing fast computations.
- GP is an easy-to-use interactive shell giving access to the PARI functions.
- gp2c, the GP-to-C compiler, combines the best of both worlds by compiling GP scripts to the C language and transparently loading the resulting functions into GP; scripts compiled by gp2c will typically run three to four times faster.
- Version of PARI/GP: 2.5.3
- Version of gp2c: 0.0.7pl4
- License: GPL v2+
- Programming language: C

5.2. GNU MPC

Participants: Andreas Enge [correspondant], Mickaël Gastineau, Philippe Théveny, Paul Zimmermann [INRIA project-team CARAMEL].

<http://mpc.multiprecision.org/>

GNU MPC is a C library for the arithmetic of complex numbers with arbitrarily high precision and correct rounding of the result. It is built upon and follows the same principles as GNU MPFR.

It is a prerequisite for the GNU compiler collection GCC since version 4.5, where it is used in the C and Fortran frontends for constant folding, the evaluation of constant mathematical expressions during the compilation of a program. Since 2011, it is an official GNU project.

2011 has seen the first release of the major version 1.0.

- Version: 1.0.1 *Fagus silvatica*
- License: LGPL v3+
- ACM: G.1.0 (Multiple precision arithmetic)
- AMS: 30.04 Explicit machine computation and programs
- APP: Dépôt APP le 2003-02-05 sous le numéro IDDN FR 001 060029 000 R P 2003 000 10000
- Programming language: C

5.3. MPFR CX

Participant: Andreas Enge.

<http://mpfrcx.multiprecision.org/>

MPFRGX is a library for the arithmetic of univariate polynomials over arbitrary precision real (MPFR) or complex (MPC) numbers, without control on the rounding. For the time being, only the few functions needed to implement the floating point approach to complex multiplication are implemented. On the other hand, these comprise asymptotically fast multiplication routines such as Toom-Cook and the FFT.

- Version: 0.4.1 *Cassava*
- License: LGPL v2.1+
- Programming language: C

5.4. CM

Participant: Andreas Enge.

<http://cm.multiprecision.org/>

The CM software implements the construction of ring class fields of imaginary quadratic number fields and of elliptic curves with complex multiplication via floating point approximations. It consists of libraries that can be called from within a C program and of executable command line applications. For the implemented algorithms, see [9].

- Version: 0.2 *Blindhühnchen*
- License: GPL v2+
- Programming language: C

5.5. AVIsogenies

Participants: Damien Robert [correspondant], Gaëtan Bisson, Romain Cosset [INRIA project-team CARAMEL].

<http://avisogenies.gforge.inria.fr/>

AVISOGENIES (Abelian Varieties and Isogenies) is a MAGMA package for working with abelian varieties, with a particular emphasis on explicit isogeny computation.

Its prominent feature is the computation of (ℓ, ℓ) -isogenies between Jacobian varieties of genus-two hyperelliptic curves over finite fields of characteristic coprime to ℓ ; practical runs have used values of ℓ in the hundreds.

It can also be used to compute endomorphism rings of abelian surfaces, and find complete addition laws on them.

- Version: 0.6
- License: LGPL v2.1+
- Programming language: Magma

5.6. Cubic

Participant: Karim Belabas.

<http://www.math.u-bordeaux1.fr/~belabas/research/software/cubic-1.2.tgz>

CUBIC is a standalone program that prints out generating equations for cubic fields of either signature and bounded discriminant. It depends on the PARI library. The algorithm has quasi-linear time complexity in the size of the output.

- Version: 1.2
- License: GPL v2+
- Programming language: C

5.7. Euclid

Participant: Pierre Lezowski.

<http://www.math.u-bordeaux1.fr/~plezowsk/euclid/index.php>

EUCLID is a C program to compute the Euclidean minimum of a number field. It uses the PARI library.

- Version: 1.0
- License: GPL v2+
- Programming language: C

5.8. KleinianGroups

Participant: Aurel Page.

<http://www.normalesup.org/~page/Recherche/Logiciels/logiciels.html>

KLEINIANGROUPS is a Magma package that computes fundamental domains of arithmetic Kleinian groups.

- Version: 1.0
- License: GPL v3+
- Programming language: Magma

LOGNET Team

4. Software

4.1. myMed

Our flagship software is called myMed. myMed is a highly innovative project in which three main orthogonal components are brought together:

- a software development kit, SDKmyMed, with which we can build social networks in “rush time”;
- a novel distributed hosting cloud, CLOUDmyMed, with which the social applications (developed by us and by third parties) can be hosted and run;
- a pull of 5-10 social network applications, aka “sociapps” developed in our team to test the SDKmyMed.

The sociapp can be enjoyed in almost all platforms, from web browsers, to mobile web, until IOS and Android devices.

4.2. myMed backbone

Participants: Luigi Liquori [contact], The myMed Engineer Team.

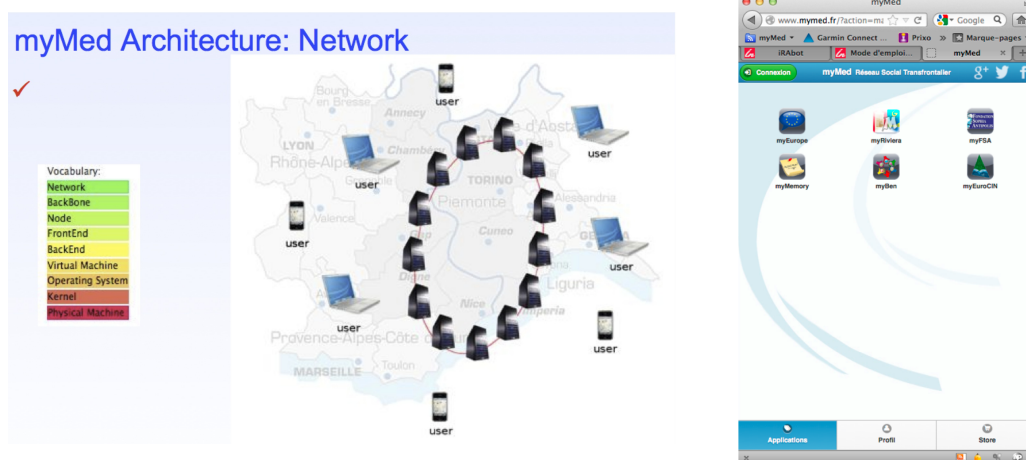


Figure 4. The myMed backbone and the myMed LaunchPad

We have implemented a “backbone” for the myMed social network using a nosql database called Cassandra <http://cassandra.apache.org>, the latter used also by social networks like Facebook and Twitter. The backbone relies on 50 PC quad code HP400, equipped with 2Tb of hard drive each.

4.3. myMed frontend

Participants: Luigi Liquori [contact], The myMed Engineer Team.

We have implemented a front-end with which all the social application can be used and downloaded via a “store” mechanism similar to the ones of Apple and Google stores. Social applications can be chosen, voted for via a reputation system, and uninstalled (including all personal data) if the user wants. We have also implemented a “template” allowing to build “proofs-of-concept” of social networks in a very short time.

4.4. Synapse simulator in Oversim

Participant: Vincenzo Ciancaglini [contact].

Synapse-Oversim is an implementation of the Synapse overlay interconnection protocol in the Oversim overlay simulator. The software presents two main contributions: first of all, a fork of the original Oversim simulator has been implemented in order to support running multiple protocol modules in a single instance of Oversim, a necessary feature in order to simulate a set of heterogeneous interconnected networks. Secondly, the whole Synapse protocol has been implemented on top of Oversim, in order to allow for the efficient inter-routing of messages between heterogeneous overlays. The Synapse code has been developed in C++, by running in Oversim, its correctness and its performances can be evaluated, while then the code can be easily ported to a real-world application.

4.5. Synapse model Erlang validator

Participant: Vincenzo Ciancaglini [contact].

During the work on the Synapse protocol, we devised a mathematical model which would allow us to estimate performance indexes of an interconnected system without having to deploy a full-scale experiment. In order to be validated, however, the model results needed to be verified against some simulation results, run under simplified conditions, but with the highest possible number of nodes. To achieve this, a dedicated simulator has been developed using Erlang, a programming language dedicated to parallel and distributed applications, which allow for the simulation of extreme systems, with a number of nodes beyond one million, in the fastest way achievable, by fully exploiting the multicore architecture of modern machines. The simulator instantiates a lightweight thread for each node, and the communication are rendered by message passing between the different node threads, thus keeping the simulation conditions as close as possible to a real world behavior.

4.6. CCN-TV Omnet++ simulator

Participant: Vincenzo Ciancaglini [contact].

CCN-TV-SIM is a software, based on the network simulation framework Omnet++, which simulates a real time video broadcast system over content-centric networks. The system is able to manage multiple streams of video at different rates, using real video traces, simulate different caching policies, different channels being transmitted concurrently, background network traffic, and different channel switch rates. Furthermore it can exploits network topologies taken from real networks, like the Deutsche Telecom network, or the Geant.

4.7. Java implementation of the OGP protocol and the experiment controller

Participant: Hoang Giang Ngo [contact].

OGP-Experiment contains Java implementation of the OGP protocol (OGP stands for overlay gateway protocol) which is used for inter-routing between heterogeneous overlay networks, and a Java implementation of the experiment controller, which is responsible for scheduling, managing and monitoring the statistics of the experiments. The software supports experiments in churn and no-churn environments. Performance metrics of the OGP protocol, such as the latency, the successful rate of data lookup and the traffic generated by a peer are reported. The experiments are performed on the Grid 5000 platform. Heterogeneous overlays which are connected by OGP can be easily plugged into the software.

4.8. Java implementation of the Synapse protocol and the experiment controller

Participant: Hoang Giang Ngo [contact].

Synapse-Experiment contains Java implementation of the Synapse overlay interconnection protocol and Java implementation of the experiment controller which is responsible for scheduling, managing and monitoring the statistics of the experiments. The software supports experiments in churn and no-churn environments. Performance metrics of the Synapse protocol, such as the latency, the successful rate of data looking up and the traffic generated by a peer are reported. The experiments are performed on the Grid 5000 platform.

4.9. Reputation Computation Engine for Social Web Platforms

Participants: Thao Nguyen [contact], Laurent Vanni.

Among the three components of a Trust and Reputation System, information gathering is most dependent on the application system, followed by the decision support component and then by the building of a robust Reputation Computation Engine and an experimental GUI, showing how bad users are segregated by the engine. To simulate the working of the reputation engine, we set up a population of N_u users, providing the same service, and undertaking N_t transactions. In each transaction, a random consumer is assigned to request the service. Other users will then be candidate providers for this request. When a user plays the role of a consumer, his behavior is modeled in the `raterType` attribute. Three types of raters include HONEST, DISHONEST and COLLUSIVE. HONEST raters share their personal experience honestly, i.e. $R_r = E_p$. DISHONEST raters provide ratings 0:5 different from their true estimation, i.e. $R_r = E_p \pm 0:5$. COLLUSIVE raters give the highest ratings ($R_r = 1$) to users in their collusion and the lowest ratings ($R_r = 0$) to the rest. Similarly, when a user acts as a provider, he can be one of the following types of providers: GOOD, NORMAL, BAD, or GOODTURNBAD. This type is denoted in `providerType` attribute. The QoS of the service provided by a BAD, NORMAL, or GOOD provider has a value in the interval $(0; 0:4]$, $(0:4; 0:7]$, or $(0:7; 1]$ respectively. A GOODTURNBAD provider will change the QoS of his service when 50% of N_t transactions have been done in the simulation. To get a transaction done, a consumer obtains a list of providers, computes reputation scores for them, chooses a provider to perform the transaction, updates his private information, and publishes his rating for the provider. The quality of service that the consumer will experience depends on the `providerType` of the chosen provider. The difference between the consumer's rating for the provider and his observation depends on the consumer's `raterType`.

To run a simulation, the user must specify 10 parameters as described above: `Simulation(Nu, Nt, %G, %N, %B, %GTB, %H, %D, %C, %dataLost)`. The simulator has been published in [22].

4.10. Ariwheels

Participants: Luigi Liquori [contact for the *Ariwheels* simulator], Claudio Casetti [Politecnico di Torino, Italy], Diego Borsetti [Politecnico di Torino, Italy], Carla-Fabiana Chiasserini [Politecnico di Torino, Italy], Diego Malandrino [Politecnico di Torino, Italy, contact for the *Ariwheels* client].

Ariwheels is an info-mobility solution for urban environments, with access points deployed at both bus stops (forming thus a wired backbone) and inside the buses themselves. Such a network is meant to provide connectivity and services to the users of the public transport system, allowing them to exchange services, resources and information through their mobile devices. *Ariwheels* is both:

- a protocol, based on *Arigatoni* and the publish/subscribe paradigm;
- a set of applications, implementing the protocol on the different types of nodes;
- a simulator, written in OMNET++ and recently ported to the ns2 simulator, see Fig 6 .

See the web page <http://www-sop.inria.fr/members/Luigi.Liquori/ARIGATONI/Ariwheels.htm> and <http://arigt.altevista.org>.

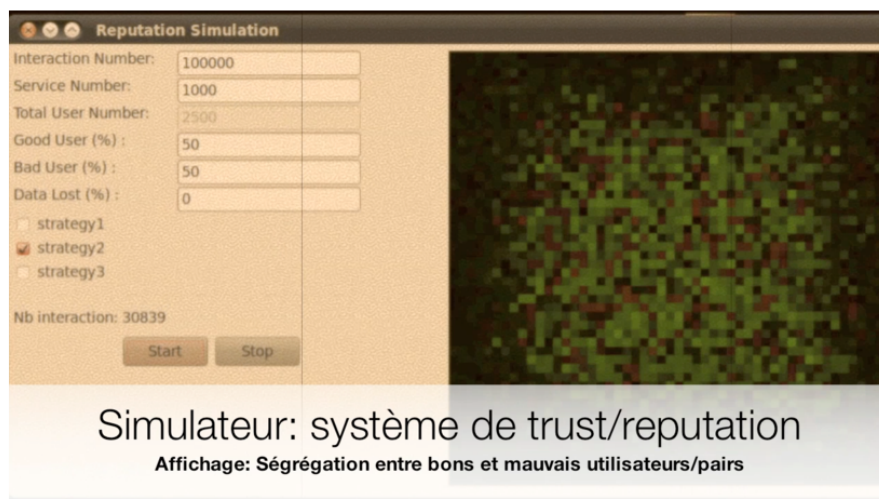


Figure 5. A prototype graphical GUI for the Reputation Computation Engine

4.11. Arigatoni simulator

Participants: Luigi Liquori [contact], Raphael Chand [Université de Geneva, Switzerland].

We have implemented in C++ (~2.5K lines of code) the Resource Discovery Algorithm and the Virtual Intermittent Protocol of the Arigatoni Overlay Network. The simulator was used to measure the load when we issued n service requests at Global Computers chosen uniformly at random. Each request contained a certain number of instances of one service, also chosen uniformly at random. Each service request was then handled by the Resource Discovery mechanism of Arigatoni networks.

4.12. Synapse client

Participants: Laurent Vanni [contact], Luigi Liquori, Cédric Tedeschi, Vincenzo Ciancaglini.

In order to test our Synapse protocol [17] on real platforms, we have initially developed JSynapse, a Java software prototype, which uses the Java RMI standard for communication between nodes, and whose purpose is to capture the very essence of our Synapse protocol. It is a flexible and ready-to-be-plugged library which can interconnect any type of overlay networks. In particular, JSynapse fully implements a Chord-based inter-overlay network. It was designed to be a lightweight and easy-to-extend software. We also provided some practical classes which help in automating the generation of the inter-overlay network and the testing of specific scenarios. We have experimented with JSynapse on the Grid'5000 platform connecting more than 20 clusters on 9 different sites. Again, Chord was used as the intra-overlay protocol. See, <http://www-sop.inria.fr/teams/lognet/synapse-net2012/>.

4.13. Open Synapse client

Participant: Bojan Marinkovic [contact].

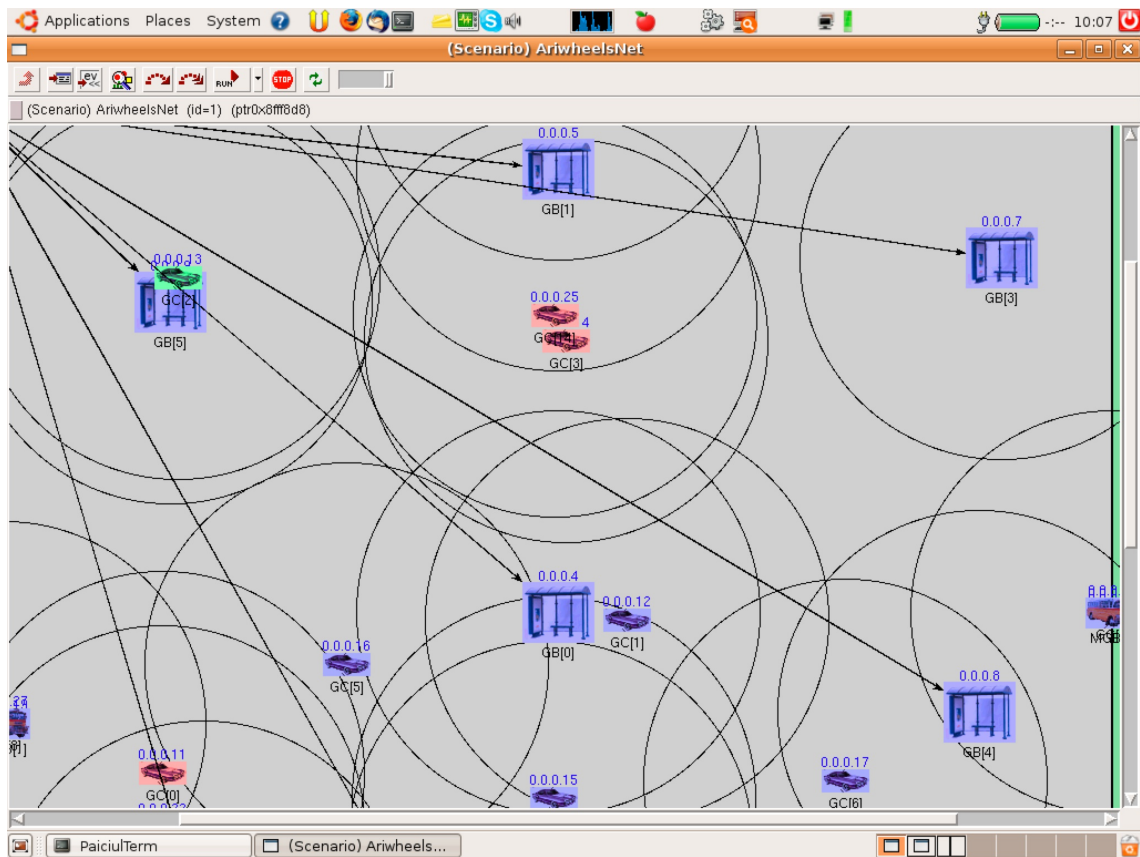


Figure 6. The Ariwheels simulator in Omnet

```

mascotte:~/ARIGATONI/CODE - Konsole
Session Edit View Bookmarks Settings Help

./a.out: /usr/local/lib/libstdc++.so.6: no version information available (required by ./a.out)
./a.out: /usr/local/lib/libstdc++.so.6: no version information available (required by ./a.out)
-t topology_1.net -s 3 -x 0 -r 0 -a 750 -S 128 -p 1234 -N 10000 -H 1 -g 100 -I 10
100%
ALL CORRECT
AVERAGE # PEERS INVOLVED PER REQUEST (GBUs ONLY | GBUs & GCUs): 19.7828 26.6448
AVERAGE # REQ MESSAGES PER REQUEST (GBUs ONLY | GBUs & GCUs): 20.6098 27.4718
AVERAGE # MESSAGES (ALL TYPES) PER REQUEST (GBUs ONLY | GBUs & GCUs): 68.0267 74.8887
MEAN # REQUESTS FOR DEPTH 0: 10000
MEAN # REQUESTS FOR DEPTH 1: 4131
MEAN # REQUESTS FOR DEPTH 2: 2067.71
MEAN # REQUESTS FOR DEPTH 3: 868.589
MEAN # REQUESTS FOR DEPTH 4: 506.055
MEAN # REQUESTS FOR DEPTH 5: 404.25
MEAN # REQUESTS FOR DEPTH 6: 313.842
MEAN # REQUESTS FOR DEPTH 7: 188.875
MEAN # REQUESTS FOR DEPTH 8: 0
MEAN: 951.096 MIN: 0 MAX: 10000 (ID2) STD DEV: 1625.7 MEDIAN: 43
ACCEPT RATE: 0.513175 REJECT RATE: 0.486825 UNARY ACCEPT RATE: 0.991244 UNARY REJECT RATE: 0.00875567
$

```

Figure 7. The Arigatoni simulator

Opensynapse is an open source implementation of [17]. It is available for free under the GNU GPL. This implementation is based on Open Chord (v. 1.0.5) - an open source implementation of the Chord distributed hash table implementation by Distributed and Mobile Systems Group Lehrstuhl fuer Praktische Informatik Universitaet Bamberg, see <http://www-sop.inria.fr/teams/lognet/synapse-net2012/>.

Opensynapse is implemented on top of an arbitrary number of overlay networks. Inter-networking can be built on top of Synapse in a very efficient way. Synapse is based on co-located nodes playing a role that is reminiscent of neural synapses. The current implementation of Opensynapse in this precise case interconnects many Chord overlay networks. The new client currently can interconnect an arbitrary number of Chord networks. This implementation follows the notation presented in [16], and so, each new Chord network is called a *Floor*.

4.14. Husky interpreter

Participants: Marthe Bonamy [contact], Luigi Liquori.

Husky is a variable-less language based on lambda calculus and term rewriting systems. Husky is based on the version 1.1 of *Snake* [13]. It was completely rewritten in CAML by Marthe Bonamy, ENSL (new parser, new syntactic constructions, like, *e.g.*, guards, anti-patterns, anti-expressions, exceptions and parametrized pattern matching). In *Husky*, all the keywords of the language are ASCII-symbols. It could be useful for teaching basic algorithms and pattern-matching to children.

4.15. myTransport Gui

Participants: Laurent Vanni [contact], Vincenzo Ciancaglini, Liquori Liquori.

myTransport is a GUI built on top of the Synapse protocol and network. Its purpose is to be a proof of concept of the future service of info-mobility to be available in the myMed social Network, see Figure 9 . The GUI is written in Java and it is fully functional in the Nokia N800 Internet tablet devices. myTransport has been ported to the myMed social network.

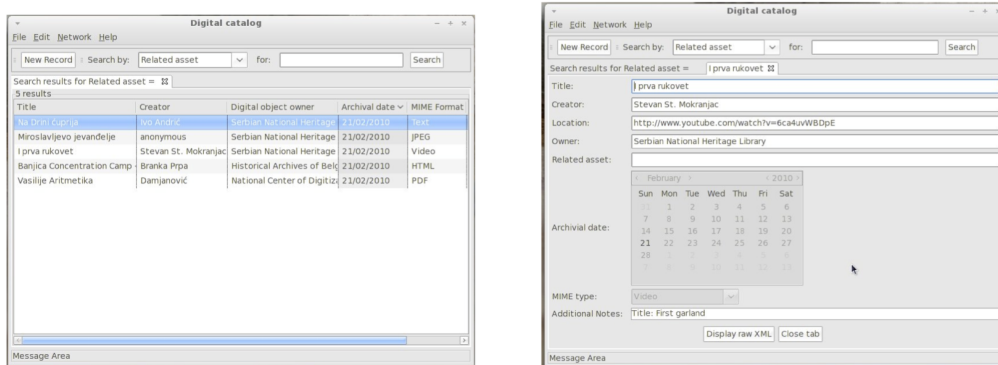


Figure 10. myDistributed Catalog

4.16. myDistributed Catalog for Digitized Cultural Heritage

Participants: Vincenzo Ciancaglini [contact], Bojan Marinkovic [MISANU, Serbia], Liquori Liquori.

Peer-to-peer networks have emerged recently as a flexible decentralized solution to handle large amount of data without the use of high-end servers. We have implemented a distributed catalog built up on an overlay network called “Synapse”. The Synapse protocol allows interconnection of different overlay networks each of them being an abstraction of a “community” of virtual providers. Data storage and data retrieval from different kind of content providers (i.e. libraries, archives, museums, universities, research centers, etc.) can be stored inside one catalog. We illustrate the concept based on the Synapse protocol: a catalog for digitized cultural heritage of Serbia, see Figure 10.

4.17. myStreaming P2P

Participants: Vincenzo Ciancaglini [contact], Rossella Fortuna [Politech Bari], Salvatore Spoto [Univ. Turin], Liquori Liquori, Luigi Alfredo Grieco [Politech Bari].

We have implemented, in Python, a fork of Goalbit <http://goalbit.sourceforge.net>, an open source video streaming platform peer-to-peer software streaming platform capable of distributing high-bandwidth live video content to everyone preserving its quality. We have aligned with the classical gossip-based distribution protocol a *DHT* that distribute contents according to a content-based strategy.

MACS Project-Team

5. Software

5.1. FELISCE

Participants: Dominique Chapelle, Sébastien Gilles [correspondant], Philippe Moireau.

FELISCE – standing for “Finite Elements for Life SCiences and Engineering” – is a new finite element code which the MACS and REO teams have decided to jointly develop in order to build up on their respective experiences concerning finite element simulations. One specific objective of this code is to provide in a unified software environment all the state-of-the-art tools needed to perform simulations of the complex cardiovascular models considered in the two teams – namely involving fluid and solid mechanics, electrophysiology, and the various associated coupling phenomena. FELISCE is written in C++, and may be later released as an opensource library. <https://gforge.inria.fr/projects/felisce/>

5.2. HeartLab

Participants: Matthieu Caruel, Dominique Chapelle, Alexandre Imperiale, Philippe Moireau [correspondant].

The heartLab software is a library written in (64 bits compatible) Matlab and C (mex functions) designed to perform both simulation and estimation (based on various types of measurements, e.g. images) of the heart mechanical behavior. Started in 2006, it is already quite large (about 60,000 lines), and is used within the CardioSense3D community.

The code relies on OpenFEM for the finite element computations, and the implementation was performed with a particular concern for modularity, since modeling and estimation use the same finite element operators. This modularity also allows to couple the code with other FEM solvers, such as LifeV and Mistral developed in the Reo team-project. In particular, we are now able to include perfusion and electrical coupling with LifeV using PVM, and fluid-structure interaction using Mistral.

We also included geometric data and tools in the code to define heart anatomical models compatible with the simulation requirements in terms of mesh quality, fiber direction data defined within each element, and referencing necessary for handling boundary conditions and estimation, in particular. These geometries are analytical or come from computerized tomography (CT) or magnetic resonance (MR) image data of humans or animals.

We recently incorporated numerous non-linear data assimilation observation operators based on medical imaging post-processing to be able to now perform estimation with a large variety of medical imaging modalities.

5.3. MITCNL

Participants: Dominique Chapelle [correspondant], Marina Vidrascu [REO team].

The package MITCNL is a set of subroutines that implements the triangular MITC3, MITC6 and quadrilateral MITC4 and MITC9 shell elements for large displacements [2]. We use it as a basis for new developments of shell elements, in particular within Modulef (<http://www-rocq.inria.fr/modulef/>). It can be easily interfaced with most finite element codes as well. We also license this package to some of our partners for use with their own codes.

5.4. OpenFEM: a Finite Element Toolbox for Matlab and Scilab

Participants: Dominique Chapelle, Philippe Moireau [correspondant].

OpenFEM (<http://www.openfem.net>) is an *opensource* finite element toolbox for linear and nonlinear structural mechanics within the Matlab and Scilab matrix computing environments. This software is developed in a collaboration between Macs and the SDTools company ¹. Performing finite element analyses within a matrix computing environment is of considerable interest, in particular as regards the ease of new developments, integration of external software, portability, post-processing, etc.

This Library is the core of the finite element computations of HeartLab where a specific version have been developed with the help of Cesare Corrado from Reo.

5.5. SHELDDON

Participants: Dominique Chapelle [correspondant], Marina Vidrascu [REO team].

SHELDDON (SHELLs and structural Dynamics with DOrain decomposition in Nonlinear analysis) is a finite element library based on the Modulef package which contains shell elements, nonlinear procedures and PVM subroutines used in domain decomposition or coupling methods.

5.6. Verdandi

Participants: Dominique Chapelle, Marc Fragu, Vivien Mallet [Clime team], Philippe Moireau [correspondant].

Verdandi is an *opensource* (LGPL) software library aiming at providing assimilation data methods and related tools. Mainly targeted at large systems arising from the discretization of PDEs, it is intentionally devised as generic, which allows for applications in a wide range of problems (biology and medicine, environment, image processing...). See also the web page <http://verdandi.gforge.inria.fr/>, with a complete documentation in English. The first stable version (1.0) was released in June 2012 and contains most of the major data assimilation algorithms of both variational and sequential types. The actual version (1.4) contains additional estimation algorithm and parallel capabilities. Note that some specific developments are performed with particular regard to cardiac modeling applications, as Verdandi is partly funded by – and distributed within – the euHeart project and is now referenced in the following peer-reviewed article [15]

- ACM: Mathematical software
- AMS: System theory; control
- Software benefit: Verdandi is the only *generic* data assimilation library
- License: LGPL (2.1 or any later version)
- Type of human computer interaction: Command line and configuration files
- OS/Middleware: Linux, MacOS ou Windows
- Required library or software: Seldon (LGPL, <http://seldon.sourceforge.net/>)
- Programming language: C++, ISO/IEC 14882: I998(E) Python, version 2.6
- Documentation: Doxygen and utilisation manual in english

¹<http://www.sdtools.com>

MADYNES Project-Team

5. Software

5.1. SecSIP

Participants: Abdelkader Lahmadi [contact], Olivier Festor.

*SecSip*¹ is developed by the team to defend SIP-based (The Session Initiation Protocol) services from known vulnerabilities. It presents a proactive point of defense between a SIP-based network of devices (servers, proxies, user agents) and the open Internet. Therefore, all SIP traffic is inspected and analyzed against authored Veto specification before it is forwarded to these devices. When initializing, the SecSIP runtime starts loading and parsing authored VeTo blocks to identify different variables, event patterns, operations and actions from each rule. Veto is a generic declarative language for attack patterns specification. SecSIP implements an input and output layer, to capture, inject, send and receive SIP packets from and to the network. Intercepted packets are moved to the SIP Packet parser module. The main function of this module is to extract different fields within a SIP message and trigger events specified within the definition blocks. During each execution cycle when a SIP message arrives, the SecSIP runtime uses a data flow acyclic graph network to find definition matching rules and triggers defined events. The paired events in each operator node are propagated over the graph until a pattern is satisfied. When the pattern is satisfied, the respective rule is fired and the set of actions is executed.

SecSIP is freely available on the Internet. It was extended to support new protocols in the area of SCADA systems in 2012.

5.2. NDPMon

Participants: Isabelle Chrisment, Olivier Festor [contact].

The Neighbor Discovery Protocol Monitor (**NDPMon**) is an IPv6 implementation of the well-known ArpWatch tool. NDPMon monitors the pairing between IPv6 and Ethernet addresses (NDP activities: new station, changed Ethernet address, flip flop...). NDPMon also detects attacks on the NDP protocol, as defined in RFC 3756 (bogon, fake Router Advertisements...). New attacks based on the Neighbor Discovery Protocol and Address Auto-configuration (RFC 2461 and RFC 2462) have been identified and integrated in the tool. An XML file describes the default behavior of the network, with the authorized routers and prefixes, and a second XML document containing the neighbors database is used. This second file can be filled during a learning phase. All NDP activities are logged in the syslog utility, and so the attacks, but these ones are also reported by mail to the administrator. Finally, NDPMon can detect stack vulnerabilities, like the assignment of an Ethernet broadcast address on an interface.

NDPMon comes along with a WEB interface acting as a GUI to display the informations gathered by the tool, and give an overview of all alerts and reports. Thanks to color codes, the WEB interface makes possible for the administrator to have an history of what happened on his network and identify quickly problems. All the XML files used or produced by the daemon (neighbor cache, configuration file and alerts list) are translated in HTML via XSL for better readability. A statistic module is also integrated and gives informations about the discovery of the nodes and their type (MAC manufacturer distribution ...).

The software package and its source code is freely distributed under an opensource license (LGPL). It is implemented in C, and is available through a SourceForge project at <http://ndpmon.sf.net>. An open source community is now established for the tool which has distributions for several Operating Systems (Linux, FreeBSD, OpenBSD, NetBSD and Mac OS X). It is also integrated in FreeBSD ports at <http://www.freebsd.org/cgi/cvsweb.cgi/ports/net-mgmt/ndpmon/>. Binary distributions are also available for .deb and .rpm based Linux flavors.

In 2012, the software underwent a complete reshaping thanks to a substantial support from the High Security Lab which dedicated us 6 months of research engineer.

¹<http://secsip.gforge.inria.fr/doku.php>

MAESTRO Project-Team (section vide)

MAGIQUE-3D Project-Team

5. Software

5.1. Hou10ni

This software, written in FORTRAN 90, simulates the propagation of acoustic waves in heterogeneous 2D and 3D media. It is based on an Interior Penalty Discontinuous Galerkin Method (IPDGM). The 2D version of the code has been implemented in the Reverse Time Migration (RTM) software of TOTAL in the framework of the Ph.D thesis of Caroline Baldassari. The 2D code allows for the use of meshes composed of cells of various order (p -adaptivity in space). For the time discretization, we used the local time stepping strategy described at section 3.2, item **High-Order Schemes in Space and Time** which permits not only the use of different time-step, but also to adapt the order of the time-discretization to the order of each cells (hp -adaptivity in time).

The main competitors of Hou10ni are codes based on Finite Differences, Spectral Element Method or other Discontinuous Galerkin Methods (such as the ADER schemes). During her Ph.D thesis, Caroline Baldassari compared the solution obtained by Hou10ni to the solution obtained by a Finite Difference Method and by a Spectral Element Method (SPECFEM). To evaluate the accuracy of the solutions, we have compared them to analytical solutions provided by the codes Gar6more (see below). The results of these comparisons is: a) that Hou10ni outperforms the Finite Difference Methods both in terms of accuracy and of computational burden and b) that its performances are similar to Spectral Element Methods. Since Hou10ni allows for the use of meshes based on tetraedrons, which are more appropriate to mesh complex topographies, and for the p -adaptivity, we decided to implement it in the RTM code of TOTAL. Of course, we also used these comparisons to validate the code. Now, it remains to compare the performances of Hou10ni to the ADER schemes.

Recently, we have extended the 2D version of Hou10ni for computing the solution of the harmonic wave equation (Helmholtz). This new version is able to deal with both acoustic and elastodynamic media, but also to model elastoacoustic problems. The surfaces between the different media can be approximated by curved elements. We can use up to P^{15} elements when dealing with curved elements and element of arbitrary order (with of course a limitation depending on the machine precision) when dealing with non-curved elements.

5.2. Gar6more3D

Participant: Julien Diaz [correspondant].

This code computes the analytical solution of problems of waves propagation in two layered 3D media such as-acoustic/acoustic- acoustic/elastodynamic- acoustic/porous- porous/porous, based on the Cagniard-de Hoop method.

See also the web page <http://web.univ-pau.fr/~jdiaz1/software.html>.

The main objective of this code is to provide reference solutions in order to validate numerical codes. They have been already used by J. Tromp and C. Morency to validate their code of poroelastic wave propagation [87]. They are freely distributed under a CECILL licence and can be downloaded on the website <http://web.univ-pau.fr/~jdiaz1/software.html>. As far as we know, the main competitor of this code is EX2DELDEL (available on <http://www.spice-rtn.org>), but this code only deals with 2D acoustic or elastic media. Our codes seem to be the only one able to deal with bilayered poroelastic media and to handle the three dimensional cases.

- ACM: J.2
- AMS: 34B27 35L05 35L15 74F10 74J05
- Programming language: Fortran 90

MAGNOME Project-Team

5. Software

5.1. Inria Bioscience Resources

Participants: Olivier Collin [correspondant], Frédéric Cazals, Mireille Régnier, Marie-France Sagot, Hélène Touzet, Hidde de Jong, David James Sherman, Marie-Dominique Devignes, Dominique Lavenier.

Inria Bioscience Resources is a portal designed to improve the visibility of bioinformatics tools and resources developed by Inria teams. This portal will help the community of biologists and bioinformaticians understand the variety of bioinformatics projects in Inria, test the different applications, and contact project-teams. Eight project-teams participate in the development of this portal. Inria Bioscience Resources is developed in an Inria Technology Development Action (ADT).

5.2. Magus: Collaborative Genome Annotation

Participants: David James Sherman [correspondant], Pascal Durrens, Natalia Golenetskaya, Florian Lajus, Tiphaine Martin.

As part of our contribution to the Génolevures Consortium, we have developed over the past few years an efficient set of tools for web-based collaborative annotation of eukaryote genomes. The MAGUS genome annotation system integrates genome sequences and sequence features, *in silico* analyses, and views of external data resources into a familiar user interface requiring only a Web navigator. MAGUS implements the annotation workflows and enforces curation standards to guarantee consistency and integrity. As a novel feature the system provides a workflow for *simultaneous annotation* of related genomes through the use of protein families identified by *in silico* analyses; this has resulted in a three-fold increase in curation speed, compared to one-at-a-time curation of individual genes. This allows us to maintain Génolevures standards of high-quality manual annotation while efficiently using the time of our volunteer curators.

MAGUS is built on: a standard sequence feature database, the Stein lab generic genome browser [61], various biomedical ontologies (<http://obo.sf.net>), and a web interface implementing a representational state transfer (REST) architecture [39].

For more information see magus.gforge.inria.fr, the MAGUS Gforge web site. MAGUS is developed in an Inria Technology Development Action (ADT).

5.3. YAGA: Yeast Genome Annotation

Participants: Tiphaine Martin, Pascal Durrens [correspondant], Elisabeth Bon, Aurélie Goulielmakis.

With the arrival of new generations of sequencers, laboratories, at a lower cost, can be sequenced groups of genomes. You can no longer manually annotate these genomes. The YAGA (Yeast Automatic Genome Annotation) software's objective is to annotate a raw sequence syntactically and functionally as well as generate EMBL files for publication. The annotation takes into account data from comparative genomics, such as protein family profiles.

After determining the constraints of the annotation, the YAGA software can automatically annotate *de novo* all genomes from their raw sequences. The predictors used by the YAGA software can also take into account the data RNAseq to reinforce the prediction of genes. The current settings of the software are intended for annotation of the genomes of yeast, but the software is adaptable for all types of species, and has been trained and used for the annotation of bacterial genomes.

5.4. BioRica: Multi-scale Stochastic Modeling

Participants: David James Sherman [correspondant], Rodrigo Assar Cuevas.

BioRica is a high-level modeling framework integrating discrete and continuous multi-scale dynamics within the same semantics field. A model in BioRica node is hierarchically composed of nodes, which may be existing models. Individual nodes can be of two types:

- Discrete nodes are composed of states, and transitions described by constrained events, which can be non deterministic. This captures a range of existing discrete formalisms (Petri nets, finite automata, etc.). Stochastic behavior can be added by associating the likelihood that an event fires when activated. Markov chains or Markov decision processes can be concisely described. Timed behavior is added by defining the delay between an event's activation and the moment that its transition occurs.
- Continuous nodes are described by ODE systems, potentially a hybrid system whose internal state flows continuously while having discrete jumps.

The system has been implemented as a distributable software package

The BioRica compiler reads a specification for hierarchical model and compiles it into an executable simulator. The modeling language is a stochastic extension to the AltaRica Dataflow language, inspired by work of Antoine Rauzy. Input parsers for SBML 2 version 4 are currently being validated. The compiled code uses the Python runtime environment and can be run stand-alone on most systems [40].

For more information see biorica.gforge.inria.fr, the BioRica Gforge web site. BioRica was developed as an Inria Technology Development Action (ADT).

5.5. Pathtastic: Inference of whole-genome metabolic models

Participants: David James Sherman [correspondant], Pascal Durrens, Nicolás Loira, Tiphaine Martin, Anna Zhukova.

Pathtastic is a software tool for inferring whole-genome metabolic models for eukaryote cell factories. It is based on *metabolic scaffolds*, abstract descriptions of reactions and pathways on which inferred reactions are hung are eventually connected by an iterative mapping and specialization process. Scaffold fragments can be repeatedly used to build specialized subnetworks of the complete model.

Pathtastic uses a consensus procedure to infer reactions from complementary genome comparisons, and an algebra for assisted manual editing of pathways.

For more information see pathtastic.gforge.inria.fr, the Pathtastic Gforge web site.

5.6. Génolevures On Line: Comparative Genomics of Yeasts

Participants: Pascal Durrens [correspondant], Natalia Golenetskaya, Tiphaine Martin, David James Sherman.

The Génolevures online database provides tools and data for exploring the annotated genome sequences of more than 20 genomes, determined and manually annotated by the Génolevures Consortium to facilitate comparative genomic studies of hemiascomycetous yeasts. Data are presented with a focus on relations between genes and genomes: conservation of genes and gene families, speciation, chromosomal reorganization and synteny. The Génolevures site includes an area for specific studies by members of its international community.

Génolevures online uses the MAGUS system for genome navigation, with project-specific extensions developed by David Sherman, Pascal Durrens, and Tiphaine Martin. An advanced query system for data mining in Génolevures is being developed by Natalia Golenetskaya. The contents of the knowledge base are expanded and maintained by the CNRS through GDR 2354 Génolevures. Technical support for Génolevures On Line is provided the CNRS through UMR 5800 LaBRI.

For more information see genolevures.org, the Génolevures web site.

MAGRIT Project-Team

5. Software

5.1. Software

Our software efforts are integrated in a library called RALib which contains our research development on image processing, registration (2D and 3D) and visualization. This library is licensed by the APP (French agency for software protection).

The visualization module is called QGLSG: it enables the visualization of images, 2D and 3D objects under a consistent perspective projection. It is based on Qt ¹ and OpenScenegraph ² libraries. The QGLSG library integrates innovative features such as online camera distortion correction, and invisible objects that can be incorporated in a scene so that virtual objects can cast shadows on real objects, and occlusion between virtual and real objects are easier to handle. The library was also ported to Mac OS and Windows and a full doxygen documentation was written.

¹<http://qt.digia.com>

²<http://www.openscenegraph.org/projects/osg>

MAIA Project-Team

5. Software

5.1. AA4MM

Participants: Vincent Chevrier [correspondant], Benjamin Camus.

Laurent Ciarletta (Madyne team, LORIA) is a collaborator and correspondant for this software.

AA4MM (Agents and Artefacts for Multi-modeling and Multi-simulation) is a framework for coupling existing and heterogeneous models and simulators in order to model and simulate complex systems. The first implementation of the AA4MM meta-model was proposed in Julien Siebert's PhD [56] and written in Java. This year we added a new coupling between models to represent multi-level modeling, and rewrote a part of the core to ease coupling of simulator.

5.2. MASDYNE

Participant: Vincent Chevrier [correspondant].

This work was undertaken in a joint PhD Thesis between MAIA and Madyne Team. Laurent Ciarletta (Madyne team, LORIA) has been co-advisor of this PhD and correspondant for this software.

Other contributors to this software were: Tom Leclerc, François Klein, Christophe Torin, Marcel Lamenu, Guillaume Favre and Amir Toly.

MASDYNE (Multi-Agent Simulator of DYnamic Networks usErs) is a multi-agent simulator for modeling and simulating users behaviors in mobile ad hoc network. This software is part of joint work with MADYNES team, on modeling and simulation of ubiquitous networks.

5.3. FiatLux

FiatLux is a discrete dynamical systems simulator that allows the user to experiment with various models and to perturb them. Its main feature is to allow users to change the type of updating, for example from a deterministic parallel updating to an asynchronous random updating. FiatLux has a Graphical User Interface and can also be launched in a batch mode for the experiments that require statistics. In 2012, the main contributions were made by Olivier Bouré, who developed a lattice-gas cellular automata module.

5.4. Cart-o-matic

Participants: Olivier Simonin [correspondant], François Charpillet, Antoine Bautin, Nicolas Beaufort.

Philippe Lucidarme (Université d'Angers, LISA) is a collaborator and the coordinator of the Cartomatic project.

Cart-o-matic is a software platform for (multi-)robot exploration and mapping tasks. It has been developed by Maia members and LISA (Univ. Angers) members during the robotics ANR/DGA Carotte challenge (2009-2012). This platform is composed of three softwares which as been protected by software copyrights (APP): Slam-o-matic a SLAM algorithm developed by LISA members, Plan-o-matic a robot trajectory planning algorithm developed by Maia and LISA members and Expl-o-matic a distributed multi-agent strategy for multi-robot exploration developed by Maia members (which is based on algorithms proposed in the PhD Thesis of Antoine Bautin). Cf. illustration at [Cart-o-matic](#)

The purchase of Cart-o-matic by some robotics companies is underway.

MANAO Team

4. Software

4.1. EIGEN

Participants: G. Guennebaud, D. Nuytsa

Keywords : Linear algebra

Efficient numerical computation is central to many computer science domains. In particular, in computer graphics, space transformations and local regressions involve dense linear algebra, data interpolation and differential equations require sparse linear algebra, while more advanced problems involve non-linear optimization or spectral analysis. On the one hand, solutions such as MatLab are limited to prototyping. On the other hand, optimized libraries coming from the HPC (high performance computing) world are often tedious to use and more adapted for very large problems running on clusters. Moreover, all these solutions are very slow at handling very small problems which often arise in computer graphics, vision, or robotics. As a result, researchers of these domains used to waste a lot of time at either implementing their own half cooked solution, or dealing with dozens of complex to use libraries.

The objective of Eigen is to fill this gap by proposing an easy to use, efficient, and versatile C++ mathematical template library for linear algebra and related algorithms. In particular it provides fixed and dynamic size matrices and vectors, matrix decompositions (LU, LLT, LDLT, QR, eigenvalues, etc.), sparse matrices and solvers, some basic geometry features (transformations, quaternions, axis-angles, Euler angles, hyperplanes, lines, etc.), some non-linear solvers, automatic differentiations, etc. Thanks to expression templates, Eigen provides a very powerful and easy to use API. Explicit vectorization is performed for the SSE, AltiVec and ARM NEON instruction sets, with graceful fallback to non-vectorized code. Expression templates allow to perform global expression optimizations, and to remove unnecessary temporary objects.

Eigen is already a well established library with about 20000 unique visitors of the website per month. Eigen is co-developed and maintained with a couple of other researchers and occasional contributors spread over the world. Its development started in 2008, and the last release is the 3.1 version in June 2012. Eigen is currently supported by Inria through an ADT started in January 2012.

Facts:

- Web: <http://eigen.tuxfamily.org/>
- License: LGPL3+

MARELLE Project-Team

4. Software

4.1. Tralics

Participant: José Grimm [correspondant].

Tralics is a Latex-to-XML translator available at <http://www-sop.inria.fr/marelle/tralics>. Version 2.15 has been released this year. Some features have been added, and some bugs corrected.

4.2. Semantics

Participant: Yves Bertot [correspondant].

This is a library for the Coq system, where the description of a toy programming language is presented. The value of this library is that it can be re-used in classrooms to teach programming language semantics or the Coq system. The topics covered include introductory notions to domain theory, pre and post-conditions, abstract interpretation, and the proofs of consistency between all these point of views on the same programming language. Standalone tools for the object programming language can be derived from this development.

See also the web page <http://coq.inria.fr/pylons/pylons/contribs/view/Semantics/v8.3>.

- ACM: F3.2 F4.1
- AMS: 68N30
- Programming language: Coq

4.3. Certicrypt and Easycrypt

Participants: Gilles Barthe [IMDEA Software Institute], Juan Manuel Crespo [IMDEA Software Institute], Benjamin Grégoire [correspondant], Sylvain Heraud, César Kunz [IMDEA Software Institute], Federico Olmedo [IMDEA Software Institute], Santiago Zanella Béguelin [IMDEA Software Institute].

CertiCrypt takes a language-based approach to cryptography: the security of a cryptographic scheme and the cryptographic assumptions upon which its security relies are expressed by means of probabilistic programs, called games; in a similar way, adversarial models are specified in terms of complexity classes, e.g. probabilistic polynomial-time programs. This code-centric view leads to statements that are amenable to formalization and tool-assisted verification. CertiCrypt instruments a rich set of verification techniques for probabilistic programs, including equational theories of observational equivalence, relational Hoare logic, data-flow analysis-based program transformations, and game-based techniques such as eager/lazy sampling and failure events.

See also the web page <http://easycrypt.gforge.inria.fr/>.

MASAIE Project-Team (section vide)

MASCOTTE Project-Team

5. Software

5.1. Grph

Participants: David Coudert, Luc Hogue [correspondant], Aurélien Lancin, Grégory Morel, Issam Tahiri.

Around 20,000 lines of code, developed in Java.

The objective of GRPH is to provide researchers and engineers a suitable graph library for graph algorithms experimentation and network simulation. GRPH is primarily a software library, but it also comes with a set of executable files for user interaction and graph format conversion; as such, it can be used autonomously. Performance and accessibility are the primary targets of the GRPH library. It allows manipulating large graphs (millions of nodes). Its model considers mixed graphs composed of directed and undirected simple- and hyper-edges. GRPH comes with a collection of base graph algorithms which are regularly augmented.

So far, known users of the GRPH library include people at Mascotte and others involved in the FP7 EULER project. It got some contribution from the Inria team GANG who contributed GRPH with an implementation of the four-sweep algorithm which provides accurate lower bound on the diameter in linear time. It has a number of other academic users including research students at Bergamo University (Italy), and University of Southern Denmark (students supervised by Jørgen Bang-Jensen).

GRPH includes bridges to other graph libraries such as JUNG, JGraphT, CORESE (a software developed by the WIMMICS team Inria-I3S), LAD (Christine Solnon, LIRIS), Nauty (Brendan D. McKay), as well as specific algorithms developed by Matthieu Latapy and Jean-Lou Guillaume (LIP6), etc.

GRPH is distributed under the terms of a license defined by its contributors and is available for download. This license allows free usage and access to the source code. See <http://www-sop.inria.fr/mascotte/software/grph>.

In 2012, numerous graph algorithms have been added to GRPH, such as maximum matching, minimum vertex cover (brute force, branching, Niedermeier), maximum independent set (Fomin/Grandoni/Kratsch). Furthermore, to answer a number of issues about the generation of graph instances with particular properties, a framework for evolutionary computing dedicated to graphs was integrated to GRPH. Moreover, a reworked version of Mascsim was integrated in GRPH.

On-going works concern the distributed execution of graph algorithms, and a bridge to Sage.

See also the web page <http://www-sop.inria.fr/mascotte/software/grph/>.

5.2. DRMSim

Participants: David Coudert, Luc Hogue [correspondant], Aurélien Lancin, Nicolas Nisse, Issam Tahiri.

Around 45,000 lines, developed in Java, collaboration between MASCOTTE and LaBRI.

DRMSim relies on a discrete-event simulation engine aiming at enabling the large-scale simulations of routing models. DRMSim is developed in the framework of the FP7 EULER project. It proposes a general routing model which accommodates any network configuration. Aside to this, it includes specific models for Generalized Linear Preference (GLP), and k-chordal network topologies, as well as implementations of routing protocols, including the routing protocol proposed in [37] and lightweight versions of BGP (Border Gateway Protocol).

The system model considers the dynamic evolution of the simulated network. This model takes as its input parameter the distribution of failure probability for both routers and links.

The metric model takes measures along a discrete-event simulation which can be performed in many ways.

Commonly, a simulation campaign consists in iterating over the set of combinations of parameter values, calling the simulation function for every combination. These combinations are most often complex, impeding their description by a set of mathematical functions. Thus DRMSim provides a simulation methodology that describes (programmatically) the way a simulation campaign should be conducted.

DRMSim stores on disk every step of the execution of a simulation campaign. In a simulation campaign, simulation runs are independent (no simulation depends on the result computed by another simulation). Consequently they can be executed in parallel. Because one simulation is most likely to use large amount of memory and to be multi-threaded, parallelizing the simulation campaign on one single computer is a poor parallelization scheme. Instead, we currently work at enabling the remote parallel execution of several simulation runs, with the same distribution framework that is used in the GRPH library.

DRMSim relies on the Mascsim abstract discrete-event simulation framework, the GRPH library and the Java4Unix integration framework.

Finally, from an object-oriented point of view of its conception model, DRMSim manipulates graph abstractions, allowing the user to force the use of a library different from the default one, i.e. GRPH.

See also the web page <http://www-sop.inria.fr/mascotte/projets/DCR/>.

5.3. SageMath

Participants: David Coudert, Leonardo Sampaio.

Developed in Python, Cython, and C++. MASCOTTE members have already contributed to the development of more than 180 patches and to the reviewing process of more than 200 patches that are now part of the standard distribution.

Sagemath is a free open-source mathematics software aiming at becoming an alternative to Maple and Matlab. Initially created by William Stein (Professor of mathematics at Washington University), Sagemath is currently developed by more than 180 contributors around the world (mostly researchers) and its source code has reached 350 MB. It is of interest for Mascotte members because it combines a large collection of graph algorithms with various libraries in algebra, calculus, combinatorics, linear programming, statistics, etc.

We use Sagemath for quickly testing algorithms, analyzing graphs, and disseminating algorithms. We also use it for teaching purposes in the Master IFI, stream UBINET.

In 2012, David Coudert has contributed to the development of the Sage releases 5.0 to 5.6 with 15 patches (from bug fix to advance graph algorithms) and participated to the reviewing process of more than 30 patches.

5.4. Utilities

5.4.1. Java4unix

Participant: Luc Hogie [correspondant].

More than 5,000 lines, developed in Java.

Java4unix proposes a development and distribution framework which simplifies the use of Java for UNIX software programming/distribution. Until now, Java could hardly be used for the development UNIX applications because invoking Java applications from the UNIX shell must be done through an explicit call to the Java virtual machine and writing simple things in Java often requires long coding. Java4unix aims at filling those two gaps by providing a UNIX installer for java applications, turning them to standard UNIX application and a framework that UNIX programmers may use to manipulate files/text, etc.

Java4unix includes a module which enables the reporting and automatic releasing of Eclipse Java projects.

See also the web page <http://www-sop.inria.fr/members/Luc.Hogie/java4unix/>.

5.4.2. Jalinopt

Participants: Luc Hogie [correspondant], Grégory Morel.

Developed in Java.

Jalinopt is a Java toolkit for building and solving linear programs. It consists of a straightforward object-oriented model for linear programs, as well as a bridge to most common solvers, including GLPK and CPLEX. It is an interface to many LP solvers allowing users to code independently of the solver effectively. Although Jalinopt is inspired by Mascopt and JavaILP, it provides a significantly different model and an utterly different approach to connecting to the solver. In particular this approach, based in inter-process piping, offers better portability, and the possibility to connect (via SSH) to solvers on remote computers.

In 2012, we refined the object-oriented model of Jalinopt and improved its portability by making it working with LPSolve as its default native solver.

See also the web page <http://www-sop.inria.fr/members/Luc.Hogie/jalinopt/>.

5.4.3. *JavaFarm*

Participant: Luc Hogie [correspondant].

More than 1,500 lines, developed in Java.

JavaFarm is a middleware enabling the distribution of Java applications across farms of servers.

Its workflow basically enables an application to locally aggregate code and data into an object, called job, that will migrate to another computer where it will be computed. When a job completes, its result is transferred back to the caller. Among other features, JavaFarm supports futures (asynchronous job executions), thereby enabling parallelization of the distributed code. The design objectives of JavaFarm are to make distribution and parallelism as transparent and easy as possible.

See also the web page <http://www-sop.inria.fr/members/Luc.Hogie/javafarm/>.

5.4.4. *Mascsim*

Participants: Luc Hogie [correspondant], Aurélien Lancin, Issam Tahiri.

Around 12,000 lines, developed in Java.

Mascsim is a distributed discrete event simulator whose main target is to be easy to use. Unlike most discrete-event simulators, the researcher who is using Mascsim is required to provide only the bare minimum material needed for the simulation: a model for the system, a set of events describing what is going on in the system, as well as a set of metrics of interest. The simulation process is then entirely automatized.

In 2012, Mascsim was adapted and integrated to GRPH.

See also the web page <http://www-sop.inria.fr/mascotte/software/mascsim/>.

5.4.5. *P2PVSIm*

Participant: Remigiusz Modrzejewski [correspondant].

Around 12,000 lines, developed in Python.

P2PVSIm is a simple discrete-event simulator created for analyzing theoretical properties of peer-to-peer live video streaming algorithms. Implemented in Python it was designed with clarity and extensibility in mind from the beginning. It is capable of simulating overlays of a few thousands of peers. Multiple control protocols have been implemented. At the same time, a lot of work was put into the performance and scalability aspects of the software. Currently it is meant for simulating overlays of a few thousand peers running multiple control protocols that have been implemented.

In 2012, a distributed version of P2PVSIm was developed. The objectives for developing a distributed version was to fasten the simulation of large campaigns, that would be too long to run on one single computer. The distributed P2PVSIm runs on an arbitrary number of computers. It has been so far used with success on a dozen computers with multiple cores all located in the same LAN.

MATHRISK Team

4. Software

4.1. PREMIA

Participants: Antonino Zanette, Mathrisk Research team, Agnès Sulem [correspondant].

Premia is a software designed for option pricing, hedging and financial model calibration. It is provided with its C/C++ source code and an extensive scientific documentation. <https://www-rocq.inria.fr/mathfi/Premia>

The Premia project keeps track of the most recent advances in the field of computational finance in a well-documented way. It focuses on the implementation of numerical analysis techniques for both probabilistic and deterministic numerical methods. An important feature of the platform Premia is the detailed documentation which provides extended references in option pricing.

Premia is thus a powerful tool to assist Research & Development professional teams in their day-to-day duty. It is also a useful support for academics who wish to perform tests on new algorithms or pricing methods without starting from scratch.

Besides being a single entry point for accessible overviews and basic implementations of various numerical methods, the aim of the Premia project is:

1. to be a powerful testing platform for comparing different numerical methods between each other;
2. to build a link between professional financial teams and academic researchers;
3. to provide a useful teaching support for Master and PhD students in mathematical finance.
 - AMS: 91B28;65Cxx;65Fxx;65Lxx;65Pxx
 - License: Licence Propriétaire (genuin license for the Consortium Premia)
 - Type of human computer interaction: Console, interface in Nsp, Web interface
 - OS/Middelware: Linux, Mac OS X, Windows
 - APP: The development of Premia started in 1999 and 14 are released up to now and registered at the APP agency.
 - Programming language: C/C++ librairie Gtk
 - Documentation: the PNL library is interfaced via doxygen
 - Size of the software: 11 Mbyte of code split into 275,000 lines. 93 Mbyte of PDF files of documentation
 - Publications: [1] [64] [78] [89] [95], [47]

4.1.1. Content of Premia

Premia contains various numerical algorithms (Finite-differences, trees and Monte-Carlo) for pricing vanilla and exotic options on equities, interest rate, credit and energy derivatives.

1. Equity derivatives:

The following models are considered:

Black-Scholes model (up to dimension 10), stochastic volatility models (Hull-White, Heston, Fouque-Papanicolaou-Sircar), models with jumps (Merton, Kou, Tempered stable processes, Variance gamma, Normal inverse Gaussian), Bates model.

For high dimensional American options, Premia provides the most recent Monte-Carlo algorithms: Longstaff-Schwartz, Barraquand-Martineau, Tsitsklis-Van Roy, Broadie-Glassermann, quantization methods and Malliavin calculus based methods.

Dynamic Hedging for Black-Scholes and jump models is available.

Calibration algorithms for some models with jumps, local volatility and stochastic volatility are implemented.

2. Interest rate derivatives

The following models are considered:

HJM and Libor Market Models (LMM): affine models, Hull-White, CIR++, Black-Karasinsky, Squared-Gaussian, Li-Ritchken-Sankarasubramanian, Bhar-Chiarella, Jump diffusion LMM, Markov functional LMM, LMM with stochastic volatility.

Premia provides a calibration toolbox for Libor Market model using a database of swaptions and caps implied volatilities.

3. Credit derivatives: CDS, CDO

Reduced form models and copula models are considered.

Premia provides a toolbox for pricing CDOs using the most recent algorithms (Hull-White, Laurent-Gregory, El Karoui-Jiao, Yang-Zhang, Schönbucher)

4. Hybrid products

PDE solver for pricing derivatives on hybrid products like options on inflation and interest or change rates is implemented.

5. Energy derivatives: swing options

Mean reverting and jump models are considered.

Premia provides a toolbox for pricing swing options using finite differences, Monte-Carlo Malliavin-based approach and quantization algorithms.

4.1.2. Premia design

Anton Kolotaev (ADT engineer), supervised by J. Lelong, has developed a web platform allowing online tests (<https://quanto.inria.fr/premia/koPremia>). This online version allow us to supply benchmarks both for professional R&D teams and academics in mathematical finance. This will considerably increase the impact and the visibility of the software. Up to now, to use the opensource version of the software, one has to download from Premia's website and install it on its own computer and this had become a brake on using Premia. Providing an online version of Premia is an original way of keeping up with the new standards of software usability without focusing too much on a dedicated solution per operation system.

To enable easy an advanced usage of Premia without being an advanced C or C++ programmer, we have started to implement Python bindings. The choice of Python has been quite obvious as Python has become over the past few years a standard cross-platform interpreted language for numerical problems.

Premia has managed to grow up over a period of more than a dozen years; this has been possible only because contributing an algorithm to Premia is subject to strict rules, which have become too stringent. To facilitate contributions, a standardized numerical library (PNL) has been developed under the LGPL since 2009, which offers a wide variety of high level numerical methods for dealing with linear algebra, numerical integration, optimization, random number generators, Fourier and Laplace transforms, and much more. Everyone who wishes to contribute is encouraged to base its code on PNL and providing such a unified numerical library has considerably eased the development of new algorithms which have become over the releases more and more sophisticated. An effort will be made to continue and stabilize the development of PNL.

4.1.3. Algorithms implemented in Premia in 2012

Premia 14 was delivered to the consortium members in March 2012. It contains the following new algorithms:

- **Interest Rate Derivatives**
 - An n-Dimensional Markov-functional Interest Rate Model
L. Kaisajuntti J. Kennedy. Preprint 2008
 - Efficient log-Levy approximations for Levy-driven Libor model.

A. Papapantoleon, J.Schoenmakers, D. Skovmand.
Preprint 2111, TU Berlin.

- **Energy and Commodities**
 - Efficient pricing of Swing options in Lévy-driven models. O. Kudryavtsev, A. Zannette.
- **Credit Risk Derivatives**
 - Calibration in a local and stochastic intensity model. A. Alfonsi, C. Labart, J. Lelong
- **Equity Derivatives**
 - Forward Variance Dynamics: Bergomi's model revisited. S.M. Ould Aly
 - Volatility of Volatility Expansion for Bergomi's model. S.M. Ould Aly
 - Robust Approximations for Pricing Asian Options and Volatility Swaps Under Stochastic Volatility. M. Forde, A. Jacquier *Applied Mathematical Finance, Volume 17 Issue 3 2010*
 - Small-time asymptotics for implied volatility under the Heston model, M. Forde, A. Mijatovic, and Jacquier, A. *International Journal of Theoretical and Applied Finance, Volume 12, issue 6, 2009*
 - Asymptotic formulae for implied volatility under the Heston model. Forde, F, Jacquier, A, and Mijatovic, A. *Proceedings of the Royal Society A, to appear.*
 - A Mean-Reverting SDE on Correlation Matrices. A. Alfonsi, A. Ahdida
 - Fast and Accurate Long Stepping Simulation of the Heston Stochastic Volatility Model J.H. Chan, M S. Joshi, Preprint
 - High order discretization schemes for stochastic volatility models. B. Jourdain, M. Sbai. *Quantitative Finance, to appear*
 - A Fourier-based Valuation Method for Bermudan and Barrier Options under Heston Model. F. Fang, C. W. Oosterlee. *SIAM J. Finan. Math. 2, 2011, 439-463.*
 - Numerical methods and volatility models for valuing cliquet options. H. Windcliff, P.A. Forsyth, K.R. Vetzal, *Applied Mathematical Finance 13 2006.*
 - Pricing Discretely Monitored Asian Options by Maturity Randomization. G. Fusai, D. Marazzina, M.arena. *SIAM Journal on Financial Mathematics, Vol. 2 2011*
 - Wiener-Hopf techniques for Lookback options in Levy models. O. Kudryavtsev
 - Computing VaR and AVar in Infinitely Divisible Distributions. Y.S. Kim, S. Rachev, M.S. Bianchi, F.J. Fabozzi. *Probability and Mathematical Statistics, Vol. 30, Fasc. 2 2010.*
 - Analytical formulas for local volatility model with stochastic rates. E. Benhamou, E. Gobet, M. Miri 2009
 - Nonparametric Variance Reduction Methods on Malliavin Calculus. B. Lapeyre, A. Turki *SIAM Journal on Financial Mathematics, to appear*
 - Stochastic expansion for the pricing of call options with discrete dividends. P. Eto, E. Gobet. *Applied Mathematical Finance, to appear*
 - American options in high dimension solving EDSR with penalization C. Labart, J. Lelong
 - Static Hedging of Standard Options. P. Carr, L. Wu. Preprint.

The software Premia 14 has been deposited at the APP (Agence pour la Protection des Programmes) with the reference ID: FR.001.190010.011.S.C.2001.000.31000.

MAVERICK Team

5. Software

5.1. Introduction

Maverick insists on sharing the software that is developed for internal use. These are all listed in a dedicated section on the web site <http://artis.imag.fr/Software>.

5.2. PlantRad

Participant: Cyril Soler [contact].

PlantRad is a software program for computing solutions to the equation of light equilibrium in a complex scene including vegetation. The technology used is hierarchical radiosity with clustering and instantiation. Thanks to the latter, PlantRad is capable of treating scenes with a very high geometric complexity (up to millions of polygons) such as plants or any kind of vegetation scene where a high degree of approximate self-similarity permits a significant gain in memory requirements. Its main domains of applications are urban simulation, remote sensing simulation (See the collaboration with Noveltis, Toulouse) and plant growth simulation, as previously demonstrated during our collaboration with the LIAMA, Beijing.

5.3. High Quality Renderer

Participant: Cyril Soler [contact].

In the context of the European project RealReflect, the Maverick team has developed the HQR software based on the photon mapping method which is capable of solving the light balance equation and of giving a high quality solution. Through a graphical user interface, it reads X3D scenes using the X3DToolkit package developed at Maverick, it allows the user to tune several parameters, computes photon maps, and reconstructs information to obtain a high quality solution. HQR also accepts plugins which considerably eases the development of new algorithms for global illumination, those benefiting from the existing algorithms for handling materials, geometry and light sources. HQR is freely available for download at <http://artis.imag.fr/~Cyril.Soler/HQR>.

5.4. MobiNet

Participants: Fabrice Neyret [contact], Joëlle Thollot.

The MobiNet software allows for the creation of simple applications such as video games, virtual physics experiments or pedagogical math illustrations. It relies on an intuitive graphical interface and language which allows the user to program a set of mobile objects (possibly through a network). It is available in public domain at <http://mobinet.inrialpes.fr> for Linux, Windows and MacOS, and originated in a collaboration with the EVASION project-team.

The main aim of MobiNet is to allow young students at high school level with no programming skills to experiment, with the notions they learn in math and physics, by modeling and simulating simple practical problems, and even simple video games. This platform has been massively used during the Grenoble INP "engineer weeks" since 2002: 150 senior high school pupils per year, doing a 3 hour practice. This work is partly funded by Grenoble INP. Various contacts are currently developed in the educational world. Besides "engineer weeks", several groups of "monitors" PhD students conducts experimentations based on MobiNet with a high school class in the frame of the courses. Moreover, presentation in workshops and institutes are done, and a web site repository is maintained.

5.5. Freestyle

Freestyle is a software for Non-Photorealistic Line Drawing rendering from 3D scenes (Figure 2). It is designed as a programmable interface to allow maximum control over the style of the final drawing: the user "programs" how the silhouettes and other feature lines from the 3D model should be turned into stylized strokes using a set of programmable operators dedicated to style description. This programmable approach, inspired by the shading languages available in photorealistic renderers such as Pixar's RenderMan, overcomes the limitations of integrated software with access to a limited number of parameters and permits the design of an infinite variety of rich and complex styles. The system currently focuses on pure line drawing as a first step. The style description language is Python augmented with our set of operators. Freestyle was developed in the framework of a research project dedicated to the study of stylized line drawing rendering from 3D scenes. This research has led to two publications [23], [24].

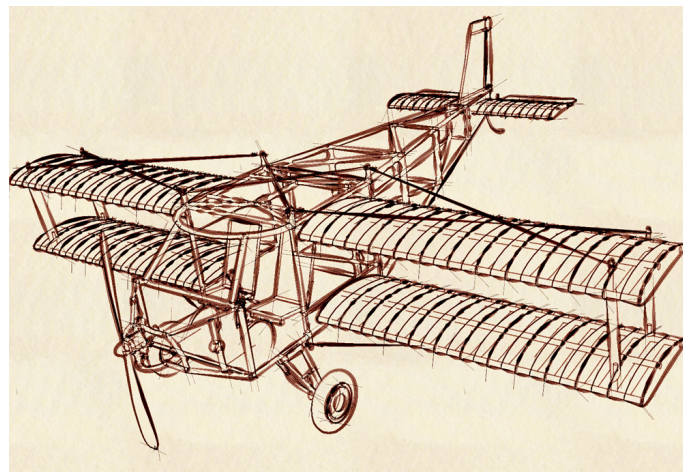


Figure 2. Stylized plane using Freestyle.

In 2008, Freestyle got a new life, completely outside Maverick or Inria: it was the basis of one of the 6 *Google Summer of Code* projects awarded to the *Blender Foundation*¹! The goal of the project was to integrate Freestyle to the well known free 3D modeler *Blender*, as its standard NPR line-drawing renderer. Maxime Curioni (under the mentoring of Jean-Luc Peurière from the *Blender Foundation*), is currently making the integration. First beta versions are publicly available, and tested by enthusiasts around the web.

5.6. Diffusion Curves

Participant: Joëlle Thollot [contact].

We provide an implementation of the vector drawing tool described in the 2008 Diffusion Curves Siggraph paper (Figure 3) This prototype is composed of the Windows binary, along with the required shader programs (ie. in source code). The software is available for download at <http://artis.imag.fr/Publications/2008/OBWBTS08> for free, for non-commercial research purposes.

5.7. VRender: vector figures

Participant: Cyril Soler [contact].

¹<http://www.blender.org/>



Figure 3. Diffusion curves freely downloadable demo.

The VRender library is a simple tool to render the content of an OpenGL window to a vectorial device such as Postscript, XFig, and soon SVG. The main usage of such a library is to make clean vectorial drawings for publications, books, etc.

In practice, VRender replaces the z-buffer based hidden surface removal of OpenGL by sorting the geometric primitives so that they can be rendered in a back-to-front order, possibly cutting them into pieces to solve cycles.

VRender is also responsible for the vectorial snapshot feature of the QGLViewer library. VRender is released under the LGPL licence and is freely available for download at <http://artis.imag.fr/Software/VRender>.

5.8. ProLand

Participants: Fabrice Neyret [contact], Eric Bruneton.

Now available at <http://proland.inrialpes.fr/> in double licencing GPL/commercial.

Proland (for procedural landscape) is a software platform originally developed at the Evasion team-project by Eric Bruneton, and currently funded by the ANR-JCJC SimOne. The goal of this platform is the real-time quality rendering and editing of large landscapes. All features can work with planet-sized terrains, for all viewpoints from ground to space. Most of the work published by Eric Bruneton and Fabrice Neyret has been done within Proland, and a large part has been integrated in the main branch. Several licences have been transferred to companies. Eric Bruneton was hired by Google-Zürich in september 2011, but will be able to keep some participation in the project.

5.9. GigaVoxel

Participants: Fabrice Neyret [contact], Morgan Armand, Eric Bruneton, Cyril Crassin, Pascal Guehl, Eric Heitz.

Soon available at <http://gigavoxels.inrialpes.fr/index.htm> in double licencing GPL/commercial.

Gigavoxel is a software platform initiated from the PhD work of Cyril Crassin, and currently funded by the ANR CONTINT RTIGE (Figure 4). The goal of this platform is the real-time rendering of very large very detailed scenes. Performances permit showing details over deep zooms and walk through very crowded scenes (which are rigid, for the moment). The principle is GPU ray-tracing of volumetric-encoded multiscale data with minimal just-in time generation of data (accounting visibility and needed resolution) kept in a cache on GPU. The representation eases the cheap management of soft shadows, depth of field, anti-aliasing and geometric LOD. Beside the representation, data management and base rendering algorithm itself, we also worked on realtime light transport, and on quality prefiltering of complex data. This work led to numerous publications ([22], [21], [20]). Several licences have been sold to companies. we also did a technical presentation of the GigaVoxels tool during Afig conference [17] in order to invit the community to use the tool.

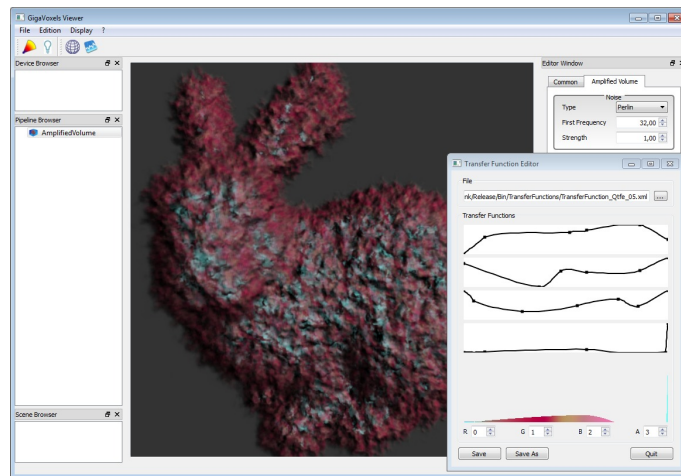


Figure 4. GigaVoxels freely downloadable demo.

MAXPLUS Project-Team

5. Software

5.1. Boîte à outil Maxplus de SCILAB/Maxplus toolbox of Scilab

Trois chercheurs du groupe (S. Gaubert, J.-P. Quadrat, et G. Cohen) ont développé (à partir d'une première version réalisée par M. Mc Gettrick) la *boîte à outils Maxplus* de Scilab, qui est **téléchargeable librement** parmi les contributions du site **Scilab**, et qui est maintenant intégrée par défaut dans **Scicoslab**. Cette boîte à outils implémente l'ensemble du calcul numérique linéaire max-plus, elle comprend en particulier le stockage creux des matrices, et des algorithmes efficaces pour le calcul de la valeur propre basées sur les itérations sur les politiques. Elle a été utilisées par plusieurs chercheurs, voir notamment [70], [134]. Il faut aussi noter que le groupe de L. Hardouin, du LISA/Istia, a complété la boîte à outils Maxplus en interfaçant leur propre librairie C++, qui permet le calcul des séries de transfert de graphes d'événements temporisés.

English version

Three researchers of the team (S. Gaubert, J.-P. Quadrat, and G. Cohen, building on a preliminary version of M. McGettrick) have developed and released the *Maxplus toolbox* of Scilab, which is freely **available** among the contributions on the **Scilab** web site, and which is now included by default in **Scicoslab**. It implements all basic linear algebra functionalities, with a special attention to large sparse matrices, including efficient algorithms for eigenvalue computation based on policy iteration. The software has been used by several researchers in their work, including [70], [134]. It should be noted that the team of L. Hardouin, from LISA/Istia, has completed the toolbox by interfacing their own C++ **library** computing the transfer series of a timed event graph.

5.2. Itérations sur les politiques pour les jeux stochastiques à somme nulle/Policy iterations for zero sum stochastic games

L'algorithme d'itérations sur les politiques pour les jeux stochastiques à somme nulle pour le cas de paiements ergodiques (gain moyen par unité de temps), et dégénérés de type "multi-chaîne" a été introduit dans [87]. Plusieurs stages ont permis l'implémentation partielle en Scilab, C ou C++, et le test de ce type d'algorithmes (voir le travail de Vishesh Dhingra [100]), ou de son couplage avec la résolution de systèmes linéaires par des méthodes multigrilles algébriques (stage de Shantanu Gangal en 2007). Le travail de thèse de Sylvie Detournay, qui porte sur le couplage entre itérations sur les politiques et méthodes multigrilles algébriques, voir le §6.4.1 ci-dessous, a permis le développement d'un programme complet. Le code écrit par Sylvie Detournay (en C) a été déposé sur InriaGForge. Pour le moment il n'est accessible qu'aux membres de l'équipe.

English version

The policy iteration algorithm for zero sum repeated games with ergodic payoff (i.e. mean payoff per time unit), and in degenerate "multichain" cases, has been introduced in [87]. Several internships allowed us to implement in Scilab, C or C++, and to test such algorithms (see the work of Vishesh Dhingra [100]), or its combinaison with the resolution of linear systems by algebraic multigrid methods (internship of Shantanu Gangal in 2007). The PhD thesis work of Sylvie Detournay, who concerns the combinaison of policy iterations with algebraic multigrid methods, see §6.4.1 below, allowed us to develop a complete program. The program written by Sylvie Detournay (in C language) has been posted on InriaGForge. For the moment it can only be seen by members of the team.

5.3. TPLib: bibliothèque pour la manipulation de polyèdres tropicaux/TPLib: tropical polyhedra library

TPLib est une bibliothèque écrite en OCaml qui permet de manipuler des polyèdres tropicaux. Elle est distribuée sous license LGPL <https://gforge.inria.fr/projects/tplib>.

Cette bibliothèque implémente notamment des algorithmes permettant de passer d'une représentation externe d'un polyèdre à une représentation interne, ou inversement (voir §6.2.1 pour plus de détails). Elle permet aussi de réaliser d'autres opérations fondamentales, comme le calcul du complexe polyédral associé à un polyèdre donné (au sens de Develin et Sturmfels [98]), ou le calcul de cônes tangents tropicaux. Enfin, elle fournit toutes les primitives permettant d'utiliser les polyèdres tropicaux en tant que domaine abstrait numérique, afin de déterminer des invariants de programmes ou systèmes faisant intervenir les opérations min et max (voir [67]).

TPLib est aujourd'hui utilisé dans le logiciel Polymake [117], développé à la Technische Universität Darmstadt (Allemagne). Ce dernier logiciel constitue une boîte à outils permettant de manipuler des nombreux objets mathématiques (polytopes convexes, complexes polyédraux, graphes, matroïdes, polytopes tropicaux).

Le développement d'interfaces avec d'autres logiciels est désormais facilité grâce à la présence de *bindings* dans le langage C. Grâce à cela, un prototype d'interface a été réalisé entre TPLib et l'outil VerifyTAPN (<https://launchpad.net/verifytapn>), qui permet la vérification de réseaux de Pétri avec arcs temporisés (voir §6.5.7). De même, une interface à la bibliothèque de domaines abstraits numériques APRON [127] est également en cours de développement.

English version

TPLib is a library written in OCaml, which allows to manipulate tropical polyhedra. It is distributed under LGPL <https://gforge.inria.fr/projects/tplib>.

This library implements algorithms allowing to pass from an external representation of a polyhedron to an internal description, or inversely (see §6.2.1 for more details). Besides, the library allows to perform several fundamental operations over tropical polyhedra, such as computing the associated polyhedral complex (see Develin and Sturmfels [98]), or determining the tropical tangent cone at any point. Finally, it provides all the primitives allowing to use tropical polyhedra as an numerical abstract domain, in order to determine program/system invariants involving the operations min and max (see [67]).

TPLib is now used in the software Polymake [117], developed in Technische Universität Darmstadt (Germany). Polymake is a toolbox allowing to manipulate mathematic objects such as convex polytopes, polyhedral complexes, graphs, matroids, and tropical polytopes.

The development of further interfaces is now easier thanks to the distribution of bindings in C language. Using these bindings, a prototype of interface has been created between TPLib and the model-checker VerifyTAPN (<https://launchpad.net/verifytapn>), which allows the verification of timed-arc Petri Nets (see §6.5.7). An interface to the numerical abstract domain APRON [127] is also under development.

MC2 Project-Team

5. Software

5.1. eLYSe

Participants: Olivier Saut [correspondant], Raphael Bahègne, Vincent Huber, Jean-Baptiste Lagaert, Mathieu Specklin.

eLYSe is a numerical platform used for our computations in Biology (tumor growth), micro-fluidics and complex Newtonian fluid flows. The platform is divided in two libraries : one is devoted to the modelling equations and the other one includes the numerical solvers. For example, we are able to treat (in 2D and 3D) transport equations, diffusion equations, Navier-Stokes equations, Maxwell system and the interaction fluid-structure by level-set and penalization methods. The solvers are based on finite volume methods on cartesian grids and allow parallel computations. See also the web page <http://www.math.u-bordeaux1.fr/~osaut/pages/eLYSe.html>.

- Version: 0.4
- ACM: ACM J.2 J.3 G.1.8 G.1.10
- AMS: AMS65Z05 35Q92
- Keywords: Modélization and numerical simulations, Finite volume methods, Level Set approach, Penalization method
- APP: En cours
- Type of human computer interaction: console
- OS/Middleware: Platform developped on Mac OS X architecture.
- Required library or software: Petsc (<http://www.mcs.anl.gov/petsc/petsc-as/>)Vtk (<http://www.vtk.org/>)Blitz++ (<http://c2.com/cgi/wiki?BlitzPlusPlus>) (optionnel)Boost (<http://www.boost.org/>)
- Programming language: C++
- Documentation: doxygen.

5.2. Kesaco

Participants: Olivier Saut [correspondant], Raphael Bahègne, Damiano Lombardi, Mathieu Specklin.

Kesaco is a set of libraries and programs aiming at applications of mathematical modeling in clinical oncology. It features:

- A library of specialized mathematical model describing the growth of different types of cancers (secondary tumors in the lung, gliomas).
- A set of programs useful to validate mathematical models (compute the various behavior they can produce) and to build databases of numerical simulations.
- Segmentation and registration routines to use medical images directly in our numerical codes.
- Calibration methods to recover the parameters of the models using sequences of medical images. Three techniques are implemented (a genetic algorithm, a technique based on reduced order models, a sensitivity technique).

All these routines are adapted to run on a MP architecture. The webpage may be found at <http://www.math.u-bordeaux1.fr/~osaut/pages/kesaco.html>.

- Version: 0.1
- Keywords: Modélization and numerical simulations
- APP: En cours
- Type of human computer interaction: console
- OS/Middelware: Platform developped on Mac OS X architecture.
- Required library or software: eLYSe, Insight Toolkit (<http://www.itk.org>)
- Programming language: C++
- Documentation: doxygen.

5.3. NaSCar

Participant: Michel Bergmann [correspondant].

This code is devoted to solve 3D-flows in around moving and deformable bodies. The incompressible Navier-Stokes equations are solved on fixed grids, and the bodies are taken into account thanks to penalization and/or immersed boundary methods. The interface between the fluid and the bodies is tracked with a level set function or in a Lagrangian way. The numerical code is fully second order (time and space). The numerical method is based on projection schemes of Chorin-Temam's type. The code is written in C language and use Petsc (<http://www.mcs.anl.gov/petsc/petsc-as/>) library for the resolution of large linear systems in parallel.

NaSCar can be used to simulate both hydrodynamic bio-locomotion as fish like swimming and aerodynamic flows such wake generated by a wind turbine.

- Version: 1
- Keywords: numerical analyse, fluid mechanics, langage C, PETSc
- Software benefit : simulate a flow around a deformable obstacle, moving into a fluid.
- APP: En cours
- Patent: non
- Type of human computer interaction: human for the moment
- OS/Middelware: unix, linux, mac os
- Required library or software: PETSc item Programming language: C
- Documentation: in progress

5.4. S-MPI-2D-3D

Participants: Charles-Henri Bruneau [correspondant], Khodor Khadra.

The software NS-MPI-2D-3D is a numerical platform devoted to the computation of the incompressible flow around bodies in two or three dimensions modelled by Stokes, Navier-Stokes or Oldroyd-B equations. It is based on finite differences or finite volumes approximations on cartesian grid using the volume penalization method to handle the obstacles. The resolution is achieved by means of the multigrid method. Dirichlet, periodic or artificial boundary conditions are implemented to solve various problems in closed or open domains.

- Version: 3
- Keywords: Numerical simulation of incompressible flows,
- Type of human computer interaction: console
- OS/Middelware: unix, linux, Mac OS X item Programming language: Fortran 95 and MPI
- Documentation: included

5.5. Other MC2 codes

- Penalization techniques on cartesian grids to solve incompressible Navier-Stokes equations
 - **Vortex**: sequential, Vortex In-Cell (VIC) scheme : hybrid vortex methods based on the combination of Lagrangian mesh-free schemes and Eulerian grid based schemes on the same flow region.
 - Unstructured body fitted meshes
 - **Richards** : 2D Unstructured finite element code, implicit solver, sequential, to solve the transport-diffusion equations through a porous media including tidal forcing and mechanisms of diagenesis.
 - development inside **FluidBox** software in collaboration with **BACCHUS**. 2D-3D unstructured meshes, Stabilized Finite Elements method (SUPG), RANS turbulence model, parallel: Domain Decomposition and MPI.
- Immersed boundary techniques for:
 - **Compressible flows** : 2D-3D finite volume scheme for compressible Euler equations with solid obstacles on cartesian grids. 3D code parallelized with MPI
 - **Elliptic problems** : 2 2D-3D finite difference scheme for elliptic interface problems, parallelized with PETSc
 - Electropermeabilization: 2D finite difference scheme, parallelized with PETSc to simulate the electropermeabilization of biological cells

MCTAO Team (section vide)

MESCAL Project-Team

5. Software

5.1. Tools for cluster management and software development

Participant: Olivier Richard [correspondant].

The KA-Tools is a software suite developed by MESCAL for exploitation of clusters and grids. It uses a parallelization technique based on spanning trees with a recursive starting of programs on nodes. Industrial collaborations were carried out with Mandrake, BULL, HP and Microsoft.

KA-DEPLOY is an environment deployment toolkit that provides automated software installation and reconfiguration mechanisms for large clusters and light grids. The main contribution of KA-DEPLOY 2 toolkit is the introduction of a simple idea, aiming to be a new trend in cluster and grid exploitation: letting users concurrently deploy computing environments tailored exactly to their experimental needs on different sets of nodes. To reach this goal KA-DEPLOY must cooperate with batch schedulers, like OAR, and use a parallel launcher like TAKTUK (see below).

TAKTUK is a tool to launch or deploy efficiently parallel applications on large clusters, and simple grids. Efficiency is obtained thanks to the overlap of all independent steps of the deployment. We have shown that this problem is equivalent to the well known problem of the single message broadcast. The performance gap between the cost of a network communication and of a remote execution call enables us to use a work stealing algorithm to realize a near-optimal schedule of remote execution calls. Currently, a complete rewriting based on a high level language (precisely Perl script language) is under progress. The aim is to provide a light and robust implementation. This development is lead by the MOAIS project-team.

5.2. OAR: Batch scheduler for clusters and grids

Participant: Olivier Richard [correspondant].

The OAR project focuses on robust and highly scalable batch scheduling for clusters and grids. Its main objectives are the validation of grid administration tools such as TAKTUK, the development of new paradigms for grid scheduling and the experimentation of various scheduling algorithms and policies.

The grid development of OAR has already started with the integration of best effort jobs whose purpose is to take advantage of idle times of the resources. Managing such jobs requires a support of the whole system from the highest level (the scheduler has to know which tasks can be canceled) down to the lowest level (the execution layer has to be able to cancel awkward jobs). The OAR architecture is perfectly suited to such developments thanks to its highly modular architecture. Moreover, this development is used for the CiGri grid middleware project.

The OAR system can also be viewed as a platform for the experimentation of new scheduling algorithms. Current developments focus on the integration of theoretical batch scheduling results into the system so that they can be validated experimentally.

See also the web page <http://oar.imag.fr>.

5.3. CiGri: Computing resource Reaper

Participant: Olivier Richard [correspondant].

CiGri is a middleware which gather the unused computing resource from intranet infrastructure and to make it available for large set of tasks. It manages the execution of large sets of parametric tasks on lightweight grid by submitting individual jobs to each batch scheduler. It's associated to the OAR resource management system (batch scheduler). Users can easily monitor and control their set of jobs through a web portal. System provides mechanisms to identify job error causes, to isolate faulty components and to resubmit job in a safer context. See also the web page <http://cigri.imag.fr/>

5.4. FTA: Failure Trace Archive

Participant: Derrick Kondo [correspondant].

The Failure Trace Archive is available at <http://fta.inria.fr>.

With the increasing functionality, scale, and complexity of distributed systems, resource failures are inevitable. While numerous models and algorithms for dealing with failures exist, the lack of public trace data sets and tools has prevented meaningful comparisons. To facilitate the design, validation, and comparison of fault-tolerant models and algorithms, we led the creation of the Failure Trace Archive (FTA), an on-line public repository of availability traces taken from diverse parallel and distributed systems.

While several archives exist, the FTA differs in several respects. First, it defines a standard format that facilitates the use and comparison of traces. Second, the archive contains traces in that format for over 20 diverse systems over a time span of 10 years. Third, it provides a public toolbox for failure trace interpretation, analysis, and modeling. The FTA was released in November 2009. It has received over 11,000 hits since then. The FTA has had national and international impact. Several published works have already cited and benefited from the traces and tools of the FTA. Simulation toolkits for distributed systems, such as SimGrid (CNRS, France) and GridSim (University of Melbourne, Australia), have incorporated the traces to allow for simulations with failures.

5.5. SimGrid: simulation of distributed applications

Participants: Arnaud Legrand [correspondant], Lucas Schnorr, Pierre Navarro, Degomme Augustin, Laurent Bobelin.

SimGrid is a toolkit that provides core functionalities for the simulation of distributed applications in heterogeneous distributed environments. The specific goal of the project is to facilitate research in the area of distributed and parallel application scheduling on distributed computing platforms ranging from simple network of workstations to Computational Grids.

We have released one new major version (3.6) of SimGrid (June 2011) and two minor versions (June and October 2011). These versions include our current work on visualization, analysis of large scale distributed systems, and extremely scalable simulation. See also the web page <http://simgrid.gforge.inria.fr/>.

5.6. TRIVA: interactive trace visualization

Participants: Lucas Schnorr [correspondant], Arnaud Legrand.

TRIVA is an open-source tool used to analyze traces (in the Pajé format) registered during the execution of parallel applications. The tool serves also as a sandbox for the development of new visualization techniques. Some features include: Temporal integration using dynamic time-intervals; Spatial aggregation through hierarchical traces; Scalable visual analysis with squarified treemaps; A Custom Graph Visualization.

See also the web page <http://triva.gforge.inria.fr/>.

5.7. ψ and ψ^2 : perfect simulation of Markov Chain stationary distributions

Participant: Jean-Marc Vincent [correspondant].

ψ and ψ^2 are two software tools implementing perfect simulation of Markov Chain stationary distributions using *coupling from the past*. ψ starts from the transition kernel to derive the simulation program while ψ^2 uses a monotone constructive definition of a Markov chain. They are available at <http://www-id.imag.fr/Logiciels/psi/>.

5.8. GameSeer: simulation of game dynamics

Participant: Panayotis Mertikopoulos [correspondant].

Mathematica toolbox (graphical user interface and functions library) for efficient, robust and modular simulations of game dynamics.

5.9. Kameleon: environment for experiment reproduction

Participants: Olivier Richard [correspondant], Joseph Emeras.

Kameleon is a tool developed to facilitate the building and rebuilding of software environment. It helps experimenter to manage his experiment's software environment which can include the operating system, libraries, runtimes, his applications and data. This tool is an element in the experimental process to obtain repeatable experiments and therefore reproducible results.

METISS Project-Team

5. Software

5.1. Audio signal processing, segmentation and classification toolkits

Participant: Guillaume Gravier.

Guillaume Gravier is now with the TEXMEX group but this software is being used by several members of the METISS group.

speech, audio, signal, analysis, processing, audio stream, detection, tracking, segmentation, audio indexing, speaker verification

The SPro toolkit provides standard front-end analysis algorithms for speech signal processing. It is systematically used in the METISS group for activities in speech and speaker recognition as well as in audio indexing. The toolkit is developed for Unix environments and is distributed as a free software with a GPL license. It is used by several other French laboratories working in the field of speech processing.

In the framework of our activities on audio indexing and speaker recognition, AudioSeg, a toolkit for the segmentation of audio streams has been developed and is distributed for Unix platforms under the GPL agreement. This toolkit provides generic tools for the segmentation and indexing of audio streams, such as audio activity detection, abrupt change detection, segment clustering, Gaussian mixture modeling and joint segmentation and detection using hidden Markov models. The toolkit relies on the SPro software for feature extraction.

Contact : guillaume.gravier@irisa.fr

<http://gforge.inria.fr/projects/spro>, <http://gforge.inria.fr/projects/audiaseg>

5.2. Irene: a speech recognition and transcription platform

Participant: Guillaume Gravier.

Guillaume Gravier is now with the TEXMEX group but this software is being used by several members of the METISS group.

speech modeling, speech recognition, broadcast news indexing, beam-search, Viterbi, HMM

In collaboration with the computer science dept. at ENST, METISS has actively participated in the past years in the development of the freely available Sirocco large vocabulary speech recognition software [91]. The Sirocco project started as an Inria Concerted Research Action now works on the basis of voluntary contributions.

The Sirocco speech recognition software was then used as the heart of the transcription modules within a spoken document analysis platform called IRENE. In particular, it has been extensively used for research on ASR and NLP as well as for work on phonetic landmarks in statistical speech recognition.

In 2009, the integration of IRENE in the multimedia indexing platform of IRISA was completed, incorporating improvements benchmarked during the ESTER 2 evaluation campaign in december 2008. Additional improvements were also carried out such as bandwidth segmentation and improved segment clustering for unsupervised acoustic model adaptation. The integration of IRENE in the multimedia indexing platform was mainly validated on large datasets extracted from TV streams.

Contact : guillaume.gravier@irisa.fr

<http://gforge.inria.fr/projects/sirocco>

5.3. MPTK: the Matching Pursuit Toolkit

Participants: Rémi Gribonval, Jules Espiau.

The Matching Pursuit ToolKit (MPTK) is a fast and flexible implementation of the Matching Pursuit algorithm for sparse decomposition of monophonic as well as multichannel (audio) signals. MPTK is written in C++ and runs on Windows, MacOS and Unix platforms. It is distributed under a free software license model (GNU General Public License) and comprises a library, some standalone command line utilities and scripts to plot the results under Matlab.

MPTK has been entirely developed within the METISS group mainly to overcome limitations of existing Matching Pursuit implementations in terms of ease of maintainability, memory footprint or computation speed. One of the aims is to be able to process in reasonable time large audio files to explore the new possibilities which Matching Pursuit can offer in speech signal processing. With the new implementation, it is now possible indeed to process a one hour audio signal in as little as twenty minutes.

Thanks to an Inria software development operation (Opération de Développement Logiciel, ODL) started in September 2006, METISS efforts have been targeted at easing the distribution of MPTK by improving its portability to different platforms and simplifying its developers' API. Besides pure software engineering improvements, this implied setting up a new website with an FAQ, developing new interfaces between MPTK and Matlab and Python, writing a portable Graphical User Interface to complement command line utilities, strengthening the robustness of the input/output using XML where possible, and most importantly setting up a whole new plugin API to decouple the core of the library from possible third party contributions.

Collaboration : Laboratoire d'Acoustique Musicale (University of Paris VII, Jussieu).

Contact : remi.gribonval@irisa.fr

<http://mptk.gforge.inria.fr>, <http://mptk.irisa.fr>

5.4. FASST

Participants: Emmanuel Vincent [correspondant], Nancy Bertin, Frédéric Bimbot.

FASST is a Flexible Audio Source Separation Toolbox in Matlab, designed to speed up the conception and automate the implementation of new model-based audio source separation algorithms.

5.5. NACHOS

Participants: Nancy Bertin [correspondant], Rémi Gribonval.

The software and associated database were developed within the ANR ECHANGE project, with the participation of Gilles Chardon, Laurent Daudet, François Ollivier and Antoine Peillot.

NACHOS (Nearfield Acoustic HOlography with Sparse regularization) is a downloadable companion software for the journal paper [38], distributed to comply with the "reproducible research" principle. It performs the reconstruction of operational deflection shapes of a vibrating structure, from acoustic measurements of the generated sound field. The software consists in Matlab source code, and automatically downloads the needed database. It allows to reproduce all results and figures of the paper, and to experiment some additional settings. It is distributed under GPL 3.0 license.

MEXICO Project-Team

5. Software

5.1. Software

5.1.1. *libalf: the Automata Learning Framework*

Participant: Benedikt Bollig [correspondant].

libalf is a comprehensive, open-source library for learning finite-state automata covering various well-known learning techniques (such as, Angluin's L^* , Biermann, and RPNI, as well as a novel learning algorithm for NFA). *libalf* is highly flexible and allows for facily interchanging learning algorithms and combining domain-specific features in a plug-and-play fashion. Its modular design and its implementation in C++ make it a flexible platform for adding and engineering further, efficient learning algorithms for new target models (e.g., Büchi automata).

Details on *libalf* can be found at <http://libalf.informatik.rwth-aachen.de/>

5.1.2. *Mole/Cunf: unfolders for Petri Nets*

Participants: Stefan Schwoon [correspondant], César Rodríguez.

Mole computes, given a safe Petri net, a finite prefix of its unfolding. It is designed to be compatible with other tools, such as PEP and the Model-Checking Kit, which are using the resulting unfolding for reachability checking and other analyses. The tool *Mole* arose out of earlier work on Petri nets. Details on *Mole* can be found at <http://www.lsv.ens-cachan.fr/~schwoon/tools/mole/>.

In the context of MEXICO, we have created a new tool called *Cunf*, which is able to handle contextual nets, i.e. Petri nets with read arcs [42],[48]. While in principle every contextual net can be transformed into an equivalent Petri net and then unfolded using *Mole*, *Cunf* can take advantage of their special features to do the job faster and produce a smaller unfolding. *Cunf* has recently been extended with a verification component that takes advantage of these features [70]. More details can be found at <http://www.lsv.ens-cachan.fr/~rodrigue/tools/cunf/>. Moreover, *Cunf* has been integrated into the *CosyVerif* environment (see section 5.1.4).

5.1.3. *COSMOS : a Statistical Model Checker for the Hybrid Automata Stochastic Logic*

Participants: Hilal Djafri, Benoît Barbot [correspondant].

COSMOS is a statistical model checker for the Hybrid Automata Stochastic Logic (HASL). HASL employs Linear Hybrid Automata (LHA), a generalization of Deterministic Timed Automata (DTA), to describe accepting execution paths of a Discrete Event Stochastic Process (DESP), a class of stochastic models which includes, but is not limited to, Markov chains. As a result HASL verification turns out to be a unifying framework where sophisticated temporal reasoning is naturally blended with elaborate reward-based analysis. *COSMOS* takes as input a DESP (described in terms of a Generalized Stochastic Petri Net), an LHA and an expression Z representing the quantity to be estimated. It returns a confidence interval estimation of Z ; recently, it has been equipped with functionalities for rare event analysis. *COSMOS* is written in C++ and is freely available to the research community.

Details on *COSMOS* can be found at <http://www.lsv.ens-cachan.fr/~barbot/cosmos/>

5.1.4. *COSYVERIF*

Participants: Serge Haddad [correspondant ?], Benoît Barbot.

CosyVerif is a software environment whose goal is the formal specification and verification of dynamic systems.

It has been designed in order to:

- support different formalisms with the ability to easily create new ones, - provide a graphical interface for every formalism, - include verification tools called via the interface as a web service, - offer the possibility for a developer to integrate his/her own tool, also allowing it to interact with the other tools.

This environment consists of two software tools: Coloane, the graphical interface, and Alligator, an integration framework based web services. It is enlarged with the existing verification tools developed in our laboratories (founding members or partners). Why ?

The development of Cosyverif has been decided and it is supported by three partners of the Parisian verification group, MeFoSyLoMa. This group is composed of seven teams. and the founding members of are LIP6, LIPN and LSV. First, these members aim at sharing their tools, comparing and supporting industrial case studies and finally making them long-lasting. Second, they also want to promote the practice of formal verification in industry and thus they intend to ease the task of integration of new formalisms and tools.

It is managed by a steering committee consisting of researchers and engineers. It decides strategic orientations as well as technical choices. Current Tools

Two formalisms are supported: automata and Petri nets, both with extensions. Most of the tools are related to Petri nets. Some of them perform structural analyses like invariant computations. while other tools perform behavioural analyses: symbolic reachability graph building, unfolding, stochastic simulations, etc. Finally some of them transform high-level nets into low-level ones. All the developed software are open source and free software tools. Alligator is published under the GNU Affero General Public License (AGPL) version 3 ; Coloane is published under the Eclipse Public License (EPL) version 1 .

Three engineers have worked or are currently working on COSYVERIF:

- Francis Hulin-Hubard, part-time (CNRS) in 2012;
- Clément Desmoulins , full-time (ANR), 6 months; and
- Alban Linard, full-time Inria engineer, for 2 years.

MICMAC Project-Team (section vide)

MIMETIC Team

5. Software

5.1. HPTS++: Hierarchical Parallel Transition System ++

Participants: Stéphane Donikian [contact], Fabrice Lamarche [contact].

HPTS++ is a platform independent toolkit to describe and handle the execution of multi-agent systems. It provides a specific object oriented language encapsulating C++ code for interfacing facilities and a runtime kernel providing automatic synchronization and adaptation facilities.

The language provides functionalities to describe state machines (states and transitions) and to inform them with user specific C++ code to call at a given point during execution. This language is object oriented and supports concepts such as polymorphism and inheritance (state machines and user defined C++ classes). The compilation phase translates a state machine in a C++ class that can be compiled separately and linked through static or dynamic libraries. The runtime kernel includes a scheduler that handles parallel state machines execution and that provides synchronization facilities such as mutual exclusion on resources, dead lock avoidance, notions of priorities and execution adaptation in accordance with resources availability.

HPTS++ also provides a task model. Thanks to this model, the user can describe primitive behaviors through atomic tasks and combine them with operators (sequence, parallelism, loops, alternatives...). These operators are fully dynamic. Hence they can be used at runtime to rapidly create complex behaviors.

5.2. MKM: Manageable Kinematic Motions

Participants: Richard Kulpa [contact], Franck Multon.

We have developed a framework for animating human-like figures in real-time, based on captured motions. This work was carried-out in collaboration with the M2S Laboratory (Mouvement, Sport, Santé) of the University Rennes 2.

In this software, we propose a morphology-independent representation of the motion that is based on a simplified skeleton which normalizes the global postural informations. This formalism is not linked to morphology and allows very fast motion retargetting and adaptation to geometric constraints that can change in real-time. This approach dramatically reduces the post production time and allows the animators to handle a general motion library instead of one library per avatar.

The framework provides an animation library which uses the motions either obtained from our off-line tool (that transforms standard formats into our morphology-independent representation) or parameterized models in order to create complete animation in real-time. Several models are proposed such as grasping, orientation of the head toward a target. We have also included a new locomotion model that allows to control the character directly using a motion database.

In order to create realistic and smooth animations, MKM uses motion synchronization, blending and adaptation to skeletons and to external constraints. All those processes are performed in real-time in an environment that can change at any time, unpredictably.

All these features have been used to anticipate and control the placement of footprints depending on high level parameters. This link between control and behavior levels will be used for reactive navigation in order to have realistic motion adaptations as well as to deal with constrained environments.

5.3. TopoPlan: Topological Planner and Behaviour Library

Participant: Fabrice Lamarche [contact].

TopoPlan (Topological Planner) is a toolkit dedicated to the analysis of a 3D environment geometry in order to generate suitable data structures for path finding and navigation. This toolkit provides a two step process: an off-line computation of spatial representation and a library providing on-line processes dedicated to path planning, environmental requests...

TopoPlan is based on an exact 3D spatial subdivision that accurately identifies floor and ceiling constraints for each point of the environment. Thanks to this spatial subdivision and some humanoid characteristics, an environment topology is computed. This topology accurately identifies navigable zones by connecting 3D cells of the spatial subdivision. Based on this topology several maps representing the environment are extracted. Those maps identify obstacle and step borders as well as bottlenecks. TopoPlan also provides a runtime library enabling the on-line exploitation of the spatial representation. This library provides several algorithms including roadmap-based path-planning, trajectory optimization, footprint generation, reactive navigation and spatial requests through customizable spatial selectors.

TopoPlan behavior is a library built on top of TopoPlan and MKM providing several behaviors described thanks to the HPTS++ task model. Its goal is to provide a high level interface handling navigation and posture adaptation within TopoPlan environments. Provided behaviors include:

- A behavior handling fully planned navigation toward an arbitrary destination. This behavior precisely handles footprint generation within constrained environments such as stairs for instance.
- A behavior controlling an MKM humanoid to follow a trajectory specified by the user.
- A behavior controlling MKM to follow a list of footprints given by the user.
- A behavior adapting the humanoid posture to avoid collision with ceiling. This behavior runs in parallel of all other behaviors and adapts humanoid motion when needed without any user intervention.
- A behavior handling reactive navigation of virtual humans. This behavior plan a path to a given target and follows the path while avoiding collisions with other navigating entities.

Those behaviors have been built using the HPTS++ task model. Thus, they can be easily combined together or with other described behaviors through task operators.

MINT Project-Team

5. Software

5.1. LibGINA

Participant: Laurent Grisoni [correspondant].

This library has been developed within the context of the ADT GINA, for one of the installation that have been made in collaboration with Le Fresnoy national studio (Damassama, Léonore Mercier). This library is currently being posted as APP, and has been used by Idées-3com small company, in the context of our join I-lab program. This library allows for use of gesture for command, and is able to handle strong variability into recognized patterns.

Current version: version 1.0

Software characterization: A-2 SO-3 SM-2-up EM-3 SDL-3 OC-DA4-CD4-MS2-TPM4

5.2. 3D interaction using mobile phone

Participants: Samuel Degrande [correspondant], Laurent Grisoni.

This work has been achieved in the context of the Idées-3com I-lab. In this context a module, that allows to use any android based smartphone to control an Explorer module for navigation and interaction with VRML-based content. This module was used as a basis by Idées-3com in their commercial product this year.

Current version: version 1.0

Software characterization: A-2 SO-3 SM-2-up EM-2-up SDL-3 OC-DA4-CD4-MS2-TPM4

5.3. tIO (tactile input & output)

Participants: Paolo Olivo, Nicolas Roussel [correspondant], Ibrahim Yapici.

tIO is a library designed to facilitate the implementation of doubly tactile interaction techniques (tactile input coupled with tactile feedback) based on the STIMTAC technology. Supporting all current STIMTAC prototypes, it makes it easy to move the system pointer of the host computer according to motions detected on them and adapt their vibration amplitude based on the color of the pointed pixel or the nature of the pointed object. The library includes a set of Qt demo applications that illustrate these two different approaches and makes it easy to “augment” existing Qt applications with tactile feedback. It also makes it possible to supplement or substitute tactile feedback with basic auditory feedback synthesized using **portaudio** (friction level is linearly mapped to the frequency of a sine wave). This not only facilitates the development and documentation of tactile-enhanced applications but also makes it easier to demonstrate them to a large audience.

Current version: 0.1 - June 2011 (IDDN.FR.001.270005.000.S.P.2011.000.10000)

Software characterization: A2, SO3-up, SM-2, EM2, SDL1.

5.4. libpointing

Participants: Géry Casiez [correspondant], Damien Marchal, Nicolas Roussel.

Libpointing is a software toolkit that provides direct access to HID pointing devices and supports the design and evaluation of pointing transfer functions [2]. The toolkit provides resolution and frequency information for the available pointing and display devices and makes it easy to choose between them at run-time through the use of URIs. It allows to bypass the system's transfer functions to receive raw asynchronous events from one or more pointing devices. It replicates as faithfully as possible the transfer functions used by Microsoft Windows, Apple OS X and Xorg (the X.Org Foundation server). Running on these three platforms, it makes it possible to compare the replicated functions to the genuine ones as well as custom ones. The toolkit is written in C++ with Python and Java bindings available. It is scheduled to be publicly released in 2012, the licence remaining to be decided.

Web site: <http://libpointing.org/>

Software characterization: A3, SO3, SM-2, EM2, SDL4

MISTIS Project-Team

5. Software

5.1. The ECMPR software

Participant: Florence Forbes.

Joint work with: Radu Horaud and Manuel Iguel.

The ECMPR (Expectation Conditional Maximization for Point Registration) package implements [57] [65]. It registers two (2D or 3D) point clouds using an algorithm based on maximum likelihood with hidden variables. The method can register both rigid and articulated shapes. It estimates both the rigid or the kinematic transformation between the two shapes as well as the parameters (covariances) associated with the underlying Gaussian mixture model. It has been registered in APP in 2010 under the GPL license.

5.2. The LOCUS and P-LOCUS software

Participants: Florence Forbes, Senan James Doyle.

Joint work with: Michel Dojat.

From brain MR images, neuroradiologists are able to delineate tissues such as grey matter and structures such as Thalamus and damaged regions. This delineation is a common task for an expert but unsupervised segmentation is difficult due to a number of artefacts. The LOCUS software and its recent extension P-LOCUS automatically perform this segmentation for healthy and pathological brains. An image is divided into cubes on each of which a statistical model is applied. This provides a number of local treatments that are then integrated to ensure consistency at a global level, resulting in low sensitivity to artifacts. The statistical model is based on a Markovian approach that enables to capture the relations between tissues and structures, to integrate a priori anatomical knowledge and to handle local estimations and spatial correlations.

The LOCUS software has been developed in the context of a collaboration between Mistis, a computer science team (Magma, LIG) and a Neuroscience methodological team (the Neuroimaging team from Grenoble Institut of Neurosciences, INSERM). This collaboration resulted over the period 2006-2008 into the PhD thesis of B. Scherrer (advised by C. Garbay and M. Dojat) and in a number of publications. In particular, B. Scherrer received a "Young Investigator Award" at the 2008 MICCAI conference. Its extension (P-LOCUS) for lesion detection is realized by S. Doyle with financial support from Gravit for possible industrial transfer.

The originality of this work comes from the successful combination of the teams respective strengths i.e. expertise in distributed computing, in neuroimaging data processing and in statistical methods.

5.3. The POPEYE software

Participant: Florence Forbes.

Joint work with: Vasil Khalidov, Radu Horaud, Miles Hansard, Ramya Narasimha, Elise Arnaud.

POPEYE contains software modules and libraries jointly developed by three partners within the POP STREP project: Inria, University of Sheffield, and University of Coimbra. It includes kinematic and dynamic control of the robot head, stereo calibration, camera-microphone calibration, auditory and image processing, stereo matching, binaural localization, audio-visual speaker localization. Currently, this software package is not distributed outside POP.

5.4. The HDDA and HDDC toolboxes

Participant: Stéphane Girard.

Joint work with: Charles Bouveyron (Université Paris 1). The High-Dimensional Discriminant Analysis (HDDA) and the High-Dimensional Data Clustering (HDDC) toolboxes contain respectively efficient supervised and unsupervised classifiers for high-dimensional data. These classifiers are based on Gaussian models adapted for high-dimensional data [53]. The HDDA and HDDC toolboxes are available for Matlab and are included into the software MixMod [52]. Recently, a R package has been developed and integrated in The Comprehensive R Archive Network (CRAN). It can be downloaded at the following URL: <http://cran.r-project.org/web/packages/HDclassif/>.

5.5. The Extremes freeware

Participant: Stéphane Girard.

Joint work with: Diebolt, J. (CNRS), Laurent Gardes (Univ Strasbourg) and Garrido, M. (INRA Clermont-Ferrand-Theix).

The EXTREMES software is a toolbox dedicated to the modelling of extremal events offering extreme quantile estimation procedures and model selection methods. This software results from a collaboration with EDF R&D. It is also a consequence of the PhD thesis work of Myriam Garrido [55]. The software is written in C++ with a Matlab graphical interface. It is now available both on Windows and Linux environments. It can be downloaded at the following URL: <http://extremes.gforge.inria.fr/>.

5.6. The SpaCEM³ program

Participants: Senan James Doyle, Florence Forbes.

SpaCEM³ (Spatial Clustering with EM and Markov Models) is a software that provides a wide range of supervised or unsupervised clustering algorithms. The main originality of the proposed algorithms is that clustered objects do not need to be assumed independent and can be associated with very high-dimensional measurements. Typical examples include image segmentation where the objects are the pixels on a regular grid and depend on neighbouring pixels on this grid. More generally, the software provides algorithms to cluster multimodal data with an underlying dependence structure accounting for some spatial localisation or some kind of interaction that can be encoded in a graph.

This software, developed by present and past members of the team, is the result of several research developments on the subject. The current version 2.09 of the software is CeCILLB licensed.

Main features. The approach is based on the EM algorithm for clustering and on Markov Random Fields (MRF) to account for dependencies. In addition to standard clustering tools based on independent Gaussian mixture models, SpaCEM³ features include:

- The unsupervised clustering of dependent objects. Their dependencies are encoded via a graph not necessarily regular and data sets are modelled via Markov random fields and mixture models (eg. MRF and Hidden MRF). Available Markov models include extensions of the Potts model with the possibility to define more general interaction models.
- The supervised clustering of dependent objects when standard Hidden MRF (HMRF) assumptions do not hold (ie. in the case of non-correlated and non-unimodal noise models). The learning and test steps are based on recently introduced Triplet Markov models.
- Selection model criteria (BIC, ICL and their mean-field approximations) that select the "best" HMRF according to the data.
- The possibility of producing simulated data from:
 - general pairwise MRF with singleton and pair potentials (typically Potts models and extensions)
 - standard HMRF, ie. with independent noise model
 - general Triplet Markov models with interaction up to order 2
- A specific setting to account for high-dimensional observations.
- An integrated framework to deal with missing observations, under Missing At Random (MAR) hypothesis, with prior imputation (KNN, mean, etc), online imputation (as a step in the algorithm), or without imputation.

The software is available at <http://spacem3.gforge.inria.fr>. A user manual in English is available on the web site above together with example data sets. The INRA Toulouse unit is more recently participating to this project for promotion among the bioinformatics community [75].

5.7. The FASTRUCT software

Participant: Florence Forbes.

Joint work with: Francois, O. (TimB, TIMC) and Chen, C. (former Post-doctoral fellow in Mistis).

The FASTRUCT program is dedicated to the modelling and inference of population structure from genetic data. Bayesian model-based clustering programs have gained increased popularity in studies of population structure since the publication of the software STRUCTURE [70]. These programs are generally acknowledged as performing well, but their running-time may be prohibitive. FASTRUCT is a non-Bayesian implementation of the classical model with no-admixture uncorrelated allele frequencies. This new program relies on the Expectation-Maximization principle, and produces assignment rivaling other model-based clustering programs. In addition, it can be several-fold faster than Bayesian implementations. The software consists of a command-line engine, which is suitable for batch-analysis of data, and a MS Windows graphical interface, which is convenient for exploring data.

It is written for Windows OS and contains a detailed user's guide. It is available at <http://mistis.inrialpes.fr/realisations.html>.

The functionalities are further described in the related publication:

- Molecular Ecology Notes 2006 [56].

5.8. The TESS software

Participant: Florence Forbes.

Joint work with: Francois, O. (TimB, TIMC) and Chen, C. (former post-doctoral fellow in Mistis).

TESS is a computer program that implements a Bayesian clustering algorithm for spatial population genetics. Is it particularly useful for seeking genetic barriers or genetic discontinuities in continuous populations. The method is based on a hierarchical mixture model where the prior distribution on cluster labels is defined as a Hidden Markov Random Field [60]. Given individual geographical locations, the program seeks population structure from multilocus genotypes without assuming predefined populations. TESS takes input data files in a format compatible to existing non-spatial Bayesian algorithms (e.g. STRUCTURE). It returns graphical displays of cluster membership probabilities and geographical cluster assignments through its Graphical User Interface.

The functionalities and the comparison with three other Bayesian Clustering programs are specified in the following publication:

- Molecular Ecology Notes 2007

MNEMOSYNE Team

5. Software

5.1. Positioning

Our previous works in the domain of well-defined distributed asynchronous adaptive computations [32], [29], [34] have already made us define a library (DANA [3]), closely related to both the notion of artificial neural networks and cellular automata. From a conceptual point of view, the computational paradigm supporting the library is grounded on the notion of a unit that is essentially a (vector of) potential that can vary along time under the influence of other units and learning. Those units can be organized into layers, maps and network.

More generally, we gather in the middleware EnaS (that stands for *Event Neural Assembly Simulation*; cf. <http://gforge.inria.fr/projects/enas>) our numerical and theoretical developments, allowing to simulate and analyze so called "event neural assemblies". EnaS has been designed as a plug-in for our simulators (e.g. DANA or MVASpike) as other existing simulators (via the NeuralEnsemble meta-simulation platform) and additional modules for computations with neural unit assembly on standard platforms (e.g. Python or the Scilab platform).

We will also have to interact with the High Performance Computing (HPC) community, since having large scale simulations at that mesoscopic level is an important challenge in our systemic view of computational neuroscience. Our approach implies to emulate the dynamics of thousands, or even millions, of integrated computational units, each of them playing the role of a whole elementary neural circuit (e.g. the microcolumn for the cortex). Mesoscopic models are considered in such an integrative approach, in order to exhibit global dynamical effects that would be hardly reachable by compartment models involving membrane equations or even spiking neuron networks.

The vast majority of high performance computing softwares for computational neuroscience addresses sub-neural or neural models [19], but coarser grained population models are also demanding for large scale simulations, with fully distributed computations, without global memory or time reference, as it is specified in (cf. § 3.2).

5.2. Dana

Participant: Nicolas Rougier.

DANA [28] is a python framework (<http://dana.loria.fr>) whose computational paradigm is grounded on the notion of a unit that is essentially a set of time dependent values varying under the influence of other units via adaptive weighted connections. The evolution of a unit's value are defined by a set of differential equations expressed in standard mathematical notation which greatly ease their definition. The units are organized into groups that form a model. Each unit can be connected to any other unit (including itself) using a weighted connection. The DANA framework offers a set of core objects needed to design and run such models. The modeler only has to define the equations of a unit as well as the equations governing the training of the connections. The simulation is completely transparent to the modeler and is handled by DANA. This allows DANA to be used for a wide range of numerical and distributed models as long as they fit the proposed framework (e.g. cellular automata, reaction-diffusion system, decentralized neural networks, recurrent neural networks, kernel-based image processing, etc.).

5.3. ENAS: Event Neural Assembly Simulation

Participants: Frédéric Alexandre, Nicolas Rougier, Thierry Viéville.

EnaS (that stands for “Event Neural Assembly Simulation”) is a middleware implementing our last numerical and theoretical developments, allowing to simulate and analyze so called "event neural assemblies". The recent achievements include (in collaboration with the Neuromathcomp EPI): spike trains statistical analysis via Gibbs distributions, spiking network programming for exact event's sequence restitution, discrete neural field parameters algorithmic adjustments and time-constrained event-based network simulation reconciling clock and event based simulation methods. It has been designed as plug-in for our simulators (e.g. DANA or Mvaspike) as other existing simulators (via the NeuralEnsemble meta-simulation platform) and additional modules for computations with neural unit assembly on standard platforms (e.g. Python or the Scilab platform).

5.4. Virtual Enaction

Participants: Frédéric Alexandre, André Garenne, Nicolas Rougier, Thierry Viéville.

The computational models studied in this project have applications that extend far beyond what is possible to experiment yet in human or non-human primate subjects. Real robotics experimentations are also impaired by rather heavy technological constraints; for instance, it is not easy to dismantle a given embedded system in the course of emerging ideas. The only versatile environment in which such complex behaviors can be studied both globally and at the level of details of the available modeling is a virtual environment, as in video games. Such a system can be implemented as «brainy-bot» (a programmed player based on our knowledge of the brain architecture) which goal is to survive in a complete manipulable environment.

In order to attain this rather ambitious objective we are going to both (i) deploy an existing open-source video game middleware in order to be able to shape the survival situation to be studied and (ii) revisit the existing models in order to be able to integrate them as an effective brainy-bot. It will consist of a platform associated to a scenario that would be the closest possible to a survival situation (foraging, predator-prey relationship, partner approach to reproduction) and in which it would be easy to integrate an artificial agent with sensory inputs (visual, touch and smell), emotional and somatosensory cues (hunger, thirst, fear, ..) and motor outputs (movement, gesture, ..) connected to a "brain" whose architecture will correspond to the major anatomical regions involved in the issues of learning and action selection (cortex areas detailed here, basal ganglia, hippocampus, and areas dedicated to sensorimotor processes). The internal game clock will be slowed down enough to be able to run non trivial brainy-bot implementations.

MOAIS Project-Team

5. Software

5.1. KAAPI

Participants: Thierry Gautier [correspondant], Vincent Danjean, François Broquedis, Pierre Neyron.

Kaapi (<http://kaapi.gforge.inria.fr>, coordinator T. Gautier) Kaapi is a middleware for high performance applications running on multi-cores/multi-processors as well as cluster or computational grid. Kaapi provides 1/ a very high level API based on macro data flow language; 2/ several scheduling algorithms for multi-threaded computations as well as for iterative applications for numerical simulation on multi-CPU / multi-GPU; 3/ fault-tolerant protocols. Publicly available at <http://kaapi.gforge.inria.fr> under CeCILL licence. Kaapi has won the 2008 Plugtest organized by Grid@Works. Kaapi provides ABI compliant implementations of Quark (PLASMA, Linear Algebra, Univ. of Tennessee) and libGOMP (GCC runtime for OpenMP). Direct competitors with 1/: Quark, StarSs (UPC, BSC), OpenMP. Direct competitors with 2/: StarSs, StarPU (Inria RUNTIME), Quark, OpenACC runtimes. Direct competitors providing 3/: Charm++, MPI.

- ACM: D.1.3
- License: CeCILL
- OS/Middleware: Unix (Linux, MacOSX, ...)
- Programming language: C/C++, Fortran
- Characterization of Software : A-3 / SO-4 / SM-3 / EM-3 / SDL-4
- Own Contribution: DA-4 / CD-4 / MS-4 / TPM-4

5.2. FlowVR

Participant: Bruno Raffin [correspondant MOAIS].

- Characterization of Software : A-3 / SO-4 / SM-3 / EM-3 / SDL-4
- Own Contribution: DA-4 / CD-3 / MS-3 / TPM-4
- Additional information: FlowVR (<http://flowvr.sf.net>, coordinator B. Raffin) provides users with the necessary tools to develop and run high performance interactive applications on PC clusters and Grids. The main target applications include virtual reality, scientific visualization and Web3D. FlowVR enforces a modular programming that leverages software engineering issues while enabling high performance executions on distributed and parallel architectures. FlowVR is the reference API for Grimage. See also the web page <http://flowvr.sf.net>. The FlowVR software suite has 3 main components:
 - FlowVR : The core middleware library. FlowVR relies on the data-flow oriented programming approach that has been successfully used by other scientific visualization tools.
 - FlowVR Render : A parallel rendering library.
 - VTK FlowVR : a VTK / FlowVR / FlowVR Render coupling library.

5.3. TakTuk - Adaptive large scale remote execution deployment

Participants: Guillaume Huard [correspondant], Pierre Neyron.

- Characterization of Software : A-2 / SO-3 / SM-5 / EM-3 / SDL-4
- Own Contribution: DA-4 / CD-4 / MS-4 / TPM-4
- Additional information:
 - web site: <http://taktuk.gforge.inria.fr>, Coordinator G. Huard
 - Objective of the software: TakTuk is a tool for deploying parallel remote executions of commands to a potentially large set of remote nodes. It spreads itself using an adaptive algorithm and sets up an interconnection network to transport commands and perform I/Os multiplexing/demultiplexing. The TakTuk mechanics dynamically adapt to environment (machine performance and current load, network contention) by using a reactive work-stealing algorithm that mixes local parallelization and work distribution.
 - Users community: TakTuk is a research open source project available in the Debian GNU/Linux distribution (package taktuk) used in lower levels of Grid5000 software architectures (nodes monitoring in OAR, environment diffusion in Kadeploy). The community is small : developers and administrators for large scale distributed platforms, but active.
 - Positioning: main competing tools are pdsh (but uses linear deployment) and gexec (not fault tolerant, requires installation), for more details : B. Claudel, G. Huard and O. Richard. TakTuk, Adaptive Deployment of Remote Executions. In Proceedings of the International Symposium on High Performance Distributed Computing (HPDC), 2009. TakTuk is the only tool to provide to deployed processes a communication layer (just like an MPIrun, but not tied to a specific environment) and synchronization capabilities.

5.4. KRASH - Kernel for Reproduction and Analysis of System Heterogeneity

Participants: Guillaume Huard [correspondant], Swann Perarnau.

- Characterization of Software : A-1 / SO-3 / SM-4 / EM-2 / SDL-3
- Own Contribution: DA-4 / CD-4 / MS-4 / TPM-4
- Additional information:
 - web site: <http://krash.ligforge.imag.fr>
 - Objective of the software: Krash is a tool to create a synthetic heterogeneity on top of a dedicated system while preserving the OS state and algorithms (no modification). It makes use of the control groups (cgroups) in Linux kernel newer than version 2.6.24 to create a dynamic CPU load enforced no matter how many applications are running in parallel.
 - Users community: Research open source project, small community: developers of parallel applications in heterogeneous contexts.
 - Positioning: Competing tool is Wreakavoc (less scalable, less precise), more details in : Swann Perarnau and Guillaume Huard. Krash: Reproducible cpu load generation on many-core machines. In IEEE International Parallel and Distributed Processing Symposium (IPDPS), 2010.

5.5. Cache Control

Participants: Guillaume Huard [correspondant], Swann Perarnau.

- Characterization of Software : A-1 / SO-3 / SM-3 / EM-2 / SDL-3
- Own Contribution: DA-4 / CD-4 / MS-4 / TPM-4
- Additional information:
 - web site: <http://ccontrol.ligforge.imag.fr/>
 - Objective of the software: Cache Control is a Linux kernel module enabling user applications to restrict their memory allocations to a subset of the hardware memory cache. This module reserves and exports available physical memory as virtual devices that can be mmap'd to. It gives to calling processes physical memory using only a subset of the cache (similarly to page coloring). It actually creates cache partitions that can be used simultaneously by a process to control how much cache a data structure can use.
 - Users community: Research open source project, small community: developers wanting to measure or tune the cache usage of their applications. Does not apply to recent NUCA caches.
 - Positioning: Competing tool is ULCC which does the same thing at the runtime level, more details in : Swann Perarnau, Marc Tchiboukdjian, and Guillaume Huard. Controlling cache utilization of hpc applications. In International Conference on Supercomputing (ICS), 2011.

5.6. GGen

Participants: Guillaume Huard [correspondant], Swann Perarnau.

- Characterization of Software : A-2 / SO-4 / SM-4 / EM-2 / SDL-3
- Own Contribution: DA-4 / CD-4 / MS-4 / TPM-4
- Additional information:
 - web site: <http://ggen.ligforge.imag.fr/>, Coordinator Swann Perarnau
 - Objective of the software: GGen is a free (GPL-compatible) command line application and library for generating and analyzing directed acyclic graphs. Designed primarily to be used in simulations of scheduling algorithms, it helps researchers understand fully the nature of the graphs generated. It implements the most known graph generation algorithms enabling comparisons between them.
 - Users community: Research open source project, task scheduling community: ggen provides a meaningful way to generate test cases.
 - Positioning: To our knowledge, there's no competing tool, more details in : Daniel Cordeiro, Grégory Mounié, Swann Perarnau, Denis Trystram, Jean-Marc Vincent, and Frédéric Wagner. Random graph generation for scheduling simulations. In International ICST Conference on Simulation Tools and Techniques (SIMUTools), 2010.

5.7. Triva

Participants: Guillaume Huard [correspondant], Lucas Schnorr.

- Characterization of Software : A-2 / SO-4 / SM-5 / EM-3 / SDL-3
- Own Contribution: DA-4 / CD-3 / MS-3 / TPM-3
- Additional information:
 - web site: <http://triva.gforge.inria.fr/>, Coordinator, Lucas Schnorr

- Objective of the software: Triva is an open-source tool used to analyze traces (in the pajé format) registered during the execution of parallel applications. The tool serves also as a sandbox to the development of new visualization techniques.
- Users community: Research open source project, applications developers, especially parallel applications.
- Positioning: Main competing tools are Vampir (classical 2D Gantt charts) and Tau (less advanced agregation techniques), more details in : A Hierarchical Aggregation Model to achieve Visualization Scalability in the analysis of Parallel Applications. Lucas Mello Schnorr, Guillaume Huard, Philippe Olivier Alexandre Navaux. Parallel Computing. Volume 38, Issue 3, March 2012.

5.8. OAR

Participants: Pierre Neyron [correspondant MOAIS], Grégory Mounié.

- Characterization of Software : A-5 / SO-3 / SM-4 / EM-4 / SDL-5
- Own Contribution: DA-3 / CD-2 / MS-1 / TPM-1
- Additional information: OAR (<http://oar.imag.fr>, Coordinator O. Richard, Inria MESCAL) is a batch scheduler. The MOAIS team develops the central automata and the scheduling module that includes successive evolutions and improvements of the policy.OAR is used to schedule jobs both on the CiGri (Grenoble region) and Grid50000 (France) grids. CiGri is a production grid that federates about 500 heterogeneous resources of various Grenoble laboratories to perform computations in physics. MOAIS has also developed the distributed authentication for access to Grid5000.

5.9. SOFA

Participant: Bruno Raffin [correspondant].

Inria category: ????

- Characterization of Software : A-5 / SO-4 / SM-4 / EM-4 / SDL-5
- Own Contribution: DA-2 / CD-2 / MS-1 / TPM-1
- Additional information: SOFA (<http://www.sofa-framework.org/>, Coordinator F. Faure, Inria IMAG-INE) is an Open Source framework primarily targeted at real-time simulation, with an emphasis on medical simulation. It is mostly intended for the research community to help develop newer algorithms, but can also be used as an efficient prototyping tool. Moais contributes to parallelization of kernel algorithms used in the simulation.
- ACM: J.3
- Programming language: C/C++

5.10. LinBox

Participants: Clément Pernet [correspondant], Thierry Gautier.

- Characterization of Software : A-3 / SO-4 / SM-2 / EM-3 / SDL-5
- Own Contribution: DA-4 / CD-3 / MS-3 / TPM-4
- Additional information:
 - web site: <http://linalg.org>
 - Objective of the software: LinBox is an open-source C++ template library for exact, high-performance linear algebra computations. It is considered as the reference library for numerous computations (such as linear system solving, rank, characteristic polynomial, Smith normal forms,...) over finite fields and integers with dense, sparse, and structured matrices.

- The LinBox group is an international collaboration (USA: NCSU, UDel; Canada: U Waterloo, U Calgary; France: LIP, LIRMM, LJK and LIG). Articles related to the library have been published in the main Conferences of the area: ISSAC, ICMS. MOAIS contributes to its development and more specifically to its parallelization in the context of ANR HPAC project. It is currently experiencing a major change of design, to better integrate parallelism.
- Users community: mostly researchers doing computational mathematics (number theory, cryptology, group theory, persistent homology). They use the library by either linking against it directly (the library is packaged in Debian, Fedora, etc) or withing the general purpose math software Sage (sagemath.org very broad diffusion) which includes LinBox as a kernel for exact linear algebra.

MODAL Project-Team

5. Software

5.1. Two advances for the MIXMOD software

Participants: Christophe Biernacki, Serge Iovleff, Remi Lebret, Parmeet Bhatia.

MIXMOD (MIXture MODelling) is an important software for the mΘdal team since it concerns its main topics: model-based supervised, unsupervised and semisupervised classification for various data situations. MIXMOD is now a well-distributed software with over 250 downloads/month are recorded for several years. MIXMOD is written in C++ (more than 10 000 lines) and distributed under GNU General Public License. Several other institutions participate in the MIXMOD development since several years: CNRS, Inria Saclay-Île de France, Université de Franche-Comté, Université Lille 1. The software already benefits from several APP deposits.

An interface between MIXMOD and R (Rmixmod) has been developed by Rémi Lebret and Serge Iovleff and is now available on the CRAN (<http://cran.r-project.org/web/packages/Rmixmod/index.html>). We expect now a wide impact of MIXMOD on the growing community familiar with R. A paper related to Rmixmod is submitted to an international journal [34].

Until December 2012, Parmeet Bhatia, under scientific supervision of Christophe Biernacki, is developing possibility in MIXMOD to cluster simultaneously continuous and categorical data with the restrictive conditional independence assumption. It is an important first step towards the long term purpose of mΘdal to cluster heterogeneous (or mixed) data sets.

5.2. The blockcluster package

Participants: Christophe Biernacki, Serge Iovleff, Parmeet Bhatia.

blockcluster is a R package for model-based simultaneous clustering of rows and columns, thanks to an Inria ADT grant (Parmeet Bhatia). It is also developed in collaboration with University of Technology of Compiègne. It offers the ability to structure very large data tables both in lines and columns for different data types (continuous, binary and contingency data). In particular, it opens wide potential applications in biology, marketing, etc. It is available online on CRAN (<http://cran.r-project.org/web/packages/blockcluster/index.html>) for all major platforms (Linux, MacOS, Windows). It also comes with utility functions to visualize data. A paper related to blockcluster is submitted to an international journal [40].

5.3. Cuvclust package

Participant: Guillemette Marot.

cuvclust is a R package dedicated to model-based curve clustering. Considered models include Functional Clustering Mixed Models (FCMM, ie functional clustering with the presence of functional random effects), but also traditional functional clustering model (FCM, without functional random effects), and functional mixed models (FMM, functional random effects without clustering). Estimation is done by maximum likelihood using the EM algorithm, and two criteria are proposed to select the number of clusters, based on integrated likelihoods.

Guillemette Marot was the main contributor of the beta version of the package during her post-doc. Due to several changes in conception and due to planning of extensions in the package by the other contributors of the package, she decided to become a regular contributor and left the maintenance to Franck Picard.

5.4. MetaMa

Participant: Guillemette Marot.

metaMA is a specialised software for microarrays. It is a R package which combines either p-values or modified effect sizes from different studies to find differentially expressed genes. The main competitor of metaMA is geneMeta. Compared to geneMeta, metaMA offers an improvement for small sample size datasets since the corresponding modelling is based on shrinkage approaches.

Guillemette Marot is the main contributor and the maintainer of these packages and spent around one year full time for this package between the conception, the implementation, and the documentation. Her PhD advisors (Florence Jaffrézic, Claus-Dieter Mayer, Jean-Louis Foulley) helped her with the conception but she implemented alone the code.

First versions were posted to the CRAN, the official website of the R software, in 2009. New versions for this package were released in August 2011 in order to take into account remarks from the main users (biologists or biostatisticians analysing gene expression data). This software is routinely used by biologists from INRA, Jouy en Josas (it has been included in a local analysis pipeline) but its diffusion on the CRAN makes it available to a wider community, as attested by the citations of publications related to the methods implemented in the software.

More information is available on the website <http://cran.r-project.org/web/packages/metaMA/>

5.5. SMVar

Participant: Guillemette Marot.

SMVar is a specialised software for microarrays. This R package implements the structural model for variances in order to detect differentially expressed genes from gene expression data. It performs gene expression differential analysis, based on a particular variance modelling. Its main competitor is the Bioconductor R package limma but limma assumes a common variance between the two groups to be compared while SMVar relaxes this assumption.

More information on the website <http://cran.r-project.org/web/packages/SMVar/index.html>

5.6. Tax3 Software

Participants: Serge Iovleff, Remi Lebret.

Tax3 implements a statistical method providing an analytical framework for high dimensional datasets and complex problems combining several variable types: genetics, genomics, biomarkers and phenotypes

5.7. aam Program

Participant: Serge Iovleff.

aam is a console based program dedicated to the estimation of the semi-linear auto-associative models in a gaussian setting. It is written in C++ and used the STK++ library as support.

5.8. STK++

Participant: Serge Iovleff.

STK++ is a multi-platform toolkit written in C++ for creating fast and easy to use data mining programs. It offers a large set of templated class in C++ which are suitable for projects ranging from small one-off projects to complete statistical application suites. A C equivalent would be gsl. However, STK++ is developed in C++ in order to get speed and reusability.

As the aim of STK++ is to aid developers to new developments, it proposes essentially interfaces classes and various concrete helping classes, like arrays, numerical methods (QR, SVD), input and output (csv files), random number generators, etc.

The software is regularly developed for 10 years by Serge Iovleff and it is a work in progress. The version 0.3 has been released. More information is available on the website <http://www.stkpp.org/> and source repository is here: <https://sourcesup.cru.fr/projects/stk/>

5.9. Scan3D

Participants: Alexandru Amarioarei, Cristian Preda.

Scan3D is a C++ software for estimating the distribution of the three-dimensional scan statistics for Bernoulli and Poisson models. It implements the most recent approximation methods available, in particular that developed by the authors providing bounds for the approximation errors [39].

MODEMIC Project-Team

5. Software

5.1. VITELBIO

Participants: Jérôme Harmand, Alain Rapaport.

VITELBIO (Virtual TELluric BIOreactors) is a simulation tool for studying networks of interconnected chemostats with the objective of mimicking microbial activities in soil. The software, developed with the help of ITK Company, is accessible on a server from any web navigator and make use of Flex for the user interface and Octave for the numerical integration. An important effort has been made for obtaining a pleasant and easy interface that is appealing for microbiologists: the network can be drawn graphically on the screen and simulation results can be easily compared between (virtual) experiments, superposing trajectories curves.

This software is used by several researchers, from LBE (INRA Narbonne), UMR Eco & Sols (Montpellier), UREP (Unité de Recherche sur l'Ecosystème Prairial, INRA Theix), Biomeco (Paris-Grignon), UMR EGC (Environnement et grandes cultures, Paris-Grignon)... and also as a teaching support. Vitelbio is presented at <http://sites.google.com/site/vitelbio/> and it is accessible at <http://vitelbio.itkweb.fr>.

5.2. SMC DEMOS

Participant: Fabien Campillo.

SMC DEMOS (Sequential Monte Carlo demos) proposes a set of demonstration Matlab procedures for nonlinear filtering approximation via particle filtering (sequential Monte Carlo): bearing-only tracking with obstacles, tracking in digital terrain model, track-before-detect in a sequence of digital picture, mobile phone tracking based on the signal strength to nearby antenna. This software is deposited with the "Agence pour la Protection des Programmes" (APP, 7/7/2009), available at <http://www-sop.inria.fr/members/Fabien.Campillo/software/smc-demos/>.

MOISE Project-Team

5. Software

5.1. Adaptive Grid Refinement

Participants: Laurent Debreu, Marc Honnorat.

AGRIF (Adaptive Grid Refinement In Fortran, [80],[6]) is a Fortran 90 package for the integration of full adaptive mesh refinement (AMR) features within a multidimensional finite difference model written in Fortran. Its main objective is to simplify the integration of AMR potentialities within an existing model with minimal changes. Capabilities of this package include the management of an arbitrary number of grids, horizontal and/or vertical refinements, dynamic regridding, parallelization of the grids interactions on distributed memory computers. AGRIF requires the model to be discretized on a structured grid, like it is typically done in ocean or atmosphere modelling. As an example, AGRIF is currently used in the following ocean models: MARS (a coastal model developed at IFREMER-France), ROMS (a regional model developed jointly at Rutgers and UCLA universities), OPA-NEMO ocean modelling system (a general circulation model used by the French and European scientific community) and HYCOM (a regional model developed jointly by University of Miami and the French Navy).

In 2012, a new contract has been signed with IFREMER to optimize parallel capabilities of the software. The software will be used operationally to attain a resolution of 500meters along the French coasts. (<http://www.previmer.org>) AGRIF is licensed under a GNU (GPL) license and can be downloaded at its web site (<http://ljk.imag.fr/MOISE/AGRIF/index.html>).

5.2. NEMOVAR

Participants: Arthur Vidard, Pierre-Antoine Bouttier, Bénédicte Lemieux-Dudon.

NEMOVAR is a state-of-the-art multi-incremental variational data assimilation system dedicated to the european ocean modelling platform NEMO for research and operational applications. It is co-developed by MOISE, CERFACS (FR), ECMWF (EU) and MetOffice (UK) under the CeCILL license, written in fortran and python. It is now in use in both ECMWF and MetOffice for their operational oceanic forecasting systems. It has also been used for specific studies in collaboration with Mercator-Ocean, LPO, LOCEAN and LEGI in France and University of Namur in Belgium. It is also a likely candidate for becoming the future Black-Sea forecasting system of the Marine Hydrographical Institute of Ukraine with whom we collaborate actively. Previously part of NEMOVAR, NEMO-TAM (Tangent and adjoint models for NEMO) that have been developed by the MOISE team will be now distributed directly by the NEMO consortium. The first official tagged release including NEMO-TAM will be published early 2013.

5.3. DatIce

Participants: Bénédicte Lemieux-Dudon, Habib Toyé Mahamadou Kele.

Antarctic and Greenland ice cores provide a mean to study the phase relationships of climate changes in both hemispheres. They also enable to study the timing between climate, and greenhouse gases or orbital forcings. One key step for such studies is to improve the absolute and relative precisions of ice core age scales (for ice and trapped gas), and beyond that, to try to reach the best consistency between chronologies of paleo-records of any kind.

The DatIce tool is designed to increase the consistency between pre-existing core chronologies (also called background). It formulates a variational inverse problem which aims at correcting three key quantities that uniquely define the core age scales: the accumulation rate, the total thinning function, and the close-off depth. For that purpose, it integrates paleo-data constraints of many types among which age markers (with for instance documented volcanoes eruptions), and stratigraphic links (with for instance abrupt changes in methane concentration). A cost function is built that enables to calculate new chronologies by making a trade-off between all the constraints (background chronologies and paleo-data).

DatIce enables to circumvent the limits encountered with other dating approaches, in particular because it controls the model errors, which are still large despite efforts to better describe the firn densification, the ice flow and the forcing fields (ice sheet elevation, temperature and accumulation rate histories). Controlling the model error makes it possible to assimilate large set of observations, to constrain both the gas and ice age scales, and to apply the process on several cores at the same time by including stratigraphic links between cores. This approach greatly improves the consistency of ice cores age scales.

The method presented in ([84], [85]) has already been applied simultaneously to EPICA EDML and EDC, Vostok and NGRIP drillings. The DatIce tool has aroused some interest in the glaciological and paleo-community since 2009.

The code has been recently applied in two publications [2] and [22] which aimed at the construction of a unified chronology for Antarctic ice cores. LGGE, LSCE and MOISE are partners to extend the code to marine and terrestrial cores. On going development efforts are made to ensure the robustness of the dating solution (diagnostics on the assimilation system, calibration of the background error covariance matrices).

5.4. SDM toolbox

Participant: Antoine Rousseau.

The computation of the wind at small scale and the estimation of its uncertainties is of particular importance for applications such as wind energy resource estimation. To this aim, we develop a new method based on the combination of an existing numerical weather prediction model providing a coarse prediction, and a Lagrangian Stochastic Model adapted from a pdf method introduced by S.B. Pope for turbulent flows. This Stochastic Downscaling Method (SDM <http://sdm.gforge.inria.fr/>) is thus aimed to be used as a refinement toolbox of large-scale numerical models. SDM requires a specific modelling of the turbulence closure, and involves various simulation techniques whose combination is totally new (such as Poisson solvers, optimal transportation mass algorithm, original Euler scheme for confined Langevin stochastic processes, and stochastic particle methods). Since 2011, we work on the comparison of the SDM model (endowed with a physical geostrophic forcing and a wall log law) with simulations obtained with a LES method (Mésos-NH code) for the atmospheric boundary layer (from 0 to 750 meters in the vertical direction), in the neutral case.

5.5. CompModSA package

Alexandre Janon is a contributor of the packages CompModSA - Sensitivity Analysis for Complex Computer Models (see <http://cran.r-project.org/web/packages/CompModSA/index.html>), and sensitivity (see <http://cran.r-project.org/web/packages/sensitivity/index.html>). These packages are useful for conducting sensitivity analysis of complex computer codes.

MORPHEME Team

4. Software

4.1. Software

4.1.1. Deposits

The software MAD V2.0 was deposited with the APP in November 2012. It deals with the melasma severity scoring from multi-spectral imaging.

4.1.2. Transfers

The software MAD V2.0 was transferred to Galderma R&D.

MORPHEO Team

5. Software

5.1. Platforms

5.1.1. The Grimage platform

The Grimage platform is an experimental multi-camera platform dedicated to spatio-temporal modeling including immersive and interactive applications. It hosts a multiple-camera system connected to a PC cluster, as well as visualization facilities including head mounted displays. This platform is shared by several research groups, most prominently Moais, Morpheo and Perception. In particular, Grimage allows challenging real-time immersive applications based on computer vision and interactions between real and virtual objects, Figure 1 . Note that the Grimage platform will be replaced by the Kinovis platform that will exhibit a larger acquisition space and better acquisition facilities.

5.1.2. Virtualization Gate

Vgate is an immersive environment that allows full-body immersion and interaction with virtual worlds. It is a joint initiative of computer scientists from computer vision, parallel computing and computer graphics from several research groups at Inria Grenoble Rhône-Alpes, and in collaboration with the company 4D View Solutions. The Morpheo team is leading this project.



Figure 1. Platforms: on the left the Grimage acquisition; on the right the vgate immersive environment.

5.1.3. Multicamera platform for video analysis of mice behavior

This project is a follow-up of the experimental set-up developed for a CNES project with Mathieu Beraneck from the CESeM laboratory (centre for the study of sensorimotor control, CNRS UMR 8194) at the Paris-Descartes University. The goal of this project was to analyze the 3D body postures of mice with various vestibular deficiencies in low gravity condition (3D posturography) during a parabolic flight campaign. The set-up has been now adapted for new experiments on motor-control disorders for other mice models. This experimental platform is currently under development for a broader deployment for high throughput phenotyping with the technology transfer project ETHOMICE. This project involves a close relationship with the CESeM laboratory and the European Mouse Clinical Institute in Strasbourg (Institut Clinique de la Souris, ICS).

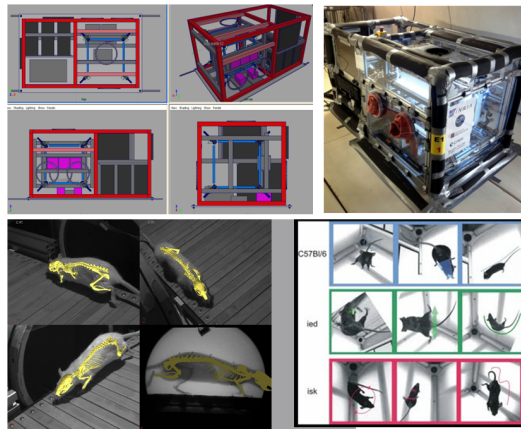


Figure 2. *Ethomice: Experimental platform for video analysis of mice behavior.*

5.2. Software packages

5.2.1. LucyViewer

Lucy Viewer http://4drepository.inrialpes.fr/lucy_viewer/ is an interactive viewing software for 4D models, i.e, dynamic three-dimensional scenes that evolve over time. Each 4D model is a sequence of meshes with associated texture information, in terms of images captured from multiple cameras at each frame. Such data is available from various websites over the world including the 4D repository website hosted by Inria Grenoble <http://4drepository.inrialpes.fr/>. The software was developed in the context of the European project iGlace, it is available as an open source software under the GNU LGPL Licence.

5.2.2. Ethomice

Ethomice <http://morpheo.inrialpes.fr/people/reveret/ethomice/> is a motion analysis software to characterize motor behavior of small vertebrates such as mice or rats. From a multiple views video input, a biomechanical model of the skeleton is registered. Study on animal model is the first important step in Biology and Clinical research. In this context, the analysis of the neuro-motor behaviour is a frequent cue to test the effect of a gene or a drug. Ethomice is a platform for simulation and analysis of the small laboratory animal, such as rat or mouse. This platform links the internal skeletal structure with 3D measurements of the external appearance of the animal under study. From a stream of multiple views video, the platform aims at delivering a three dimensional analysis of the body posture and the behaviour of the animal. The software was developed by Lionel Reveret and Estelle Duveau.

5.3. Databases

5.3.1. 4D repository (<http://4drepository.inrialpes.fr/>)

This website hosts dynamic mesh sequences reconstructed from images captured using a multi-camera set up. Such mesh-sequences offer a new promising vision of virtual reality, by capturing real actors and their interactions. The texture information is trivially mapped to the reconstructed geometry, by back-projecting from the images. These sequences can be seen from arbitrary viewing angles as the user navigates in 4D (3D geometry + time) . Different sequences of human / non-human interaction can be browsed and downloaded from the data section. A software to visualize and navigate these sequences is also available for download.

MOSTRARE Project-Team

5. Software

5.1. FXP

Participants: Joachim Niehren [correspondant], Denis Debarbieux, Tom Sebastian.

Software Self-Assessment: A-3, SO-4, SM-3, EM-3, SDL-4

The FXP language is a temporal logic for a fragment of Forward XPath that is suitable for querying XML streams. The FXP library of the Mostrare project of Inria Lille provides rewriting tool that removes backward axis, a compiler of the FXP library to nested word automata and an efficient query answering algorithm for nested word automata on XML streams.

FXP is developed in the Inria transfer project QuiXProc in cooperation with Innovimax. Both a professional and a free version are available. The owner is Inria. A new release was published in October 2012.

See also the web page <http://fxp.lille.inria.fr/>.

- Version: FXP v1.1.0

5.2. QuixPath

Participants: Joachim Niehren [correspondant], Denis Debarbieux, Tom Sebastian.

Software Self-Assessment: A-3, SO-4, SM-3, EM-3, SDL-4

The QuiXPath language is a large fragment of XPath with full support for the XML data model. The QuiXPath library provides a compiler from QuiXPath to FXP. QuiXPath is developed in the Inria transfer project QuiXProc in cooperation with Innovimax. Both, a free open source and a professional version are available. The ownership of QuiXPath is shared between Inria and Innovimax. The main application of QuiXPath is its usage in QuiXProc, an professional implementation of the W3C pipeline language XProc own by Innovimax. A new release was published in October 2012.

See also the web page <http://code.google.com/p/quixpath/>.

- Version: QuixPath v1.1.0

5.3. VOLATA

Participant: Fabien Torre [correspondant].

Software Self-Assessment: A-2, SO-4, SM-2, EM-2, SDL-2

VOLATA provides several machine learning algorithms for attribute-value inference, grammatical inference and inductive logic programming.

See also the web page <http://www.grappa.univ-lille3.fr/~torre/Recherche/Softwares/volata/>.

- ACM: I.2.6

5.4. JProGraM

Participant: Antonino Freno [correspondant].

Software Self-Assessment: A-3, SO-3-up, SM-2, EM-3, SDL-4.

JProGraM is a GPL-licensed Java library for machine learning and statistical analysis *over* graphs and *through* graphs. Supported models for vectorial data include e.g. Bayesian networks, Markov random fields, Gaussian mixtures, kernel density estimators, and neural networks, whereas random graph tools include small-world networks, preferential-attachment, exponential random graphs, and spectral models (as well as subgraph sampling algorithms). One strong point of the library is the extensive support for continuous random variables.

See also the webpage <http://researchers.lille.inria.fr/~freno/JProGraM.html>.

MUTANT Project-Team

5. Software

5.1. Antescofo

Participants: Arshia Cont, Jean-Louis Giavitto, Florent Jacquemard, José Echeveste.

Antescofo is a modular polyphonic Score Following system as well as a Synchronous Programming language for musical composition. The module allows for automatic recognition of music score position and tempo from a realtime audio Stream coming from performer(s), making it possible to synchronize an instrumental performance with computer realized elements. The synchronous language within Antescofo allows flexible writing of time and interaction in computer music.

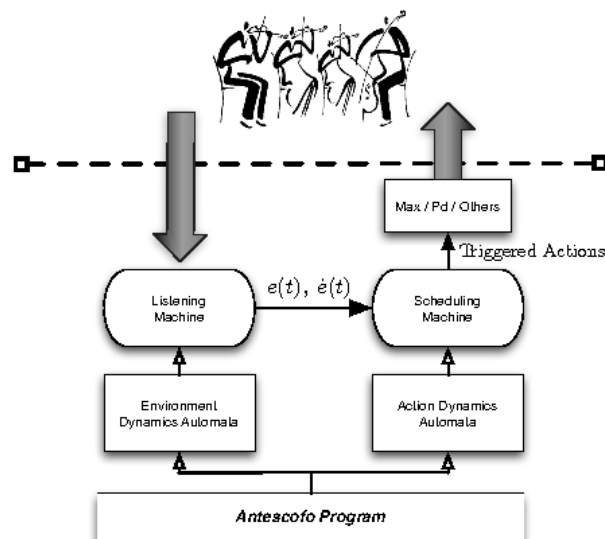


Figure 2. General scheme of Antescofo virtual machine

Antescofo is developed as modules for *Max* and *PureData* real-time programming environments.

5.2. Antescofo Visual Editor

Participants: Thomas Coffy [ADT], Arshia Cont, José Echeveste.

The Antescofo programming language can be extended to visual programming to better integrate existing scores and to allow users to construct complex and embedded temporal structures that are not easily integrated into text. This project has started in October 2012 thanks to Inria ADT Support. The foundations of a visual editor is set and the goal is to release a standalone editor for Antescofo programs in 2013.

MYRIADS Project-Team

5. Software

5.1. SAFDIS

Contact: Jean-Louis Pazat, Jean-Louis.Pazat@irisa.fr

URL: <http://www.irisa.fr/myriads/software/folder.2011-12-13.8949308917/>

Status: Version 1.0

License: TBD

Presentation: SAFDIS (Self Adaptation for Distributed Services) is a generic framework allowing the self-adaptation of distributed service based applications within a highly volatile context. Compared to other adaptation frameworks, the main advantages of SAFDIS are its genericity, its distributed nature and the focus on SOAs. SAFDIS is in its final implementation and testing phase within the Myriads team and is being used with a real life use case for emergency services.

The current implementation of SAFDIS is based on a Java OSGi implementation. SAFDIS is written in Java and organized into OSGi bundles. SAFDIS is not tight to any specific operating system and work within any JAVA 1.6 platform. An OSGi implementation is needed (such as the Apache Felix <http://felix.apache.org> or Equinox eclipse.org/equinox implementations). In order to benefit from the reactive adaptation tools, the Jess engine is also needed as an OSGi bundle (<http://www.jessrules.com>).

Active contributors (from Myriads project-team): Erwan Daubert, Guillaume Gauvrit, Jean-Louis Pazat.

5.2. HOCL-tools

Contact: Cédric Tedeschi, Cedric.Tedeschi@irisa.fr

Status: Version 1.0 to be released

License: TBD

Presentation: HOCL (Higher Order Chemical Language) is a chemical programming language based on the chemical metaphor presented before (see Section 3.5). It was developed for several years within the PARIS team. Within HOCL, following the chemical metaphor, computations can be regarded as chemical reactions, and data can be seen as molecules which participate in these reactions. If a certain condition is held, the reaction will be triggered, thus continuing until it gets inert: no more data can satisfy any computing conditions. To realize this program paradigm, a multiset is implemented to act as a chemical tank, containing necessary data and rules. An HOCL program is then composed of two parts: *chemical rule definitions* (reaction rules) and *multiset definition* (data). More specifically, HOCL provides the high order: reaction rules are molecules that can be manipulated like any other molecules. In other words, HOCL programs can manipulate other HOCL programs.

An HOCL compiler was developed using java to execute some chemical programs expressed with HOCL. This compiler is based on the translation of HOCL programs to java code. As a support for service coordination and service adaptation (refer to Section 6.3), we recently extended the HOCL compiler with the support of decentralized workflow execution. Works around the implementation of a distributed multiset gave birth to an underlying layer for this compiler, making it able to deploy HOCL programs transparently over large scale platforms. This last part is currently considered to be interfaced with the current HOCL compiler. All these features are planned to be released under the common name of *HOCL-tools*.

Active contributors (from Myriads project-team): Héctor Fernández, Marko Obrovac, Cédric Tedeschi.

Impact: The compiler is used as a tool within the team to develop HOCL programs. The decentralized workflow execution support has been used extensively to produce results published and presented at several conferences.

5.3. XtreamOS

Contact: Yvon Jégou, Yvon.Jegou@inria.fr

URL: <http://www.xtreemos.eu>, <http://gforge.inria.fr/projects/xtreemos>

Status: Version 3.0

License: GPL-2/BSD depending on software packages composing the system

Presentation: XTREEMOS is a Grid Operating system based on Linux with native support for virtual organizations. Three flavours of XTREEMOS were developed for individual PCs, clusters and mobile devices (PDA, notebooks and smartphones). XTREEMOS has been developed by the XTREEMOS consortium.

XTREEMOS software is a set of services developed in Java, C++ and C. XTREEMOS cluster version leverages KERRIGHED single system image operating system. A permanent testbed composed of computers provided by several XTREEMOS partners has been public since fall 2010. The third public version of XTREEMOS has been released in February 2012 for the OpenSuse Linux distribution. Ready-to-use XtreamOS virtual machine images have been made available for the community.

Active contributors (from Myriads project-team): Amine Belhaj, Rémy Garrigue, Yvon Jégou, Christine Morin, Yann Radenac.

Impact: XtreamOS software has been used as part of the COOP ANR project. It was also used in the ANR CLOUD project. Some services such as XtreamFS are used in various R&D projects including Contrail European project.

5.4. Contrail Virtual Execution Platform (VEP)

Contact: Yvon Jégou, Yvon.Jegou@inria.fr

URL: http://vep.gforge.inria.fr/index.php?title=Main_Page

Status: Version 1.0

License: BSD

Presentation: Virtual Execution Platform (VEP)[32] is a Contrail service that sits just above IaaS layer at the service provider end of the Contrail cloud federation. The VEP provides a uniform interface for managing the whole lifecycle of elastic applications on the cloud and hides the details of the IaaS layer to the user. VEP applications are described in OVF (Open Virtualization Format) standard format. Resource usage is controlled by CEE (Constrained Execution Environment) rules which can be derived from SLAs (Service Level Agreement). The VEP integrates a monitoring system where the major events about the application, mainly resource usage, are made available to the user.

The VEP service provides a RESTful interface and can be exploited directly by users on top of the provider IaaS. OpenNebula and OCCl-based IaaS interfaces are currently supported.

Active contributors (from Myriads project-team): Roberto Cascella, Florian Dudouet, Filippo Gaudenzi, Piyush Harsh, Yvon Jégou, Christine Morin.

Impact: VEP is part of Contrail software stack. Several Contrail partners experiment use cases on top of VEP. External users can experiment with it using the open testbed operated by Myriads team.

5.5. Snooze

Contact: Christine Morin, Christine.Morin@inria.fr

URL: <http://snooze.inria.fr>

Status: Version 1.0

License: GPLv2

Presentation: Snooze [25], [26], a novel Infrastructure-as-a-Service (IaaS) cloud management system, which is designed to scale across many thousands of servers and virtual machines (VMs) while being easy to configure, highly available, and energy efficient. For scalability, Snooze performs distributed VM management based on a hierarchical architecture. To support ease of configuration and high availability Snooze implements self-configuring and self-healing features. Finally, for energy efficiency, Snooze integrates a holistic energy management approach via VM resource (i.e. CPU, memory, network) utilization monitoring, underload/overload detection and mitigation, VM consolidation (by implementing a modified version of the Sercon algorithm [69]), and power management to transition idle servers into a power saving mode. Snooze is a highly modular Software. It has been extensively evaluated on the Grid'5000 testbed using realistic applications.

Snooze is fully implemented from scratch in Java and currently comprises approximately 15.000 lines of maintainable abstractions-based code. In order to provide a uniform interface to the underlying hypervisors and support transparent VM monitoring and management, Snooze integrates the *libvirt* virtualization library. Snooze provides a RESTful command line interface (CLI) to support virtual cluster (VC) definitions and management (i.e., start, shutdown, destroy, suspend, etc.) as well hierarchy visualization and exporting in GraphML format.

Active contributors (from Myriads team): Eugen Feller, Christine Morin.

Impact: Snooze has been used by students at LIFL, IRIT in France and LBNL in the US in the framework of internships during the summer 2012. It has also been deployed and experimented at EDF R&D. Finally, we know that it was experimented by external users from academia and industry as we received feed-back from them.

5.6. Resilin

Contact: Christine Morin, Christine.Morin@inria.fr

URL: <http://resilin.inria.fr>

Status: Version 1.0

License: GNU Affero GPL

Presentation: Resilin [51], [31] is an open-source system for creating and managing MapReduce execution platforms over clouds. Resilin is compatible with the Amazon Elastic MapReduce (EMR) API, but it goes beyond Amazon's proprietary EMR solution in allowing users (e.g. companies, scientists) to leverage resources from one or more public and/or private clouds. This enables performing MapReduce computations over a large number of geographically-distributed and diverse resources. Resilin can be deployed across most of the open-source and commercial IaaS cloud management systems (e.g., OpenStack, OpenNebula, Amazon EC2). Once deployed, Resilin takes care of provisioning Hadoop clusters and submitting MapReduce jobs, allowing users to focus on writing their MapReduce applications rather than managing cloud resources. Resilin is implemented in the Python language and uses the Apache Libcloud library to interact with IaaS clouds. Resilin has been evaluated on multiple clusters of the Grid'5000 experimentation testbed. The results show that Resilin enables the use of geographically distributed resources with a limited impact on MapReduce job execution time.

Active contributors (from the Myriads project-team): Ancuta Iordache, Nikos Parlavantzas, Christine Morin.

Impact: Resilin is being used in the MOAIS project-team at Inria Grenoble - Rhône Alpes.

5.7. QU4DS

Contact: Jean-Louis Pazat, Jean-Louis.Pazat@inria.fr

URL: <http://www.irisa.fr/myriads/software/quads/>

Status: Version 0.1

License: TBD

Presentation: The QU4DS framework provides PaaS (Platform-as-a-Service) support that fills the gap between the higher-level SaaS (Software-as-a-Service) and the underlying IaaS (Infrastructure-as-a-Service). Qu4DS aids service administrators to define high-level objectives that guide execution management in an automatic and transparent fashion. Moreover, Qu4DS supports the full SLA life-cycle while increasing the service provider profit. SLA support includes service negotiation, instantiation and management on the infrastructure. Orthogonally to these features, complementary actions are in charge of increasing the provider profit guided by SLA constraints.

Currently, Qu4DS targets the development of service providers that use the Master/Worker pattern. Qu4DS assists the development of such services by freeing developers from managing workers and by ensuring their proper execution in accordance with time constraints and by reacting to job failures and delays at runtime. At development time, service developers use the Qu4DS library to develop applications and create a Java jar file. The Qu4DS framework uses this jar file to deploy and manage the service instance on the infrastructure according to SLA constraints.

Active contributors (from Myriads team): André Lage Freitas, Nikos Parlavantzas, Jean-Louis Pazat

5.8. Themis

Contact: Nikos Parlavantzas, Nikos.Parlavantzas@irisa.fr

URL: <http://www.irisa.fr/myriads/software/themis/>

Status: Version 1.0

License: TBD

Presentation: Themis is a market-based private PaaS (Platform-as-a-Service) system, supporting dynamic, fine-grained resource allocation and automatic application management[19], [20]. Themis implements a proportional-share auction that ensures maximum resource utilization while providing incentives to applications to regulate their resource usage. Themis includes generic mechanisms for application deployment and automatic scaling. These mechanisms can be adapted to support diverse performance goals and application types, such as master-worker, MPI, or MapReduce applications. Themis is implemented in Python and uses OpenNebula for virtual machine management. Experimental results on the Grid'5000 testbed show that using Themis increases resource utilization and improves application performance. Themis is currently installed and being evaluated by EDF R&D using EDF high-performance applications.

Active contributors (from the Myriads team): Stefania Costache, Nikos Parlavantzas, Christine Morin.

Impact: Themis is not yet distributed in open source. However, it has been integrated in EDF R&D portal providing access to internal computing resources and is currently experimented on a testbed at EDF R&D.

NACHOS Project-Team

5. Software

5.1. MAXW-DGTD

Participants: Stéphane Lanteri [correspondant], Loula Fezoui.

MAXW-DGTD is a software suite for the simulation of time domain electromagnetic wave propagation. It implements a solution method for the Maxwell equations in the time domain. MAXW-DGTD is based on a discontinuous Galerkin method formulated on unstructured triangular (2D case) or tetrahedral (3D case) meshes [14]. Within each element of the mesh, the components of the electromagnetic field are approximated by a arbitrary high order nodal polynomial interpolation method. This discontinuous Galerkin method combines a centered scheme for the evaluation of numerical fluxes at a face shared by two neighboring elements, with an explicit Leap-Frog time scheme. The software and the underlying algorithms are adapted to distributed memory parallel computing platforms thanks to a parallelization strategy that combines a partitioning of the computational domain with message passing programming using the MPI standard. Besides, a peripheral version of the software has been recently developed which is able to exploit the processing capabilities of a hybrid parallel computing system comprising muticore CPU and GPU nodes [27].

- AMS: AMS 35L50, AMS 35Q60, AMS 35Q61, AMS 65N08, AMS 65N30, AMS 65M60
- Keywords: Computational electromagnetics, Maxwell equations, discontinuous Galerkin, tetrahedral mesh.
- OS/Middelware: Linux
- Required library or software: MPI (Message Passing Interface), CUDA
- Programming language: Fortran 77/95

5.2. MAXW-DGFD

Participant: Stéphane Lanteri [correspondant].

MAXW-DGFD is a software suite for the simulation of time harmonic electromagnetic wave propagation. It implements a solution method for the Maxwell equations in the frequency domain. MAXW-DGFD is based on a discontinuous Galerkin method formulated on unstructured triangular (2D case) or tetrahedral (3D case) meshes. Within each element of the mesh, the components of the electromagnetic field are approximated by a arbitrary high order nodal polynomial interpolation method. The resolution of the sparse, complex coefficients, linear systems resulting from the discontinuous Galerkin formulation is performed by a hybrid iterative/direct solver whose design is based on domain decomposition principles. The software and the underlying algorithms are adapted to distributed memory parallel computing platforms thanks to a parallelization strategy that combines a partitioning of the computational domain with a message passing programming using the MPI standard. Some recent achievements have been the implementation of non-uniform order DG method in the 2D case [18] and of a new hybridizable discontinuous Galerkin (HDG) formulation also in the 2D case [22].

- AMS: AMS 35L50, AMS 35Q60, AMS 35Q61, AMS 65N08, AMS 65N30, AMS 65M60
- Keywords: Computational electromagnetics, Maxwell equations, discontinuous Galerkin, tetrahedral mesh.
- OS/Middelware: Linux
- Required library or software: MPI (Message Passing Interface)
- Programming language: Fortran 77/95

5.3. SISMO-DGTD

Participants: Nathalie Glinsky [correspondant], Stéphane Lanteri.

SISMO-DGTD is a software for the simulation of time domain seismic wave propagation. It implements a solution method for the velocity-stress equations in the time domain. SISMO-DGTD is based on a discontinuous Galerkin method formulated on unstructured triangular (2D case) or tetrahedral (3D case) meshes [5]. Within each element of the mesh, the components of the electromagnetic field are approximated by an arbitrary high order nodal polynomial interpolation method. This discontinuous Galerkin method combines a centered scheme for the evaluation of numerical fluxes at a face shared by two neighboring elements, with an explicit Leap-Frog time scheme. The software and the underlying algorithms are adapted to distributed memory parallel computing platforms thanks to a parallelization strategy that combines a partitioning of the computational domain with a message passing programming using the MPI standard.

- AMS: AMS 35L50, AMS 35Q74, AMS 35Q86, AMS 65N08, AMS 65N30, AMS 65M60
- Keywords: Computational geoseismics, elastodynamic equations, discontinuous Galerkin, tetrahedral mesh.
- OS/Middleware: Linux
- Required library or software: MPI (Message Passing Interface)
- Programming language: Fortran 77/95

5.4. NUM3SIS

Participants: Nora Aissiouene, Thibaud Kloczko [SED³ team], Régis Duvigneau [OPALE project-team], Thibaud Kloczko [SED team], Stéphane Lanteri, Julien Wintz [SED team].

NUM3SIS <http://num3sis.inria.fr> is a modular platform devoted to scientific computing and numerical simulation. It is designed to handle complex multidisciplinary simulations involving several fields such as Computational Fluid Dynamics (CFD), Computational Structural Mechanic (CSM) and Computational ElectroMagnetics (CEM). In this context, the platform provides a comprehensive framework for engineers and researchers that speeds up implementation of new models and algorithms. From a software engineering point of view, num3sis specializes and extends some layers of the meta-platform dtk, especially its core and composition layers. The core layer enables the user to define generic concepts used for numerical simulation such as mesh or finite-volume schemes which are then implemented through a set of plugins. The composition layer provides a visual programming framework that wraps these concepts inside graphical items, nodes. These nodes can then be connected to each other to define data flows (or compositions) corresponding to the solution of scientific problems. NUM3SIS provides a highly flexible, re-usable and efficient approach to develop new computational scenarios and takes advantage of existing tools. The team participates to the development of the NUM3SIS platform through the adaptation and integration of the MAXW-DGTD simulation software. This work is being carried out with the support of two engineers in the framework of an ADT (Action de Développement Technologique) program.

5.5. Medical Image Extractor

Participants: Stéphane Lanteri, Julien Wintz [SED team].

Medical Image Extractor <http://num3sis.inria.fr/software/apps/extractor> provides functionalities needed to extract meshes from labeled MR or PET-CT medical images. It puts the emphasis on consistence, by generating both boundary surfaces, and volume meshes for each label (ideally identifying a tissue) of the input image, using the very same tetrahedrization. As this process requires user interaction, images and meshes are visualized together with tools allowing navigation and both easy and accurate refinement of the generated meshes, that can then be exported to serve as an input for other tools, within a multidisciplinary software toolchain. Using both DTK <http://dtk.inria.fr> and NUM3SIS SDKs, Medical Image Extractor comes within NUM3SIS' framework. Using cutting edge research algorithms developed by different teams at Inria, spread among different research topics, namely, visualization algorithms from medical image processing, meshing algorithms from algorithmic geometry, it illustrates the possibility to bridge the gap between software that come from different communities, in an innovative and highly non invasive development fashion.

³Service d'Experimentation et de Développement

NANO-D Team

5. Software

5.1. SAMSON



Figure 6. SAMSON's architecture.

A major objective of NANO-D is to try and integrate a variety of adaptive algorithms into a unified framework. As a result, NANO-D is developing SAMSON (Software for Adaptive Modeling and Simulation Of Nanosystems), a software platform aimed at including all developments from the group, in particular those described below.

The objective is to make SAMSON a generic application for computer-aided design of nanosystems, similar to existing applications for macrosystem prototyping (CATIA, SolidWorks, etc.).

The current architecture of SAMSON is visible in Figure 6. The code is organized into four main parts: a) the Base (in which “Core” contains, in particular, the heart of the adaptive algorithms: signaling mechanisms specifically designed for SAMSON), b) the Software Development Kit (SDK: a subset of the base that will be provided to module developers), c) Modules, and d) the SAMSON application itself.

Similar to the concept of Mathematica *toolboxes*, for example, the goal has been to make it possible to personalize the user interface of SAMSON for potentially many distinct applications. For example, we may want to personalize the interface of SAMSON for crystallography, drug design, protein folding, electronics, material science, nano-engineering, etc., by loading different modules at startup, depending on the user application domain.

NECS Project-Team

5. Software

5.1. ORCCAD

Participants: Daniel Simon [correspondant], Soraya Arias [SED], Roger Pissard-Gibollet [SED].

ORCCAD is a software environment that allows for the design and implementation of the continuous and discrete time components of complex control systems, e.g. robotics systems which provided it first ground [59]. It also allows the specification and validation of complex missions to be performed by the system. It is mainly intended for critical real-time applications, in which automatic control aspects (servo loops) have to interact narrowly with the handling of discrete events (exception handling, mode switching). ORCCAD offers a complete and coherent vertical solution, ranging from the high level specification to real-time code generation. The ORCCAD V3 software was designed with proprietary tools that moreover are now becoming obsolete. ORCCAD V4 is currently deeply re-engineered to be compliant with open-source and free software tools (Java/Eclipse). Current targets are Linux (Posix threads) and Xenomai, a real-time development framework cooperating with the Linux kernel (<http://www.xenomai.org>). ORCCAD is supported by the *Support Expérimentations & Développement (SED)* service of INRIA-Rhône-Alpes. ORCCAD is used by the experimental robotics platforms of INRIA-Rhône-Alpes and by the Safenecs ANR project in a real-time simulator of a X4 drone. New functionalities and updates are developed jointly by the *SED* service and researchers of the NECS team. Web page: <http://orccad.gforge.inria.fr>.

5.2. MASim

Participants: J. Dumon [contact person], P. Bellemain [GIPSA-Lab], S. Nicolas [PROLEXIA], N. Maciol [PROLEXIA], F. Martinez [ROBOSOFT], J. Caquas [ROBOSOFT].

MASIM is a tool that has been adapted from our former multiagent simulator MUSim (MUSim=MASim + ConnectSim + ConnectIHM). It integrates agent's models, communication media including their limitations, heterogeneous network, and all the variants of the multi-agent control strategies. Besides the models and simulation engine, the simulation can be replayed through a GUI, an interactive graphical interface which is used to visualise and interpret the state of the multi-agent control system and communication topology. The validation scenario is a real-size application enough complex to enforce the pertinence of our results. The simulator MASim is now being used as an open research tool for various applications in the field of multi-agents networked systems, particularly within the FeedNetBack project (see Fig. 4).

Web page: <http://www.gipsa-lab.grenoble-inp.fr/projet/connect/simulator.php>

5.3. GTL

Participants: C. Canudas de Wit [contact person], I. Bellicot [contact person], L. Leon Ojeda, D. Pisarski.

The team has created a software demonstrator to have a showcase including cutting-edge model developments. This software is a global services platform for end-user traffic use, providing accurate density calculation on traffic color vision, and integration of estimator and prediction motor for travel-time calculation. GTL (Grenoble Traffic Lab) is a real-time traffic data center platform intended to collect traffic road infrastructure information in real-time with minimum latency and fast sampling periods. The main elements of the GTL are: a real-time database, a show room, and a suit of traffic forecasting software. Sensed informations come from a dense wireless sensor network providing macroscopic traffic signals such as flows, velocities, densities, and magnetic signatures. This sensor network was set in place in collaboration with Inria spin-off Karrus-ITS, local traffic authorities (DIR-CE, CG38, La Metro), and specialized traffic research centers.

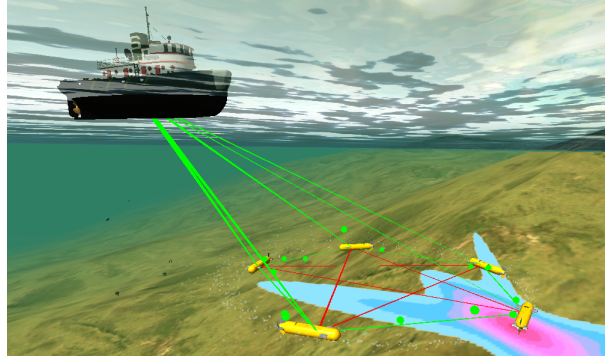


Figure 4. A scenario's view obtained with MASIM.

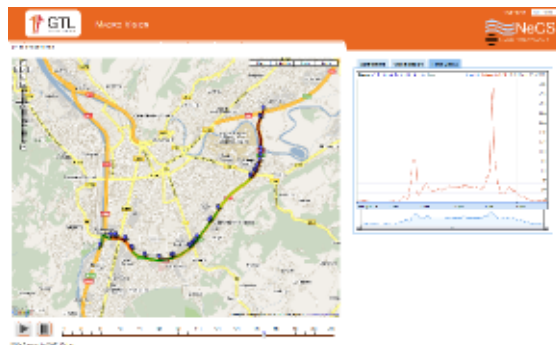


Figure 5. The GTL Macro-simulator.

NEUROMATHCOMP Project-Team

4. Software

4.1. Virtual Retina: A Large-Scale Simulator of Biological Retina

Participants: Bruno Cessac, Maria-Jose Escobar [Universidad Técnica Federico Santa María, Valparaiso, Chile], Christobal Nettle [Universidad Técnica Federico Santa María, Valparaiso, Chile], Pierre Kornprobst [correspondent], Adrien Wohrer [Group for Neural Theory - ENS, Paris, France].

Virtual Retina is a simulation software developed by Adrien Wohrer during his PhD [79], [78] that allows large-scale simulations of biologically-plausible retinas.

Virtual Retina has a variety of biological features implemented such as (i) spatio-temporal linear filter implementing the basic center/surround organization of retinal filtering, (ii) non-linear contrast gain control mechanism providing instantaneous adaptation to the local level of contrast; (iii) spike generation by one or several layers of ganglion cells paving the visual field.

Virtual Retina is under Inria CeCill C open-source licence, so that one can download it, install it and run it on one's own sequences. Virtual Retina also offers a web service (v 2.1), so that users may test directly the main software on their own data, without any installation. This webservice was developed in collaboration with Nicolas Debeissat (engineer, 2002).

We are now interested in the analysis of the collective behavior of ganglion cells responses. To take this collective behavior into account, Virtual Retina needs to be extended since in its current version, ganglion cells are independent. Other evolutions of Virtual Retina are also investigated by external partners like the role/implementation of starburst amacrine cells involved in direction selectivity (collaboration with Universidad Técnica Federico Santa María, Valparaiso, Chile, and Centro de urociencia de Valparaiso) (see also e.g., [70]).

- IDDN number: IDDN.FR.001.210034.000.S.P.2007.000.31235
- Version: v 2.2.2 (September 2011)
- Link: <http://www-sop.inria.fr/neuromathcomp/public/software/virtualretina>

4.2. Event Neural Assembly Simulation

Participants: Bruno Cessac [correspondent], Sélim Kraria [Inria DREAM], Olivier Marre [Institut de la vision, Paris], Hassan Nasser, Thierry Viéville [Inria Mnemosyne Bordeaux].

Enas is a library providing numerical tools for the simulation of neural networks and the analysis of spike trains either coming from neural simulators or from biological experiments.

It is designed mainly as

- An existing simulator plug-in (e.g. MVASpike or other simulators via the NeuralEnsemble meta-simulation platform),
- Additional modules for computations with neural unit assembly on standard platforms (e.g. Python, Matlab or the Scilab platform).
- Original modules for the analysis of spike train statistics intended to be used by the neuroscientists community.

Achievements include:

- Spike trains statistical analysis via Gibbs distributions. They are based on the estimation of a parametric Gibbs potential optimally characterizing the statistics of empirical spike trains (by minimisation of the Kullback-Leibler divergence between the empirical measure and the Gibbs measure). From this, classical statistical indicators such as firing rate, correlations, higher order moments and statistical entropy are obtained. Also, the form of the Gibbs potential provides essential informations on the underlying neural network and its structure. This method does not only allows us to estimate the spikes statistics but also to compare different models, thus answering such questions about the neural code as: are correlations (or time synchrony or a given set of spike patterns, . . .) significant with respect to rate coding?
- Spiking network programming for exact event's sequence restitution;
- Discrete neural field parameters algorithmic adjustments and time-constrained event-based network simulation reconciling clock and event based simulation methods.

Compared to existing libraries Enas offers new computational methods taking into account time constraints in neural networks (such as memory effects), based on theoretical methods rooted in statistical physics and applied mathematics. The algorithms used are based on linear programming, nonlinear parameter estimations, statistical methods. The C/C++ code has been organized as "bean java" to ease its use by programmers non specialized in advanced object programming. As a consequence the code is distributed in the form of an include source for the lightest and the most universal integration into users codes. The standard algorithms are based on the best free libraries in the domain such as gsl <http://www.gnu.org/software/gsl>.

Event neural assembly simulation is developed in gForge. It is under CeCILL C licence
APP logiciel Enas: IDDN.FR.001.360008.000.S.P.2009.000.10600.

Its development as a friendly software designed for the neuroscience community is our present purpose. This is done with the support of an ADT Inria.

Website: <http://enas.gforge.inria.fr/>

NON-A Project-Team (section vide)

NUMED Project-Team

5. Software

5.1. Zebre

Participant: Thierry Dumont [correspondant].

Thierry Dumont is currently developing a toolbox to solve stiff reaction diffusion equations using splitting methods, together with refined numerical schemes for ODEs (RADO 5).

5.2. OptimChemo

Participants: Violaine Louvet [correspondant], Emmanuel Grenier.

OptimChemo is a userfriendly software designed to study numerically the effect of multiple chemotherapies on simple models of tumour growth and to optimize chemotherapy schedules.

OAK Team

5. Software

5.1. Amada

Name: Amada (<https://team.inria.fr/oak/amada/>)

Contact: Jesús Camacho-Rodríguez (jesus.camacho-rodriguez@inria.fr)

Other contacts: Zoi Kaoudi (zoi.kaoudi@inria.fr), Ioana Manolescu (ioana.manolescu@inria.fr), Dario Colazzo (dario.colazzo@lri.fr), François Goasdoué (fg@lri.fr)

Presentation: A platform for Web data management in the Amazon cloud

5.2. Nautilus Analyzer

Name: Nautilus Analyzer (<http://nautilus.saclay.inria.fr/>)

Contact: Melanie Herschel (melanie.herschel@lri.fr)

Other contacts: n.a.

Presentation: A tool for analyzing and debugging SQL queries using why-provenance and why-not provenance.

5.3. RDFViewS

Name: RDFViewS (<http://tripleo.saclay.inria.fr/rdfvs/>)

Contact: Konstantinos Karanasos (konstantinos.karanasos@inria.fr)

Other contacts: François Goasdoué (fg@lri.fr), Julien Leblay (julien.leblay@inria.fr), and Ioana Manolescu (ioana.manolescu@inria.fr)

Presentation: a storage tuning wizard for RDF applications

5.4. ViP2P

Name: ViP2P (views in peer-to-peer, <http://vip2p.saclay.inria.fr>)

Contact: Ioana Manolescu (ioana.manolescu@inria.fr)

Other contacts: Jesús Camacho_Rodriguez (jesus.camacho-rodriguez@inria.fr), Asterios Katsifodimos (asterios.katsifodimos@inria.fr), Konstantinos Karanasos (konstantinos.karanasos@inria.fr)

Presentation: a P2P platform for disseminating and querying XML and RDF data in large-scale distributed networks.

5.5. XUpOp

Name: XUpOp (XML Update Optimization)

Contact: Dario Colazzo (colazzo@lri.fr)

Other contacts: Nicole Bidoit (bidoit@lri.fr), Marina Sahakian (Marina.Sahakyan@lri.fr), and Mohamed Amine Baazizi (baazizi@lri.fr)

Presentation: a general purpose type based optimizer for XML updates

5.6. XUpIn

Name: XUpIn (XML Update Independence)

Contact: Federico Ulliana (Federico.Ulliana@lri.fr)

Other contacts: Dario Colazzo (colazzo@lri.fr), Nicole Bidoit (bidoit@lri.fr)

Presentation: an XML query-update independence tester

5.7. XUpTe

Name: XUpTe (XML Update for Tempora documents)

Contact: Dario Colazzo (colazzo@lri.fr)

Other contacts: Nicole Bidoit (bidoit@lri.fr), Mohamed-Amine Baazizi (amine.baazizi@gmail.com)

Presentation: a type-based optimizer for representing and updated XML temporal sata

5.8. XPUQ

Name: XPUQ (XML Partitioning for Updates and Queries)

Contact: Dario Colazzo (colazzo@lri.fr)

Other contacts: Nicole Bidoit (bidoit@lri.fr), Noor Malla (noorwm@hotmail.com)

Presentation: a static analyzer and partitioner for XML queries and updates

OASIS Project-Team

5. Software

5.1. ProActive

Participants: F. Baude, D. Caromel, L. Henrio, F. Huet [correspondant], F. Viale, O. Smirnov, B. Sauvan, A. Bourdin.

Proactive Parallel Suite

ProActive is a Java library (Source code under AGPL license) for parallel, distributed, and concurrent computing, also featuring mobility and security in a uniform framework. With a reduced set of simple primitives, ProActive provides a comprehensive API to simplify the programming of applications that are distributed on a Local Area Network (LAN), on cluster of workstations, Clouds, or on Internet Grids.

The library is based on an Active Object pattern that is a uniform way to encapsulate:

- a remotely accessible object,
- a thread,
- an actor with its own script,
- a server of incoming requests,
- a mobile and potentially secure agent.

and has an architecture to inter-operate with (de facto) standards such as:

- Web Service exportation (Apache Axis2 and CXF),
- HTTP transport,
- ssh, rsh, RMI/ssh tunnelling,
- Globus: GT2, GT3, GT4, gsi, Unicore, ARC (NordGrid)
- LSF, PBS, Sun Grid Engine, OAR, Load Leveler

ProActive is only made of standard Java classes, and requires **no changes to the Java Virtual Machine**, no preprocessing or compiler modification; programmers write standard Java code. Based on a simple Meta-Object Protocol, the library is itself extensible, making the system open for adaptations and optimisations. ProActive currently uses the RMI Java standard library as default portable transport layer, but others such as Ibis or HTTP can be used instead, in an adaptive way.

ProActive is particularly well-adapted for the development of applications distributed over the Internet, thanks to reuse of sequential code, through polymorphism, automatic future-based synchronisations, migration of activities from one virtual machine to another. The underlying programming model is thus innovative compared to, for instance, the well established MPI programming model.

In order to cope with the requirements of large-scale distributed and heterogeneous systems like the Grid, many features have been incorporated into ProActive, including support for many transport and job submission protocols, GCM component support, graphical visualization interface, object migration, distributed and non-functional exception handling, fault-tolerance and checkpointing mechanisms; file transfer capabilities, a job scheduler, a resource manager able to manage various hosting machines, support for JMX and OSGi capabilities, web service object exposition, an SCA personality, etc.

ProActive is a project of the former ObjectWeb, now OW2 Consortium. OW2 is an international consortium fostering the development of open-source middleware for cutting-edge applications: EAI, e-business, clustering, grid computing, managed services and more. For more information, refer to [5] [55] and to the web pages <http://www.objectweb.org> and <http://proactive.inria.fr/> which list several white papers.

ProActive management, distribution, support, and commercialisation is now ensured by the start-up company ActiveEon (<http://www.activeeon.com>), in the context of a collaboration with Inria and UNS.

5.2. Vercors platform

Participants: E. Madelaine, L. Henrio, A. Savu, M. Alexe.

The Vercors tools (<http://www-sop.inria.fr/oasis/Vercors>) include front-ends for specifying the architecture and behaviour of components in the form of UML diagrams. We translate these high-level specifications, into behavioural models in various formats, and we also transform these models using abstractions. In a final step, abstract models are translated into the input format for various verification toolsets. Currently we mainly use the various analysis modules of the CADP toolset.

- We have pursued last year experiments in distributed model-checking, and were able to generate explicit state-spaces of (sub-systems) for a new distributed use-case of several billion states. The challenges here lie in the structure of the verification workflow, and in finding strategies for separating the sub-systems in an intelligent way.
- We have also conducted intensive experiments within the Papyrus environment, aiming at the definition of a graphical specification environment combining some of the standard UML formalisms (typically class diagrams and state-machines), with a dedicated graphical formalism for the architecture of GCM components.

5.3. Open Simulation Architecture (OSA)

Participants: O. Dalle, V.D. Nguyen.

OSA stands for Open Simulation Architecture. OSA is primarily intended to be a federating platform for the simulation community: it is designed to favor the integration of new or existing contributions at every level of its architecture. The platform core supports discrete-event simulation engine(s) built on top of the ObjectWeb Consortium's Fractal component model. In OSA, the systems to be simulated are modeled and instrumented using Fractal components. In OSA, the event handling is mostly hidden in the controller part of the components, which alleviates noticeably the modeling process, but also eases the replacement of any part of the simulation engine. Apart the simulation engine, OSA aims at integrating useful tools for modeling, developing, experimenting, and analysing simulations. OSA is also a platform for experimenting new techniques and approaches in simulation, such as aspect oriented programming, separation of concerns, innovative component architectures, and so on.

5.4. BtrPlace

Participant: F. Hermenier.

Btrplace (<http://btrp.inria.fr>) is an open source virtual machine (VM) placement algorithm for datacenters. BtrPlace has been designed to be extensible. It can be customized by plugins from third party developers to address new SLAs or optimization objectives. Its extensibility is possible thanks to a composable core reconfiguration algorithm implemented using Constraint Programming.

Btrplace is a part of the OW2 project Entropy. It has been originally developed by Fabien Hermenier during its PhD at the Ecole des Mines of Nantes. BtrPlace is now a standalone project that is currently used to address fault tolerance, isolation, infrastructure management, performance, and energy efficiency concerns inside the national project OpenCloudWare (<http://opencloudware.org/>) and the European project Fit4Green.

This year, our development has been guided by our collaborations. The Fit4Green project chose to rely on BtrPlace to compute an energy-efficient placement for their VMs while some partners inside OpenCloudWare required new placement constraints. The inferring capabilities of BtrPlace and its catalog of placement constraint have then been upgraded accordingly.

OPALE Project-Team

5. Software

5.1. NUM3SIS

Participants: Régis Duvigneau [correspondant], Nora Aïssiouene, Babett Lekouta.

NUM3SIS (<http://num3sis.inria.fr>) is a modular platform devoted to scientific computing and numerical simulation. It is not restricted to a particular application field, but is designed to host complex multidisciplinary simulations. Main application fields are currently Computational Fluid Dynamics (CFD), Computational Electro-Magnetics (CEM, in collaboration with Nachos Project-Team) and pedestrian traffic simulation.

The most important concept in NUM3SIS is the concept of node. It is a visual wrapper around derivatives of fundamental concepts such as data, algorithm or viewer. Atomic nodes are provided for convenience in order to manipulate computational data (such as grids or fields), apply computational methods (such as the building of a finite-element matrix or the construction of a finite-volume flux) and visualize computational results (such as vector or tensor fields, on a screen or in an immersive space). For a given abstract node, different implementations can be found, each of them being embedded in a plugin system that is managed by a factory.

The second important concept in NUM3SIS is the concept of composition. It consists of the algorithmic pipeline used to link the nodes together. The use of these two concepts, composition and nodes, provides a highly flexible, re-usable and efficient approach to develop new computational scenarii and take benefit from already existing tools. This is a great advantage with respect to classical monolithic softwares commonly used in these fields.

This work is being carried out with the support of two engineers in the framework of an ADT (Action de Développement Technologique) program.

5.2. FAMOSA

Participant: Régis Duvigneau [correspondant].

Opale team is developing the software platform FAMOSA (C++), that is devoted to multidisciplinary design optimization in engineering. It integrates the following components:

- an optimization library composed of various algorithms : several descent methods from steepest-descent method to quasi-Newton BFGS method (deterministic, smooth), the Multi-directional Search Algorithm (deterministic, noisy), the Covariance Matrix Adaption Evolution Strategy (semi-stochastic, multi-modal) and the Efficient Global Optimization method (deterministic, multi-modal). It also contains the Pareto Archived Evolution Strategy to solve multi-objective optimization problems ;
- an evaluation library managing the performance estimation process (communication with external simulation tools) ;
- a metamodel library that contains tools to build a database and kriging models that are used to approximate the objective function for different purposes;
- a scenario library that allows to use the previous components to achieve various tasks:
 - Construct a design of experiments ;
 - Construct a metamodel ;
 - Find the design that minimizes a cost functional ;
 - Find the Pareto front for two cost functionals
 - Play a Nash game to find the equilibrium between two criteria ;
 - Apply a multiple gradient descent strategy to improve simultaneously two criteria.

The FAMOSA platform is employed by Opale Project-Team to test its methodological developments in multidisciplinary design optimization (MDO). The platform is also used by the Fluid Mechanics Laboratory at Ecole Centrale de Nantes and by the K-Epsilon company (<http://www.k-epsilon.com>) for hydrodynamic design applications. Moreover, it is presently tested by Peugeot Automotive industry for external aerodynamic design purpose.

5.3. Plugins for AXEL

Participants: Régis Duvingneau [correspondant], Louis Blanchard.

Opale team is developing plugins in the framework of the algebraic modeler Axel, in collaboration with GALAAD team. These developments correspond to two research axes :

- methods for isogeometric analysis and design. In particular, two simulation tools for heat conduction and compressible flows have been implemented, in conjunction with some deterministic and semi-stochastic optimization algorithms for optimum-shape design ;
- methods for geometrical modeling of bow shapes for trawler ships.

5.4. Integration platform for multidiscipline optimization applications

Participants: Toan Nguyen, Laurentiu Trifan.

A prototype software integration platform is developed and tested for multidiscipline optimization applications. It is based on a workflow management system called YAWL (<http://www.yawlfoundation.org>). The goal is to design, develop and assess high-performance distributed scientific workflows featuring resilience, i.e., fault-tolerance and exception-handling capabilities. The platform is used to experiment new resilience algorithms, including monitoring and management of application-level errors. Errors include time-outs and out of bounds data values. They can be added and modified by the users. The platform is tested against use-cases provided by the industry partners in the OMD2 project supported by the French Agence Nationale de la Recherche. For example, an optimization of a car air-conditioning pipe was implemented and deployed on the Grid5000 infrastructure. It also takes into account run-time errors related to resource consumption, e.g., memory overflow, to automatically and dynamically relocate the applications tasks involved on the various clusters. This work is part of Laurentiu Trifan's PhD thesis that is to be defended in 2013. (See Fig. 1 .)

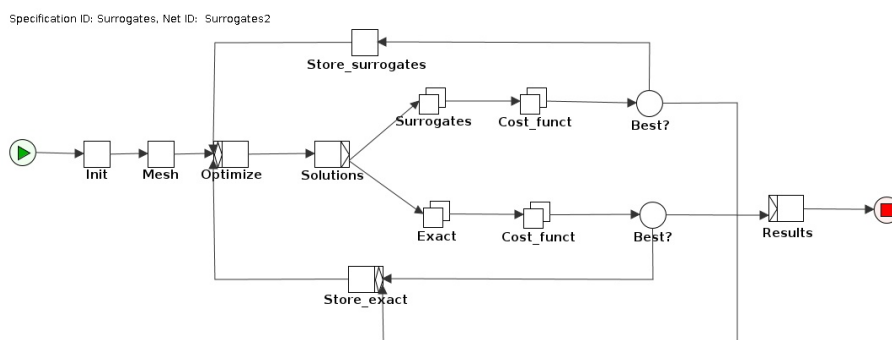


Figure 1. Testcase deployment on the Grid5000 infrastructure.

ORPAILLEUR Project-Team

5. Software

5.1. Generic Symbolic KDD Systems

5.1.1. The Coron Platform

Participants: Victor Codocedo, Adrien Coulet, Amedeo Napoli, Yannick Toussaint, Jérémie Bourseau [contact person].

data mining, frequent itemsets, frequent closed itemsets, frequent generators, association rule extraction, rare itemsets

The Coron platform [120], [102] is a KDD toolkit organized around three main components: (1) Coron-base, (2) AssRuleX, and (3) pre- and post-processing modules. The software was registered at the “Agence pour la Protection des Programmes” (APP) and is freely available (see <http://coron.loria.fr>). The Coron-base component includes a complete collection of data mining algorithms for extracting itemsets such as frequent itemsets, frequent closed itemsets, frequent generators. In this collection we can find APriori, Close, Pascal, Eclat, Charm, and, as well, original algorithms such as ZART, Snow, Touch, and Talky-G. The Coron-base component contains also algorithms for extracting rare itemsets and rare association rules, e.g. APriori-rare, MRG-EXP, ARIMA, and BTB. AssRuleX generates different sets of association rules (from itemsets), such as minimal non-redundant association rules, generic basis, and informative basis. In addition, the Coron system supports the whole life-cycle of a data mining task and proposes modules for cleaning the input dataset, and for reducing its size if necessary. The Coron toolkit is developed in Java, is operational, and was already used in several research projects.

5.1.2. Orion: Skycube Computation Software

Participant: Chedy Raïssi [contact person].

skyline, skycube algorithms

This program implements the algorithms described in a research paper published last year at VLDB 2010 [112]. The software provides a list of four algorithms discussed in the paper in order to compute skycubes. This is the most efficient –in term of space usage and runtime– implementation for skycube computation (see <https://github.com/leander256/Orion>).

5.2. Stochastic systems for knowledge discovery and simulation

5.2.1. The CarottAge system

Participants: Florence Le Ber, Jean-François Mari [contact person].

Hidden Markov Models, stochastic process

The system CarottAge is based on Hidden Markov Models of second order and provides a non supervised temporal clustering algorithm for data mining. It is freely available under GPL license (see <http://www.loria.fr/~jfmari/App/>).

It provides a synthetic representation of temporal and spatial data. CarottAge is currently used by INRA researchers interested in mining the changes in territories related to the loss of biodiversity (projects ANR BiodivAgrim and ACI Ecoger) and/or water contamination. A new version incorporating a graphic user interface was released and is now running on Windows systems.

CarottAge has been used for mining hydromorphological data. Actually a comparison was performed with three other algorithms classically used for the delineation of river continuum and CarottAge proved to give very interesting results for that purpose [17].

5.2.2. *The ARPEnTAge system*

Participants: Florence Le Ber, Jean-François Mari [contact person].

Hidden Markov Models, stochastic process

ARPEnTAge¹ (for *Analyse de Régularités dans les Paysages: Environnement, Territoires, Agronomie* is a software based on stochastic models (HMM2 and Markov Field) for analyzing spatio-temporal data-bases [106]. ARPEnTAge is built on top of the CarottAge system to fully take into account the spatial dimension of input sequences. It takes as input an array of discrete data in which the columns contain the annual land-uses and the rows are regularly spaced locations of the studied landscape. It performs a Time-Space clustering of a landscape based on its time dynamic Land Uses (LUS). Displaying tools and the generation of Time-dominant shape files have also been defined.

We model the spatial structure of the landscape by a Potts model with external field whose sites are LUS located in the parcels. The dynamics of these LUS are modeled by a temporal HMM2. This leads to the definition of a Potts model where the underlying mean field is approximated by a hierarchical hidden Markov model that processes a Hilbert-Peano fractal curve spanning the image.

Those stochastic models have been used to segment the landscape into patches, each of them being characterized by a temporal HMM2. The patch labels, together with the geographic coordinates, determine a clustered image of the landscape that can be coded within an ESRI shapefile. ARPEnTAge can locate in a 2-D territory time regularities and implements a Time-dominant approach in Geographic Information Systems.

ARPEnTAge is freely available (GPL license) and is currently used by INRA researchers interested in mining the changes in territories related to the loss of biodiversity (projects ANR BiodivAgrim and ACI Ecoger) and/or water contamination.

In these practical applications, CarottAge and ARPEnTAge aim at building a partition –called the hidden partition– in which the inherent noise of the data is withdrawn as much as possible. The estimation of the model parameters is performed by training algorithms based on the Expectation Maximization and Mean Field theories. The ARPEnTAge system takes into account: (i) the various shapes of the territories that are not represented by square matrices of pixels, (ii) the use of pixels of different size with composite attributes representing the agricultural pieces and their attributes, (iii) the irregular neighborhood relation between those pixels, (iv) the use of shape files to facilitate the interaction with GIS (geographical information system).

ARPEnTAge and CarottAge have been used for mining decision rules in a territory holding environmental issues. They provide a way of visualizing the impact of farmers decision rules in the landscape and revealing new extra hidden decision rules [23].

5.2.3. *GenExp-LandSiTes: KDD and simulation*

Participants: Sébastien Da Silva, Florence Le Ber [contact person], Jean-François Mari.

simulation, Hidden Markov Models

In the framework of the project “Impact des OGM” initiated by the French ministry of research, we have developed a software called GenExp-LandSiTes for simulating bidimensional random landscapes, and then studying the dissemination of vegetable transgenes. The GenExp-LandSiTes system is linked to the CarottAge system, and is based on computational geometry and spatial statistics. The simulated landscapes are given as input for programs such as “Mapod-Maïs” or “GeneSys-Colza” for studying the transgene diffusion. Other landscape models based on tessellation methods are under studies. The last version of GenExp allows an interaction with R and deals with several geographical data formats.

¹ <http://www.loria.fr/~jfmari/App/>

This work is now part of an INRA research network about landscape modeling, PAYOTE, that gathers several research teams of agronomists, ecologists, statisticians, and computer scientists. Sébastien da Silva is preparing his PhD thesis within this framework and is conducted both by Claire Lavigne (DR in ecology, INRA Avignon) and Florence Le Ber [46], [40].

GenExp-LandSiTes was part of a survey about innovative tools for geographical information [74], [73]. This survey has been conducted within the GDR Magis and has been presented in a book both in French and in English.

5.3. KDD in Systems Biology

5.3.1. IntelliGO online

The IntelliGO measure computes semantic similarity between terms from a structured vocabulary (Gene Ontology: GO) and uses these values for computing functional similarity between genes annotated by sets of GO terms [83]. The IntelliGO measure is made available on line (<http://plateforme-mbi.loria.fr/intelligo/>) to be used by members of the community for exploitation and evaluation purposes. It is possible to compute the functional similarity between two genes, the intra-set similarity value in a given set of genes, and the inter-set similarity value for two given sets of genes.

5.3.2. WAFOBI : KNIME nodes for relational mining of biological data

KNIME (for “Konstanz Information Miner”) is an open-source visual programming environment for data integration, processing, and analysis. KNIME has been developed using rigorous software engineering practices and is used by professionals in both industry and academia. The KNIME environment includes a rich library of data manipulation tools (import, export) and several mining algorithms which operate on a single data matrix (decision trees, clustering, frequent itemsets, association rules...). The KNIME platform aims at facilitating the data mining experiment settings as many tests are required for tuning the mining algorithms. The evaluation of the mining results is also an important issue and its configuration is made easier.

A position of engineer (“Ingénieur Jeune Diplômé Inria”) was granted to the Orpailleur team to develop some extra KNIME nodes for relational data mining using the ALEPH program (<http://www.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/aleph.pl>). The developed KNIME nodes include a data preparation node for defining a set of first-order predicates from a set of relation schemes and then a set of facts from the corresponding data tables (learning set). A specific node allows to configure and run the ALEPH program to build a set of rules. Subsequent nodes allow to test the first-order rules on a test set and to perform configurable cross validations. An Inria APP procedure is currently pending.

5.3.3. Model-driven Data Integration for Mining (MODIM)

Participants: Marie-Dominique Devignes [contact person], Malika Smaïl-Tabbone.

The MODIM software (MModel-driven Data Integration for Mining) is a user-friendly data integration tool which can be summarized along three functions: (i) building a data model taking into account mining requirements and existing resources; (ii) specifying a workflow for collecting data, leading to the specification of wrappers for populating a target database; (iii) defining views on the data model for identified mining scenarios. A steady-version of the software has been deposited through Inria APP procedure in December, 2010.

Although MODIM is domain independent, it was used so far for biological data integration in various internal research studies. A poster was presented at the last JOBIM conference (Paris, June 2011). Recently, MODIM was used by colleagues from the LIFL for organizing data about non ribosomal peptide syntheses. Feedback from users led to extensions of the software. The sources can be downloaded at <https://gforge.inria.fr/projects/modim/>.

5.4. Knowledge-Based Systems and Semantic Web Systems

5.4.1. *The Kasimir System for Decision Knowledge Management*

Participants: Nicolas Jay, Jean Lieber [contact person], Amedeo Napoli, Thomas Meilender.

classification-based reasoning, case-based reasoning, edition and maintenance of knowledge, decision knowledge management, semantic portal

The objective of the Kasimir system is decision support and knowledge management for the treatment of cancer. A number of modules have been developed within the Kasimir system for editing of treatment protocols, visualization, and maintenance. Kasimir is developed within a semantic portal, based on OWL. KatexOWL (Kasimir Toolkit for Exploiting OWL Ontologies, <http://katexowl.loria.fr>) has been developed in a generic way and is applied to Kasimir. In particular, the user interface EdHibou of KatexOWL is used for querying the protocols represented within the Kasimir system.

The software CabamakA (case base mining for adaptation knowledge acquisition) is a module of the Kasimir system. This system performs case base mining for adaptation knowledge acquisition and provides information units to be used for building adaptation rules. Actually, the mining process in CabamakA is implemented thanks to a frequent close itemset extraction module of the Coron platform (see §5.1.1).

The Oncologik system is a collaborative editing tool aiming at facilitating the management of medical guidelines [49], [48]. Based on a semantic wiki, it allows the acquisition of formalized decision knowledge. A production version was released this year (<http://www.oncologik.fr/>). Oncologik also includes a graphical decision tree editor, KcatoS [61].

5.4.2. *Taaable: a system for retrieving and creating new cooking recipes by adaptation*

Participants: Valmi Dufour-Lussier, Emmanuelle Gaillard, Laura Infante Blanco, Florence Le Ber, Jean Lieber, Amedeo Napoli, Emmanuel Nauer [contact person].

knowledge acquisition, ontology engineering, semantic annotation, case-based reasoning, hierarchical classification, text mining

Taaable is a system whose objectives are to retrieve textual cooking recipes and to adapt these retrieved recipes whenever needed. Suppose that someone is looking for a “leek pie” but has only an “onion pie” recipe: how can the onion pie recipe be adapted?

The Taaable system combines principles, methods, and technologies of knowledge engineering, namely case-based reasoning (CBR), ontology engineering, text mining, text annotation, knowledge representation, and hierarchical classification. Ontologies for representing knowledge about the cooking domain, and a terminological base for binding texts and ontology concepts, have been built from textual web resources. These resources are used by an annotation process for building a formal representation of textual recipes. A CBR engine considers each recipe as a case, and uses domain knowledge for reasoning, especially for adapting an existing recipe w.r.t. constraints provided by the user, holding on ingredients and dish types.

The Taaable system is available since 2008 on line at <http://taaable.fr>, but is constantly evolving. This year, Taaable has been extended by two new features, both concerning knowledge acquisition.

The first feature uses closed itemsets for extracting adaptation knowledge in order to better adapt recipes. A first approach integrates a previous work about adaptation rule extraction [93] into a collaborative environment, in which humans and machines may now collaborate to better acquire adaptation rules [38]. This environment integrates also the results of a new work on knowledge extraction where specific cooking adaptation rules that can be applied to a single recipe, are generalized using close itemsets into generic adaptation rules, to make them usable on other recipes [60].

The second feature addresses the improvement of the formal representation of the preparation part of recipes, using a semi-automatic annotation process [59]. In Taaable, the procedural text describing the preparation is formalized in a graph, where cooking actions and ingredients, among others, are represented as vertexes, and semantic relations between those, shown as arcs. As the automatic annotation process that transforms, using natural language processing, a procedural text into a graph, produces incomplete annotation (disconnected graphs) or other annotation errors, a validating and correcting step is required. A specific graphical interface has been built to provide the users with a way to correct the graph representation of the cooking process, improving at the same time the quality of the knowledge about cooking procedures.

5.4.3. Tuurbine: a generic ontology guided case-based inference engine

Participants: Laura Infante Blanco, Jean Lieber, Emmanuel Nauer [contact person].

case-based reasoning, inference engine, knowledge representation, ontology engineering, semantic web

The experience acquired since 5 years with the Taaable system conducted to the creation of a generic case-based reasoning system, whose reasoning procedure is based on a domain ontology. This new system, called Tuurbine, takes into account the retrieval step, the case base organization, but also an adaptation procedure which is not addressed by other generic case-based reasoning tools. Moreover, Tuurbine is built over semantic web standards that will ensure facilities for being plugged over data available on the web. The domain knowledge is considered to be represented in a RDF store, which could be additionally be interfaced with a semantic wiki, in order to benefit from the collaborative edition and management of the knowledge involved in the reasoning system (cases, ontology, adaptation rules). This development is support by an Inria ADT funding.

PAREO Project-Team

5. Software

5.1. ATerm

Participant: Pierre-Etienne Moreau [correspondant].

ATerm (short for Annotated Term) is an abstract data type designed for the exchange of tree-like data structures between distributed applications.

The ATerm library forms a comprehensive procedural interface which enables creation and manipulation of ATerms in C and Java. The ATerm implementation is based on maximal subterm sharing and automatic garbage collection.

A binary exchange format for the concise representation of ATerms (sharing preserved) allows the fast exchange of ATerms between applications. In a typical application—parse trees which contain considerable redundant information—less than 2 bytes are needed to represent a node in memory, and less than 2 bits are needed to represent it in binary format. The implementation of ATerms scales up to the manipulation of ATerms in the giga-byte range.

The ATerm library provides a comprehensive interface in C and Java to handle the annotated term data-type in an efficient manner.

We are involved (with the CWI) in the implementation of the Java version, as well as in the garbage collector of the C version. The Java version of the ATerm library is used in particular by *Tom*.

The ATerm library is documented, maintained, and available at the following address: <http://www.meta-environment.org/Meta-Environment/ATerms>.

5.2. Tom

Participants: Jean-Christophe Bach, Christophe Calvès, Horatiu Cirstea, Pierre-Etienne Moreau [correspondant], Claudia Tavares.

Since 2002, we have developed a new system called *Tom* [33], presented in [17], [18]. This system consists of a pattern matching compiler which is particularly well-suited for programming various transformations on trees/terms and XML documents. Its design follows our experiments on the efficient compilation of rule-based systems [30]. The main originality of this system is to be language and data-structure independent. This means that the *Tom* technology can be used in a C, C++ or Java environment. The tool can be seen as a Yacc-like compiler translating patterns into executable pattern matching automata. Similarly to Yacc, when a match is found, the corresponding semantic action (a sequence of instructions written in the chosen underlying language) is triggered and executed. *Tom* supports sophisticated matching theories such as associative matching with neutral element (also known as list-matching). This kind of matching theory is particularly well-suited to perform list or XML based transformations for example.

In addition to the notion of *rule*, *Tom* offers a sophisticated way of controlling their application: a strategy language. Based on a clear semantics, this language allows to define classical traversal strategies such as *innermost*, *outermost*, *etc.*. Moreover, *Tom* provides an extension of pattern matching, called *anti-pattern matching*. This corresponds to a natural way to specify *complements* (*i.e.* what should not be there to fire a rule). *Tom* also supports the definition of cyclic graph data-structures, as well as matching algorithms and rewriting rules for term-graphs.

Tom is documented, maintained, and available at <http://tom.loria.fr> as well as at <http://gforge.inria.fr/projects/tom>.

PARIETAL Project-Team

5. Software

5.1. Mayavi

Participant: Gaël Varoquaux [Correspondant].

Mayavi is the most used scientific 3D visualization Python software (<http://mayavi.sourceforge.net/>). It has been developed by Prabhu Ramachandran (IIT Bombay) and Gaël Varoquaux (PARIETAL, Inria Saclay). Mayavi can be used as a visualization tool, through interactive command line or as a library. It is distributed under Linux through Ubuntu, Debian, Fedora and Mandriva, as well as in PythonXY and EPD Python scientific distributions. Mayavi is used by several software platforms, such as PDE solvers (fipy, sfepy), molecule visualization tools (<http://pyrx.scripps.edu>) and brain connectivity analysis tools (connectomeViewer).

See also the web page <http://mayavi.sourceforge.net/> and the following paper <http://hal.inria.fr/inria-00528985/en>.

- Version: 3.4.0

5.2. Nipy

Participants: Bertrand Thirion [correspondant], Virgile Fritsch, Elvis Dohmatob, Gaël Varoquaux.

Nipy is an open-source Python library for neuroimaging data analysis, developed mainly at Berkeley, Stanford, MIT and Neurospin. It is open to any contributors and aims at developing code and tools sharing. Some parts of the library are completely developed by Parietal and LNAO (CEA, DSV, Neurospin). It is devoted to algorithmic solutions for various issues in neuroimaging data analysis. All the nipy project is freely available, under BSD license. It is available in NeuroDebian.

See also the web page <http://nipy.org>.

- Version: 0.3

5.3. MedInria

Participants: Pierre Fillard [correspondant], Sergio Medina, Viviana Siless.

MedInria is a free collection of softwares developed within the ASCLEPIOS, ATHENA and VISAGES research projects. It aims at providing to clinicians state-of-the-art algorithms dedicated to medical image processing and visualization. Efforts have been made to simplify the user interface, while keeping high-level algorithms. MedInria is available for Microsoft windows XP/Vista, Linux Fedora Core, MacOSX, and is fully multi-threaded.

See also the web page <http://med.inria.fr/>.

- Version: 2.0

5.4. Scikit learn

Participants: Bertrand Thirion, Gaël Varoquaux [correspondant], Jaques Grobler, Alexandre Gramfort, Fabian Pedregosa, Virgile Fritsch.

Scikit-learn is an open-source machine learning toolkit written in Python/C that provides generic tools to learn information for the classification of various kinds of data, such as images or texts. It is tightly associated to the scientific Python software suite (numpy/scipy) for which it aims at providing a complementary toolkit for machine learning (classification, clustering, dimension reduction, regression). There is an important focus on code quality (API consistency, code readability, tests, documentation and examples), and on efficiency, as the scikit-learn compares favorably to state-of-the-art modules developed in R in terms of computation time or memory requirements. Scikit-learn is currently developed by more than 60 contributors, but the core developer team has been with the Parietal Inria team at Saclay-Île-de-France since January 2010. The scikit-learn has recently become the reference machine learning library in Python.

- Version: 0.12
- Programming language: Python, C/Cython

PARKAS Project-Team

5. Software

5.1. Lucid Sychrone

Participant: Marc Pouzet [contact].

Synchronous languages, type and clock inference, causality analysis, compilation

Lucid Sychrone is a language for the implementation of reactive systems. It is based on the synchronous model of time as provided by Lustre combined with features from ML languages. It provides powerful extensions such as type and clock inference, type-based causality and initialization analysis and allows to arbitrarily mix data-flow systems and hierarchical automata or flows and valued signals.

It is distributed under binary form, at URL <http://www.di.ens.fr/~pouzet/lucid-sychrone/>.

The language was used, from 1996 to 2006 as a laboratory to experiment various extensions of the language Lustre. Several programming constructs (e.g. merge, last, mix of data-flow and control-structures like automata), type-based program analysis (e.g., typing, clock calculus) and compilation methods, originally introduced in Lucid Sychrone are now integrated in the new SCADE 6 compiler developed at Esterel-Technologies and commercialized since 2008.

Three major release of the language has been done and the current version is V3 (dev. in 2006). The language is still used for teaching and in our research but we do not develop it anymore. Nonetheless, we have integrated several features from Lucid Sychrone in new research prototypes described below.

5.2. ReactiveML

Participants: Mehdi Dogguy, Louis Mandel [contact], Cédric Pasteur.

Programming language, synchronous reactive programming, concurrent systems, dedicated type-systems.

ReactiveML is a programming language dedicated to the implementation of interactive systems as found in graphical user interfaces, video games or simulation problems. ReactiveML is based on the synchronous reactive model due to Boussinot, embedded in an ML language (OCaml).

The Synchronous reactive model provides synchronous parallel composition and dynamic features like the dynamic creation of processes. In ReactiveML, the reactive model is integrated at the language level (not as a library) which leads to a safer and a more natural programming paradigm.

ReactiveML is distributed at URL <http://rml.lri.fr>. The compiler is distributed under the terms of the Q Public License and the library is distributed under the terms of the GNU Library General Public License. The development of ReactiveML started at the University Paris 6 (from 2002 to 2006).

The language was mainly used for the simulation of mobile ad hoc networks at the Pierre and Marie Curie University and for the simulation of sensor networks at France Telecom and Verimag (CNRS, Grenoble).

In 2012, a new automatic build system for ReactiveML program based on ocamlbuild has been implemented. A new static analysis which checks that programs cooperate has been developed. A full ReactiveML toplevel compiled into JavaScript has been made available at <http://rml.lri.fr/tryrml>. The ReactiveML distribution has also been cleaned up.

5.3. Heptagon

Participants: Cédric Pasteur [contact], Brice Gelineau, Léonard Gérard, Adrien Guatto, Marc Pouzet.

Synchronous languages, compilation, optimizing compilation, parallel code generation, behavioral synthesis.

Heptagon is an experimental language for the implementation of embedded real-time reactive systems. It is developed inside the Synchronics large-scale initiative, in collaboration with Inria Rhones-Alpes. It is essentially a subset of Lucid Sychrone, without type inference, type polymorphism and higher-order. It is thus a Lustre-like language extended with hierchical automata in a form very close to SCADE 6. The intention for making this new language and compiler is to develop new aggressive optimization techniques for sequential C code and compilation methods for generating parallel code for different platforms. This explains much of the simplifications we have made in order to ease the development of compilation techniques.

Some extensions have already been made, most notably automata. It's currently used to experiment with linear typing for arrays and also to introduce a concept of asynchronous parallel computations. The compiler developed in our team generates C, java and VHDL code.

Heptagon is jointly developed by Gwenael Delaval and Alain Girault from the Inria POP ART team (Grenoble).

5.4. Lucy-n: an n-synchronous data-flow programming language

Participants: Louis Mandel [contact], Adrien Guatto, Marc Pouzet.

Lucy-n is a language to program in the n-synchronous model. The language is similar to Lustre with a buffer construct. The Lucy-n compiler ensures that programs can be executed in bounded memory and automatically computes buffer sizes. Hence this language allows to program Kahn networks, the compiler being able to statically compute bounds for all FIFOs in the program.

The language compiler and associated tools are available in a binary form at <http://www.lri.fr/~mandel/lucy-n>.

In 2012, a first version of the code generator has been distributed. The typing algorithms has been improved.

5.5. ML-Sundials

Participants: Timothy Bourke, Marc Pouzet [contact].

ML-Sundials library provides an Ocaml interface to the Sundials numerical suite ⁷ (version 2.4.0). This library is used for solving and initial value problem and includes a zero-crossing detection mechanism. Only the CVODE solver with serial nectors is currently supported. The structure and naming conventions largely follow the original libraries, both for ease of reading the existing documentation and for converting existing source code, but several changes have been made for programming convenience, namely:

- solver sessions are configured through algebraic data types rather than through multiple function calls,
- error conditions are signalled by exceptions rather than return codes (including in user-supplied callback routines),
- closures (partial applications of higher-order functions) are used to share user data between callback routines, and,
- explicit free commands are not necessary nor provided since Ocaml is a garbage-collected language.

The library is in use in a new synchronous hybrid language we are currently developping.

5.6. GCC

Participants: Albert Cohen [contact], Tobias Grosser, Antoniu Pop, Feng Li, Riyadh Baghdadi, Nhat Minh Le.

Compilation, optimizing compilation, parallel data-flow programming automatic parallelization, polyhedral compilation. <http://gcc.gnu.org>

Licence: GPLv3+ and LGPLv3+

⁷<https://computation.llnl.gov/casc/sundials/main.html>

The GNU Compiler Collection includes front ends for C, C++, Objective-C, Fortran, Java, Ada, and Go, as well as libraries for these languages (libstdc++, libgcj,...). GCC was originally written as the compiler for the GNU operating system. The GNU system was developed to be 100% free software, free in the sense that it respects the user's freedom.

PARKAS contributes to the polyhedral compilation framework, also known as Graphite. We also distribute an experimental branch for a stream-programming extension of OpenMP, parallel data-flow programming, and automatic parallelization to a data-flow runtime or architecture. This experiment borrows key design elements to synchronous data-flow languages.

Tobias Grosser is the maintainer of the Graphite optimization pass of GCC.

5.7. isl

Participants: Sven Verdoolaege [contact], Tobias Grosser, Albert Cohen.

Presburger arithmetic, integer linear programming, polyhedral library, automatic parallelization, polyhedral compilation. <http://freshmeat.net/projects/isl>

Licence: MIT

isl is a library for manipulating sets and relations of integer points bounded by linear constraints. Supported operations on sets include intersection, union, set difference, emptiness check, convex hull, (integer) affine hull, integer projection, transitive closure (and over-approximation), computing the lexicographic minimum using parametric integer programming. It also includes an ILP solver based on generalized basis reduction. isl also supports affine transformations for polyhedral compilation.

5.8. ppcg

Participants: Sven Verdoolaege [contact], Tobias Grosser, Riyadh Baghdadi, Albert Cohen.

Presburger arithmetic, integer linear programming, polyhedral library, automatic parallelization, polyhedral compilation. <http://freshmeat.net/projects/ppcg>

Licence: LGPLv2.1+

More tools are being developed, based on isl. PPCG is our source-to-source research tool for automatic parallelization in the polyhedral model. It serves as a test bet for many algorithms and heuristics published by our group, and is currently the best automatic parallelizer for CUDA (on the Polybench suite).

5.9. Ott: tool support for the working semanticist

Participant: Francesco Zappa Nardelli [contact].

Languages, semantics, tool support, theorem provers.

Ott is a tool for writing definitions of programming languages and calculi. It takes as input a definition of a language syntax and semantics, in a concise and readable ASCII notation that is close to what one would write in informal mathematics. It generates output:

1. a LaTeX source file that defines commands to build a typeset version of the definition;
2. a Coq version of the definition;
3. an Isabelle version of the definition; and
4. a HOL version of the definition.

Additionally, it can be run as a filter, taking a LaTeX/Coq/Isabelle/HOL source file with embedded (symbolic) terms of the defined language, parsing them and replacing them by typeset terms.

The main goal of the Ott tool is to support work on large programming language definitions, where the scale makes it hard to keep a definition internally consistent, and to keep a tight correspondence between a definition and implementations. We also wish to ease rapid prototyping work with smaller calculi, and to make it easier to exchange definitions and definition fragments between groups. The theorem-prover backends should enable a smooth transition between use of informal and formal mathematics.

In collaboration with Peter Sewell (Cambridge University).

The current version of Ott is about 30000 lines of OCaml. The tool is available from <http://moscova.inria.fr/~zappa/software/ott> (BSD licence). It is widely used in the scientific community.

In 2012 we implemented several bug-fixes and we kept the theorem prover backends up-to date with the prover evolution. We have also been working toward a closer integration with the Lem tool.

The currently relased version is 0.21.2.

5.10. Lem: a tool for lightweight executable semantics

Participant: Francesco Zappa Nardelli [contact].

Languages, semantics, tool support, theorem provers.

Lem is a lightweight tool for writing, managing, and publishing large scale semantic definitions. It is also intended as an intermediate language for generating definitions from domain-specific tools, and for porting definitions between interactive theorem proving systems (such as Coq, HOL4, and Isabelle). As such it is a complementary tool to Ott.

Lem resembles a pure subset of Objective Caml, supporting typical functional programming constructs, including top-level parametric polymorphism, datatypes, records, higher-order functions, and pattern matching. It also supports common logical mechanisms including list and set comprehensions, universal and existential quantifiers, and inductively defined relations. From this, Lem generates OCaml, HOL4 and Isabelle code; the OCaml backend uses a finite set library (and does not yet support inductive relations). A Coq backend is in development.

Lem is already in use at Cambridge and Inria for research on relaxed-memory concurrency. We are currently preparing a feature-complete release with back-ends for HOL4, Isabelle/HOL, Coq, OCaml, and LaTeX. The project web-page is <http://www.cl.cam.ac.uk/~so294/lem/>.

In collaboration with Scott Owens (U. Kent, UK) and Peter Sewell (U. Cambridge, UK).

5.11. Cmmtest: a tool for hunting concurrency compiler bugs

Participants: Francesco Zappa Nardelli [contact], Robin Morisset, Pankaj Pawan.

Languages, concurrency, memory models, C11/C++11, compiler, bugs.

The cmmtest tool performs random testing of C and C++ compilers against the C11/C++11 memory model. A test case is any well-defined, sequential C program; for each test case, cmmtest:

1. compiles the program using the compiler and compiler optimisations that are being tested;
2. runs the compiled program in an instrumented execution environment that logs all memory accesses to global variables and synchronisations;
3. compares the recorded trace with a reference trace for the same program, checking if the recorded trace can be obtained from the reference trace by valid eliminations, reorderings and introductions.

Although not yet publicly distributed, cmmtest already identified several mistaken write introductions and other unexpected behaviours in the latest release of the gcc compiler. These have been promptly fixed by the gcc developers.

PAROLE Project-Team

5. Software

5.1. WinSnoori

contact : Yves Laprie (Yves.Laprie@loria.fr)

WinSnoori is a speech analysis software that we have been developing for 15 years. It is intended to facilitate the work of the scientist in automatic speech recognition, phonetics or speech signal processing. Basic functions of Snoori enable several types of spectrograms to be calculated and the fine edition of speech signals (cut, paste, and a number of filters) as the spectrogram allows the acoustical consequences of all the modifications to be evaluated. Beside this set of basic functions, there are various functionalities to annotate phonetically or orthographically speech files, to extract fundamental frequency, to pilot the Klatt synthesizer and to utilize PSOLA resynthesis.

The main improvement concerns automatic formant tracking which is now available with other tools for copy synthesis. It is now possible to determine parameters for the formant synthesizer of Klatt quite automatically. The first step is formant tracking, then the determination of F0 parameters and finally the adjustment of formant amplitudes for the parallel branch of the Klatt synthesizer enable a synthetic speech signal to be generated. The automatic formant tracking that has been implemented is an improved version of the concurrent curve formant tracking [49]. One key point of this tracking algorithm is the construction of initial rough estimates of formant trajectories. The previous algorithm used a mobile average applied onto LPC roots. The window is sufficiently large (200 ms) to remove fast varying variations due to the detection of spurious roots. The counterpart of this long duration is that the mobile average prevents formants fairly far from the mobile average to be kept. This is particularly sensitive in the case of F2 which presents low frequency values for back vowels. A simple algorithm to detect back vowels from the overall spectral shape and particularly energy levels has been added in order to keep extreme values of F2 which are relevant.

Together with other improvements reported during the last years, formant tracking enables copy synthesis. The current version of WinSnoori is available on <http://www.winsnoori.fr>.

5.2. JSnoori

contact : Yves Laprie (Yves.Laprie@loria.fr)

JSnoori is written in Java and uses signal processing algorithms developed within WinSnoori software with the double objective of being a platform independent signal visualization and manipulation tool, and also for designing exercises for learning the prosody of a foreign language. JSnoori thus focused the calculation of F0, the forced alignment of non native English uttered by French speakers and the correction of prosody parameters (F0, rhythm and energy). Since phonetic segmentations and annotations play a central role in the derivation of diagnosis concerning the realization of prosody by learners, several tools have been incorporated to segment and annotate speech. In particular, a complete phonetic keyboard is available, several kinds of annotation can be used (phonemes, syllables and words) and forced alignment can exploit variants to cope with non native accents. In addition, JSnoori offers real time F0 calculation which can be useful from a pedagogical point of view.

5.3. Xarticulator

contact : Yves Laprie (Yves.Laprie@loria.fr)

Xarticulators software is intended to delineate contours of speech articulators in X-ray images, construct articulatory models and synthesize speech from X-ray films. This software provide tools to track contours automatically, semi-automatically or by hand, to make the visibility of contours easier, to add anatomical landmarks to speech articulators and to synchronize images together with the sound.

It also enables the construction of adaptable linear articulatory models from the X-ray images.

This year we particularly worked on the possibility of synthesizing speech from X-ray images. We thus designed an algorithm to compute the centerline of the vocal tract in order to segment the vocal tract into elementary tubes approximating the propagation of a one-dimensional wave. In addition we also added the possibility of processing digitized manual delineation results made on sheet of papers when no software was available

5.4. SUBWEB

contacts : David Langlois (langlois@loria.fr) and Kamel Smaïli (smaïli@loria.fr).

We published in 2007 a method which allows to align sub-titles comparable corpora [50]. In 2009, we proposed an alignment web tool based on the developed algorithm. It allows to: upload a source and a target files, obtain an alignment at a sub-title level with a verbose option, and a graphical representation of the course of the algorithm. This work has been supported by CPER/TALC/SUBWEB ².

5.5. SELORIA

contact : Odile Mella (Odile.Mella@loria.fr).

SELORIA is a toolbox for speaker diarization.

The system contains the following steps:

- Speaker change detection: to find points in the audio stream which are candidates for speaker change points, a distance is computed between two Gaussian modeling data of two adjacent given-length windows. By sliding both windows on the whole audio stream, a distance curve is obtained. A peak in this curve is thus considered as a speaker change point.
- Segment recombination: too many speaker turn points detected during the previous step results in a lot of false alarms. A segment recombination using BIC is needed to recombine adjacent segments uttered by the same speaker.
- Speaker clustering: in this step, speech segments of the same speaker are clustered. Top-down clustering techniques or bottom-up hierarchical clustering techniques using BIC can be used.
- Viterbi re-segmentation: the previous clustering step provides enough data for every speaker to estimate multi-gaussian speaker models. These models are used by a Viterbi algorithm to refine the boundaries between speakers.
- Second speaker clustering step (called cluster recombination): This step uses Universal Background Models (UBM) and the Normalized Cross Likelihood Ratio (NCLR) measure.

This toolbox is derived from mClust designed by LIUM.

5.6. ANTS

contacts : Dominique Fohr (fohr@loria.fr) and Denis Jouvét (denis.jouvet@inria.fr).

The aim of the Automatic News Transcription System (ANTS) is to transcribe radio or TV shows. ANTS is composed of several stages. The first processing steps aim at splitting the audio stream into homogeneous segments of a manageable size and at identifying the segment characteristics in order to allow the use of specific algorithms or models according to the nature of the segment. This includes broad-band/narrow-band speech segmentation, speech/music classification, speaker segmentation and clustering, detection of silences/breathing segments and generally speaker gender classification.

²<http://wikitalc.loria.fr/dokuwiki/doku.php?id=operations:subweb>

Each segment is then decoded using a large vocabulary continuous speech recognition engine, either the Julius engine or the Sphinx engine. The Julius engine operates in two passes: in the first pass, a frame-synchronous beam search algorithm is applied on a tree-structured lexicon assigned with bigram language model probabilities. The output of this pass is a word-lattice. In the second pass, a stack decoding algorithm using a trigram language model gives the N-best recognition sentences. The Sphinx engine processes the speech input segment in a single forward pass using a trigram language model.

Further processing passes are usually run in order to apply unsupervised adaptation processes on the feature computations (VTLN: vocal tract length normalization) and/or on the model parameters (MLLR: maximum likelihood linear regression), or to use speaker adaptive training (SAT) based models. Moreover decoding results of both systems can be efficiently combined for improved decoding performance.

The latest version which relies on a perl script exploits the multiple CPUs available on a computer to reduce the processing time, and runs on both a stand alone linux machine and on the cluster.

5.7. JTrans

Contact : Christophe Cerisara (Christophe.Cerisara@loria.fr).

JTrans is an open-source software for semi-automatic alignment of speech and textual corpus. It is written 100% in JAVA and exploits libraries developed since several years in our team. Two algorithms are available for automatic alignment: a block-viterbi and standard forced-alignment Viterbi. The latter is used when manual anchors are defined, while the former is used for long audio files that do not fit in memory. It is designed to be intuitive and easy to use, with a focus on GUI design. The rationale behind JTrans is to let the user control and check on-the-fly the automatic alignment algorithms. It is bundled for now with a French phonetic lexicon and French models.

Recent improvements include its integration within the JSafran platform and its release as a Java applet that can be demonstrated on web pages. During the last three months, JTrans has been downloaded about 120 times and seven users of JTrans, outside LORIA, have directly contacted the team for requests about JTrans.

JTrans is developed in the context of the CPER MISN TALC project, in collaboration between the Parole and Talaris Inria teams, and CNRS researchers from the ATILF laboratory. It is distributed under the Cecill-C licence, and can be downloaded at <http://synalp.loria.fr/?n=Research.Software>

5.8. CoALT

contacts : Dominique Fohr (dominique.fohr@loria.fr) and Odile Mella (odile.mella@loria.fr).

CoALT (Comparing Automatic Labelling Tools) compares two automatic labellers or two speech-text alignment tools, ranks them and displays statistics about their differences. The main feature of our software is that a user can define its own criteria for evaluating and comparing two speech- text alignment tools. With CoALT, a user can give more importance to either phoneme labels or phoneme boundaries because the CoALT elastic comparison algorithm takes into account time boundaries. Moreover, by providing a set of phonetic rules, a user can define the allowed discrepancies between the automatic labelling result and the hand-labelling one.

5.9. TTS SoJA

contact : Vincent Colotte (Vincent.Colotte@loria.fr).

TTS SoJA (Speech synthesis platform in Java) is a software of text-to-speech synthesis system. The aim of this software is to provide a toolkit to test some steps of natural language processing and to provide a whole system of TTS based on non uniform unit selection algorithm. The software performs all steps from text to the speech signal. Moreover, it provides a set of tools to elaborate a corpus for a TTS system (transcription alignment, ...). Currently, the corpus contains 1800 sentences (about 3 hours of speech) recorded by a female speaker.

Most of the modules are developed in Java. Some modules are in C. The platform is designed to make easy the addition of new modules. The software runs under Windows and Linux (tested on Mandriva, Ubuntu). It can be launch with a graphical user interface or directly integrated in a Java code or by following the client-server paradigm.

The software license should easily allow associations of impaired people to use the software. A demo web site has been built: <http://soja-tts.loria.fr>

5.10. Corpus Recorder

contact : Vincent Colotte (Vincent.Colotte@loria.fr).

Corpus Recorder is a software for the recording of audio corpora. It provides a easy tool to record with a microphone. The gain of the audio input is controlled during the recording. From a list of sentences, the output is a set of wav files automatically renamed with textual information given in input (nationality, speaker language, gender...). An easy syntactic tagging allows to display a textual context of the sentence to pronounce. This software is suitable for recording sentences with information to guide the speaker.

The software is developed in Tcl/Tk (tested under Windows and Linux). It was used for the recording of sentences for the TTS system SOJA and during the Intonale Project (Prosody Modeling).

5.11. VisArtico

contact : Slim Ouni (Slim.Ouni@loria.fr).

VisArtico is intended to visualize articulatory data acquired using an articulograph [30], [29]. It is intended for researchers that need to visualize data acquired from the articulograph with no excessive processing. It is well adapted to the data acquired using the AG500 and AG501 (developed by Carstens Medizinelektronik GmbH), and the articulograph NDI Wave, developed by Northern Digital Inc.

The software allows displaying the positions of the sensors that are simultaneously animated with the speech signal. It is possible to display the tongue contour and the lips contour. The software helps to find the midsagittal plane of the speaker and find the palate contour. In addition, VisArtico allows labeling phonetically the articulatory data.

All this information is very useful to researchers working in the field of speech production, as phoneticians for instance. VisArtico provides several possible views: (1) temporal view, (2) 3D spatial view and (3) 2D midsagittal view. In the temporal view, it is possible to display different articulatory trajectories in addition to the acoustic signal and eventually labels. The midsagittal view can display the tongue contour, the jaw, the lips and the palate.

VisArtico provides several tools to help to improve the quality of interpreting the data. It is cross-platform software as it is developed in JAVA and does not need any external library or framework to be additionally installed. It was tested and worked on Windows, Mac OS, and Linux. It should work on any system having JAVA installed. VisArtico is freely distributed via a dedicated website <http://visartico.loria.fr>.

PARSIFAL Project-Team

5. Software

5.1. Abella

Participants: Kaustuv Chaudhuri [correspondant], Matteo Cimini, Dale Miller.

Abella is an interactive theorem prover based on the two-level logic logic approach. It consists of a sophisticated reasoning logic that supports induction, co-induction, and generic reasoning, and a specification logic that is based on logic programming. Abella was initially designed to reason about simple second-order Lambda Prolog programs, which is sufficient for the computational specifications.

During 2012, as part of the RAPT Associated Team, Chaudhuri and Yuting Wang (intern from Univ. Minnesota) have been working on extending the expressive power of both levels of the Abella system. The following modifications have been made.

- We have extended the specification logic to support the full Lambda Prolog, which can be used to provide succinct higher-order specifications that tend to be unnatural and difficult to reason about with only second-order Lambda Prolog programs.
- We have extended the type system of Abella from simple types to parametrically polymorphic types. This is a significant improvement in the user-friendliness of the system as a lot of code does not have to be manually monomorphised and duplicated any more.
- We have experimented with extending the type system of Abella even further to higher-order predicate quantification. The theoretical basis of this work is a part of ongoing research, although we already have a number of examples of practical benefits of this extension.
- Finally, several improvements have been made to Abella's proof language to make the proofs more robust and reusable. We intend to make a more drastic change to the proof language in the future that will make proofs more declarative and high level.

The core development of Abella has also been centralized, with a single canonical repository and a new webpage: <http://abella-prover.org>. These resources are managed jointly by members of Parsifal and our colleagues at the University of Minnesota.

The next version of Abella, version 2.0, is in beta testing with expected release early in 2013.

5.2. Bedwyr

Participants: Quentin Heath, Dale Miller [correspondant].

During 2012, Quentin Heath has made the following important improvements to the Bedwyr model checking system.

- The concrete syntax for Bedwyr and Abella have been unified. Now, both systems can load the definitions and theorems developed in the other system. Eventually, we expect to have our model checker (Bedwyr) and interactive theorem prover (Abella) share theories and proofs.
- The documentation, distribution, and testing of Bedwyr were all improved, greatly increasing the usability of this system.
- The underlying support for logic has also been increased. In particular, the Bedwyr system contains a *tabling* mechanism which is capable of remembering past successful proofs (it can even support a finite failure as a successful proof of a negation). The most recent version of Bedwyr allows one to actually program the table in rather sophisticated ways. For example, simple lemmas can be loaded into the table and these lemmas can be used to greatly extend the range of what is tabled (remembered). We are currently examining different trade-offs between different styles of reasoning in the table (backchaining vs forwardchaining).

The work of Heath is being done in the context of the BATT ADJ project funded by Inria.

See also the web page <http://slimmer.gforge.inria.fr/bedwyr/>.

5.3. Psyche

Participants: Mahfuza Farooque, Stéphane Graham-Lengrand [correspondant].

Psyche (*Proof-Search factorY for Collaborative HEuristics*) is a modular programme for universal proof-search in classical logic. The motivation is twofold:

On the one hand, prove some mathematics of the broadest range while making the most of problem-specific techniques; On the other hand, gain high confidence about the correctness of the proofs produced without having to rely on a proof-checker.

The architecture is that of an interaction between a trusted universal kernel and smart plugins that are meant to be efficient at solving certain kinds of problems:

The kernel contains the mechanisms for exploring the proof-search space in a sound and complete way, taking into account branching and backtracking. The output of Psyche comes from the (trusted) kernel and is therefore correct by construction. The plugins then drive the kernel by specifying how the branches of the search space should be explored, depending on the kind of problem that is being treated. The quality of the plugin is then measured by how fast it drives the kernel towards the final answer.

Version 1.0 of Psyche (released 4/9/2012) handles classical propositional logic, and its proof-search mechanism is simply the incremental construction of proof-trees in the polarised and focussed sequent calculus. The mechanism is driven by a plugin that emulates the behaviour of a SAT-solver (DPLL), with non-trivial features such as the eager application of the Unit Propagation rule, conflict analysis, backjumping and clause learning.

Psyche's input for that kind of SAT-problem is a file given in the standard DIMACS format.

See also the web page <http://www.lix.polytechnique.fr/~lengrand/PSI/index.php?page=Psyche/index>.

PERCEPTION Team

5. Software

5.1. Mixed camera platform

We started to develop a multiple camera platform composed of both high-definition color cameras and low-resolution depth cameras. This platform combines the advantages of the two camera types. On one side, depth (time-of-flight) cameras provide relatively accurate 3D scene information. On the other side, color cameras provide information allowing for high-quality rendering. The software package developed during the year 2011 contains the calibration of TOF cameras, alignment between TOF and color cameras, and image-based rendering. These software developments are performed in collaboration with the Samsung Advanced Institute of Technology. The multi-camera platform and the basic software modules are products of 4D Views Solutions SAS, a start-up company issued from the PERCEPTION group.



Figure 3. The mixed multi-camera system composed of four TOF-stereo sensor units.

5.2. Audiovisual robot head

We have developed two audiovisual (AV) robot heads: the POPEYE head and the NAO stereo head. Both are equipped with a binocular vision system and four microphones. The software modules comprise stereo matching and reconstruction, sound-source localization and audio-visual fusion. POPEYE has been developed within the European project POP (<http://perception.inrialpes.fr/POP>) in collaboration with the project-team MISTIS and with two other POP partners: the Speech and Hearing group of the University of Sheffield and the Institute for Systems and Robotics of the University of Coimbra. The NAO stereo head is being developed under the European project HUMAVIPS (<http://humavips.inrialpes.fr>) in collaboration with Aldebaran Robotics (which manufactures the humanoid robot NAO) and with the University of Bielefeld, the Czech Technical Institute, and IDIAP. The software modules that we develop are compatible with both these robot heads.

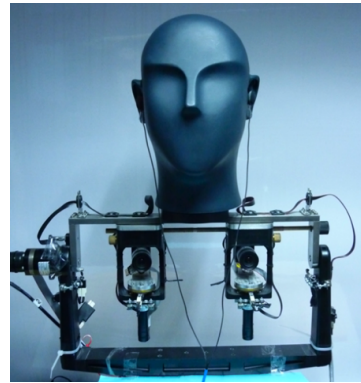


Figure 4. Left: The consumer humanoid robot NAO is equipped with a binocular-binaural head specially designed for human-humanoid interaction; Right: The binocular-binaural robot head POPEYE equipped with a four degrees of freedom stereo camera pair and with a dummy head.

PHOENIX Project-Team

5. Software

5.1. DiaSuite: a Development Environment for Sense/Compute/Control

Applications

Participants: Charles Consel [correspondent], Julien Bruneau, Amélie Marzin, Damien Martin-Guillerez, Emilie Balland.

Despite much progress, developing a pervasive computing application remains a challenge because of a lack of conceptual frameworks and supporting tools. This challenge involves coping with heterogeneous devices, overcoming the intricacies of distributed systems technologies, working out an architecture for the application, encoding it in a program, writing specific code to test the application, and finally deploying it.

DIASUITE is a suite of tools covering the development life-cycle of a pervasive computing application:

- *Defining an application area.* First, an expert defines a catalog of entities, whether hardware or software, that are specific to a target area. These entities serve as building blocks to develop applications in this area. They are gathered in a taxonomy definition, written in the taxonomy layer of the DIASPEC language.
- *Designing an application.* Given a taxonomy, the architect can design and structure applications. To do so, the DIASPEC language provides an application design layer [40]. This layer is dedicated to an architectural pattern commonly used in the pervasive computing domain [30]. Describing the architecture application allows to further model a pervasive computing system, making explicit its functional decomposition.
- *Implementing an application.* We leverage the taxonomy definition and the architecture description to provide dedicated support to both the entity and the application developers. This support takes the form of a Java programming framework, generated by the DIAGEN compiler. The generated programming framework precisely guides the developer with respect to the taxonomy definition and the architecture description. It consists of high-level operations to discover entities and interact with both entities and application components. In doing so, it abstracts away from the underlying distributed technologies, providing further separation of concerns.
- *Testing an application.* DIAGEN generates a simulation support to test pervasive computing applications before their actual deployment. An application is simulated in the DIASIM tool, without requiring any code modification. DIASIM provides an editor to define simulation scenarios and a 2D-renderer to monitor the simulated application. Furthermore, simulated and actual entities can be mixed. This hybrid simulation enables an application to migrate incrementally to an actual environment.
- *Deploying a system.* Finally, the system administrator deploys the pervasive computing system. To this end, a distributed systems technology is selected. We have developed a back-end that currently targets the following technologies: Web Services, RMI, SIP and OSGI. This targeting is transparent for the application code. The variety of these target technologies demonstrates that our development approach separates concerns into well-defined layers.

This development cycle is summarized in the Figure 1 .

See also the web page <http://diasuite.inria.fr>.

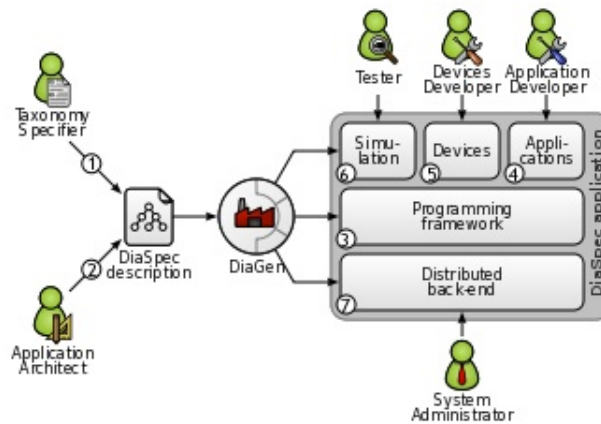


Figure 1. DIASUITE Development Cycle

5.1.1. DiaSpec: a Domain-Specific Language for Networked Entities

The core of the DIASUITE development environment is the domain specific language called DIASPEC and its compiler DIAGEN:

- DIASPEC is composed of two layers:
 - The *Taxonomy Layer* allows the declaration of entities that are relevant to the target application area. An entity consists of sensing capabilities, producing data, and actuating capabilities, providing actions. Accordingly, an entity description declares a data source for each one of its sensing capabilities. As well, an actuating capability corresponds to a set of method declarations. An entity declaration also includes attributes, characterizing properties of entity instances. Entity declarations are organized hierarchically allowing entity classes to inherit attributes, sources and actions. A taxonomy allows separation of concerns in that the expert can focus on the concerns of cataloging area-specific entities. The entity developer is concerned about mapping a taxonomical description into an actual entity, and the application developer concentrates on the application logic.
 - The *Architecture Layer* is based on an architectural pattern commonly used in the pervasive computing domain [30]. It consists of context components fueled by sensing entities. These components process gathered data to make them amenable to the application needs. Context data are then passed to controller components that trigger actions on entities. Using an architecture description enables the key components of an application to be identified, allowing their implementation to evolve with the requirements (e.g., varying light management implementations in a controller component to optimize energy consumption).
- DIAGEN is the DIASPEC compiler that performs both static and runtime verifications over DIASPEC declarations and produces a dedicated programming framework that guides and eases the implementation of components. The generated framework is independent of the underlying distributed technology. As of today, DIAGEN supports multiple targets: Local, RMI, SIP, Web Services and OSGI.

5.1.2. DiaSim: a Parametrized Simulator for Pervasive Computing Applications

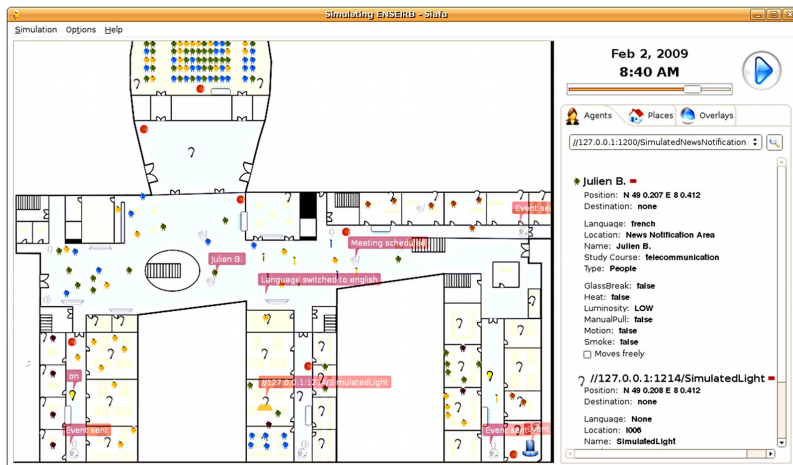


Figure 2. A screenshot of the DIASIM simulator

Pervasive computing applications involve both software and integration concerns. This situation is problematic for testing pervasive computing applications because it requires acquiring, testing and interfacing a variety of software and hardware entities. This process can rapidly become costly and time-consuming when the target environment involves many entities.

To ease the testing of pervasive applications, we are developing a simulator for pervasive computing applications: DIASIM. To cope with widely heterogeneous entities, DIASIM is parameterized with respect to a DIASPEC specification describing a target pervasive computing environment. This description is used to generate with DIAGEN both a programming framework to develop the simulation logic and an emulation layer to execute applications. Furthermore, a simulation renderer is coupled to DIASIM to allow a simulated pervasive system to be visually monitored and debugged. The simulation renderer is illustrated in Figure 2.

5.2. DiaSuiteBox: an Open Service Platform

Participants: Julien Bruneau [correspondent], Damien Martin-Guillerez, Charles Consel, Emilie Balland.

The DiaSuiteBox platform runs an open-ended set of applications leveraging a range of appliances and web services. Our solution consists of a dedicated development environment, a certifying application store, and a lightweight runtime platform. This solution is based on the DIASUITE project.

The DiaSuiteBox platform can be embedded in a small plug-computer or deployed in the cloud. Thanks to the application store and the developer community, the platform is fed by a full offer of new innovative applications. During the submission process, an application is automatically analyzed and checked in order to be certified. The user is ensured of the behavior of its applications are innocuous and correct beside the provided information. This box relies on several technology standards like UPnP, Bluetooth, USB, etc. As shown in Figure 3, this platform can be easily extended by plugging appliances directly on the box or by connecting devices on the local network.

See also the web page <http://diasuitebox.inria.fr>.

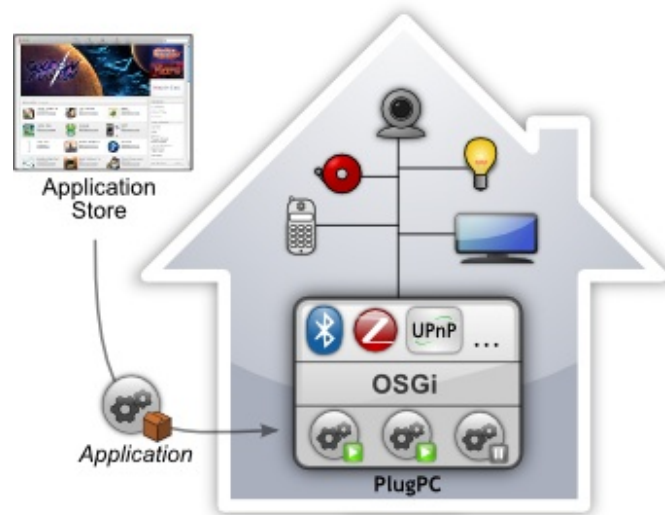


Figure 3. DiaSuiteBox platform architecture

PI.R2 Project-Team

5. Software

5.1. <http://coq.inria.fr> COQ

Participants: Bruno Barras [TypiCal team, Saclay], Yves Bertot [Marelle team, Sophia], Pierre Boutillier, Xavier Clerc [SED team], Pierre Courtieu [CNAM], Maxime Dénès [Marelle team, Sophia], Julien Forest [CNAM], Stéphane Glondu [CAMEL team, Nancy Grand Est], Benjamin Grégoire [Marelle team, Sophia], Vincent Gross [Consultant at NBS Systems], Hugo Herbelin [correspondant], Pierre Letouzey, Assia Mahboubi [TypiCal team, Saclay], Julien Narboux [University of Strasbourg], Jean-Marc Notin [TypiCal team, Saclay], Christine Paulin [Proval team, Saclay], Pierre-Marie Pédrot, Loïc Pottier [Marelle team, Sophia], Matthias Puech, Yann Régis-Gianas, François Ripault, Matthieu Sozeau, Arnaud Spiwack, Pierre-Yves Strub [Formes team, Beijing], Enrico Tassi [TypiCal team, Saclay], Benjamin Werner [TypiCal team, Saclay].

5.1.1. *Version 8.4*

Version 8.4 was released in August 2012. It introduced a new proof engine designed and implemented by Arnaud Spiwack and a new extensive modular library of arithmetic contributed by Pierre Letouzey. It also included an extension of the underlying logic with “ η -conversion” by Hugo Herbelin and “commutative-cuts compliant guard condition” by Pierre Boutillier, an extension of the pattern-matching compilation algorithm by Hugo Herbelin, an extension of the procedure of simplification of polynomial expressions by Loïc Pottier, a refinement of the type classes mechanism by Matthieu Sozeau, a new communication model by Vincent Gross for the graphical user interface CoqIDE, that Pierre Letouzey, Pierre Boutillier and Pierre-Marie Pédrot further extended.

Several users gracefully contributed improvements of various features (Tom Prince, Enrico Tassi, Daniel Grayson, Hendrik Tews, ...).

5.1.2. *Graphical user interface*

Pierre Letouzey has finalised and extended the work initiated by Vincent Gross (former ADT engineer) concerning the CoqIDE user interface: CoqIDE and Coq are now separate Unix processes, enhancing the reliability and improving the user experience.

In Fall 2012, Pierre Letouzey also revised the event infrastructure of CoqIDE, from a thread-based model to pure GTK event-loop. This way, CoqIDE is more reactive, less subject to deadlocks (especially under Windows), and the source code is more idiomatic and easier to understand. Interestingly, this work takes advantage of deeper notions such as C.P.S. (continuation passing style).

Pierre Boutillier and Pierre-Marie Pédrot built an abstract communication interface between Coq and CoqIDE based on XML syntax. They also refined the ability to customise CoqIDE. Pierre Boutillier made CoqIDE rely on Gtksourceview.

5.1.3. *Proof engine*

Arnaud Spiwack has proposed an extension of the expressiveness of tactics based on his previous work for a new proof engine. It allows for more atomic tactics, has a primitive support for backtracking, and allows for tactics which manipulate several goals.

5.1.4. *Evaluation algorithms*

Pierre Boutillier has proposed a new unfolding algorithm for global constants so that the definition of these ones are unfolded only if it triggers extra reductions. This helps users to keep goals concise during interactive proofs.

5.1.5. Type classes, internal representation

Matthieu Sozeau is adapting the type-classes mechanism to benefit from the new tactic engine and avoid reimplementing a whole proof-search algorithm with backtracking on top of the tactic language. This will bring high benefits in terms of efficiency and ease of use to the users. Forward proof-search for type classes was stabilised and is now used in libraries for better control on the search space, notably in the `MathClasses` library developed in Nijmegen [69].

An important shortcoming of type classes is the verbosity of the representation of projections from a class, as was illustrated in François Garillot's PhD thesis [48]. Matthieu Sozeau has developed a branch of Coq supporting an efficient representation of these projections based on the idea of bidirectional type checking which is now under stabilisation. This support will also enhance the performance of the assistant on developments using regular parameterised records and dependent sums like the HoTT library on homotopy type theory and the Forcing plugin developed by Sozeau *et al* [32].

5.1.6. Universes

While visiting the Institute for Advanced Study, Matthieu Sozeau implemented a new system of universe polymorphism that makes it possible to develop highly generic theories in the Coq system. Based on ideas from Harper and Pollack's [53] design of polymorphism as an elaboration in the Lego theorem prover, he developed an original algorithm for type inference of universes and implemented it in Coq. Its first application is inside the Homotopy Type Theory (HoTT) research program, as the formalisation of HoTT requires a high level of polymorphism that was not available before. Many other theories will benefit from this, including Sozeau's work on Forcing, B. Barras' (Typical) work on models of type theory or in the `Math Classes` library mentioned before. It also opens up possibilities to formalise category-theoretic notions without being limited by the universe system, a long-standing barrier in the Coq proof assistant.

5.1.7. The Equations plugin

Matthieu Sozeau continued the maintenance of the Equations plugin and developed a new Forcing plugin for Coq (see below).

5.1.8. Tools

Pierre-Marie Pédrot has written a program using the internal representation of libraries to compact Coq object files. It is based on well-known automata algorithms, representing memory as transition systems. The idea underlying this program is generalisable to any OCaml data structure, provided some conditions on its use are satisfied, and was formalised in a paper that was accepted at JFLA 2013.

5.1.9. Internal architecture of the Coq software

Pierre Letouzey continued a large reorganisation of the internal components of Coq, since these components are currently too much interdependent. This work brought better isolation between some of the Coq components and explicit interfaces between them. This allowed to simplify the compilation of Coq, since it is now easier to build the OCaml syntax extension used when compiling many advanced parts of Coq. Moreover, this clearer architecture should also help new contributors when they discover and interact with this large and complex code-base.

Pierre-Marie Pédrot also made some reorganisations of the code. This includes a clean generic library superseding the one of OCaml, pushing the CAMLP4/5 dependent parts out of the lower strata, as well as benefiting from the OCaml module system to get more uniformity in the naming of usual data structures.

Pierre Letouzey proposed a nicer backtracking infrastructure to Coq, used when the user wants to cancel some recent commands and go back before them. This new infrastructure unifies and improves what was used earlier by ProofGeneral and CoqIDE, the two main user interfaces for Coq.

Pierre Letouzey also dedicated many efforts to improve the support of the Windows platform by Coq.

5.1.10. Efficiency

Pierre-Marie Pédrot has been trying to optimise various parts of the Coq system, including the new tactical system designed by Arnaud Spiwack. Some neat tricks on garbage collection permitted to reach a substantial time improvement in compilation of object files. Various architectural modifications were also made in the process, like trying to get rid of the generic comparison in the code base.

Pierre Letouzey investigated an alternative implementation of the code dealing with Coq universes. These universes are a critical part of Coq: they have direct consequences on Coq safety, and handling them is time-consuming (between 10% to 20% depending on the Coq usage). This alternative implementation looks promising, but still requires some more work and stress-tests before being integrated in mainstream Coq.

5.1.11. Documentation generation

François Ripault and Yann Régis-Gianas developed a new version of coqdoc, the documentation generator of Coq. This new implementation is based on the interaction protocol with the Coq system and should be more robust with respect to the evolution of Coq.

5.1.12. General maintenance

Pierre Letouzey has been the main maintainer of Coq with extra contributions from Hugo Herbelin, Pierre Boutillier, Matthieu Sozeau, Pierre-Marie Pédrot, ...

5.1.13. Development Action

An “Action de Développement Technologique” about Coq started September 2011 and continued this year. It gathers the πr^2 team, the Marelle team and the CPR team from CNAM, Hugo Herbelin acting as the coordinator. It supports visits and meetings between developers and aims at strengthening the community of Coq users and contributors.

Yann Régis-Gianas set up an “osqa” server for Frequently Asked Questions.

The ADT Coq supported the internship of François Ripault.

Hugo Herbelin formalised a type-theoretic construction of semi-simplicial sets answering a problem raised early this year by Steve Awodey, Peter LeFanu Lumsdaine and others, in relation with the homotopy models of type theory.

5.2. Pangolin

Participant: Yann Régis-Gianas.

Yann Régis-Gianas maintained a prototype version of Pangolin. He used it to prove concrete complexity bounds for a set of functional programs using the method described in his FOPARA 2011 paper [19].

5.3. Other software developments

In collaboration with François Pottier (Inria Gallium), Yann Régis-Gianas maintained Menhir, an LR parser generator for OCaml.

PLANETE Project-Team

5. Software

5.1. ns-3

Participant: Daniel Camara [correspondant].

ns-3 is a discrete-event network simulator for Internet systems, targeted primarily for research and educational use. ns-3 is free software, licensed under the GNU GPLv2 license, and is publicly available for research, development, and use. ns-3 includes a solid event-driven simulation core as well as an object framework focused on simulation configuration and event tracing, a set of solid 802.11 MAC and PHY models, an IPv4, UDP, and TCP stack and support for nsc (integration of Linux and BSD TCP/IP network stacks).

See also the web page <http://www.nsnam.org>.

- Version: ns-3.7
- Keywords: networking event-driven simulation
- License: GPL (GPLv2)
- Type of human computer interaction: programmation C++/python, No GUI
- OS/Middleware: Linux, cygwin, osX
- Required library or software: standard C++ library: GPLv2
- Programming language: C++, python
- Documentation: doxygen

5.2. EphPub

Participants: Mohamed Ali Kaafar [correspondant], Claude Castelluccia.

EphPub (Ephemeral Publishing) (previously called EphCom) implements a novel key storage mechanism for time-bounded content, that relies on the caching mechanism of the Domain Name System (DNS). Features of EphPub include: EphPub exploits the fact that DNS servers temporarily cache the response to a recursive DNS query for potential further requests. EphPub provides higher security than Vanish, as it is immune to Sybil attacks. EphPub is easily deployable and does not require any additional infrastructure, such as Distributed Hash Tables. EphPub comes with high usability as it does not require users to install and execute any extra additional software. EphPub lets users define data lifetime with high granularity. We provide EphPub as an Android Application to provide ephemeral exchanged SMS, emails, etc. and as a Firefox or Thunderbird extensions so as to support ephemeral publication of any online document.

For more details about the different software products, see <http://planete.inrialpes.fr/projects/ephemeral-publication/>.

- Version: v0.1.2-beta
- ACM: K.4.1
- AMS: 94Axx
- Keywords: Ephemeral communications, Right to Forget, Future Internet Architecture, Privacy
- Software benefit: We provide a Firefox Extension that easily allows users to manage disappearing emails. We also provide a command-line tool to manage disappearing files.
- APP: Under APP deposit internal process
- License: GPL
- Type of human computer interaction: Firefox extension + Unix Console
- OS/Middelware: Firefox under any OS
- Required library or software: Python Ext
- Programming language: Python
- Documentation: No detailed documentation has been released so far. A detailed howto can be consulted however at: http://code.google.com/p/disappearingdata/source/browse/wiki/EphCOM_Firefox_Extension.wiki?r=77

5.3. Username Tester

Participants: Claude Castelluccia [correspondant], Mohamed Ali Kaafar, Daniele Perito.

Usernames are ubiquitous on the Internet. Almost every web site uses them to identify its users and, by design, they are unique within each service. In web services that have millions or hundreds of millions of users, it might become difficult to find a username that has not already been taken. For instance, you might have experienced that a specific username you wanted was already taken. This phenomenon drives users to choose increasingly complex and unique usernames.

We built a tool to estimate how unique and linkable usernames are and made it available on this page for you to check. For example, according to our tool, “ladygaga” or “12345678” only carry 24 and 17 bits of entropy, respectively. They are therefore not likely to be unique on the Internet. On the other hand, usernames such as “pdjkwelr!” or “yourejerky” carry about 40 bits of entropy and are therefore very good identifiers.

Type your username (for example “zorro1982” or “dan.perito”) to discover how unique it is. This tool can help you to select an username that has low entropy and can’t be used to track you on the Internet.

Alternatively, try typing two usernames separated by a space. The tool will give an estimation on whether the two usernames are linkable. The tool is accessible here: <http://planete.inrialpes.fr/projects/how-unique-are-your-usernames/>

5.4. DroidMonitor

Participants: Claude Castelluccia [correspondant], Mohamed Ali Kaafar.

In nowadays world the technological progress evolves very quickly. There are more and more new devices, fully equipped with the latest innovations. The question is: do we adopt our main privacy concerns according to these new technologies as quickly as they grow and become widely available for us?...

We developed a novel tool, private data leakage monitoring tool, DroidMonitor. It aims to serve as an educational tool for regular Android Smartphones users to make them aware of existing privacy threats while they are using Location-Based Services. It can be downloaded here: <http://planete.inrialpes.fr/android-privacy/>

5.5. NEPI

Participants: Thierry Turletti [correspondant], Alina Quereilhac.

NEPI stands for Network Experimentation Programming Interface. NEPI implements a new experiment plane used to perform ns-3 simulations, planetlab and emulation experiments, and, more generally, any experimentation tool used for networking research. Its goal is to make it easier for experimenters to describe the network topology and the configuration parameters, to specify trace collection information, to deploy and monitor experiments, and, finally, collect experiment trace data into a central datastore. NEPI is a python API (with an implementation of that API) to perform all the above-mentioned tasks and allows users to access these features through a simple yet powerful graphical user interface called NEF.

During the year 2012 we improved support for PlanetLab experiments in NEPI, adding the ability to create customized routing overlays on top of PlanetLab. Details on these improvements can be found in [48]. We also included the ability to easily conduct CCNx <http://www.ccnx.org/> experiments using PlanetLab nodes. This work was presented at the CCNx 2012 community meeting [73], and has had a good impact on the number of NEPI users.

Additionally, ongoing work on the context of the Openlab, Fed4Fire and Simulbed projects, has lead to a number of interesting extensions to NEPI. We are currently developing support to conduct experiments on OMF wireless testbeds (<http://mytestbed.net/>). We are also working to support DCE enabled experimentation, using the ns-3 simulator, in NEPI. Furthermore, recent work on improving NEPI’s experiment control architecture, to enable both easier extension to new experimentation platforms and improve the user ability to control of experiment tasks, was presented at the CoNEXT’12 Students Workshop (see [61]).

For more information, see also the web page <http://nepi.inria.fr>.

- Version: 2.0
- ACM: C.2.2, C.2.4
- Keywords: networking experimentation
- License: GPL (2)
- Type of human computer interaction: python library, QT GUI
- OS/Middleware: Linux
- Required library or software: python – <http://www.python.org> – <http://rpyc.sourceforge.net>
- Programming language: python

5.6. Reference implementation for SFA Federation of experimental testbeds

Participants: Thierry Parmentelat [correspondant], Julien Tribino.

We are codevelopping with Princeton University a reference implementation for the Testbed-Federation architecture known as SFA for Slice-based Federation Architecture. During 2011 we have focused on the maturation of the SFA codebase, with several objectives in mind, better interoperability between the PlanetLab world and the EmuLab, a more generic shelter that other testbeds can easily leverage in order to come up with their own SFA-compliant wrapper and support for 'reservable' mode, which breaks the usual best-effort PlanetLab model. For more details about this contribution see section

See also the web page <http://planet-lab.eu>

- Version: myplc-5.0-rc26
- Keywords: networking testbed virtual machines
- License: Various Open Source Licences
- Type of human computer interaction: Web-UI, XMLRPC-based API, Qt-based graphical client
- OS/Middleware: Linux-Fedora
- Required library or software: Fedora-14 for the infrastructure side; the software comes with a complete software suite for the testbed nodes
- Programming languages: primarily python, C, ocaml
- Documentation: most crucial module plcapi is self-documented using a local format & related tool. See e.g. <https://www.planet-lab.eu/db/doc/PLCAPI.php>
- Codebase: <http://git.onelab.eu>

5.7. SfaWrap

Participants: Thierry Parmentelat [correspondant], Mohamed Larabi.

The SfaWrap is a reference implementation of the Slice-based Federation Architecture (SFA), the emerging standard for networking experimental testbed federation. We are codevelopping the SfaWrap with Princeton University, and during 2012, we have focused on:

- Participating in the discussions about the future and evolutions of the architecture of SFA, as part of the architecture working group of the GENI project.
- Turning this initially Planet-Lab specific implementation into a generic one, that testbed providers can easily leverage for bringing SFA-compliance to their own testbeds.
- Supporting the allocation and provisioning of both 'Exclusive' and 'Shared' testbed resources.
- Enlarging the federation scheme by federating various testbeds with heterogeneous resources, in order to allow researchers to combine all available resources and run advanced networking experiments of significant scale and diversity.

- Version: sfa-2.1-22, myplc-5.0-rc33
- Keywords: networking testbed federation
- License: Various Open Source Licenses
- Type of human computer interaction: Web-UI, XMLRPC-based API, Qt-based graphical client
- OS/Middleware: Linux
- Required library or software: python2.5 or superior
- Programming languages: python
- Documentation: <http://svn.planet-lab.org/#SFAUser-leveldocumentation>
- Codebase: <http://git.onelab.eu/?p=sfa.git;a=summary>

5.8. MultiCast Library Version 3

Participant: Vincent Roca [correspondant].

MultiCast Library Version 3 is an implementation of the ALC (Asynchronous Layered Coding) and NORM (NACK-Oriented Reliable Multicast Protocol) content delivery Protocols, and of the FLUTE/ALC file transfer application. This software is an implementation of the large scale content distribution protocols standardized by the RMT (Reliable Multicast Transport) IETF working group and adopted by several standardization organizations, in particular 3GPP for the MBMS (Multimedia Broadcast/Multicast Service), and DVB for the CBMS (Convergence of Broadcast and Mobile Services). Our software is used in operational, commercial environments, essentially in the satellite broadcasting area and for file delivery over the DVB-H system where FLUTE/ALC has become a key component. See <http://planetec-bcast.inrialpes.fr/> for more information.

5.9. OpenFEC.org: because open, free AL-FEC codes and codecs matter

Participants: Vincent Roca [correspondant], Jonathan Detchart [engineer], Ferdaouss Mattoussi [PhD student].

The goals of the OpenFEC.org <http://openfec.org> are:

1. to share IPR-free, open, AL-FEC codes,
2. to share high performance, ready-to-use, open, free, C-language, software codecs
3. to share versatile and automated performance evaluation environments.

This project can be useful to users who do not want to know the details of AL-FEC schemes but do need to use one of them in the software they are designing, or by users who want to test new codes or new encoding or decoding techniques, and who do know what they are doing and are looking for, or by users who need to do extensive tests for certain AL-FEC schemes in a given use-case, with a well defined channel model.

5.10. BitHoc

Participants: Chadi Barakat [correspondant], Thierry Turletti.

BitHoc (BitTorrent for wireless ad hoc networks) enables content sharing among spontaneous communities of mobile users using wireless multi-hop connections. It is an open source software developed under the GPLv3 licence. A first version of BitHoc has been made public. We want BitHoc to be the real testbed over which we evaluate our solutions for the support and optimization of file sharing in a mobile wireless environment where the existence of an infrastructure is not needed. The proposed BitHoc architecture includes two principal components: a membership management service and a content sharing service. In its current form it is composed of PDAs and smartphones equipped with WIFI adapters and Windows Mobile 6 operating system.

See also the web page <http://planete.inria.fr/bithoc>

- Version: 1.2
- Keywords: Tracker-less BitTorrent for mobile Ad Hoc networks
- License: GPL (GPLv3)
- Type of human computer interaction: Windows Mobile 6 GUI
- OS/Middleware: Windows Mobile 6
- Required library or software: OpenSSL(<http://www.openssl.org/>, GPL), C++ Sockets, GPL)
- Programming languages: C++, C#
- Documentation: doxygen

5.11. TICIP

Participant: Chadi Barakat [correspondant].

TICIP is a TCP-friendly reliable transport protocol to collect information from a large number of network entities. The protocol does not impose any constraint on the nature of the collected information: availability of network entities, statistics on hosts and routers, quality of reception in a multicast session, weather monitoring, etc. TICIP ensures two main things: (i) the information to collect arrives entirely and correctly to the collector where it is stored and forwarded to upper layers, and (ii) the implosion at the collector and the congestion of the network are avoided by controlling the rate of sending probes. The congestion control part of TICIP is designed with the main objective to be friendly with applications using TCP. Experimental results show that TICIP can achieve better performance than using parallel TCP connections for the data collection. The code of TICIP is available upon request, it is an open source software under the GPLv3 licence.

See also the web page <http://planete.inria.fr/ticip/>

- Version: 1.0
- Keywords: Information Collection, Congestion and Error Control
- License: GPL (GPLv3)
- Type of human computer interaction: XML file
- OS/Middleware: Linux/Unix
- Required library or software: C/C++ Sockets
- Programming languages: C/C++
- Documentation: Text

5.12. Private Data Publication

Participants: Gergely Acs, Claude Castelluccia.

We are developing a set of tools to privately publish different types of datasets. For example, we are developing a software that can be used to sanitize sequential data (described in our CCS paper [41]). The code generates the set of noisy n-grams and generate a synthetic, and private, dataset. We are also developing a tool that implement the histogram sanitization algorithm described in our ICDM paper [33].

These tools are accessible here: <http://planete.inrialpes.fr/projects/p-publication/>

5.13. Experimentation Software

ACQUA

ACQUA stands for Application for Collaborative Estimation of the Quality of Internet Access. It has been developed within the French National project ANR CMON on Collaborative Monitoring in conjunction with Grenouille.com. ACQUA consists of a tool that lets the user have an estimation of the anomalies of the Internet based on active measurements of end-to-end delay metrics among a predefined set of landmarks (i.e. test points). When an anomaly is detected it is expressed in terms of how many destinations are affected by this anomaly, and how important in terms of delay variation is this anomaly for these affected destinations. See also <http://planete.inria.fr/acqua/> for more information and for a java version of the code.

WisMon

WisMon is a Wireless Statistical Monitoring tool that generates real-time statistics from a unified list of packets, which come from possible different probes. This tool fulfills a gap on the wireless experimental field: it provides physical parameters on realtime for evaluation during the experiment, records the data for further processing and builds a single view of the whole wireless communication channel environment. WisMon is available as open source under the Cecill license, at <http://planete.inria.fr/software/WisMon/>.

WEX Toolbox

The Wireless Experimentation (WEX) Toolbox aims to set up, run and make easier the analysis of wireless experiments. It is a flexible and scalable open-source set of tools that covers all the experimentation steps, from the definition of the experiment scenario to the storage and analysis of results. Sources and binaries of the WEX Toolbox are available under the GPLv2 licence at <https://twiki-sop.inria.fr/twiki/bin/view/Projets/Planete/WEXToolkit>. WEX Toolbox includes the CrunchXML utility, which aims to make easier the running and the analysis of wireless experimentations. In a nutshell, it implements an efficient synchronization and merging algorithm, which takes XML (or PDML) input trace files generated by multiple probes, and stores only the packets fields that have been marked as relevant by the user in a MySQL database –original pcap traces should be first formatted in XML using Wireshark. These operations are done in a smart way to balance the CPU resources between the central server (where the database is created) and the different probes (i.e., PC stations where the capture traces are located). CrunchXML is available under the GNU General Public License v2 at <http://twiki-sop.inria.fr/twiki/bin/view/Projets/Planete/CrunchXML>.

WiMAX ns-3

This simulation module for the ns-3 network simulator is based on the IEEE 802.16-2004 standard. It implements the PMP topology with TDD mode and aims to provide detailed and standard compliant implementation of the standard, supporting important features including QoS scheduling services, bandwidth management, uplink request/grant scheduling and the OFDM PHY layer. The module is available under the GNU General Public License at <http://code.nsnam.org/jamine/ns-3-wimax>. It will be included in the official 3.8v release of ns-3.

MonLab

Monitoring Lab is a platform for the emulation and monitoring of traffic in virtual ISP networks. It is supported by the FP7 ECODE project and is available for download at the web page of the tool <http://planete.inria.fr/MonLab/> under the terms of the GPL licence. MonLab presents a new approach for the emulation of Internet traffic and for its monitoring across the different routers of the emulated ISP network. In its current version, the traffic is sampled at the packet level in each router of the platform, then monitored at the flow level. We put at the disposal of users real traffic emulation facilities coupled to a set of libraries and tools capable of Cisco NetFlow data export, collection and analysis. Our aim is to enable running and evaluating advanced applications for network wide traffic monitoring and optimization. The development of such applications is out of the scope of this research. We believe that the framework we are proposing can play a significant role in the systematic evaluation and experimentation of these applications' algorithms. Among the direct

candidates figure algorithms for traffic engineering and distributed anomaly detection. Furthermore, methods for placing monitors, sampling traffic, coordinating monitors, and inverting sampling traffic will find in our platform a valuable tool for experimentation.

MobiTrade

MobiTrade is the ns-3 and Android implementation of our solution for trading content between wireless devices. The application provides a utility driven trading system for efficient content dissemination on top of a disruption tolerant network. While simple tit-for-tat (TFT) mechanisms can force nodes to *give one to get one*, dealing with the inherent tendency of peers to take much but give back little, they can quickly lead to deadlocks when some (or most) of interesting content must be somehow fetched across the network. To resolve this, MobiTrade proposes a trading mechanism that allows a node (*merchant*) to buy, store, and carry content for other nodes (its *clients*) so that it can later trade it for content it is personally interested in. To exploit this extra degree of freedom, MobiTrade nodes continuously profile the type of content requested and the collaboration level of encountered devices. An appropriate utility function is then used to collect an optimal inventory that maximizes the expected value of stored content for future encounters, matched to the observed mobility patterns, interest patterns, and collaboration levels of encountered nodes. See also <http://planete.inria.fr/MobiTrade>.

POEMS Project-Team

5. Software

5.1. Introduction

We are led to develop two types of software. The first one is prototype softwares : various softwares are developed in the framework of specific research contracts (and sometimes sold to the contractor) or during PhD theses. They may be also contributions to already existing softwares developed by other institutions such as CEA, ONERA or EDF. The second category is an advanced software which are intended to be developed, enriched and maintained over longer periods. Such software is devoted to help us for our research and/or promote our research. We have chosen to present here only our advanced software.

5.2. XLIFE++

XLIFE++ is a new Finite Element library in C++ based on philosophy of the previous library MELINA in Fortran but with new capabilities (boundary element, discontinuous Galerkin methods, more integrated tools (in particular mesh tools) and high performance computing skills (multithread and GPU computation). It is licensed under LGPL and it is developed in the context of the european project SIMPOSIUM (FP7/ICT, leader CEA/LIST, from september 2011 to august 2014). There are also academic partners : Irmar-univ. Rennes and Lama-univ. Paris-Est .

In 2012, all development tools have been set up : versioning using Git, repository on Inria-Gforge, compiling and installing tools using Cmake, documentation in TeX and using Doxygen), test processing. All fundamentals library have been developed and checked (String, Function, Messages, Matrix, Vector, ...) and major libraries are done : geometry (Mesh description and tools), space,form, operators to deal with variational descriptions of PDE problem, finiteElements describing all finite elements description, term wich deals with sparse matrix representation of linear or bilinear forms involved in variational problems and finally the solvers library. A first version of the library should be published soon.

POLSYS Project-Team

5. Software

5.1. FGb

Participant: Jean-Charles Faugère [contact].

FGb is a powerful software for computing Groebner bases. It includes the new generation of algorithms for computing Gröbner bases polynomial systems (mainly the F4, F5 and FGLM algorithms). It is implemented in C/C++ (approximately 250000 lines), standalone servers are available on demand. Since 2006, FGb is dynamically linked with Maple software (version 11 and higher) and is part of the official distribution of this software.

See also the web page <http://www-polsys.lip6.fr/~jcf/Software/FGb/index.html>.

- ACM: I.1.2 Algebraic algorithms
- Programming language: C/C++

5.2. RAGlib

Participant: Mohab Safey El Din [contact].

RAGlib is a Maple library for computing sampling points in semi-algebraic sets.

5.3. Epsilon

Participant: Dongming Wang [contact].

Epsilon is a library of functions implemented in Maple and Java for polynomial elimination and decomposition with (geometric) applications.

POMDAPI Project-Team

3. Software

3.1. LifeV

Participant: Michel Kern.

LifeV is a finite element (FE) library providing implementations of state of the art mathematical and numerical methods. It serves both as a research and production library. It has been used already in medical and industrial context to simulate fluid structure interaction and mass transport. LifeV is the joint collaboration between four institutions: École Polytechnique Fédérale de Lausanne (CMCS) in Switzerland, Politecnico di Milano (MOX) in Italy, Inria (Pomdapi) in France and Emory University (Sc. Comp) in the U.S.A.

- Version 3.1.1
- Programming language: C++
- <http://www.lifev.org/>

3.2. M1cg1

- Participant: J. Ch. Gilbert.
- Version: 1.2.
- Programming language: Fortran 77.
- Solves a convex quadratic optimization problem and builds a preconditioning matrix, 8 downloads in 2012.
- <https://who.rocq.inria.fr/Jean-Charles.Gilbert/modulopt/optimization-routines/m1cg1/m1cg1.html>.

3.3. M1qn3

- Participants: J. Ch. Gilbert, Cl. Lemaréchal.
- Version: 3.3.
- Programming language: Fortran 77.
- Solves a very large scale differentiable optimization problem, 34 downloads in 2012.
- <https://who.rocq.inria.fr/Jean-Charles.Gilbert/modulopt/optimization-routines/m1qn3/m1qn3.html>.

3.4. Sklml

Participants: François Clément, Pierre Weis.

Easy coarse grain parallelization.

See also the web page <http://sklml.inria.fr/>.

- Version: 1.0+pl1
- Programming language: OCaml

3.5. SQPlab

- Participant: J. Ch. Gilbert.
- Version: 0.4.5.
- Programming language: Matlab.
- Solves a constrained differentiable optimization problem, 211 downloads in 2012.
- <https://who.rocq.inria.fr/Jean-Charles.Gilbert/modulopt/optimization-routines/sqplab/sqplab.html>.

POP ART Project-Team

5. Software

5.1. NBac

Participant: Bertrand Jeannot.

NBAC (Numerical and Boolean Automaton Checker) ¹⁵ is a verification/slicing tool for reactive systems containing combination of Boolean and numerical variables, and continuously interacting with an external environment. NBAC can also handle the same class of hybrid systems as the HyTech tool [63]. It aims at handling efficiently systems combining a non-trivial numerical behaviour with a complex logical (Boolean) behaviour.

NBAC is connected to two input languages: the synchronous dataflow language LUSTRE, and a symbolic automaton-based language, AUTOC/AUTO, where a system is defined by a set of symbolic hybrid automata communicating via valued channels. It can perform reachability analysis, co-reachability analysis, and combination of the above analyses. The result of an analysis is either a verdict to a verification problem, or a set of states together with a necessary condition to stay in this set during an execution. NBAC is founded on the theory of abstract interpretation.

It has been used for verifying and debugging LUSTRE programs [65] [52] [36]. It is connected to the LUSTRE toolset ¹⁶. It has also been used for controller synthesis of infinite-state systems. The fact that the analyses are approximated results simply in the obtention of a possibly non-optimal controller. In the context of conformance testing of reactive systems, it has been used by the test generator STG ¹⁷ [42] [66] for selecting test cases.

It has recently been superseded by REAVER (see Section 5.2).

5.2. ReaVer

Participant: Peter Schrammel.

REAVER (REActive VERifier ¹⁸) is a tool framework for the safety verification of discrete and hybrid systems specified by logico-numerical data-flow languages, like LUSTRE, LUCIDSYNCHRONE or ZELUS. It provides time-unbounded analysis based on abstract interpretation techniques. In many aspects it is the successor of NBAC (see Section 5.1).

It features partitioning techniques and several logico-numerical analysis methods based on Kleene iteration with widening and descending iterations, abstract acceleration, max-strategy iteration, and relational abstractions; logico-numerical product and power domains (based on the APRON and BddApron domain libraries) with convex polyhedra, octagons, intervals, and template polyhedra; and frontends for the hybrid NBAC format, LUSTRE via lus2nbac, and ZELUS/LUCIDSYNCHRONE. Compared to NBAC, it is connected to higher-level, more recent synchronous and hybrid languages, and provides much more options regarding analysis techniques.

It has been used for several experimental comparisons published in papers and it integrates all the methods developed by Peter Schrammel in its PhD.

5.3. Implementations of Synchronous Programs

Participant: Alain Girault.

¹⁵<http://pop-art.inrialpes.fr/people/bjeannot/nbac/>

¹⁶<http://www-verimag.imag.fr/The-Lustre-Toolbox.html>

¹⁷<http://www.irisa.fr/prive/ployette/stg-doc/stg-web.html>

¹⁸<http://members.ktvam.at/schrammel/research/reaver>

5.3.1. Fault Tolerance

We have been cooperating for several years with the INRIA team AOSTE (INRIA Sophia-Antipolis and Rocquencourt) on the topic of fault tolerance and reliability of safety critical embedded systems. In particular, we have implemented several new heuristics for fault tolerance and reliability within their software SYNDEX¹⁹. Our first scheduling heuristic produces static multiprocessor schedules tolerant to a specified number of processor and communication link failures [55]. The basic principles upon which we rely to make the schedules fault tolerant is, on the one hand, the active replication of the operations [56], and on the other hand, the active replication of communications for point-to-point communication links, or their passive replication coupled with data fragmentation for multi-point communication media (*i.e.*, buses) [57]. Our second scheduling heuristic is multi-criteria: it produces a static schedule multiprocessor schedule such that the reliability is maximized, the power consumption is minimized, and the execution time is minimized [3] [33][17], [11]. Our results on fault tolerance are summarized in a web page²⁰.

5.4. Apron and BddApron Libraries

Participant: Bertrand Jeannot.

5.4.1. Principles

The APRON library²¹ is dedicated to the static analysis of the numerical variables of a program by abstract interpretation [43]. Many abstract domains have been designed and implemented for analysing the possible values of numerical variables during the execution of a program (see Figure 1). However, their API diverge largely (datatypes, signatures, ...), and this does not ease their diffusion and experimental comparison *w.r.t.* efficiency and precision aspects.

The APRON library provides:

- a uniform API for existing numerical abstract domains;
- a higher-level interface to the client tools, by factorizing functionalities that are largely independent of abstract domains.

From an abstract domain designer point of view, the benefits of the APRON library are:

- the ability to focus on core, low-level functionalities;
- the help of generic services adding higher-level services for free.

For the client static analysis community, the benefits are a unified, higher-level interface, which allows experimenting, comparing, and combining abstract domains.

In 2011, the Taylor1plus domain [53], which is the underlying abstract domain of the tool FLUCTUAT [51] has been improved. Glue code has also been added to enable the connection of an abstract domain implemented in OCaml to the APRON infrastructure written in C (this requires callbacks from C to OCaml that are safe *w.r.t.* garbage collection). This will enable the integration in APRON of the MaxPlus polyhedra library written by X. Allamigeon [30] in the context of the ANR ASOPT project.

The BDDAPRON library²² aims at a similar goal, by adding finite-types variables and expressions to the concrete semantics of APRON domains. It is built upon the APRON library and provides abstract domains for the combination of finite-type variables (Booleans, enumerated types, bitvectors) and numerical variables (integers, rationals, floating-point numbers). It first allows the manipulation of expressions that freely mix, using BDDs and MTBDDs, finite-type and numerical APRON expressions and conditions. It then provides abstract domains that combines BDDs and APRON abstract values for representing invariants holding on both finite-type variables and numerical variables.

¹⁹<http://www-rocq.inria.fr/syndex>

²⁰<http://pop-art.inrialpes.fr/~girault/Projets/FT>

²¹<http://apron.cri.enscm.fr/library/>

²²<http://pop-art.inrialpes.fr/~bjeannot/bjeannot-forge/bddapron/index.html>

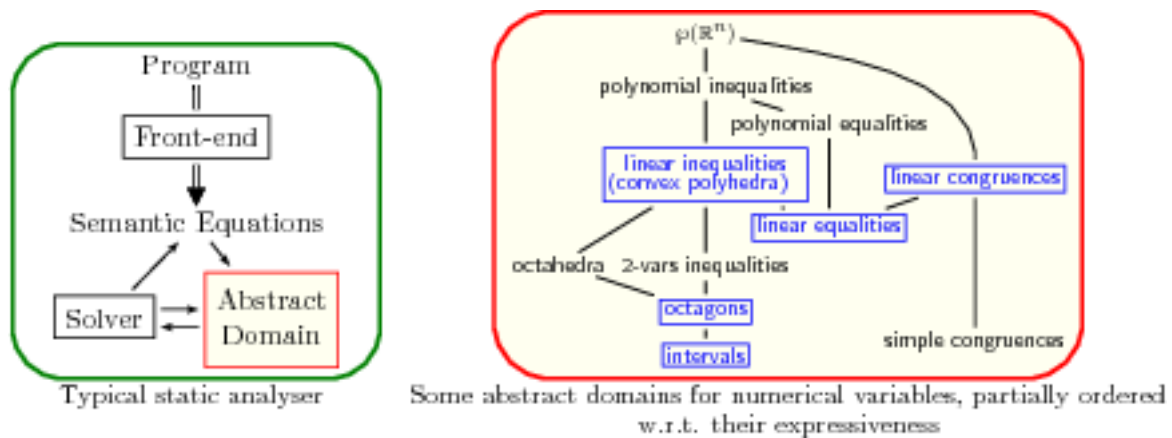


Figure 1. Typical static analyser and examples of abstract domains

5.4.2. Implementation and Distribution

The APRON library (Fig. 2) is written in ANSI C, with an object-oriented and thread-safe design. Both multi-precision and floating-point numbers are supported. A wrapper for the OCAML language is available, and a C++ wrapper is on the way. It has been distributed since June 2006 under the LGPL license and available at <http://apron.cri.enscm.fr>. Its development has still progressed much since. There are already many external users (ProVal/Démons, LRI Orsay, France — CEA-LIST, Saclay, France — Analysis of Computer Systems Group, New-York University, USA — Sierum software analysis platform, Kansas State University, USA — NEC Labs, Princeton, USA — EADS CCR, Paris, France — IRIT, Toulouse, France) and is currently packaged as a REDHAT and DEBIAN package.

The BDDAPRON library is written in OCAML, using polymorphism features of OCAML to make it generic. It is also thread-safe. It provides two different implementations of the same domain, each one presenting pros and cons depending on the application. It is currently used by the CONCURINTERPROC interprocedural and concurrent program analyzer.

5.5. Prototypes

5.5.1. Logical Causality

Participant: Gregor Goessler [contact person].

We have developed LOCA, a new prototype tool written in Scala that implements the analysis of logical causality described in 6.6.2. LOCA currently supports causality analysis in BIP. The core analysis engine is implemented as an abstract class, such that support for other models of computation (MOC) can be added by instantiating the class with the basic operations of the MOC.

5.5.2. Cosyma

Participants: Gregor Goessler [contact person], Sebti Mouelhi.

We have developed COSYMA, a tool for automatic controller synthesis for incrementally stable switched systems based on multi-scale discrete abstractions (see 6.2.1). The tool accepts a description of a switched system represented by a set of differential equations and the sampling parameters used to define an approximation of the state-space on which discrete abstractions are computed. The tool generates a controller — if it exists — for the system that enforces a given safety or time-bounded reachability specification.

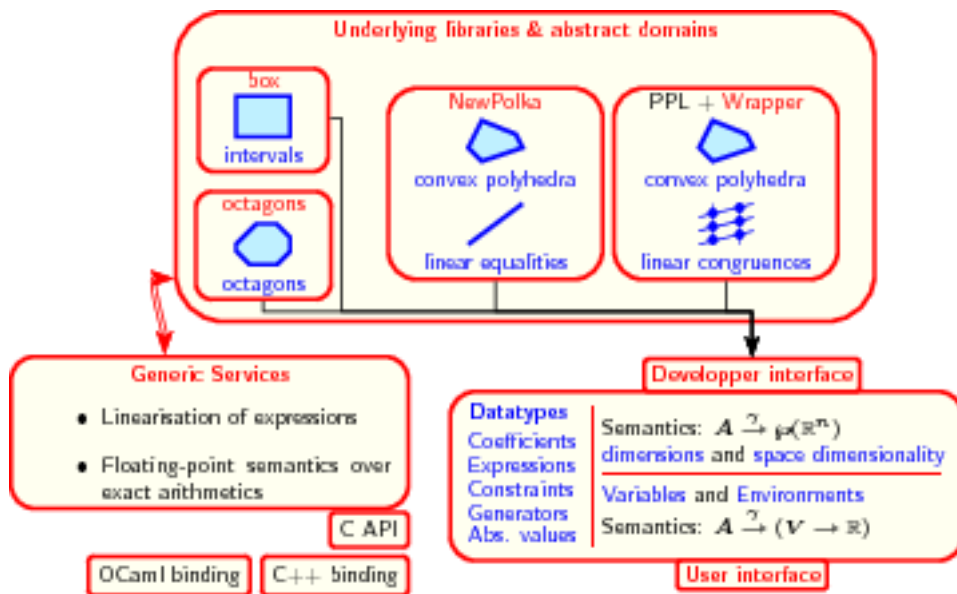


Figure 2. Organisation of the APRON library

5.5.3. Automatic Controller Generation

Participants: Emil Dumitrescu, Alain Girault [contact person].

We have developed a software tool chain to allow the specification of models, the controller synthesis, and the execution or simulation of the results. It is based on existing synchronous tools, and thus consists primarily in the use and integration of SIGALI²³ and Mode Automata²⁴. It is the result of a collaboration with Eric Rutten from the SARDES team.

Useful component templates and relevant properties can be materialized, on one hand by libraries of task models, and, on the other hand, by properties and synthesis objectives.

5.5.4. Rapture

Participant: Bertrand Jeannot.

RAPTURE²⁵ [64] [46] is a verification tool that was developed jointly by BRICS (Denmark) and INRIA in years 2000–2002. The tool is designed to verify reachability properties on Markov Decision Processes (MDP), also known as Probabilistic Transition Systems. This model can be viewed both as an extension to classical (finite-state) transition systems extended with probability distributions on successor states, or as an extension of Markov Chains with non-determinism. We have developed a simple automata language that allows the designer to describe a set of processes communicating over a set of channels *à la* CSP. Processes can also manipulate local and global variables of finite type. Probabilistic reachability properties are specified by defining two sets of initial and final states together with a probability bound. The originality of the tool is to provide two reduction techniques that limit the state space explosion problem: automatic abstraction and refinement algorithms, and the so-called essential states reduction.

²³<http://www.irisa.fr/vertecs/Logiciels/sigali.html>

²⁴<http://www-verimag.imag.fr>

²⁵<http://pop-art.inrialpes.fr/people/bjeannot/rapture/rapture.html>

5.5.5. The Interproc family of static analyzers

Participant: Bertrand Jeannet [contact person].

These analyzers and libraries are of general use for people working in the static analysis and abstract interpretation community, and serve as an experimental platform for the ANR project ASOPT (see §8.1.2.1).

- **FIXPOINT**²⁶: a generic fix-point engine written in OCAML. It allows the user to solve systems of fix-point equations on a lattice, using a parameterized strategy for the iteration order and the application of widening. It also implements recent techniques for improving the precision of analysis by alternating post-fixpoint computation with widening and descending iterations in a sound way [59].
- **INTERPROC**²⁷: a simple interprocedural static analyzer that infers properties on the numerical variables of programs in a toy language. It is aimed at demonstrating the use of the previous library and the above-described APRON library, and more generally at disseminating the knowledge in abstract interpretation. It is also deployed through a web-interface²⁸. It is used as the experimental platform of the ASOPT ANR project.
- **CONCURINTERPROC** extends INTERPROC with concurrency, for the analysis of multithreaded programs interacting via shared global variables. It is also deployed through a web-interface²⁹.
- **PINTERPROC** extends INTERPROC with pointers to local variables. It is also deployed through a web-interface³⁰.

5.5.6. Heptagon/BZR

Participant: Gwenaël Delaval.

HEPTAGON is a dataflow synchronous language, inspired from LUCIDSYNCHRON³¹. Its compiler is meant to be simple and modular, allowing this language to be a good support for the prototyping of compilation methods of synchronous languages. It is developed within the SYNCHRONICSINRIA large-scale action.

HEPTAGON has been used to built BZR³², which is an extension of the former with contracts constructs. These contracts allow to express dynamic temporal properties on the inputs and outputs of HEPTAGON node. These properties are then enforced, within the compilation of a BZR program, by discrete controller synthesis, using the SIGALI tool³³. The synthesized controller is itself generated in HEPTAGON, allowing its analysis and compilation towards different target languages (C, JAVA, VHDL).

²⁶<http://pop-art.inrialpes.fr/people/bjeannet/bjeannet-forge/fixpoint>

²⁷<http://pop-art.inrialpes.fr/people/bjeannet/bjeannet-forge/interproc>

²⁸<http://pop-art.inrialpes.fr/interproc/interprocweb.cgi>

²⁹<http://pop-art.inrialpes.fr/interproc/concurinterprocweb.cgi>

³⁰<http://pop-art.inrialpes.fr/interproc/pinterprocweb.cgi>

³¹<http://www.di.ens.fr/~pouzet/lucid-synchrone/>

³²<http://bzs.inria.fr>

³³<http://www.irisa.fr/vertecs/Logiciels/sigali.html>

POPIX Exploratory Action

5. Software

5.1. MONOLIX

Participant: Marc Lavielle.

MONOLIX is an easy, fast and powerful tool for parameter estimation in non-linear mixed effect models, model diagnosis and assessment, and advanced graphical representation. It is a platform of reference for model-based drug development. Pharmacometricians and biostatisticians can rely on MONOLIX for population analysis and to model PK/PD and other complex biochemical and physiological processes.

MONOLIX was developed by Inria until June 2011. The spin-off Lixoft now develops and supports MONOLIX.

POPIX collaborates closely with Lixoft to convert research results into new user features available in MONOLIX.

5.2. MLXTRAN

Participant: Marc Lavielle.

MONOLIX is associated with MLXTRAN, a powerful and immediately readable declarative language for describing complex pharmacometric and statistical models. MLXTRAN can be used and interfaced with various environments, eg. R, Matlab, etc.

POPIX collaborates closely with LIXOFT on the definition of the specifications and the syntax of MLXTRAN. Implementation is then ensured by LIXOFT.

5.3. Clinical trial simulator

Participants: Marc Lavielle, Elodie Maillot, Laura Brocco.

A clinical trial simulator (CTS) enables effective implementation of the learn-and-confirm paradigm in drug development. Through simulations the anticipated success rate of a future trial can be estimated. For various reasons industry has not embraced currently available software for trial simulation. A new tool is essential for Model Based Drug Development (MBDD).

POPIX is responsible for developing a new CTS within the DDMoRe project (see below). Version 2 of the CTS is available since June 2012. The capabilities of this new version comprise:

- Flexible study designs used in Phase 2 of clinical drug development: parallel group studies, crossover studies, complex treatments defined as a combination of different treatments
- Simulation of patients sampled from a joint distributions or using an external data file
- Simulation of exposure to the investigated drug and several types of drug effects related to drug exposure (continuous, categorical, count, time-to-event)
- Graphics and statistical tests
- Automatic reporting

POTIOC Team

5. Software

5.1. OpenViBE

Participants: Fabien Lotte [local correspondent], Alison Cellard [engineer].

As part of our research work on BCI, we contribute to the development of the OpenViBE⁴ software, which is an open source platform dedicated to the design, evaluation and use of BCI for real and virtual applications. OpenViBE development is led by Inria, and Potioc will be one of the Inria team contributing to its evolution. Moreover, Potioc is implied in an Inria ADT (Technological Development Action) project that has just started, an that is dedicated to the development of OpenViBE together with 3 other Inria teams (Hybrid, Athena, Cortex).

5.2. Drile

Participant: Florent Berthaut.

As part of the research on Virtual Reality for Musical Performance, notably the Drile system, various software are being developed and made available to the community. These software pieces are the following:

- Pure-Data external to access data from the Virtual Reality Peripheral Network : <https://github.com/scrime/vrpd>
- Drile: <http://hitmuri.net/index.php/Research/Drile>

⁴<http://openvibe.inria.fr>

PRIMA Project-Team

5. Software

5.1. OMiSCID Middleware for Distributed Multimodal Perception

Participants: Rémi Barraquand, Amaury Nègre, Patrick Reignier, Dominique Vaufreydaz [correspondant].

Middleware, Distributed perceptual systems

OMiSCID is new lightweight middleware for dynamic integration of perceptual services in interactive environments. This middleware abstracts network communications and provides service introspection and discovery using DNS-SD (*DNS-based Service Discovery* [31]). Services can declare simplex or duplex communication channels and variables. The middleware supports the low-latency, high-bandwidth communications required in interactive perceptual applications. It is designed to allow independently developed perceptual components to be integrated to construct user services. Thus our system has been designed to be cross-language, cross-platform, and easy to learn. It provides low latency communications suitable for audio and visual perception for interactive services.

OMiSCID has been designed to be easy to learn in order to stimulate software reuse in research teams and is revealing to have a high adoption rate. To maximize this adoption and have it usable in projects involving external partners, the OMiSCID middleware has been released under an open source licence. To maximize its target audience, OMiSCID is available from a wide variety of programming languages: C++, Java, Python and Matlab. A website containing informations and documentations about OMiSCID has been set up to improve the visibility and promote the use of this middleware.

The OMiSCID graphical user interface (GUI) is an extensible graphical application that facilitates analysis and debugging of service oriented applications. The core functionality of this GUI is to list running services, their communication channels and their variables. This GUI is highly extensible and many modules (i.e. plugins) have been created by different members of the team: figure 2 shows an example of some of these modules. OMiSCID GUI is based on the Netbeans platform and thus inherits from its dynamic installation and update of modules.

5.2. Detection and Tracking of Pedestrians in INRETS Intelligent Urban Spaces Platform

Participants: Claudine Combe, James Crowley [correspondant], Lukas Rummelhard.

Visual detection and tracking of pedestrians, Intelligent Urban Space

The project ANR-07-TSFA-009-01 CIPEBUS ("Carrefour Intelligent - Pole d'Echange - Bus) has been proposed by INRETS-IFSTTAR, in collaboration with Inria, Citilog, Fareco, and the city of Versailles. The Objective of the CIPEBUS project is to develop an experimental platform for observing activity in a network of urban streets in order to experiment with techniques for optimizing circulation by context aware control of traffic lights.

Within CipeBus, Inria has developed a real time multi-camera computer vision system to detect and track people using a network of surveillance cameras. The CipeBus combines real time pedestrian detection with 2D and 3D Bayesian tracking to record the current position and trajectory of pedestrians in an urban environment under natural view conditions. The system extends the sliding window approach to use a half-octave Gaussian Pyramid to explore hypotheses of pedestrians at different positions and scales. A cascade classifier is used to determine the probability that a pedestrian can be found at a particular position and scale. Detected pedestrians are then tracked using a particle filter.

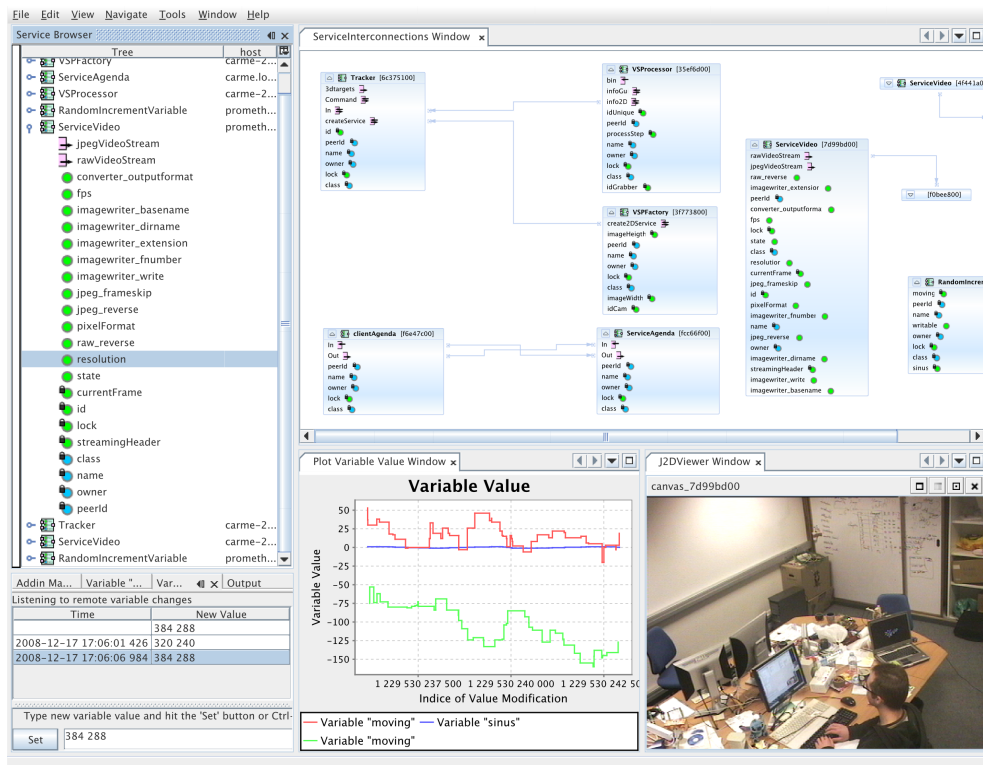


Figure 2. OMiSCID GUI showing a list of running services and some modules for service interconnections, variable plotting, live video stream display and variable control



Figure 3. Cipebus: pedestrian tracking system.

The resulting software system has been installed and tested at the INRETS CipeBus platform and is currently used for experiments in controlling the traffic lights to optimize the flow of pedestrians and public transportation while minimizing the delay imposed on private automobiles.

5.3. Multisensor observation of human activity for integrated energy and comfort management

Participants: Claudine Combe, James Crowley [correspondant], Lucas Nacsa, Amaury Nègre, Lukas Rummelhard.

multimodal tracking of human activity

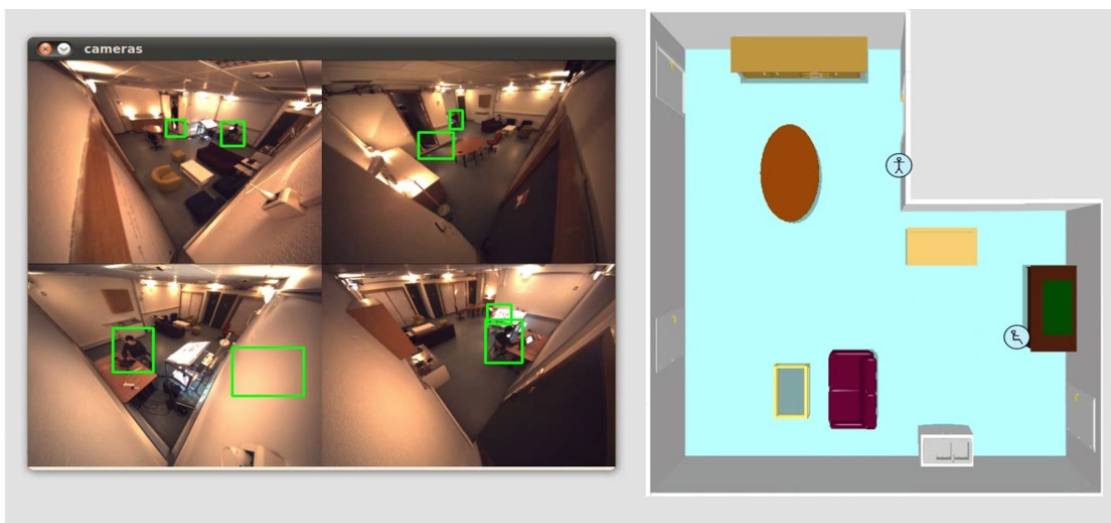


Figure 4. The 3D tracker integrates observations from multiple sensors

As part of Inria's contribution of ICTLabs Action TSES - Smart Energy Systems, we have constructed a system that integrates information from multiple environmental sensors to detect and track people in indoor environments. This system, constructed as part of activity 11831 Open SES Experience Labs for Prosumers and New Services, has been released to ICTLabs partners in June 2012. It has also been used for construction of a smart spaces testbed at Schneider Electric.

This software, named MultiSensor activity tracker, integrates information from multiple environmental sensors to keep track of the location and activity of people in a smart environment. This model is designed to be used by a home energy broker that would work in conjunction with a smart grid to manage the energy consumption of home appliances, balancing the needs of inhabitants with opportunities for savings offered by electricity rates. This database will also be used for by advisor services that will offer advice to inhabitants on the consequences to energy consumption and energy cost that could potentially result from changes to lifestyle or home energy use.

Work in this task draws from earlier results from a number of development projects at Inria. In the ANR Casper project Inria created a Bayesian tracking system for human activity using a voxel-based occupancy grid. Within the INRA ADT PAL project, Inria is creating methods for plug-and-play installation of visual and acoustic sensors for tracking human activity within indoor environments.

While a voxel based Bayesian tracker has served well for a number of applications, a number of limitations have been observed. For example, under certain circumstances, the sensor data can provide contradictory or ambiguous data about the location and activities of people. Resolving such cases required the Bayesian tracker to choose between a numbers of competing hypotheses, potentially resulting in errors. Several members of our group have argued that an alternative integration approach based on the use of a Particle filter would solve these problems and provide a more reliable tracking system. This task has been undertaken to evaluate this hypothesis. The system configured and optimized for detecting and tracking people within rooms using multiple calibrated cameras. The system currently uses corner mounted cartesian cameras, ceiling mounted cameras with wide angle lenses and panoramic cameras placed on tables. Cameras may be connected and disconnected while the component is running, but they must be pre-calibrated to a common room reference frame. We are currently experimenting with techniques for Bayesian estimation of camera parameters for auto-calibration. Cameras may be connected dynamically.

The original system 3DBT has been declared with the APP "Agence pour la Protection des Programmes" under the Interdeposit Digital number IDDN.FR.001.490023.000.S.P.2006.000.10000. A revised declaration for the latest version of the system is currently being prepared.

5.4. Stereo Viewfinder

Participants: Frédéric Devernay [correspondant], Loic Lefort, Elise Mansilla, Sergi Pujades-Rocamora.

Stereoscopy, Auto-calibration, Real-time video processing, Feature matching

This software has been filed with the APP "Agence pour la Protection des Programmes" under the Interdeposit Digital number IDDN.FR.001.370083.000.S.P.2007.000.10000

5.5. Tracking Focus of Attention for Large Screen Interaction

Participants: Rémi Barraquand, Claudine Combe, James Crowley [correspondant], Varun Jain, Sergi Pujades-Rocamora, Lukas Rummelhard.

Embedded Detection and Tracking of Faces for AttentionEstimation.

Large multi-touch screens may potentially provide a revolution in the way people can interact with information in public spaces. Technologies now exist to allow inexpensive interactive displays to be installed in shopping areas, subways and urban areas. Thesis displays can provide location aware access to information including maps and navigation guidance, information about local businesses and and commercial activities. While location information is an important component of a users context, information about the age and gender of a user, as well as information about the number of users present can greatly enhance the value of such interaction for both the user and for local commerce and other activities.

The objective of this task is to leverage recent technological advances in real time face detection developed for cell phones and mobile computing to provide a low-cost real time visual sensor for observing users of large multi-touch interactive displays installed in public spaces.

People generally look at things that attract their attention. Thus it is possible to estimate the subject of attention by estimating where people look. The location of visual attention is manifested by a region of space known as the horopter where the optical axis of the two eyes intersect. However estimating the location of attention from human eyes is notoriously difficult, both because the eyes are small relative to the size of the face, and because eyes can rotate in their socket with very high accelerations. Fortunately, when a human attends to something, visual fixation tends to remain at or near that subject of attention, and the eyes are relaxed to a symmetric configuration by turning the face towards the subject of attention. Thus it is possible to estimate human attention by estimating the orientation of the human face.

We have constructed an embedded software system for detecting, tracking and estimating the orientation of human faces. This software has been designed to be embedded on mobile computing devices such as laptop computers, tablets and interactive display panels equipped with a camera that observes the user. Noting the face orientation with respect to the camera makes it possible to estimate the region of the display screen to which the user is attending.

The system uses a Bayesian Particle filter tracker operating on a Scale invariant Gaussian pyramid to provide integrated tracking and estimation of face orientation. The use of Bayesian tracking greatly improves both the reliability and the efficiency for face detection and orientation estimation. The scale invariant Gaussian pyramid provides automatic adaptation to image scale (as occurs with a change in camera optics) and makes it possible to detect and track faces over a large range of distances. Equally important the Gaussian Pyramid provides a very fast computation of a large number of image features that can be used by a variety of image analysis algorithms.

The software developed for this activity builds on face detections software that has recently been developed by Inria for the French OSEO project MinImage. MinImage was a five year, multi-million euro project to develop next generation technologies for integrated digital imaging devices to be used in cellphones, mobile and laptop computing devices, and digital cameras, that has begun in February of 2007. The project scope included research on new forms of retinas, integrated optics, image formation and embedded image processing. Inria was responsible for embedded algorithms for real time applications of computer vision.

Within MinImage, Inria developed embedded image analysis algorithms using image descriptors that are invariant to position, orientation and scale and robust to changes in viewing angle and illumination intensity. Inria proposed use of a simple hardware circuit to compute a scale invariant Gaussian pyramid as images acquired by the retina. Sums and differences of image samples from the pyramid provide invariant image descriptors that can be used for a wide variety of computer vision applications including detection, tracking and recognition of visual landmarks, physical objects, commercial logos, human bodies and human faces. Detection and tracking of human faces was selected as benchmark test case.

This work has been continued with support from EIT ICTlabs, to provide context information for interaction with large multi-touch interactive displays installed in public spaces.

Multitouch interactive displays are increasingly used in outdoor and public spaces. This objective of this task is to provide a visual observation system that can detect and count users of a multitouch display and to estimate information such as the gender, and age category of each user. us rendering the system sensitive to environmental context.

A revised software package has recently been released to our ICTlab partners for face detection, face tracking, gender and age estimation, and orientation estimation, as part of ICTlabs Smart Spaces action line, Activity 11547 : Pervasive Information interfaces and interaction. With Task 1207 of this activity we have constructed and released an "Attention Recognition Module". This software has been protected with an APP declaration.

An similar software was released in 2007 using face color rather than appearance. The system SuiviDeCiblesCouleur located individuals in a scene for video communications. FaceStabilisationSystem renormalised the position and scale of images to provide a stabilised video stream. SuiviDeCiblesCouleur has been declared with the APP "Agence pour la Protection des Programmes" under the Interdeposit Digital number IDDN.FR.001.370003.000.S.P.2007.000.21000.

5.6. Visual Emotion Recognition for Health and Well Being.

Participants: Rémi Barraquand, Claudine Combe, James Crowley [correspondant], Varun Jain, Sergi Pujades-Rocamora, Lukas Rummelhard.

Visual Emotion Recognition

People express and feel emotions with their face. Because the face is the both externally visible and the seat of emotional expression, facial expression of emotion plays a central role in social interaction between humans. Thus visual recognition of emotions from facial expressions is a core enabling technology for any effort to adapt ICT to improve Health and Wellbeing.

Constructing a technology for automatic visual recognition of emotions requires solutions to a number of hard challenges. Emotions are expressed by coordinated temporal activations of 21 different facial muscles assisted by a number of additional muscles. Activations of these muscles are visible through subtle deformations in the surface structure of the face. Unfortunately, this facial structure can be masked by facial markings, makeup,

facial hair, glasses and other obstructions. The exact facial geometry, as well as the coordinated expression of muscles is unique to each individual. In additions, these deformations must be observed and measured under a large variety of illumination conditions as well as a variety of observation angles. Thus the visual recognition of emotions from facial expression remains a challenging open problem in computer vision.

Despite the difficulty of this challenge, important progress has been made in the area of automatic recognition of emotions from face expressions. The systematic cataloging of facial muscle groups as facial action units by Ekman [45] has let a number of research groups to develop libraries of techniques for recognizing the elements of the FACS coding system [33]. Unfortunately, experiments with that system have revealed that the system is very sensitive to both illumination and viewing conditions, as well as the difficulty in interpreting the resulting activation levels as emotions. In particular, this approach requires a high-resolution image with a high signal-to-noise ratio obtained under strong ambient illumination. Such restrictions are not compatible with the mobile imaging system used on tablet computers and mobile phones that are the target of this effort.

As an alternative to detecting activation of facial action units by tracking individual face muscles, we propose to measure physiological parameters that underlie emotions with a global approach. Most human emotions can be expressed as trajectories in a three dimensional space whose features are the physiological parameters of Pleasure-Displeasure, Arousal-Passivity and Dominance-Submission. These three physiological parameters can be measured in a variety of manners including on-body accelerometers, prosody, heart-rate, head movement and global face expression.

The PRIMA Group at Inria has developed robust fast algorithms for detection and recognition of human faces suitable for use in embedded visual systems for mobile devices and telephones. The objective of the work described in this report is to employ these techniques to construct a software system for measuring the physiological parameters commonly associated with emotions that can be embedded in mobile computing devices such as cell phones and tablets.

As part of Inria's contribution to ICT labs Action THWB Health and Wellbeing, Inria has participated in Activity 12100 "Affective Computing". In this activity we have provided a software system for detection, tracking of faces, and for visual measurement of Valence, Arousal and Dominance.

A software library, named PrimaCV has been designed, debugged and tested, and released to ICTLabs partners for real time image acquisition, robust invariant multi-scale image description, highly optimized face detection, and face tracking. This software has been substantially modified so as to run on an mobile computing device using the Tegra 3 GPU.

PROSECCO Project-Team

5. Software

5.1. ProVerif

Participants: Bruno Blanchet [correspondant], Xavier Allamigeon [April–July 2004], Vincent Cheval [Sept. 2011–], Ben Smyth [Sept. 2009–Feb. 2010].

PROVERIF (proverif.inria.fr) is an automatic security protocol verifier in the symbolic model (so called Dolev-Yao model). In this model, cryptographic primitives are considered as black boxes. This protocol verifier is based on an abstract representation of the protocol by Horn clauses. Its main features are:

- It can handle many different cryptographic primitives, specified as rewrite rules or as equations.
- It can handle an unbounded number of sessions of the protocol (even in parallel) and an unbounded message space.

The **PROVERIF** verifier can prove the following properties:

- secrecy (the adversary cannot obtain the secret);
- authentication and more generally correspondence properties, of the form “if an event has been executed, then other events have been executed as well”;
- strong secrecy (the adversary does not see the difference when the value of the secret changes);
- equivalences between processes that differ only by terms.

PROVERIF is widely used by the research community on the verification of security protocols (see <http://proverif.inria.fr/proverif-users.html> for references).

PROVERIF is freely available on the web, at proverif.inria.fr, under the GPL license.

5.2. CryptoVerif

Participants: Bruno Blanchet [correspondant], David Cadé [Sept. 2009–].

CRYPTOVERIF (cryptoverif.inria.fr) is an automatic protocol prover sound in the computational model. In this model, messages are bitstrings and the adversary is a polynomial-time probabilistic Turing machine. **CRYPTOVERIF** can prove secrecy and correspondences, which include in particular authentication. It provides a generic mechanism for specifying the security assumptions on cryptographic primitives, which can handle in particular symmetric encryption, message authentication codes, public-key encryption, signatures, hash functions, and Diffie-Hellman key agreements.

The generated proofs are proofs by sequences of games, as used by cryptographers. These proofs are valid for a number of sessions polynomial in the security parameter, in the presence of an active adversary. **CRYPTOVERIF** can also evaluate the probability of success of an attack against the protocol as a function of the probability of breaking each cryptographic primitive and of the number of sessions (exact security).

CRYPTOVERIF has been used in particular for a study of Kerberos in the computational model, and as a back-end for verifying implementations of protocols in F# and C.

CRYPTOVERIF is freely available on the web, at cryptoverif.inria.fr, under the CeCILL license.

5.3. Tookan

Participants: Graham Steel [correspondant], Romain Bardou.

See also the web page <http://tookan.gforge.inria.fr/>.

Tookan is a security analysis tool for cryptographic devices such as smartcards, security tokens and Hardware Security Modules that support the most widely-used industry standard interface, RSA PKCS#11. Each device implements PKCS#11 in a slightly different way since the standard is quite open, but finding a subset of the standard that results in a secure device, i.e. one where cryptographic keys cannot be revealed in clear, is actually rather tricky. Tookan analyses a device by first reverse engineering the exact implementation of PKCS#11 in use, then building a logical model of this implementation for a model checker, calling a model checker to search for attacks, and in the case where an attack is found, executing it directly on the device. Tookan has been used to find at least a dozen previously unknown flaws in commercially available devices.

The first results using Tookan were published in 2010 [47] and a six-month licence was granted to Boeing to use the tool. In 2011, a contract was signed with a major UK bank. Tookan is now the subject of a CSATT transfer action resulting in the hiring of an engineer, Romain Bardou, who started on September 1st, 2011. During 2012 Bardou and Steel implemented a new version of Tookan that is intended to form the technological basis for a spin-off company to be created in 2013.

5.4. miTLS

Participants: Alfredo Pironti [correspondant], Karthikeyan Bhargavan, Cedric Fournet [Microsoft Research], Pierre-Yves Strub [IMDEA], Markulf Kohlweiss [Microsoft Research].

miTLS is a verified reference implementation of the TLS security protocol in F#, a dialect of OCaml for the .NET platform. It supports SSL version 3.0 and TLS versions 1.0-1.2 and interoperates with mainstream web browsers and servers. miTLS has been verified for functional correctness and cryptographic security using the refinement typechecker F7.

A paper describing the miTLS library is under review, and the software is being prepared for imminent release in January 2013.

5.5. WebSpi

Participants: Karthikeyan Bhargavan [correspondant], Sergio Maffei [Imperial College London], Chetan Bansal [BITS Pilani-Goa], Antoine Delignat-Lavaud.

WebSpi is a library that aims to make it easy to develop models of web security mechanisms and protocols and verify them using ProVerif. It captures common modeling idioms (such as principals and dynamic compromise) and defines a customizable attacker model using a set of flags. It defines an attacker API that is designed to make it easy to extract concrete attacks from ProVerif counterexamples.

WebSpi has been used to analyze social sign-on and social sharing services offered by prominent social networks, such as Facebook, Twitter, and Google, on the basis of new open standards such as the OAuth 2.0 authorization protocol.

WebSpi has also been used to investigate the security of a number of cryptographic web applications, including password managers, cloud storage providers, an e-voting website and a conference management system.

WebSpi is under development and released as an open source library at <http://prosecco.inria.fr/webspi/>

5.6. Defensive JavaScript

Participants: Antoine Delignat-Lavaud [correspondant], Karthikeyan Bhargavan, Sergio Maffei [Imperial College London].

Defensive JavaScript (DJS) is a subset of the JavaScript language that guarantees the behaviour of trusted scripts when loaded in an untrusted web page. Code in this subset runs independently of the rest of the JavaScript environment. When properly wrapped, DJS code can run safely on untrusted pages and keep secrets such as decryption keys. DJS is especially useful to write security APIs that can be loaded in untrusted pages, for instance an OAuth library such as the one used by "Login with Facebook". It is also useful to write secure host-proof web applications, and more generally for cryptography that happens on the browser.

The DJS type checker and various libraries written in DJS are available from <http://www.defensivejs.com>.

RAP Project-Team (section vide)

REALOPT Project-Team

5. Software

5.1. BaPCod – a generic Branch-and-Price Code

Participants: Romain Leguay [Software Engineer], Pierre Pesneau, Ruslan Sadykov, François Vanderbeck [correspondant].

BaPCod is a prototype code that solves Mixed Integer Programs (MIP) by application of a Dantzig-Wolfe reformulation technique. The reformulated problem is solved using a branch-and-price (column generation) algorithm. This software platform, made of C++ classes, offers a “*black-box*” implementation that does not require user input and is not application specific. The features are

(i) the automation of the Dantzig-Wolfe reformulation process (the user defines a mixed integer programming problem in a pseudo modeling language, defining variables and constraints, identifying subproblems. He can provide subproblem solvers if available, but he does not need to explicitly define the reformulation, the explicit form of the columns, their reduced cost, or the Lagrangian bounds.

(ii) a default column generation procedure with standard initialization and stabilization [1], [23] [25] [29] [32] and

(iii) a default branching scheme that is generic to all applications [9],

(iv) default primal heuristics specially developed for use in a decomposition framework [49], [27], [30].

The prototype software was/is used as background solver for 5 PhD thesis. It also served as the framework for our comparative study in a Inria collaborative research action [1]. It has been experimented by two of our industrial partners, Exeo Solutions (Bayonne), on an inventory routing problem, and Orange Lab (France Telecom, Paris) on network design problems. The prototype also enables us to be very responsive in our industrial contact.

See also the web page <https://wiki.bordeaux.inria.fr/realopt/pmwiki.php/Project/BaPCod>.

REGAL Project-Team

5. Software

5.1. Coccinelle

Participants: Christian Clausen, Julia Lawall [correspondent], Arie Middlekoop, Gilles Muller [correspondent], Gaël Thomas, Suman Saha.

Coccinelle is a program matching and transformation engine which provides the language SmPL (Semantic Patch Language) for specifying desired matches and transformations in C code. Coccinelle was initially targeted towards performing collateral evolutions in Linux. Such evolutions comprise the changes that are needed in client code in response to evolutions in library APIs, and may include modifications such as renaming a function, adding a function argument whose value is somehow context-dependent, and reorganizing a data structure.

Beyond collateral evolutions, Coccinelle has been successfully used for finding and fixing bugs in systems code. One of the main recent results is an extensive study of bugs in Linux 2.6 that has permitted us to demonstrate that the quality of code has been improving over the last six years, even though the code size has more than doubled.

<http://coccinelle.lip6.fr>

5.2. SwiftCloud

Participants: Marc Shapiro [correspondent], Marek Zawirski, Annette Bieniusa, Valter Balegas.

SwiftCloud is a platform for deploying large-scale distributed applications on the edge, close to the users. Internet delays are a problem for interactive distributed applications. Truly optimal responsiveness and availability require mutable shared state replicated near the client, at the network edge. This raises serious challenges of consistency, fault tolerance, and programmability. The SwiftCloud system is designed to address these challenges. Our data model is based on client-side caching of shared mutable objects, made practical with a library of synchronisation-free, yet provably correct object types (CRDTs). The consistency model combines eventual consistency, absence of roll-backs, transactional consistency and session guarantees, but stops short of serialisability. This programming model is practically useful, yet does not require synchronisation, thus ensuring scalability. Maintaining the guarantees at a reasonable cost is especially challenging at large scale. Scalability and programmability are helped by several design decisions detailed in the paper. We validated our approach by building the SwiftCloud platform, by deploying three significant applications, and by measuring their performance in different configurations, in order to explore the benefits of replication at different locations.

SwiftCloud was supported by the ConcoRDanT ANR project (Section 7.1.4) and a Google European Doctoral Fellowship (Section 7.2.2.1).

The code is freely available on <http://gforge.inria.fr/> under a BSD license.

5.3. Treedoc

Participants: Marc Shapiro [correspondent], Marek Zawirski, Nuno Preguiça.

A Commutative Replicated Data Type (CRDT) is one where all concurrent operations commute. The replicas of a CRDT converge automatically, without complex concurrency control. We designed and developed a novel CRDT design for cooperative text editing, called Treedoc. It is designed over a dense identifier space based on a binary trees. Treedoc also includes an innovative garbage collection algorithm based on tree rebalancing. In the best case, Treedoc incurs no overhead with respect to a linear text buffer. The implementation has been validated with performance measurements, based on real traces of social text editing in Wikipedia and SVN.

Work in 2010 has focused on studying large-scale garbage collection for Treedoc, and design improvements. Future work includes engineering a large-scale collaborative Wiki, and studying CRDTs more generally.

TreeDoc is supported by the Prose, Streams and ConcoRDanT ANR projects (Sections 7.1.7 , 7.1.6 and 7.1.4 respectively) and by a Google European Doctoral Fellowship (Section 7.2.2.1).

The code is freely available on <http://gforge.inria.fr/> under a BSD license.

5.4. Telex

Participants: Marc Shapiro [correspondent], Lamia Benmouffok, Pierre Sutra, Pierpaolo Cincilla.

Developing write-sharing applications is challenging. Developers must deal with difficult problems such as managing distributed state, disconnection, and conflicts. Telex is an application-independent platform to ease development and to provide guarantees. Telex is guided by application-provided parameters: actions (operations) and constraints (concurrency control statements). Telex takes care of replication and persistence, drives application progress, and ensures that replicas eventually agree on a correct, common state. Telex supports partial replication, i.e., sites only receive operations they are interested in. The main data structure of Telex is a large, replicated, highly dynamic graph; we discuss the engineering trade-offs for such a graph and our solutions. Our novel agreement protocol runs Telex ensures, in the background, that replicas converge to a safe state. We conducted an experimental evaluation of the Telex based on a cooperative calendar application and on benchmarks.

The code is freely available on <http://gforge.inria.fr/> under a BSD license.

5.5. Java and .Net runtimes for LLVM

Participants: Harris Bakiras, Bertil Folliot, Julia Lawall, Jean-Pierre Lozi, Gaël Thomas [correspondent], Gilles Muller, Thomas Preud homme, Koutheir Attouchi.

Many systems research projects now target managed runtime environments (MRE) because they provide better productivity and safety compared to native environments. Still, developing and optimizing an MRE is a tedious task that requires many years of development. Although MREs share some common functionalities, such as a Just In Time Compiler or a Garbage Collector, this opportunity for sharing has not been yet exploited in implementing MREs. We are working on VMKit, a first attempt to build a common substrate that eases the development and experimentation of high-level MREs and systems mechanisms. VMKit has been successfully used to build two MREs, a Java Virtual Machine and a Common Language Runtime, as well as a new system mechanism that provides better security in the context of service-oriented architectures.

VMKit is an implementation of a JVM and a CLI Virtual Machines (Microsoft .NET is an implementation of the CLI) using the LLVM compiler framework and the MMTk garbage collectors. The JVM, called J3, executes real-world applications such as Tomcat, Felix or Eclipse and the DaCapo benchmark. It uses the GNU Classpath project for the base classes. The CLI implementation, called N3, is its in early stages but can execute simple applications and the “pnetmark” benchmark. It uses the pnetlib project or Mono as its core library. The VMKit VMs compare in performance with industrial and top open-source VMs on CPU-intensive applications. VMKit is publicly available under the LLVM license.

<http://vmkit2.gforge.inria.fr/>

REGULARITY Project-Team

5. Software

5.1. FracLab

Participants: Paul Balança, Jacques Lévy Véhel [correspondant].

FracLab was developed for two main purposes:

1. propose a general platform allowing research teams to avoid the need to re-code basic and advanced techniques in the processing of signals based on (local) regularity.
2. provide state of the art algorithms allowing both to disseminate new methods in this area and to compare results on a common basis.

FracLab is a general purpose signal and image processing toolbox based on fractal, multifractal and local regularity methods. FracLab can be approached from two different perspectives:

- (multi-) fractal and local regularity analysis: A large number of procedures allow to compute various quantities associated with 1D or 2D signals, such as dimensions, Hölder and 2-microlocal exponents or multifractal spectra.
- Signal/Image processing: Alternatively, one can use FracLab directly to perform many basic tasks in signal processing, including estimation, detection, denoising, modeling, segmentation, classification, and synthesis.

A graphical interface makes FracLab easy to use and intuitive. In addition, various wavelet-related tools are available in FracLab.

FracLab is a free software. It mainly consists of routines developed in MatLab or C-code interfaced with MatLab. It runs under Linux, MacOS and Windows environments. In addition, a “stand-alone” version (*i.e.* which does not require MatLab to run) is available.

FracLab has been downloaded several thousands of times in the last years by users all around the world. A few dozens laboratories seem to use it regularly, with more than two hundreds registered users. Our ambition is to make it the standard in fractal softwares for signal and image processing applications. We have signs that this is starting to become the case. To date, its use has been acknowledged in more than two hundreds research papers in various areas such as astrophysics, chemical engineering, financial modeling, fluid dynamics, internet and road traffic analysis, image and signal processing, geophysics, biomedical applications, computer science, as well as in mathematical studies in analysis and statistics (see <http://fraclab.saclay.inria.fr/> for a partial list with papers). In addition, we have opened the development of FracLab so that other teams worldwide may contribute. Additions have been made by groups in Australia, England, France, the USA, and Serbia.

We have produced this year a major release of FracLab (version 2.1).

REO Project-Team

5. Software

5.1. LiFE-V library

Participants: Miguel Ángel Fernández Varela [correspondant], Jean-Frédéric Gerbeau.

LiFE-V² is a finite element library providing implementations of state of the art mathematical and numerical methods. It serves both as a research and production library. LiFE-V is the joint collaboration between three institutions: Ecole Polytechnique Fédérale de Lausanne (CMCS) in Switzerland, Politecnico di Milano (MOX) in Italy and Inria (REO) in France. It is a free software under LGPL license.

5.2. Mistral library

Participants: Cristóbal Bertoglio Beltran, Jean-Frédéric Gerbeau [correspondant], Vincent Martin.

Mistral is a finite element library which implements in particular fluid-structure interaction algorithms (ALE and Fictitious domain formulations), fluid surface flow (ALE) and incompressible magnetohydrodynamics equations. Mistral results from a collaboration between Inria and ENPC (CERMICS).

5.3. FELiScE

Participants: Grégory Arbia, Cesare Corrado, Miguel Ángel Fernández Varela, Justine Fouchet-Incaux, David Froger, Jean-Frédéric Gerbeau [correspondant], Damiano Lombardi, Elisa Schenone, Saverio Smaldone, Marina Vidrascu, Irène Vignon-Clementel.

FELiScE – standing for “Finite Elements for Life Sciences and Engineering” – is a new finite element code which the MACS and REO project-teams have decided to jointly develop in order to build up on their respective experiences concerning finite element simulations. One specific objective of this code is to provide in a unified software environment all the state-of-the-art tools needed to perform simulations of the complex cardiovascular models considered in the two teams – namely involving fluid and solid mechanics, electrophysiology, and the various associated coupling phenomena. FELISCE is written in C++, and may be later released as an opensource library. <https://gforge.inria.fr/projects/felisce/>

5.4. SHELDDON

Participant: Marina Vidrascu [correspondant].

SHELDDON (SHELLs and structural Dynamics with DOrain decomposition in Nonlinear analysis) is a finite element library based on the Modulef package which contains shell elements, nonlinear procedures and PVM subroutines used in domain decomposition or coupling methods, in particular fluid-structure interaction. (<https://gforge.inria.fr/projects/shelddon>)

²<http://www.lifev.org/>

REVES Project-Team

4. Software

4.1. RID: Rich Intrinsic Decomposer

Participants: Pierre-Yves Laffont, Adrien Bousseau, George Drettakis.

We developed a software platform to perform rich intrinsic decomposition methods from photographs of outdoor scenes, as described in [18]. It includes main scripts and functions in Matlab for treatment of the input data, interfaces to software for multi-view reconstruction (Bundler, PMVS) and meshing from point clouds (method developed by Julie Digne, a postdoc in the GEOMETRICA project team). We then interface software for image matting using the Matting Laplacian, and User-Assisted Intrinsic Images. The system also includes an interface with Adobe Photoshop, for visualization and demonstration of our results in end-user image editing software. The method performs the computation of sun, sky and indirect lighting received at 3D points of an automatically reconstructed scene, using a modified version of the PBRT stochastic raytracer. Finally, there is a scene calibration module and an OpenGL viewer.

4.2. Imerse: Inria Multi-Environment Realistic Simulation Engine

Participants: Adrien David, George Drettakis.

In the context of the ADT Interact3D and the ARC NIEVE, we developed Imerse, a middleware to be used as a VR engine, helping in the implementation of realistic simulations for immersive installations. Imerse provides a wrapper to OSG's (OpenSceneGraph) deep scene graph and its traversals abilities into an abstracted collection of high level objects which directly represent realistic entities (such as indoor elements, machines and realistic characters). It provides capacities such as skeletal animations or spatialized audio by interfacing with APF, while its clear composite pattern allows implementing more behaviors easily.

Finally, a generic design based on triggers and functors lets the final user implement complex scenarios of VR applications with the feeling of writing a script in C++. Applications developed on top of Imerse plug transparently into osgVR developed in the DREAM group (i.e., the research support development group of our Inria center). We are using osgVR to render OSG's scene graph in a distributed manner, since rendering clusters are available in an increasing number of installations. osgVR is a software layer developed by the DREAM research support group, ensuring synchronization and events/inputs distribution among a list of rendering slaves. These two libraries are available on GForge.

4.3. APF: state-of-the-art 3D audio library

Participants: Adrien David, George Drettakis.

This work was performed in collaboration with Jean-Christophe Lombardo of the DREAM group (i.e., the research support development group of our Inria center). REVES has several audio research publications over the last 10 years, which correspond to a class of functionalities. The first component is the masking or culling algorithm, which aims at removing all the inaudible audio sources from a virtual scene based on perceptual metrics. The second component, called clustering, aims at grouping audio sources that are spatially close to each other and pre-mix them to a representative cluster source, so that all spatialization related processing can be applied only on the representative pre-mixed source [9]. Other audio topics were also considered and developed, like progressive and scalable frequency domain mixing, sound propagation, scalable reverberation, modal sound synthesis and contact sounds generation [1].

In order to maintain all the knowledge in the group and re-use these technologies in the Immersive Space, a previous young engineer, a previous engineer (David Grelaud) wrote a fully documented audio library (APF) which gathers about 10 audio publications and 1 US patent. APF is a cross-platform, object oriented C++ API available on GForge. All the code has been re-implemented and a completely new software architecture resulted in a twofold increase in the speed of our algorithms. APF runs in the Immersive Space and uses the tracking system to spatialize virtual audio sources around the listener. It can also exploit personal Head Related Transfer Functions (HRTF).

We have implemented a network communications layer to create an audio rendering server on a separate machine, and the library is fully integrated into the osgVR platform.

APF has also been critical in establishing collaborations in the context of various grant proposals (EU and national).

4.4. GaborNoise Software

Participants: Ares Lagae, George Drettakis.

We proposed a new procedural noise function last year, Gabor noise [6]. In the context of this project, we have developed a software package, which includes a CPU reference implementation of the 2D noise, and a complete GPU implementation of the 2D noise, surface noise, and 3D noise. This software package has been filed for APP protection and is in the process of being transferred to industrial partners.

This work is a collaboration with Sylvain Lefebvre, former member of the team, now in the ALICE project-team, Inria Nancy - Grand Est.

RMOD Project-Team

4. Software

4.1. Moose

Participants: Stéphane Ducasse [correspondant], Muhammad Bhatti, Andre Hora, Nicolas Anquetil, Tudor Gîrba [University of Bern].

Web: <http://www.moosetechnology.org/>

The platform. Moose is a language-independent environment for reverse- and re-engineering complex software systems. Moose provides a set of services including a common meta-model, metrics evaluation and visualization, a model repository, and generic GUI support for querying, browsing and grouping. The development of Moose began at the Software Composition Group in 1997, and is currently contributed to and used by researchers in at least seven European universities. Moose offers an extensible meta-described metamodel, a query engine, a metric engine and several visualizations. Moose is currently in its fourth major release and comprises 55,000 lines of code in 700 classes.

The RMoD team is currently the main maintainer of the Moose platform. There are 200 publications (journal, international conferences, PhD theses) based on execution or use of the Moose environment.

The first version running on top of Pharo (Moose 4.0) was released in June 2010. In February 2012, Moose 4.6 was released.

Here is the self-assessment of the team effort following the grid given at <http://www.inria.fr/institut/organisation/instances/commission-d-evaluation>.

- **(A5)** Audience : 5 – Moose is used by several research groups, a consulting company, and some companies using it in ad-hoc ways.
- **(SO4)** Software originality : 4 – Moose aggregates the last results of several research groups.
- **(SM4)** Software Maturity : 4 – Moose is developed since 1996 and got two main redesign phases.
- **(EM4)** Evolution and Maintenance : 4 – Moose will be used as a foundation of our Synectique start up so its maintenance is planned.
- **(SDL4)** Software Distribution and Licensing : 4 – Moose is licensed under BSD
- **(OC)** Own Contribution : (Design/Architecture)DA-4, (Coding/Debugging)-4, (Maintenance/Support)-4, (Team/Project Management)-4

4.2. Pharo

Participants: Marcus Denker [correspondant], Damien Cassou, Stéphane Ducasse, Esteban Lorenzano, Mariano Martinez-Peck, Damien Pollet, Igor Stasenko, Veronica Uquillas-Gomez.

Web: <http://www.pharo-project.org/>

The platform. Pharo is a new open-source Smalltalk-inspired language and environment. It provides a platform for innovative development both in industry and research. By providing a stable and small core system, excellent developer tools, and maintained releases, Pharo's goal is to be a platform to build and deploy mission critical Smalltalk applications.

The first stable version, Pharo 1.0, was released in 2010. The development of Pharo accelerated in 2011 and 2012: Versions 1.2 to 1.4 have been released (with more than 2400 closed issues), and the development branch (2.0) has seen already over 398 incremental releases as of mid November 2012. In 2012, RMoD organized the first *Pharo Conference* during two days in May with 60 participants.

Additionally, in November 2012 RMoD launched the Pharo Consortium (<http://www.pharo-project.org/community/consortium>) and the Pharo Association (<http://association.pharo.org/>). 25 companies already shown interest in supporting the consortium.

RMoD is the main maintainer and coordinator of Pharo.

Here is the self-assessment of the team effort following the grid given at <http://www.inria.fr/institut/organisation/instances/commission-d-evaluation>.

- (A5) Audience: 5 – Used in many universities for teaching, more than 25 companies.
- (SO3) Software originality : 3 – Pharo offers a classical basis for some aspects (UI). It includes new frameworks and concepts compared to other Smalltalk implementations.
- (SM4) Software Maturity: 4 – Bug tracker, continuous integration, large test suites are on place.
- (EM4) Evolution and Maintenance: 4 – Active user group, consortium and association had just been set up.
- (SDL4) Software Distribution and Licensing: 4 – Pharo is licensed under MIT.
- (OC5) Own Contribution: (Design/Architecture) DA-5, (Coding/Debugging) CD-5, (Maintenance/Support) MS-5, (Team/Project Management) TPM-5

4.3. Fuel

Participants: Martin Dias [Correspondant], Mariano Martinez-Peck.

Web: <http://rmod.lille.inria.fr/web/pier/software/fuel>

Objects in a running environment are constantly being born, mutating their status and dying in the volatile memory of the system. The goal of serializers is to store and load objects either in the original environment or in another one. Fuel is a general-purpose serializer based on four principles: (1) speed, through a compact binary format and a pickling algorithm which obtains the best performance on materialization; (2) good object-oriented design, without any special help from the virtual machine; (3) specialized for Pharo, so that core objects (such as contexts, block closures and classes) can be serialized too; (4) flexible about how to serialize each object, so that objects are serialized differently depending on the context.

Here is the self-assessment of the team effort following the grid given at <http://www.inria.fr/institut/organisation/instances/commission-d-evaluation>.

- (A4) Audience: 4 – Large audience software, usable by people inside and outside the field with a clear and strong dissemination, validation, and support action plan.
- (SO3) Software originality : 3.
- (SM4) Software Maturity: 4 – Bug tracker, continuous integration, large test suites are on place.
- (EM4) Evolution and Maintenance: 4.
- (SDL4) Software Distribution and Licensing: 4 – Fuel is licensed under MIT.
- (OC5) Own Contribution: (Design/Architecture) DA-5, (Coding/Debugging) CD-5, (Maintenance/Support) MS-5, (Team/Project Management) TPM-5

4.4. Athens

Participant: Igor Stasenko [Correspondant].

Athens is a vector graphics framework for Pharo.

4.5. Citezen

Participants: Damien Pollet [Correspondant], Stéphane Ducasse.

Web: <http://people.untyped.org/damien.pollet/software/citezen/>

Citezen is a suite of tools for parsing, validating, sorting and displaying BibTeX databases. This tool suite is integrated within the Pier Content Management System (CMS) and both are implemented on top of Pharo. Citezen aims at replacing and extending BibTeX, in Smalltalk; ideally, features would be similar to BibTeX, CrossTeX, and CSL.

4.6. Handles

Participant: Jean-Baptiste Arnaud [Correspondant].

Web: <http://jeanbaptiste-arnaud.eu/handles/>

An Handle is a first-class reference to a target object. Handles can alter the behavior and isolate the state of the target object. Handles provide infrastructure to automatically create and wrap new handles when required. A real-time control of handles is possible using a special object called metaHandle.

4.7. Hazelnut

Participants: Guillermo Polito [Correspondant], Benjamin van Ryseghem, Nicolas Paez, Igor Stasenko.

Web: <http://rmod.lille.inria.fr/web/pier/software/Seed>

Traditionally, Smalltalk-based systems are not bootstrapped because of their ability to evolve by self-modification. Nevertheless, the absence of a bootstrap process exposes many problems in these systems, such as the lack of reproducibility and the impossibility to reach certain evolution paths. Hazelnut is a tool that aims to introduce a bootstrap process into these systems, in particular Pharo.

4.8. Jet

Participant: Veronica Uquillas-Gomez [Correspondant].

Jet is a tool to analyze streams of changes. Jet identifies dependencies between changes and sets of changes and supports cherry picking. Moreover, Jet classifies sets of changes based on their dependencies as a way to ease the analysis of changes within the stream and guide system integrators.

4.9. LegacyParsers

Participants: Muhammad Bhatti [Correspondant], Nicolas Anquetil, Guillaume Larcheveque, Esteban Lorenzano, Gogui Ndong.

As part of our research on legacy software and also for the Synectique company), we started to define several parsers for old languages like Cobol for example. This work is important to help us validate our meta-model and tools against a larger range of existing technologies and to discover the limits of our approach. From our initial results, and the in-depth understanding that it gave us, we are formulating new research objectives in meta-model driven reverse engineering. This work is also important for the spin-off company, as being able to work with such technologies is fundamental.

4.10. Mate

Participants: Marcus Denker [Correspondant], Clement Bera, Camillo Bruni.

Mate is the future research-oriented virtual machine for Pharo. Its goal is to serve as a prototype for researchers to experiment with. As a result, the design of Mate is very simple to understand. As of today, Mate consists of an AST interpreter, a new object memory layout, and a simple garbage collector.

4.11. NativeBoost

Participant: Igor Stasenko [Correspondant].

Web: <http://code.google.com/p/nativeboost/>

NativeBoost is a Smalltalk framework for generating and running machine code from the language side of Pharo. As part of it comes a foreign function interface that enables calling external C functions from Smalltalk code with minimal effort.

4.12. Nabujito

Participants: Camillo Bruni [Correspondant], Marcus Denker.

Nabujito is a new Just In Time compiler implemented as a Smalltalk application, based on NativeBoost, that does not require changes in the virtual machine.

4.13. Nautilus

Participants: Benjamin Van Ryseghem [Correspondant], Stéphane Ducasse, Igor Stasenko, Camillo Bruni, Esteban Lorenzano.

Nautilus is a new source code browser based on the latest infrastructure representations. Its goal is mainly to replace the current system browser that was implemented in the 80s and that doesn't provide optimal tools for the system as it has evolved.

4.14. SourceCity

Participants: Erwan Douaille [Correspondant], Igor Stasenko, Guillaume Larcheveque, Stéphane Ducasse.

Modern systems are too complex. Understanding and analyzing these systems is very hard and tedious (thousand of classes, millions of lines of code). One needs an overview of the system that allows to discover important parts in the system, weak points, suspicious components. SourceCity is a powerful 3D tool that can help to understand quickly how a system works by taking the metaphor of a city buildings. By looking at tall, large, low building, one can identify different properties of the software components being represented.

4.15. Spec

Participants: Benjamin Van Ryseghem [Correspondant], Stéphane Ducasse, Johan Fabry.

Spec is a programming framework for generating graphical user interfaces inspired by VisualWorks' Subcanvas. The goal of Spec is to tackle the lack of reuse experienced in existing tools. Spec serves as a pluggable layer on top of multiple lower-level graphical frameworks. Many improvements have been noticed in Pharo after the introduction of Spec in terms of speed or number of lines of code while we re-implemented existing tools using Spec.

4.16. VerveineJ

Participants: Nicolas Anquetil [Correspondant], Andre Hora, Guillaume Larcheveque.

Web: Inria project <https://gforge.inria.fr/projects/verveinej/>.

VerveineJ is a tool to export Java projects into the MSE format, which can then be imported inside Moose (see above). Although VerveineJ is not a research project in itself, it is an important building block for our research in that it allows us to run the Moose platform on legacy Java projects. Another similar tool, Infusion, already existed to fulfil the same needs, but it was closed sources and presented some errors that tainted the results we could obtain.

ROMA Team

4. Software

4.1. MUMPS

Participants: Patrick Amestoy, Alfredo Buttari, Jean-Yves L'Excellent [correspondent], Mohamed Sid-Lakhdar, François-Henry Rouet, Bora Uçar, Clément Weisbecker.

MUMPS (for *MULTifrontal Massively Parallel Solver*, see <http://graal.ens-lyon.fr/MUMPS>) is a software package for the solution of large sparse systems of linear equations. The development of MUMPS was initiated by the European project PARASOL (Esprit 4, LTR project 20160, 1996-1999), whose results and developments were public domain. Since then, MUMPS has been supported by CERFACS, CNRS, ENS Lyon, INPT(ENSEEIH)-IRIT (main contributor), Inria, and University of Bordeaux. In the context of an ADT project (Action of Technological Development), Maurice Brémond ("SED" service) also works part-time on MUMPS.

MUMPS implements a direct method, the multifrontal method; it is a parallel code capable of exploiting distributed-memory computers; its main originalities are its numerical robustness and the wide range of functionalities available.

The latest release is MUMPS 4.10.0 (May 2011).

More information on MUMPS is available at <http://graal.ens-lyon.fr/MUMPS/> and <http://mumps.enseiht.fr>.

RUNTIME Project-Team

5. Software

5.1. Common Communication Interface

Participant: Brice Goglin.

- The *Common Communication Interface* aims at offering a generic and portable programming interface for a wide range of networking technologies (Ethernet, InfiniBand, ...) and application needs (MPI, storage, low latency UDP, ...).
- CCI is developed in collaboration with the *Oak Ridge National Laboratory* and several other academics and industrial partners.
- CCI is in early development and currently composed of 19 000 lines of C.
- <http://www.cci-forum.org>

5.2. Hardware Locality

Participants: Brice Goglin, Samuel Thibault.

- *Hardware Locality* (HWLOC) is a library and set of tools aiming at discovering and exposing the topology of machines, including processors, cores, threads, shared caches, NUMA memory nodes and I/O devices.
- It builds a widely-portable abstraction of these resources and exposes it to the application so as to help them adapt their behavior to the hardware characteristics.
- HWLOC targets many types of high-performance computing applications [2], from thread scheduling to placement of MPI processes. Most existing MPI implementations, several resource managers and task schedulers already use HWLOC.
- HWLOC is developed in collaboration with the OPEN MPI project. The core development is still mostly performed by Brice GOGLIN and Samuel THIBAULT from the RUNTIME team-project, but many outside contributors are joining the effort, especially from the OPEN MPI and MPICH2 communities.
- HWLOC is composed of 40 000 lines of C.
- <http://runtime.bordeaux.inria.fr/hwloc/>

5.3. KNem

Participant: Brice Goglin.

- KNEM (*Kernel Nemesis*) is a Linux kernel module that offers high-performance data transfer between user-space processes.
- KNEM offers a very simple message passing interface that may be used when transferring very large messages within point-to-point or collective MPI operations between processes on the same node.
- Thanks to its kernel-based design, KNEM is able to transfer messages through a single memory copy, much faster than the usual user-space two-copy model.
- KNEM also offers the optional ability to offload memory copies on INTEL I/O AT hardware which improves throughput and reduces CPU consumption and cache pollution.
- KNEM is developed in collaboration with the MPICH2 team at the Argonne National Laboratory and the OPEN MPI project. These partners already released KNEM support as part of their MPI implementations.
- KNEM is composed of 7000 lines of C. Its main contributor is Brice GOGLIN.
- <http://runtime.bordeaux.inria.fr/knem/>

5.4. Marcel

Participants: Olivier Aumage, Yannick Martin, Samuel Thibault.

- MARCEL is the two-level thread scheduler (also called N:M scheduler) of the PM² software suite.
- The architecture of MARCEL was carefully designed to support a large number of threads and to efficiently exploit hierarchical architectures (e.g. multicore chips, NUMA machines).
- MARCEL provides a *seed* construct which can be seen as a precursor of thread. It is only when the time comes to actually run the seed that MARCEL attempts to reuse the resources and the context of another, dying thread, significantly saving management costs.
- In addition to a set of original extensions, MARCEL provides a POSIX-compliant interface which thus permits to take advantage of it by just recompiling unmodified applications or parallel programming environments (API compatibility), or even by running already-compiled binaries with the Linux NPTL ABI compatibility layer.
- For debugging purpose, a trace of the scheduling events can be recorded and used after execution for generating an animated movie showing a replay of the execution.
- The MARCEL thread scheduling library is made of 80 000 lines of code.
- <http://runtime.bordeaux.inria.fr/marcel/>
- Marcel has been supported for 2 years (2009-2011) by the Inria ADT Visimar.

5.5. ForestGOMP

Participants: Olivier Aumage, Yannick Martin, Pierre-André Wacrenier.

- FORESTGOMP is an OPENMP environment based on both the GNU OPENMP run-time and the MARCEL thread library.
- It is designed to schedule efficiently nested sets of threads (derived from nested parallel regions) over hierarchical architectures so as to minimize cache misses and NUMA penalties.
- The FORESTGOMP runtime generates nested MARCEL bubbles each time an OPENMP parallel region is encountered, thereby grouping threads sharing common data.
- Topology-aware scheduling policies implemented by BUBBLESCHED can then be used to dynamically map bubbles onto the various levels of the underlying hierarchical architecture.
- FORESTGOMP allowed us to validate the BUBBLESCHED approach with highly irregular, fine grain, divide-and-conquer parallel applications.
- <http://runtime.bordeaux.inria.fr/forestgomp/>

5.6. Open-MX

Participant: Brice Goglin.

- The OPEN-MX software stack is a high-performance message passing implementation for any generic ETHERNET interface.
- It was developed within our collaboration with Myricom, Inc. as a part of the move towards the convergence between high-speed interconnects and generic networks.
- OPEN-MX exposes the raw ETHERNET performance at the application level through a pure message passing protocol.
- While the goal is similar to the old GAMMA stack [35] or the recent iWarp [34] implementations, OPEN-MX relies on generic hardware and drivers and has been designed for message passing.
- OPEN-MX is also wire-compatible with Myricom MX protocol and interface so that any application built for MX may run on any machine without Myricom hardware and talk other nodes running with or without the native MX stack.
- OPEN-MX is also an interesting framework for studying next-generation hardware features that could help ETHERNET hardware become legacy in the context of high-performance computing. Some innovative message-passing-aware stateless abilities, such as multiqueue binding and interrupt coalescing, were designed and evaluated thanks to OPEN-MX [5].
- Brice GOGLIN is the main contributor to OPEN-MX. The software is already composed of more than 45 000 lines of code in the Linux kernel and in user-space.
- <http://open-mx.org/>

5.7. StarPU

Participants: Cédric Augonnet, Olivier Aumage, Nicolas Collin, Nathalie Furmento, Cyril Roelandt, Ludovic Stordeur, Samuel Thibault, Ludovic Courtès.

- STARPU permits high performance libraries or compiler environments to exploit heterogeneous multicore machines possibly equipped with GPGPUs or Cell processors.
- STARPU offers a unified offloadable task abstraction named codelet. In case a codelet may run on heterogeneous architectures, it is possible to specify one function for each architecture (e.g. one function for CUDA and one function for CPUs).
- STARPU takes care to schedule and execute those codelets as efficiently as possible over the entire machine. A high-level data management library enforces memory coherency over the machine: before a codelet starts (e.g. on an accelerator), all its data are transparently made available on the compute resource.
- STARPU obtains portable performances by efficiently (and easily) using all computing resources at the same time.
- STARPU also takes advantage of the heterogeneous nature of a machine, for instance by using scheduling strategies based on auto-tuned performance models.
- STARPU can also leverage existing parallel implementations, by supporting *parallel tasks*, which can be run concurrently over the machine.
- STARPU provides a *reduction* mode, which permit to further optimize data management when results have to be reduced.
- STARPU provides integration in MPI clusters through a lightweight DSM over MPI.
- STARPU comes with a plug-in for the GNU Compiler Collection (GCC), which extends languages of the C family with syntactic devices to describe STARPU's main programming concepts in a concise, high-level way.
- <http://runtime.bordeaux.inria.fr/StarPU/>

5.8. NewMadeleine

Participants: Alexandre Denis, François Trahay, Raymond Namyst.

- NEWMADELEINE is communication library for high performance networks, based on a modular architecture using software components.
- The NEWMADELEINE optimizing scheduler aims at enabling the use of a much wider range of communication flow optimization techniques such as packet reordering or cross-flow packet aggregation.
- NEWMADELEINE targets applications with irregular, multiflow communication schemes such as found in the increasingly common application conglomerates made of multiple programming environments and coupled pieces of code, for instance.
- It is designed to be programmable through the concepts of optimization *strategies*, allowing experiments with multiple approaches or on multiple issues with regard to processing communication flows, based on basic communication flows operations such as packet merging or reordering.
- The reference software development branch of the NEWMADELEINE software consists in 90 000 lines of code. NEWMADELEINE is available on various networking technologies: Myrinet, Infiniband, Quadrics and ETHERNET. It is developed and maintained by Alexandre DENIS.
- <http://runtime.bordeaux.inria.fr/newmadeleine/>

5.9. PadicoTM

Participant: Alexandre Denis.

- PadicoTM is a high-performance communication framework for grids. It is designed to enable various middleware systems (such as CORBA, MPI, SOAP, JVM, DSM, etc.) to utilize the networking technologies found on grids.
- PadicoTM aims at decoupling middleware systems from the various networking resources to reach transparent portability and flexibility.
- PadicoTM architecture is based on software components. Puk (the PadicoTM micro-kernel) implements a light-weight high-performance component model that is used to build communication stacks.
- PadicoTM component model is now used in NEWMADELEINE. It is the cornerstone for networking integration in the projects “LEGO” and “COOP” from the ANR.
- PadicoTM is composed of roughly 60 000 lines of C.
- PadicoTM is registered at the APP under number IDDN.FR.001.260013.000.S.P.2002.000.10000.
- <http://runtime.bordeaux.inria.fr/PadicoTM/>

5.10. MAQAO

Participants: Denis Barthou, Andres Charif-Rubial.

- MAQAO is a performance tuning tool for OpenMP parallel applications. It relies on the static analysis of binary codes and the collection of dynamic information (such as memory traces). It provides hints to the user about performance bottlenecks and possible workarounds.
- MAQAO relies on binary codes and inserts probes for instrumentation directly inside the binary. There is no need to recompile. The static/dynamic approach of MAQAO analysis is the main originality of the tool, combining performance model with values collected through instrumentation.
- MAQAO has a static performance model for x86 architecture and Itanium. This model analyzes performance of the predecoder, of the decoder and of the different pipelines of the x86 architecture, in particular for SSE instructions.
- The dynamic collection of data in MAQAO enables the analysis of thread interactions, such as false sharing, amount of data reuse, runtime scheduling policy, ...
- MAQAO is in the project “ProHMPT” from the ANR. A demo of MAQAO has been made in Jan. 2010 for SME/Inria days and in Nov. 2010 at SuperComputing, Inria Booth.
- <http://www.maqao.org/>

5.11. QIRAL

Participant: Denis Barthou.

- QIRAL is a high level language (expressed through LaTeX) that is used to describe Lattice QCD problems. It describes matrix formulations, domain specific properties on preconditionings, and algorithms.
- The compiler chain for QIRAL can combine algorithms and preconditionings, checking validity of the composition automatically. It generates OpenMP parallel code, using libraries, such as BLAS.
- This code is developed in collaboration with other teams participating to the ANR PetaQCD project.

5.12. TreeMatch

Participants: Emmanuel Jeannot, Guillaume Mercier, François Tessier.

- TREEMATCH is a library for performing process placement based on the topology of the machine and the communication pattern of the application.
- TREEMATCH provides a permutation of the processes to the processors/cores in order to minimize the communication cost of the application.
- Important features are : the number of processors can be greater than the number of applications processes ; it assumes that the topology is a tree and does not require valuation of the topology (e.g. communication speeds) ; it implements different placement algorithms that are switched according to the input size.
- TREEMATCH is integrated into various software such as the Charm++ programming environment as well as in both major open-source MPI implementations: Open MPI and MPICH2.
- TREEMATCH is available at: <http://treematch.gforge.inria.fr>.

S4 Project-Team

5. Software

5.1. Mica: A Modal Interface Compositional Analysis Toolbox

Participant: Benoît Caillaud.

<http://www.irisa.fr/s4/tools/mica/>

Mica is an Ocaml library developed by Benoît Caillaud implementing the Modal Interface algebra published in [8]. The purpose of Modal Interfaces is to provide a formal support to contract based design methods in the field of system engineering. Modal Interfaces enable compositional reasoning methods on I/O reactive systems.

In Mica, systems and interfaces are represented by extension. However, a careful design of the state and event heap enables the definition, composition and analysis of reasonably large systems and interfaces. The heap stores states and events in a hash table and ensures structural equality (there is no duplication). Therefore complex data-structures for states and events induce a very low overhead, as checking equality is done in constant time.

Thanks to the Inter module and the mica interactive environment, users can define complex systems and interfaces using Ocaml syntax. It is even possible to define parameterized components as Ocaml functions.

Mica is available as an open-source distribution, under the CeCILL-C Free Software License Agreement (http://www.cecill.info/licences/Licence_CeCILL-C_V1-en.html).

5.2. Synet: A General Petri-Net Synthesis Toolbox

Participant: Benoît Caillaud.

<http://www.irisa.fr/s4/tools/synet/>

Synet is a software tool for the synthesis of bounded and unbounded Petri-nets, based on the theory of regions [33]. It can synthesize Petri-nets from automata or regular expressions and can be configured by command-line options to synthesize nets modulo graph isomorphism or language equality. Petri nets computed by Synet can be displayed using the GraphViz 2D graph layout software, or saved to a file for further transformation and analysis.

The tool actually implements two linear-algebraic synthesis methods: a first method uses the simplex algorithm and the second one is based on the computation of extremal rays of polyhedral cones, using Chernikova's algorithm [35]. Both methods imply that the input graphs are given by extension. Nevertheless, Synet yields good performances on many practical use-cases and is the only tool supporting unbounded net synthesis.

The main application of Synet is the synthesis of communicating distributed protocols and controllers [32]. Synthesis is constrained to produce so-called distributable nets [34], a class of nets that can be turned into networks of communicating automata by automated methods. This allows to divide the synthesis problem in two steps: Given the specification of a protocol as a finite automaton, (i) synthesize (if it exists) a distributable net, and then (ii) derive a network of communicating automata from the distributable net. While the second step is automatic and straightforward, the first step is in essence a computer assisted design task, where the distributed Petri-net synthesis algorithm helps the designer to refine the protocol specification into a graph isomorphic to the marking graph of a distributable net.

SAGE Project-Team

5. Software

5.1. H2OLab

Participants: Thomas Dufaud, Jocelyne Erhel [correspondant], Grégoire Lecourt, Aurélien Le Gentil, Géraldine Pichot.

The software platform H2OLab is devoted to stochastic simulations of groundwater flow and contaminant transport in highly heterogeneous porous and fractured geological media. It contains a database which is interfaced through the web portal H2OWeb. It contains also software modules which can be used through the interface H2OGuilde. The platform H2OLab is an essential tool for the dissemination of scientific results. Currently, software and database are shared by the partners of the h2mno4 project (see 8.1.1). Software integrated in the platform and registered at APP are GW-UTIL, GW-NUM, PARADIS, MP-FRAC.

See also the web page <http://h2olab.inria.fr>.

5.2. GW-UTIL

Participants: Jocelyne Erhel, Grégoire Lecourt, Aurélien Le Gentil, Géraldine Pichot [correspondant].

- Version: version 1.0, May 2008
- APP: registered
- Programming language: C++
- See also: <http://h2olab.inria.fr>.
- Abstract: The software GW-UTIL allows to discretize PDE for flow and transport in aquifers and to deal with stochastic models. It contains a set of utility modules for geometry, input, output, random numbers, visualization, parallel computing, numerical algorithms, etc. A package is devoted to launch applications.
- Current work: refactoring.

5.3. GW-NUM

Participants: Thomas Dufaud, Jocelyne Erhel, Grégoire Lecourt, Aurélien Le Gentil, Géraldine Pichot [correspondant].

- Version: version 1.0, May 2008
- APP: registered
- Programming language: C++
- See also: <http://h2olab.inria.fr>.
- Abstract: The software GW-NUM is a set of generic modules to discretize PDE of flow and transport in 2D computational domains in order to deal with stochastic models. Methods for flow simulations are either Finite Volume on structured meshes or Mixed Finite Element with unstructured meshes. Method for transport simulations is a particle tracker for advection and a random walker for diffusion. Uncertainty Quantification method is Monte-Carlo. For flow computations, the involved linear system is solved by external software devoted to sparse matrices.
- Current work: refactoring.

5.4. MP-FRAC

Participants: Thomas Dufaud, Jocelyne Erhel, Aurélien Le Gentil, Géraldine Pichot [correspondant].

- Version: version 1.0, May 2008
- APP: registered
- Programming language: C++
- See also: <http://h2olab.inria.fr>.
- Abstract: The software MP-FRAC aims at modelling and simulating numerically flow in a fractured aquifer. The physical domain is a network of fractures, either deterministic or stochastic, with a permeability field either deterministic or stochastic. The software computes the velocity field in the aquifer, by assuming that the medium is saturated and that flow is steady-state. Physical equations are stochastic PDEs, handled by a Monte-Carlo method. This non intrusive approach generates a set of random samples, which are used for simulations. Then, the software analyzes statistically the flow in the stochastic case. The objective is to characterize hydraulic properties in Discrete Fracture Networks. The software MP-FRAC handles a simulation corresponding to one sample, whereas Monte-Carlo method is implemented in a generic way by the software GW-NUM. The software is specific of the physical model (Discrete Fracture Network) and of the application (steady-state flow). Generic numerical methods to discretize PDE are implemented in the software GW-NUM.
- Current work: refactoring and design of libraries.

5.5. PARADIS

Participants: Jocelyne Erhel, Grégoire Lecourt, Aurélien Le Gentil, Géraldine Pichot [correspondant].

- Version: version 1.0, May 2008
- APP: registered
- Programming language: C++
- See also: <http://h2olab.inria.fr/>.
- Abstract: The software PARADIS aims at modelling and simulating numerically flow in a porous aquifer and transport by convection-diffusion of an inert solute. The porous medium is heterogeneous, with a stochastic or deterministic permeability field. A first step computes the velocity field in the aquifer, by assuming that the medium is saturated and that flow is steady-state. A second step computes the distribution of solute concentration, by assuming a transport by convection and by molecular diffusion. Physical equations are stochastic PDEs, handled by a Monte-Carlo method and discretized by numerical methods. This non intrusive approach generates a set of random samples, which are used for simulations. Then, the software analyzes statistically the flow in the stochastic case. The objectives are to determine asymptotic laws of transport, to characterize pre-asymptotic behavior and to define global laws.

The software PARADIS handles a simulation corresponding to one sample, whereas Monte-Carlo method is implemented in a generic way by the software GW-NUM. The software is specific of the physical model (heterogeneous porous medium) and of the application (steady-state flow then transport with macro-dispersion). Generic numerical methods to discretize PDE are implemented in the software GW-NUM.

- Current work: refactoring and design of libraries.

5.6. GRT3D

Participants: Édouard Canot, Jocelyne Erhel [correspondant], Souhila Sabit.

- Version: version 1.0, April 2011
- APP: registered
- Programming language: C
- Abstract: Reactive transport modeling has become an essential tool for understanding complex environmental problems. It is an important issue for MoMaS partners (see section 8.1.7), in particular Andra (see section 7.1). We have developed a method coupling transport and chemistry, based on a method of lines such that spatial discretization leads to a semi-discrete system of algebraic differential equations (DAE system). The main advantage is to use a complex DAE solver, which controls simultaneously the timestep and the convergence of Newton algorithm. Another approach, called SIA, is to use a fixed-point method to solve the nonlinear system at each timestep.

The software suite GRT3D has four executable modules:

- SIA1D: Sequential Iterative Approach for 1D domains;
- GDAE1D: Global DAE approach for 1D domains;
- SNIA3D: Sequential Non Iterative Approach for 1D, 2D or 3D domains.
- GDAE3D: Global DAE approach for 1D, 2D or 3D domains.
- Current work: extension of the chemistry module and reduction of CPU time.

5.7. GPREMS

Participants: Édouard Canot, Jocelyne Erhel [correspondant].

- Version: version 1.0, May 2008
- APP: registered
- Programming language: C++
- See also: <http://www.irisa.fr/sage/>.
- Abstract: GPREMS implements a robust hybrid solver for large sparse linear systems that combines a Krylov subspace method as accelerator with a Schwarz-based preconditioner. This preconditioner uses an explicit formulation associated to one iteration of the multiplicative Schwarz method. The Newton-basis GMRES, which aims at expressing a good data parallelism between subdomains is used as accelerator.

5.8. DGMRES

Participant: Jocelyne Erhel [correspondant].

- Version: version 1.0, June 2011
- APP: distributed with the free software PETSC
- Programming language: C
- See also: <http://www.irisa.fr/sage/>.
- Abstract: DGMRES implements a preconditioner based on adaptive deflation, which can be used with any preconditioner for the GMRES algorithm.

5.9. AGMRES

Participant: Jocelyne Erhel [correspondant].

- Version: version 1.0, November 2011
- APP: distributed with the free software PETSC
- Programming language: C
- See also: <http://www.irisa.fr/sage/>.
- Abstract: AGMRES implements an augmented subspace approach, based on adaptive deflation, which can be used with any preconditioner for the GMRES algorithm. It also implements a Newton basis for enhancing parallelism.

5.10. PPAT: pseudo-spectrum

Participants: Édouard Canot [corresponding author], Bernard Philippe.

PPAT (Parallel PATH following software) is a parallel code, developed by D. Mezher, W. Najem (University of Saint-Joseph, Beirut, Lebanon) and B. Philippe. This tool can follow the contours of a functional from \mathbb{C} to \mathbb{R}^+ . The present version is adapted for determining the level curves of the function $f(z) = \sigma_{\min}(A - zI)$ which gives the pseudospectrum of matrix A .

The algorithm is reliable: it does not assume that the curve has a derivative everywhere. The process is proved to terminate even when taking into account roundoff errors. The structure of the code spawns many independent tasks which provide a good efficiency in the parallel runs.

The software can be downloaded under the GPL licence from: <http://sourceforge.net/projects/ppat>.

5.11. MUESLI: Scientific computing

Participant: Édouard Canot [corresponding author].

Doing linear algebra with sparse and dense matrices is somehow difficult in scientific computing. Specific libraries do exist to deal with this area (*e.g.* BLAS and LAPACK for dense matrices, SPARSKIT for sparse ones) but their use is often awful and tedious, mainly because of the large number of arguments which must be used. Moreover, classical libraries do not provide dynamic allocation. Lastly, the two types of storage (sparse and dense) are so different that the user must know in advance the storage used in order to declare correctly the corresponding numerical arrays.

MUESLI is designed to help in dealing with such structures and it provides the convenience of coding in Fortran with a matrix-oriented syntax; its aim is therefore to speed-up development process and to enhance portability. It is a Fortran 95 library split in two modules: (i) FML (Fortran Muesli Library) contains all necessary material to numerically work with a dynamic array (dynamic in size, type and structure), called `mfArray`; (ii) FGL (Fortran Graphics Library) contains graphical routines (some are interactive) which use the `mfArray` objects.

MUESLI includes some parts of the following numerical libraries: Arpack, Slatec, SuiteSparse, Triangle, BLAS and LAPACK.

Linux is the platform which has been used for developing and testing MUESLI. Whereas the FML part (numerical computations) should work on any platform (*e.g.* Win32, Mac OS X, Unix), the FGL part is intended to be used only with X11 (*i.e.* under all UNIXes).

Last version of MUESLI is 2.6.6 (2012-08-29). More information can be found at: <http://people.irisa.fr/Edouard.Canot/muesli>

5.12. CANARD: BEM for surface flows

Participant: Édouard Canot [corresponding author].

When dealing with non-linear free-surface flows, mixed Eulerian-Lagrangian methods have numerous advantages, because we can follow marker particles distributed on the free-surface and then compute with accuracy the surface position without the need of interpolation over a grid. Besides, if the liquid velocity is large enough, Navier-Stokes equations can be reduced to a Laplace equation, which is numerically solved by a Boundary Element Method (BEM); this latter method is very fast and efficient because computing occur only on the fluid boundary. This method has been applied to the spreading of a liquid drop impacting on a solid wall and to the droplet formation at a nozzle; applications take place, among others, in ink-jet printing processes.

The code used (CANARD) has been developed with Jean-Luc Achard (LEGI, Grenoble) for fifteen years and is used today mainly through collaborations with Carmen Georgescu at UPB (University Polytechnica of Bucarest, Romania), and with Alain Glière (CEA-LETI, Grenoble).

SARDES Project-Team

4. Software

4.1. AAC_tactics

Participants: Thomas Braibant, Damien Pous [correspondant].

AAC_tactics is a plugin for the Coq proof-assistant that implements new proof tactics for rewriting modulo associativity and commutativity. It is available at http://sardes.inrialpes.fr/~braibant/aac_tactics and as part of the Coq distribution.

- ACM: D.2.4 Software/Program Verification
- Keywords: Rewriting, rewriting modulo AC, proof tactics, proof assistant
- Software benefit: AAC_tactics provides novel efficient proof tactics for rewriting modulo associativity and commutativity.
- License: LGPL
- Type of human computer interaction: N/A
- OS/Middleware: Windows, Linux, MacOS X
- Programming language: Coq

4.2. ATBR

Participants: Thomas Braibant, Damien Pous [correspondant].

ATBR (Algebraic Tools for Binary Relations) is library for the Coq proof assistant that implements new proof tactics for reasoning with binary relations. Its main tactics implements a decision procedure for inequalities in Kleene algebras. It is available at <http://sardes.inrialpes.fr/~braibant/atbr> and as part of the Coq distribution contributed modules.

- ACM: D.2.4 Software/Program Verification
- Keywords: Binary relations, Kleene algebras, proof tactics, proof assistant
- Software benefit: ATBR provides new proof tactics for reasoning with binary relations.
- License: LGPL
- Type of human computer interaction: N/A
- OS/Middleware: Windows, Linux, MacOS X
- Programming language: Coq

4.3. MoKa

Participant: Sara Bouchenak [correspondant].

MOKA is a software framework for the modeling and capacity planning of distributed systems. It first provides a set of tools to build analytical models that describe the behavior of distributed computing systems, in terms of performance, availability, cost. The framework allows to include several model algorithms and to compare them regarding their accuracy and their efficiency. Furthermore, MOKA provides a set of tools to build capacity planning methods. A capacity planning method allows to find a distributed system configuration that guarantee given quality-of-service objectives. MOKA is able to include different capacity planning algorithms and to compare them regarding their efficiency and the optimality of their results. MOKA is available at: <http://sardes.inrialpes.fr/research/moka>.

- ACM: C.2.4 Distributed Systems, C.4 Performance of Systems, D.2.9 Management
- Keywords: Caching, multi-tier systems, consistency, performance
- Software benefit: a novel end-to-end caching protocol for multi-tier services.
- License: TBD
- Type of human computer interaction: command-line interface
- OS/Middleware: Windows, Linux, MacOS X
- Programming language: Java

4.4. ConSer

Participant: Sara Bouchenak [correspondant].

CONSER is a software framework for the modeling and the concurrency and admission control of servers systems. It implements a fluid-based model that exhibits the dynamics and behavior of a server system in terms of service performance and availability. CONSER implements various novel admission control laws for servers such as *AM-C*, *PM-C*, *AA-PM-C* and *PA-AM-C*. A control law produces the server concurrency level that allows to trade-off and meet given service level objectives. CONSER's modeling and control laws algorithms are implemented following a proxy-based approach for more transparency.

- ACM: C.4 Performance of Systems; D.2.9 Management
- Keywords: System management, capacity planning, performance management
- Software benefit: MoKa provides modeling, capacity planning and performance management facilities for application server clusters. Thanks to its model-based capacity planning, MoKa is able to enforce service level objectives while minimizing the service cost.
- License: LGPL
- Type of human computer interaction: web interface
- OS/Middleware: Windows, Linux, MacOS X
- Programming language: Java, AspectJ

4.5. e-Caching

Participants: Damian Serrano-Garcia, Sara Bouchenak [correspondant].

E-CACHING is a software framework for higher scalability of multi-tier Internet services through end-to-end caching of dynamic data. It provides a novel caching solution that allows to cache different types of data (e.g. Web content, database query results, etc.), at different locations of multi-tier Internet services. The framework allows to combine different caches and, thus, to provide higher scalability of Internet services. E-CACHING maintains the integrity of the cached data through novel distributed caching algorithms that guarantee the consistency of the underlying data.

- ACM: C.2.4 Distributed Systems, C.4 Performance of Systems
- Keywords: Caching, multi-tier systems, consistency, performance
- Software benefit: a novel end-to-end caching protocol for multi-tier services, consistency management, performance improvement.
- License: TBD
- Type of human computer interaction: command-line interface
- OS/Middleware: Windows, Linux, MacOS X
- Programming language: Java

4.6. MRB

Participants: Amit Sangroya, Dàmian Serrano-Garcia, Sara Bouchenak [correspondant].

MRB is a software framework for benchmarking the performance and dependability of MapReduce distributed systems. It includes five benchmarks covering several application domains and a wide range of execution scenarios such as data-intensive vs. compute-intensive applications, or batch applications vs. interactive applications. MRB allows to characterize application workload, faultload and dataload, and it produces extensive performance and dependability statistics.

- ACM: C.2.4 Distributed Systems, C.4 Performance of Systems
- Keywords: Benchmark, performance, dependability, MapReduce, Hadoop, Cloud Computing
- Software benefit: the first performance and dependability benchmark suite for MapReduce systems.
- License: TBD
- Type of human computer interaction: GUI and command-line interface
- OS/Middleware: Windows, Linux, MacOS X
- Programming language: Java, Unix Shell scripts

4.7. BZR

Participants: Eric Rutten [correspondant], Gwenaël Delaval [POP ART team].

BZR is a reactive language, belonging to the synchronous languages family, whose main feature is to include discrete controller synthesis within its compilation. It is equipped with a behavioral contract mechanisms, where assumptions can be described, as well as an enforce property part: the semantics of the latter is that the property should be enforced by controlling the behaviour of the node equipped with the contract. This property will be enforced by an automatically built controller, which will act on free controllable variables given by the programmer.

BZR is now further developed with the Pop-Art team, where G. Delaval got a position. It has been designed and developed in the Sardes team in relation with the research topic on Model-based Control of Adaptive and Reconfigurable Systems. It is currently applied in different directions: component-based design and the Fractal framework; real-time control systems and the Orccad design environment; operating systems and administration loops in virtual machines; hardware and reconfigurable architecture (FPGAs).

See also the web page <http://bzs.inria.fr>.

- ACM: D.3.3 [Programming Languages]: Language Constructs and Features—Control structures; C.3 [Special-purpose and Application-based Systems]: Real-time and embedded systems; D.2.2 [Software Engineering]: Design Tools and Techniques—Computer-aided software engineering, State diagrams; D.2.4 [Software Engineering]: Software / Program Verification—Formal methods, Programming by contract
- Keywords: Discrete controller synthesis, modularity, components, contracts, reactive systems, synchronous programming, adaptive and reconfigurable systems
- Software benefit: the first integration of discrete control synthesis in a compiler, making it usable at the level of the programming language.
- License: TBD
- Type of human computer interaction: programming language and command-line interface
- OS/Middleware: Linux
- Programming language: Caml; generates C or Java or Caml executable code

SCIPORT Team

5. Software

5.1. AIRONUM

Participant: Alain Dervieux [correspondant].

AIRONUM is an experimental software that solves the unsteady compressible Navier-Stokes equations with $k - \epsilon$, LES-VMS and hybrid turbulence modelling on parallel platforms with Mpi as parallel programming concept. The mesh model is unstructured tetrahedrization, with possible mesh motion. See also <http://www-sop.inria.fr/tropics/aironum>

- Version: v 1.0
- Programming language: Fortran95 (mostly). About 100,000 lines.

AIRONUM was developed by Inria and university of Montpellier. It is used by Inria, university of Montpellier and university of Pisa (I). AIRONUM is used as an experimental platform for:

- Numerical approximation of compressible flows, such as upwind mixed element volume approximation with superconvergence on regular meshes.
- Numerical solution algorithms for the implicit time advancing of the compressible Navier-Stokes equations, such as parallel scalable deflated additive Schwarz algorithms.
- Turbulence modelling such as the Variational Multiscale Large eddy Simulation and its hybridization with RANS statistical models.

5.2. TAPENADE

Participants: Laurent Hascoët [correspondant], Valérie Pascual.

TAPENADE is an Automatic Differentiation tool that transforms an original program into a new program that computes derivatives of the original program. Automatic Differentiation produces analytical derivatives, that are exact up to machine precision. Adjoint-mode AD can compute gradients at a cost which is independent from the number of input variables. TAPENADE accepts source programs written in Fortran77, Fortran90, or C. It provides differentiation in the following modes: tangent, vector tangent, and adjoint. Documentation is provided on the web site of the reserch team and as the Inria technical report RT-0300. TAPENADE runs under most operating systems and requires installation of Java jdk1.6 or upward. See also <http://www-sop.inria.fr/tropics/>

- Version: v3.6, r4343, February 2012
- ACM: D.3.4 Compilers; G.1.0 Numerical algorithms; G.1.4 Automatic differentiation; I.1.2 Analysis of algorithms
- AMS: 65K10; 68N20
- APP: IDDN.FR.001.040038.002.S.P.2002.000.10600
- Keywords: automatic differentiation, adjoint, gradient, optimisation, inverse problems, static analysis, data-flow analysis, compilation
- Programming language: Java

TAPENADE implements the results of our research about models and static analyses for AD. TAPENADE can be downloaded and installed on most architectures. Alternatively, it can be used as a web server. TAPENADE differentiates computer programs according to the model described in section 3.1 and in [19] Higher-order derivatives can be obtained through repeated application of tangent AD on tangent- and/or adjoint-mode AD.

TAPENADE performs sophisticated data-flow analysis, flow-sensitive and context-sensitive, on the complete source program to produce an efficient differentiated code. Analyses include Type-Checking, Read-Write analysis, and Pointer analysis. AD-specific analysis include:

- **Activity analysis:** Detects variables whose derivative is either null or useless, to reduce the number of derivative instructions.
- **Adjoint Liveness analysis:** Detects the source statements that are dead code for the computation of derivatives.
- **TBR analysis:** In adjoint-mode AD, reduces the set of source variables that need to be recovered.

TAPENADE is not open-source. Academic usage is free. Industrial or commercial usage require a paying license, as detailed on the team's web page. The software has been downloaded several hundred times, and the web tool served several thousands of true connections (not counting robots). The tapenade-users mailing list is over one hundred registered users.

SCORE Team

5. Software

5.1. Wiki3.0

Participants: Luc André, Bogdan Flueraș, Claudia-Lavinia Ignat [contact], Gérald Oster.

In the context of the Wiki 3.0 project (<http://wiki30.xwikisas.com/>) (december 2009 - june 2012) sponsored by the call for projects “Innovative Web” launched by the French Ministry of Economy, SCORE team designed and integrated real-time editing features into the XWiki system (<http://www.xwiki.org>). We designed solutions for a raw text editor as well as for a WYSIWYG editor for XWiki pages. The real-time wiki editor has been released as an extension of XWiki (<http://extensions.xwiki.org/xwiki/bin/view/Extension/RealTime+Wiki+Editor>).

SECRET Project-Team (section vide)

SECSI Project-Team

5. Software

5.1. Tookan

Participants: Graham Steel [correspondant], Romain Bardou.

See also the web page <http://tookan.gforge.inria.fr/>.

Tookan is a security analysis tool for cryptographic devices such as smartcards, security tokens and Hardware Security Modules that support the most widely-used industry standard interface, RSA PKCS#11. Each device implements PKCS#11 in a slightly different way since the standard is quite open, but finding a subset of the standard that results in a secure device, i.e. one where cryptographic keys cannot be revealed in clear, is actually rather tricky. Tookan analyses a device by first reverse engineering the exact implementation of PKCS#11 in use, then building a logical model of this implementation for a model checker, calling a model checker to search for attacks, and in the case where an attack is found, executing it directly on the device. Tookan has been used to find at least a dozen previously unknown flaws in commercially available devices.

The first results using Tookan were published in 2010 [48] and a six-month licence was granted to Boeing to use the tool. In 2011, a contract was signed with a major UK bank. Tookan is now the subject of a CSATT transfer action resulting in the hiring of an engineer, Romain Bardou, who started on September 1st, 2011. During 2012 Bardou and Steel implemented a new version of Tookan that is intended to form the technological basis for a spin-off company to be created in 2013. As a result of the transfer of Graham Steel and Romain Bardou to team Prosecco, this project is being continued in that team.

5.2. Orchids

Participants: Jean Goubault-Larrecq [correspondant], Hedi Benzina, Nasr-Eddine Yousfi.

The ORCHIDS real-time intrusion detection system was created in 2003-04 at SECSI. After a few years where research and development around ORCHIDS was relatively quiet, several new things happened, starting from the end of 2010.

First, several companies and institutions expressed interest in ORCHIDS, among which, notably, EADS Cassidian, Thalès, Galois Inc. (USA), the French Direction Générale de l'Armement (DGA).

Second, Baptiste Gourdin was hired as a development engineer (Dec. 2010-Nov. 2011) on an Action de Développement Technologique (ADT). He improved Orchids in several ways.

Nasr-Eddine Yousfi followed up on Baptiste Gourdin, starting from December 2011, on an ITI engineer position allotted by Inria's CSATT. He mostly explored ways of writing security meta-policies for confidentiality of sensitive data.

Orchids will be the core of a contract between Inria and DGA, to be signed in December 2012, for three years.

SELECT Project-Team

5. Software

5.1. MIXMOD software

Participants: Gilles Celeux [Correspondant], Erwan Le Pennec.

Mixture model, cluster analysis, discriminant analysis

MIXMOD is being developed in collaboration with Christophe Biernacki, Florent Langrognet (Université de Franche-Comté) and Gérard Govaert (Université de Technologie de Compiègne). MIXMOD (MIXture MODelling) software fits mixture models to a given data set with either a clustering or a discriminant analysis purpose. MIXMOD uses a large variety of algorithms to estimate mixture parameters, e.g., EM, Classification EM, and Stochastic EM. They can be combined to create different strategies that lead to a sensible maximum of the likelihood (or completed likelihood) function. Moreover, different information criteria for choosing a parsimonious model, e.g. the number of mixture component, some of them favoring either a cluster analysis or a discriminant analysis view point, are included. Many Gaussian models for continuous variables and multinomial models for discrete variable are available. Written in C++, MIXMOD is interfaced with SCILAB and MATLAB. The software, the statistical documentation and also the user guide are available on the Internet at the following address: <http://www.mixmod.org>.

Since this 2010, MIXMOD has a proper graphical user interface (Version 1) which has been presented at the MIXMOD day in Lyon in December 2010. A version of MIXMOD in R is now available <http://cran.r-project.org/web/packages/Rmixmod/index.html>.

Erwan Le Pennec with the help of Serge Cohen has proposed a spatial extension in which the mixture weights can vary spatially.

SEMAGRAMME Team

5. Software

5.1. Leopard

Participants: Bruno Guillaume [correspondant], Guy Perrier.

Interaction Grammar, parsing

5.1.1. Software description

Leopard is a parser for natural languages which is based on the formalism of Interaction Grammars [30]. It uses a parsing principle, called “electrostatic parsing” which consists in neutralizing opposite polarities. A positive polarity corresponds to an available linguistic feature and a negative one to an expected feature.

Parsing a sentence with an Interaction Grammar consists in first selecting a lexical entry for each of its words. A lexical entry is an underspecified syntactic tree, a tree description in other words. Then, all selected tree descriptions are combined by partial superposition guided by the aim of neutralizing polarities: two opposite polarities are neutralized by merging their support nodes. Parsing succeeds if the process ends with a minimal and neutral tree. As IGs are based on polarities and under-specified trees, Leopard uses some specific and non-trivial data-structures and algorithms.

The electrostatic principle has been intensively considered in Leopard. The theoretical problem of parsing IGs is NP-complete; the nondeterminism usually associated to NP-completeness is present at two levels: when a description for each word is selected from the lexicon, and when a choice of which nodes to merge is made. Polarities have shown their efficiency in pruning the search tree:

- In the first step (tagging the words of the sentence with tree descriptions), we forget the structure of descriptions, and only keep the bag of their features. In this case, parsing inside the formalism is greatly simplified because composition rules reduce to the neutralization of a negative feature-value pair $f \leftarrow v$ by a dual positive feature-value pair $f \rightarrow v$. As a consequence, parsing reduces to a counting of positive and negative polarities present in the selected tagging for every pair (f, v) : every positive occurrence counts for +1 and every negative occurrence for -1, the sum must be 0.
- Again in the tagging step, original methods were developed to filter out bad taggings. Each unsaturated polarity p in the grammar induces constraints on the set of contexts in which it can be used: the unsaturated polarity p must find a *companion* (i.e. a tree description able to saturate it); and the set of companions for the polarity p can be computed statically from the grammar. Each lexical selection which contains an unsaturated polarity without one of its companions can be safely removed.
- In the next step (node-merging phase), polarities are used to cut off parsing branches when their trees contain too many non neutral polarities.

5.1.2. Current state of the implementation

Leopard is presented and documented at <http://leopard.loria.fr>; an online demonstration page can be found at <http://leopard.loria.fr/demo>.

It is open-source (under the CECILL License <http://www.cecill.info>) and it is developed using the InriaGforge platform (<http://gforge.inria.fr/projects/semagramme/>)

The main features of current software are:

- automatic parsing of a sentence or a set of sentences,
- dependency and parse-tree representation of sentences,
- interactive parsing (the user chooses the couple of nodes to merge),
- visualization of grammars produced by XMG-2 or of sets of description trees associated to some word in the linguistic resources,

5.2. ACG Development Toolkit

Participants: Sylvain Pogodalla [correspondant], Philippe de Groot.

In order to support the theoretical work on ACG, we have been developing a support system. The objectives of such a system are twofold:

1. to make possible to implement and experiment grammars the modeling of linguistic phenomena;
2. to make possible to implement and experiment results related to the ACG formalisms. Such results can concern parsing algorithms, type extensions, language extensions, etc.

The ACG Development toolkit development effort is part of the POLYMNIE project (see Section 7.2.1.1). It will support the experimentation and evaluation parts of the project.

The current version of the the ACG development toolkit prototype ¹ issues from a first release published in October 2008. Further releases have been published before the ESSLLI 2009 course on ACG. It focuses on providing facilities to develop grammars. To this end, the type system currently implemented is the linear core system plus the (non-linear) intuitionistic implication, and a special attention has been paid to type error management. As a major limitation, this version only considers transformation from abstract terms to object terms, and not the other way around.

Enabling transformation from the object terms to the abstract terms is the first step of future development for the ACG support system. A parsing algorithm based on [32]'s methods is being implemented for second-order ACGs. It is based on a translation of ACG grammars into Datalog programs and is well-suited to fine-grained optimization.

However, since we're interested not only by recognizability (hence whether some fact is provable) but also by the parsing structure (hence the proof), the Datalog solver requires further adaptations. Note however that in the general case, the decidability of translating an object term to an abstract one is still an open problem.

5.3. Grew

Participants: Bruno Guillaume [correspondant], Guy Perrier.

Graph rewriting, Interface syntaxe-sémantique

Grew is a Graph Rewriting tools dedicated to applications in NLP. It is freely-available (from the page <http://grew.loria.fr>) and it is developed using the InriaGforge platform (<http://gforge.inria.fr/projects/semagramme/>)

We list below some of the major specificities of the GREW software.

- Graph structures can use a build-in notion of feature structures.
- The left-hand side of a rule is described by a graph called a pattern; injective graph morphisms are used in the pattern matching algorithm.
- Negative pattern can be used for a finer control on the left-hand side of rules.
- The right-hand side or rules is described by a sequence of atomic commands that describe how the graph should be modified during the rule application.
- Rules can be parametrized by lexical information.
- Filters can be used at the output of each module to control the structure produced are well-formed.
- Subset of rules are grouped in modules; the full rewriting process being a sequence of module applications.
- The Grew software has support both for confluent and non-confluent modules; when a non-confluent modules is used, all normal forms are returned and then ambiguity is handled in a natural way.
- Grew can be used on Corpus mode with statistics about rules usage or with an a Graphical User Interface which can show all intermediate graphs used during the rewriting process (useful either to debug rewriting system or for demonstrations).

¹ Available at <http://acg.gforge.inria.fr> with a CeCILL license.

A demonstration of the Grew Software was presented at the TALN conference in June in Grenoble.[15]

The Grew software were used for several kind of applications manipulating syntactic and/or semantic graph representations. It was used to build DMRS semantic representation from syntactic dependency trees in the French TreeBank [12], [14]. More recently, it was used on the Sequoia TreeBank, to produce deep syntax annotation and DMRS Semantic representations.

Another application of the Grew software which is currently investigated is the detection of annotation errors in corpora. Graph Rewriting is used to detect ill-formed structures that don't fit the annotation guide requirements. In collaboration with the Alpage team, this was applied to the Sequoia Corpus and the reported errors were corrected in version 3.2 and 3.3 of the corpus².

5.4. Other developments

Participant: Bruno Guillaume [correspondant].

Concordancer, Dependencies, Graphical tools Other peripheral developments of the team are available either as web service or as downloadable code:

- A concordancer named CONDOR which is usable online: <http://condor.loria.fr>. With Condor, it is possible to search for all inflexions (given by a lexicon) of some lemma; it is possible to search for a couple of lemmas to find collocations.
- A program (named DEP2PICT) to build graphical representations (PNG, SVG or PDF) of dependency structures. It is presented in <http://dep2pict.loria.fr>; it is usable online <http://dep2pict.loria.fr/demo>.

²<https://www.rocq.inria.fr/alpage-wiki/tiki-index.php?page=CorpusSequoia>

SEQUEL Project-Team

5. Software

5.1. Introduction

In 2012, SEQUEL continued the development of software for computer games (notably Go) and also developed two novel libraries for functional regression and data mining.

5.2. Computer Games

Participant: Rémi Coulom.

We continued the development of three main softwares for computer games:

- *Crazy Stone* is a top-level Go-playing program that has been developed by Rémi Coulom since 2005. Crazy Stone won several major international Go tournaments in the past. In 2012, a new version was released in Japan. This new version won a game with a 4-stone handicap against a professional player during the European Go Congress in Bonn, Germany. It is distributed as a commercial product by *Unbalance Corporation* (Japan). 6-month work in 2012. URL: <http://remi.coulom.free.fr/CrazyStone/>
- *Crazy Hanafuda* is a program to play the Japanese card game of Hanafuda. One month of work in 2012. A licence agreement was signed with Unbalance Corporation in January. The Windows 8 version of the program was released commercially in November.
- *CLOP* [30] is a tool for automatic parameter optimization of game-playing programs. Distributed as freeware (GPL). One month of work in 2012. Available at: <http://remi.coulom.free.fr/CLOP/>

5.3. Vowpal Wabbit

Participants: Jérémie Mary, Romaric Gaudel, Thomas Chabin.

Vowpal Wabbit is a GPL project led by John Langford at Yahoo! Research and now at Microsoft. The goal is to build a very fast, distributed and large scale machine learning software. https://github.com/JohnLangford/vowpal_wabbit/wiki. We worked on the optimization of the parser and on the memory structures of the i/o. The modifications have been accepted for commit in the main branch and allow an average division by two of all execution times.

SERPICO Team

5. Software

5.1. Software for live cell imaging

Participants: Charles Kervrann, Patrick Bouthemy, Tristan Lecorgne.

Motion2d: parametric motion model estimation

The MOTION2D software written in C++ (APP deposit number: FR.001.520021.001.S.A.1998.000.21000 / release 1.3.11, January 2005) and JAVA (plug-in IMAGEJ (<http://rsbweb.nih.gov/ij/>)) is a multi-platform object-oriented library to estimate 2D parametric motion models in an image sequence. It can handle several types of motion models, namely, constant (translation), affine, and quadratic models. Moreover, it includes the possibility of accounting for a global variation of illumination. The use of such motion models has been proved adequate and efficient for solving problems such as optic flow computation, motion segmentation, detection of independent moving objects, object tracking, or camera motion estimation, and in numerous application domains (video surveillance, visual servoing for robots, video coding, video indexing), including biological imaging (image stack registration, motion compensation in videomicroscopy). Motion2D is an extended and optimized implementation of the robust, multi-resolution and incremental estimation method (exploiting only the spatio-temporal derivatives of the image intensity function). Real-time processing is achievable for motion models involving up to six parameters. Motion2D can be applied to the entire image or to any pre-defined window or region in the image.

Free academic software distribution: Motion2D Free Edition is the version of Motion2D available for development of Free and Open Source software only. More information on Motion2D can be found at <http://www.irisa.fr/vista/Motion2D> and the software can be downloaded at the same Web address.

On-line demo: Mobylye@GenOuest Bioinformatics <http://mobylye.genouest.org/cgi-bin/Mobylye/portal.py#forms:Motion2D> (see Fig. 5).

Partner: Fabien Spindler (Inria Lagadic team).

ND-Safir and Fast2D-SAFIR: Image denoising software

The ND-SAFIR software (APP deposit number: IDDN.FR.001.190033.002.S.A.2007.000.21000 / new release 3.0 in 2013) written in C++, JAVA and MATLAB, removes additive Gaussian and non-Gaussian noise in still 2D or 3D images or in 2D or 3D image sequences (with no motion computation) (see Figure 2) [4]. The method is unsupervised and is based on a pointwise selection of small image patches of fixed size (a data-driven adapted way) in spatial or space-time neighbourhood of each pixel (or voxel). The main idea is to modify each pixel (or voxel) using the weighted sum of intensities within an adaptive 2D or 3D (or 2D or 3D + time) neighbourhood and to use image patches to take into account complex spatial interactions. The neighbourhood size is selected at each spatial or space-time position according to a bias-variance criterion. The algorithm requires no tuning of control parameters (already calibrated with statistical arguments) and no library of image patches. The method has been applied to real noisy images (old photographs, JPEG-coded images, videos, ...) and is exploited in different biomedical application domains (time-lapse fluorescence microscopy, video-microscopy, MRI imagery, X-ray imagery, ultrasound imagery, ...).

The FAST-2D-SAFIR software (APP deposit number: IDDN.FR.001.190033.001.S.A.2007.000.21000) written in C++ removes mixed Gaussian-Poisson noise in large 2D images, typically $10^3 \times 10^3$ pixels, in a few seconds. The method is unsupervised and is a simplified version of the method related to the SAFIR-nD software. The software dedicated to microarrays image denoising, was licensed to the INNOPSYS company which develops scanners for disease diagnosis and multiple applications (gene expression, genotyping, aCGH, ChIP-chip, microRNA, ...).

On-line demo: Mobylye@GenOuest Bioinformatics

<http://mobylye.genouest.org/cgi-bin/Mobylye/portal.py#forms::NDSafir>

Free download binaries: Binaries of the software ND-SAFIR are freely and electronically distributed. Developed in standard C/C++ under Linux using the CImg library, it has been tested over several platforms such as Linux/Unix, Windows XP and Mac OS.

Academic licence agreements: Institut Curie, CNRS, ENS Ulm, Oxford university, Weizmann Institute, UCSF San-Francisco, Harvard university, Berkeley university, Stanford university, Princeton university, Georgia-Tech, Kyoto university, IMCB Singapore ...

Partners: J. Boulanger, J. Salamero (UMR 144 CNRS Institut Curie), P. Elbau (RICAM Linz, Austria), J.B. Sibarita (UMR 5091 University of Bordeaux 2).

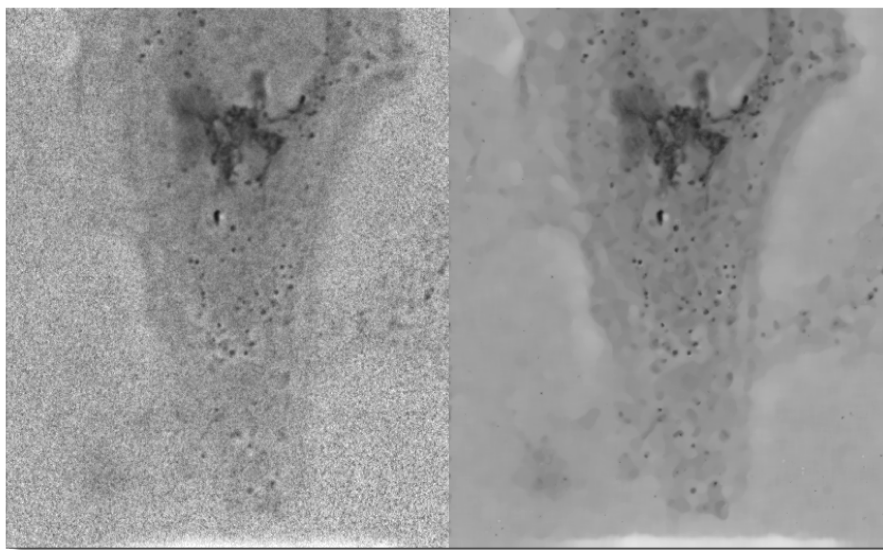


Figure 2. ND-SAFIR software: denoising of a 3D image sequence in wide-field (WF) microscopy (GFP-Rab6A (Hela cell), UMR 144 CNRS Institut Curie).

HullkGround: Background subtraction by convex hull estimation

The HULLKGROUND software (APP deposit number: IDDN.FR.001.400005.000.S.P.2009.000.21000) written in JAVA (plug-in IMAGEJ, see Fig. 3) decomposes a fluorescence microscopy image sequence into two dynamic components: i/ an image sequence showing mobile objects; ii/ an image sequence showing the slightly moving background. Each temporal signal of the sequence is processed individually and analyzed with computational geometry tools. The convex hull is estimated automatically for each pixel and subtracted to the

original signal. The method is unsupervised, requires no parameter tuning and is a simplified version of the α shapes-based scale-space method [25].

On-line demo: MobyLe@GenOuest Bioinformatics <http://mobyLe.genouest.org/cgi-bin/MobyLe/portal.py#forms::Hullkground>

Partners: A. Chessel and J. Salamero (UMR 144 CNRS Institut Curie)

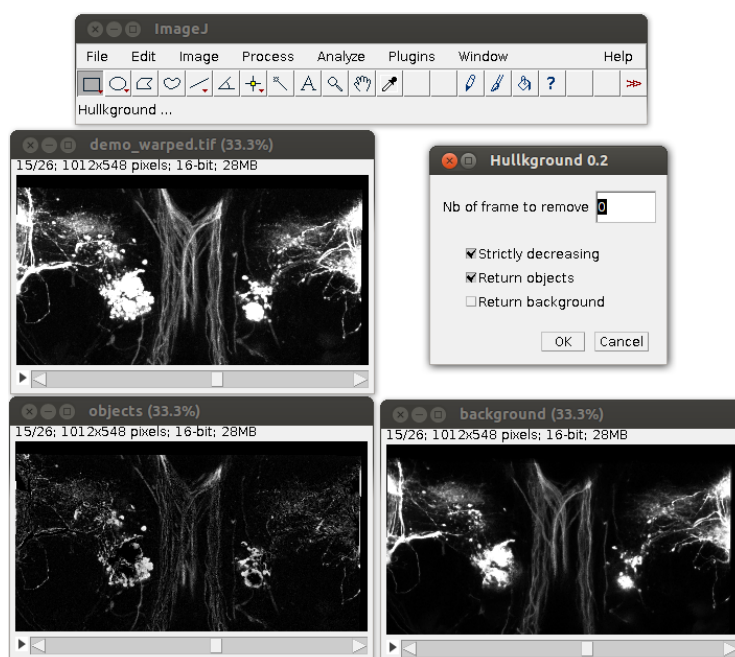


Figure 3. HULLKGROUND software: plug-in IMAGEJ.

5.2. Software for Cyo-electron tomography

Participant: Charles Kervrann.

TubuleJ: Straightening of microtubule cryo-EM projection views

The TUBULEJ software (APP deposit number: IDDN.FR.001.240023.000.S.P.2011.000.21000) written in JAVA (plug-in IMAGEJ) is devoted to the analysis of microtubules and helical structures in 2D cryo-electron microscope images. The software straightens curved microtubule images by estimating automatically points locations on the microtubule axis. The estimation of microtubule principal axis relies on microtubule cylindrical shape analyzed in the Fourier domain. A user-friendly interface enables to filter straight fiber images by selecting manually the layer lines of interest in the Fourier domain. This software can be used to generate a set of 2D projection views from a single microtubule projection view and a few parameters of this microtubule structure. These projection views are then back projected, by using the IMOD plug-in (<http://rsbweb.nih.gov/ij/>), to reconstruct 3D microtubules.

On-line demo: see <http://equipes.igdr.univ-rennes1.fr/en/tips/Software/TubuleJ/>

Partners: S. Blestel and D. Chrétien (UMR 6290 CNRS University of Rennes 1)

Cryo-Seg: Segmentation of tomograms in cryo-electron microscopy

The CRYO-SEG software written in C++ and JAVA (plug-in MAGEJ) has been developed to detect microtubule structures and helical structures in 2D cryo-electron microscope images (see Figure 4). Cryo-electron tomography allows 3D observation of biological specimens in their hydrated state. Segmentation is formulated as Maximum A Posteriori estimation problem and exploits image patches to take into account spatial contexts (Markov Random Fields). Because of the contrast anisotropy in the specimen thickness direction, the whole tomogram is segmented section by section, with an automatic update of reference patches. This algorithm has been evaluated on synthetic data and on cryo-electron tomograms of in vitro microtubules. On real data, this segmentation method extracts the most contrasted regions of microtubules, and 3D visualization is improved.

Partners: S. Blestel and D. Chrétien (UMR 6290 CNRS University of Rennes 1)

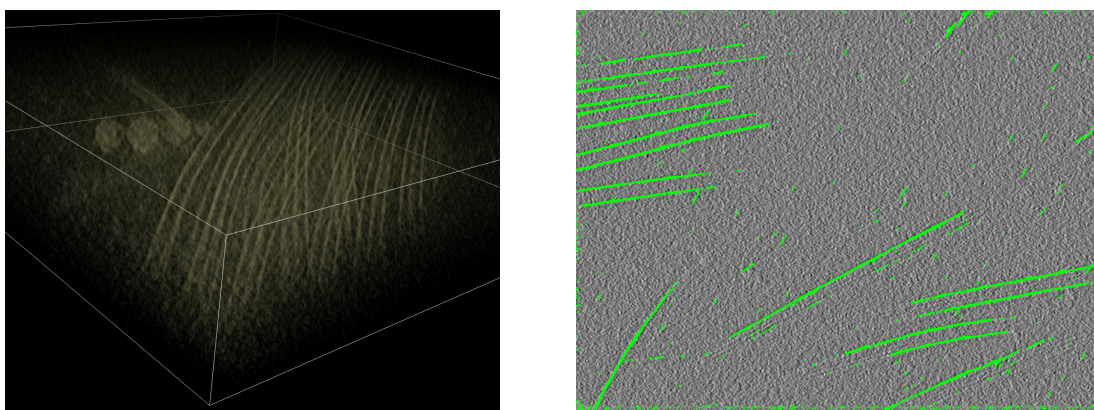


Figure 4. CRYO-SEG software: Segmentation of 3D microtubules in a cryo-EM tomogram (left) and 2D view (right) (UMR 6290 CNRS University of Rennes 1).

5.3. Image Processing software distribution

Participants: Tristan Lecorgne, Charles Kervrann.

The objective is to disseminate the distribution of SERPICO image processing software for biologist users:

- *Free binaries:* software packages have been compiled for the main operating systems (Linux, MacOS, Windows) using CMake (see <http://www.cmake.org/>). They are freely available on the team website under a proprietary license (e.g. ND-SAFIR and HULLKGROUND are distributed this way at <http://serpico.rennes.inria.fr/doku.php?id=software:index>).
- *Mobyly web portal:* An online version of the software has been released using the Mobyly framework (see <http://mobyly.pasteur.fr/>) developed at Institut Pasteur. The main role of this web portal is to demonstrate the performance of the programs developed by the team. The web interface makes our image processing methods available for biologist users without any installation or configuration (see ND-SAFIR, HULLKGROUND, MOTION2D (see Fig. 5) at <http://mobyly.genouest.org/>). The size of submitted images is limited by network bandwidth. We use the computing facility of the GenOuest platform to run calculations.

- **IMAGEJ plug-ins:** IMAGEJ (see <http://rsb.info.nih.gov/ij/>) is a widely used image visualization and analysis software for biologist users. We have developed IMAGEJ plug-in JAVA versions of the following software: ND-SAFIR, HULLKGROUND (see Fig. 3), MOTION2D, HOTSPOTDETECTION and OPTICALFLOW.
- **Institut Curie database:** Institut Curie is currently acquiring a new database system to store mass of data. The database can be searched via meta-data and includes menu selections that enable to run remote processing. We have integrated ND-SAFIR in the interface environment to allow the database users to denoise images easily.

Partners: C. Deltel (Inria Rennes SED) and Perrine Paul-Gilloteux (UMR 144 PICT IBISA CNRS Institut Curie)

The screenshot displays the Moby@GenOuest Bioinformatics program interface. The top navigation bar includes a search box, a 'Welcome' tab, and other tabs like 'Forms', 'Data Bookmarks', 'Jobs', and 'Tutorials'. The left sidebar contains a 'Programs' menu with categories such as alignment, assembly, Bio-Imaging, and Motion2D. The main content area is titled 'Motion 2D' and 'Estimate 2D parametric motion model'. It features a 'Run' button and a 'Reset' button. Below the title, there is a 'Copyright' section with a notice from INRIA. The 'Input' section includes an 'Image sequence TIFF file' upload area with a 'CLEAR' button and a file selection box. The 'Input Options' section contains several configuration fields: 'Model type' set to 'AC (affine) (6 parameters (c1, c2, a1...a4))', 'First Frame number' set to 0, 'Step between two frames in the video sequence' set to 1, and 'Number of motion estimation iterations to process' set to 0.

Figure 5. Motion2D on Moby@GenOuest Bioinformatics.

SHACRA Project-Team

5. Software

5.1. SOFA

SOFA, the Simulation Open Framework Architecture, is an international, multi-institution, collaborative initiative, aimed at developing a flexible and open source framework for interactive simulations. This will eventually establish new grounds for a widely usable standard system for long-term research and product prototyping, ultimately shared by many academic and industrial sites. Over the last two years, the SOFA framework has evolved from an informal collaborative work between the Sim Group at CIMIT, the Alcove, Asclepios and Evasion teams at Inria into a more structured development project. By proposing a unique architecture allowing the integration of the multiple competencies required for the development of a medical training system, we believe it will be possible to accelerate and foster research activities in the field of interactive medical simulation. The main objectives of the SOFA framework are:

- Simplify the development of medical simulation systems by improving interoperability
- Evaluate and validate new algorithms
- Accelerate the prototyping of simulation systems by promoting component reusability
- Promote collaboration between research groups
- Facilitate technology transfer between research and industry

Our activities around the SOFA framework will be twofold. We will remain one of the leading teams contributing to the design of SOFA, the development of its architecture and its distribution to research groups and industrial partners. In addition, we will use SOFA as a core element of most of our simulations, as a mean to facilitate the integration of results from partners of the national initiative, and to simplify the development of prototypes of simulation systems. For the past few years, there have been a few attempts at designing software toolkits for medical simulation. Examples include [36], GiPSi [25], SPORE [35] or SSTML [22]. These different solutions aim at the same goal: providing an answer (usually Open Source) to the various challenges of medical simulation research and development. Although our aim is similar, we propose a different approach, through a very modular and flexible software framework, while minimizing the impact of this flexibility on the computation overhead. To achieve these objectives, we have developed a new architecture that implements a series of innovative concepts. Also, by developing the SOFA framework collaboratively with scientific experts in the different areas of medical simulation, we believe we can provide state-of-the-art solutions that are generically applicable, yet computationally efficient. The following sections describe in more details our approach to the development of this framework, from a technical standpoint and from the perspective of a collaborative work.

5.1.1. SOFA architecture

Medical simulation relies on a variety of interacting physics-based models, such as rigid structures (e.g. bones), deformable structures (e.g. soft-tissues) and fluids. It also involves anatomical representations through geometrical models, used for visual rendering, collision detection or meshes that will support various computational models. Finally, interactions between these different models need to be efficient, accurate and capable of handling a variety of representations. In some instances, a hierarchy also exists between the various anatomical structures, and needs to be taken into account in the description of the simulated environment. The design of the SOFA architecture, by supporting these various requirements, brings the flexibility needed for academic research. Yet, its very efficient implementation makes it also suitable for professional applications and potentially for product development. This architecture relies on several innovative concepts, in particular the notion of multi-model representation. In SOFA, most simulation components (deformable models, collision models, medical devices, etc.) can have several representations, connected through a mechanism called mapping. Each representation is optimized for a particular task (e.g. collision detection, visualization) while at the same time

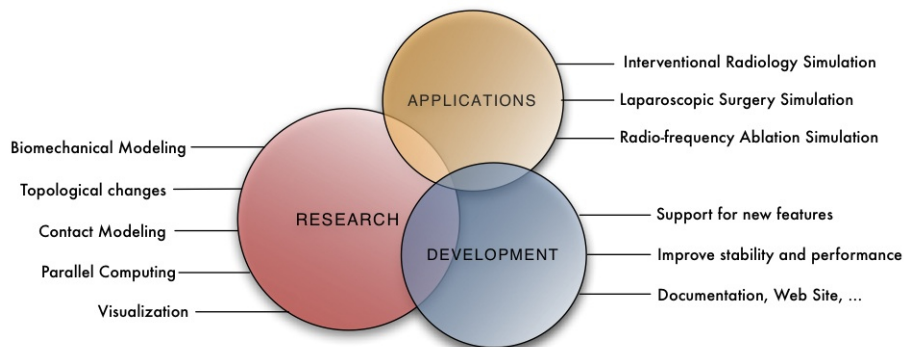


Figure 3. Multidisciplinary research and development of the SOFA framework need to take place simultaneously to quickly advance research in the field of computer-based interactive medical simulation

improving interoperability by creating a clear separation between the functional aspects of the simulation components. As a consequence, it is possible to have models of very different nature interact together, for instance rigid bodies, deformable objects, and fluids. This is an essential aspect of SOFA, as it will help the integration of new research components. This modular design also facilitates the rapid prototyping of simulation systems, allowing various combinations of algorithms to be tested and compared against each other. At a finer level of granularity, we also propose a decomposition of physical models (i.e. any model that behaves according to the laws of physics) into a set of basic components. In the case of (bio)mechanical models, which are computationally expensive, many strategies have been used to improve computation times or to reduce the complexity of the original model: linear elastic models have often been used instead of more complex non-linear representations, mass-spring methods as an alternative to finite element methods, etc. Each of these simplifications induces drawbacks, yet the importance of these drawbacks depends largely on the context in which they are applied. It becomes then very difficult to choose which particular method is most likely to provide the best results for a given simulation. To address this issue in SOFA we have introduced a finer level of granularity which permits to independently test and compare each component, such as time integration schemes, to see the change in performance or robustness of the simulation, or to test different constitutive models. These changes can be made in a matter of seconds, without having to recompile any of the code, by simply editing an XML file.

5.1.2. Current Results

Version 1.0 RC1 of SOFA was released in December 2011 but since October 2012, SOFA is now available through a public and anonymous SVN. More than 137,000 downloads of SOFA have been counted as of November 2012. More than 70 researchers, students, engineers have contributed at various degrees to SOFA, for a total of about 1,200,000 lines of code. Currently, thanks to its advanced architecture, SOFA allows to:

- Create complex and evolving simulations by combining new algorithms with existing algorithms
- Modify most parameters of the simulation by simply editing a XML file
- Build complex models from simpler ones using a scene-graph description
- Efficiently simulate the dynamics of interacting objects using abstract equation solvers
- Reuse and easily compare a variety of available methods
- Transparently parallelize complex computations using semantics based on data dependencies
- Use new generations of GPUs through the CUDA API to greatly improve computation times
- Use embedded Python environment to create interactive and parametric scenes, and interact with 3rd party software

Various results and information can be obtained on the SOFA website at <http://www.sofa-framework.org>. Most of the current results are generic and only aim at validating the different aspects of the SOFA framework. Developments of complex medical simulations have recently started, in particular in the areas of ophthalmic surgery and interventional radiology. We have also started a collaboration with a few companies (Digital Trainers, Didhaptics, B.K.) which are in the process of developing medical applications based on SOFA.



Figure 4. Animation of a chain combining a FEM model, a mass-spring model, a FFD grid, and a rigid body. This example is a perfect illustration of the flexibility of SOFA. Not only several algorithms for rigid or deformable bodies can be part of the same simulation, but they can also interact in a physically correct manner. No constraints between links were pre-defined, instead we relied on collision detection and stiff contact forces to handle the contacts. Using implicit integrator handling dynamically-created groups of interacting objects resulted in a stable simulation

SIERRA Project-Team

5. Software

5.1. SPAMS (SPArse Modeling Software)

Participants: Jean-Paul Chieze [correspondant], Guillaume Obozinski [correspondant].

SPAMS (SPArse Modeling Software) is an optimization toolbox for solving various sparse estimation problems: dictionary learning and matrix factorization, solving sparse decomposition problems, solving structured sparse decomposition problems. It is developed by Julien Mairal (former Willow PhD student, co-advised by F. Bach and J. Ponce), with the collaboration of Francis Bach (Inria), Jean Ponce (Ecole Normale Supérieure), Guillermo Sapiro (University of Minnesota), Rodolphe Jenatton (Inria) and Guillaume Obozinski (Inria). It is coded in C++ with a Matlab interface. Recently, interfaces for R and Python have been developed by Jean-Paul Chieze (Inria). Currently 650 downloads and between 1500 and 2000 page visits per month. See <http://spams-devel.gforge.inria.fr/>.

5.2. SiGMa - Simple Greedy Matching: a tool for aligning large knowledge-bases

Participant: Simon Lacoste-Julien [correspondant].

SiGMa - Simple Greedy Matching: a tool for aligning large knowledge-bases

Version 1. Webpage: <http://mlg.eng.cam.ac.uk/slacoste/sigma/>.

The tool SiGMa (Simple Greedy Matching) is a knowledge base alignment tool implemented in Python. It takes as input two knowledge bases, each represented as a list of triples of (entity, relationship, entity), in addition to a partial alignment between the relationships from one knowledge base to the other, and gives as output an ordered list of proposed entity matches between the two knowledge base (where the order corresponds heuristically to a notion of certainty about these matches). The matching decisions are made in a greedy fashion, combining information about the relationship graph as well as a pairwise similarity scores defined between the entities. The code handles various sources of information to be used for this score, such as a similarity defined on strings, dates, and other entity properties – and gives a few options to the user.

We also provide two large-scale knowledge base alignment benchmark datasets with tens of thousands of ground truth pairs: YAGO aligned to IMDb as well as Freebase aligned to IMDb.

Participants outside of Sierra: Konstantina Palla, Alex Davies, Zoubin Ghahramani (Machine Learning Group, Department of Engineering, University of Cambridge); Gjergji Kasneci, Thore Graepel (Microsoft Research Cambridge)

See <http://mlg.eng.cam.ac.uk/slacoste/sigma/>.

5.3. minFunc (2012 version)

Participant: Mark Schmidt [correspondant].

minFunc is a Matlab function for unconstrained optimization of differentiable real-valued multivariate functions using line-search methods. It uses an interface very similar to the Matlab Optimization Toolbox function `fminunc`, and can be called as a replacement for this function. On many problems, minFunc requires fewer function evaluations to converge than `fminunc` (or `minimize.m`). Further it can optimize problems with a much larger number of variables (`fminunc` is restricted to several thousand variables), and uses a line search that is robust to several common function pathologies.

The default parameters of `minFunc` call a quasi-Newton strategy, where limited-memory BFGS updates with Shanno-Phua scaling are used in computing the step direction, and a bracketing line-search for a point satisfying the strong Wolfe conditions is used to compute the step direction. In the line search, (safeguarded) cubic interpolation is used to generate trial values, and the method switches to an Armijo back-tracking line search on iterations where the objective function enters a region where the parameters do not produce a real valued output (i.e. complex, NaN, or Inf). See <http://www.di.ens.fr/~mschmidt/Software/minFunc.html>.

5.4. prettyPlot

Participant: Mark Schmidt [correspondant].

The `prettyPlot` function is a simple wrapper to Matlab's `plot` function for quickly making nicer-looking plots. Here are the features: Made the default line styles bigger, and the default fonts nicer. Options are passed as a structure, instead of through `plot`'s large number of different functions. You can pass in cell arrays to have lines of different lengths. You can pass an $n \times 3$ matrix of colors, and cell arrays of line-styles and/or markers. It will cycle through the given choices. All markers are placed on top of (all) lines, you do not have to put a marker on every data point, and you can use different spacing between markers for different lines. You can change only the upper or lower x-limit (y-limit), rather than having to specify both. There is some support for making nicer-looking error lines. See <http://www.di.ens.fr/~mschmidt/Software/prettyPlot.html>.

5.5. SegAnnot

Participant: Toby Hocking [correspondant].

`SegAnnot`: an R package for fast segmentation of annotated piecewise constant signals. Tech report and R package. Standard segmentation models for piecewise constant signals do not always agree with an expert's visual interpretation of the signal, as encoded using a set of annotations. This R package implements a dynamic programming algorithm which can be used to quickly find a segmentation model in agreement with expert annotations. Collaboration with Guillem Rigau (Inria - AgroParisTech). See <http://hal.inria.fr/hal-00759129> and <http://segannot.r-forge.r-project.org/>.

SIMPAF Project-Team

5. Software

5.1. ns2ddv-M

Participants: Caterina Calgaro [correspondant (Univ. Lille 1)], Emmanuel Creusé [correspondant (Univ. Lille 1)].

Incompressible Navier-Stokes, Variable Density, Rayleigh-Taylor Instability The NS2DDV-M code is based on a hybrid method coupling FV and FE approaches for solving the variable density Navier-Stokes equation in dimension 2. This original approach for variable density flows is described in [56]. The NS2DDV-M code will be available on the SIMPAF team web page before the end of 2011.

Here is the self-assessment of the team effort following the grid provided by Inria (see : <http://www.inria.fr/institut/organisation/instances/commission-d-evaluation>): A3, SO3-up4, SM2-up3, EM3, SDL4, DA1, CD4, MS4, TPM4.

Software web site : <http://math.univ-lille1.fr/~simpaf/SITE-NS2DDV/home.html>

5.2. ns2ddv-C++

Participants: Caterina Calgaro [correspondant (Univ. Lille 1)], Emmanuel Creusé [correspondant (Univ. Lille 1)], Thierry Goudon.

Incompressible Navier-Stokes, Variable Density, Kazhikhov-Smagulov model, Rayleigh-Taylor Instability, avalanches phenomena The NS2DVD-C++ code is based on a hybrid method coupling FV and FE approaches for solving the variable density Navier-Stokes equation in dimension 2. The code is developed around the GetFem++ and the Bamg softwares. It allows in particular mesh refinement strategies so that very relevant simulations can be reached (as the falling droplet with very high density ratios, see for example [54]). The current version of the code consider the additional terms in the Kazhikhov-Smagulov model.

Webpage : <http://math.univ-lille1.fr/~simpaf/SITE-NS2DDV>

Here is the self-assessment of the team effort following the grid provided by Inria (see : <http://www.inria.fr/institut/organisation/instances/commission-d-evaluation>): A1, SO3-up4, SM1, EM2, SDL1, DA1, CD4, MS4, TPM1.

5.3. RTcodes

Participants: Pauline Lafitte [correspondant (ECP)], Jean-François Coulombel [(CNRS & Univ. Nantes)], Christophe Besse [(Univ. Lille 1)], Thierry Goudon [(Inria)], Giovanni Samaey [(KU Leuven)].

Radiative Transfer, Radiative shocks, AP schemes

We have developed a set of numerical codes, written in Scilab, to compute the solutions of the system coupling the Euler equations to the radiation through energy exchanges, in the non equilibrium regime. This covers several situations in the hierarchy of asymptotic problems. The code treats the one-dimensional framework. In particular the code can be used to investigate radiative shocks profiles. The main advantage of our numerical codes is that they do not require any refinement near the singularities. The numerical tests show a very good agreement with the theoretical predictions. See reference [27].

Here is the self-assessment of the team effort following the grid provided by Inria (see : <http://www.inria.fr/institut/organisation/instances/commission-d-evaluation>): A2, SO3, SM2, EM1, SDL1.

5.4. FPcodes

Participants: Pauline Lafitte [correspondant (ECP)], Thierry Goudon [(Inria)], Benjamin Boutin [(Univ. Rennes)].

Fluid-Particles flows, Gravity driven flows, AP schemes

We have developed a numerical code, written in Scilab, to compute the solutions of the two-phase flows equations describing particles interacting with a fluid through friction forces. The code treats one-dimensional situation and is well adapted to describe gravity driven flows in either bubbling or flowing regimes. In particular, it can be used to describe the evolution of pollutants in the atmosphere. The numerical strategy, based on an asymptotic-based scheme, is described in details in [57].

Here is the self-assessment of the team effort following the grid provided by Inria (see : <http://www.inria.fr/institut/organisation/instances/commission-d-evaluation>): A2, SO3, SM2, EM1, SDL1.

5.5. CLAToolBox

Participant: Christophe Besse [correspondant].

Absorbant boundary conditions, Schrödinger equation

As a byproduct of the review paper [45], a user-friendly interface is offered ¹ to trial and compare various numerical methods to solve the 1D Schrödinger equation with absorbant boundary conditions. We also mention [50] for a numerical investigation of blow-up phenomena in the nonlinear Schrödinger equation.

5.6. SPARCS

Participant: Christophe Besse.

Vlasov-Poisson system, Euler-Poisson system. Back-Trajectory method

SPARCS is the code developed by Thales Alenia Space for the simulation of the charge phenomena the spacecrafts are subject to. The current version of the code, according to the PhD thesis of O. Chanrion and M. Chane-Yook performed in collaboration with the team Caiman at Sophia Antipolis, is specialized to geostationary atmospheres. The model consists in the stationary Vlasov-Poisson system, but where instationary effects are taken into account with the boundary condition for the electric field. We participate, in particular through the post doc of N. Vauchelet, to the elaboration of an improved version of the code which includes parallization optimized procedures, the modelling of the natural difference of potential between different dielectric surfaces of the spacecraft, as well as the possible presence of devices emitting charged particles.

5.7. Code-Carmel3D

Participant: Emmanuel Creusé [correspondant (Univ. Lille 1)].

This numerical code, developed in collaboration between EDF R&D and Lille 1 University, is devoted to the electromagnetic fields computation by the use of finite element methods. This code allows in particular to perform nondestructive control by the use of Foucault currents in steam generator pipes, and should be soon coupled with the thermal simulation of Code-Aster. Code-Carmel3D uses the Salomé platform (mesh-generator and post-processing) and Open Turns (uncertainties computation). It will consequently allow to solve multi-physics problems, both for the temporal and harmonic formulations.

¹<http://math.univ-lille1.fr/~besse/site/recherche/logiciels/index.html>

SIROCCO Project-Team

5. Software

5.1. Oriented wavelet based image codec

Participant: Christine Guillemot [contact person].

This still image codec is based on oriented wavelet transforms developed in the team. The transform is based on wavelet lifting locally oriented according to multiresolution image geometry information. The lifting steps of a 1D wavelet are applied along a discrete set of local orientations defined on a quincunx sampling grid. To maximize energy compaction, the orientation minimizing the prediction error is chosen adaptively. This image codec outperforms JPEG-2000 for lossy compression. This software has been registered at the APP (Agence de Protection des Programmes) under the number IDDN.FR.001.260024.000.S.P.2008.000.21000.

5.2. M3DPlayer: 3D video player

Participant: Laurent Guillo [contact person].

A 3D player - named M3DPlayer - supporting rendering of a 3D scene and navigation within the scene has been developed. It integrates as a plug-in the 3D model-based video codec of the team. From a video sequence of a static scene viewed by a monocular moving camera, the 3D model-based video codec allows the automatic construction of a representation of a video sequence as a stream of textured 3D models. 3D models are extracted using stereovision and dense matching maps estimation techniques. A virtual sequence is reconstructed by projecting the textured 3D models on image planes. This representation enables 3D functionalities such as synthetic objects insertion, lightning modification, stereoscopic visualization or interactive navigation. The codec allows compression at very low bit-rates (16 to 256 kb/s in 25Hz CIF format) with a satisfactory visual quality. It also supports scalable coding of both geometry and texture information. The first version of the software was registered at the Agency for the Protection of Programmes (APP) under the number IDDN.FR.001.130017.000S.P.2003.000.41200.

A second version of the player has been registered at the APP (Agence de Protection des Programmes) under the number IDDN.FR.001.090023.000.S.P.2008.000.21000. In 2009-2010, we focused on improving the rendering engine, based on recent OpenGL extensions, to be able to render the viewed scenes on an auto-stereoscopic display with low-end graphic cards. In our case, auto-stereoscopic display requires the rendering of eight 1920x1200 frames instead of just one for a standard display. This player is also used to render LDI (Layered Depth Images) and LDV (Layered Depth Videos) and to visualize 3D scenes on autostereoscopic displays taking multiple input views rendered from the LDI representation.

5.3. Depth maps extractor in mono-view (M3dAnalyzer2)

Participant: Laurent Guillo [contact person].

This software estimates depth maps from a video captured by a unique camera moving in a static 3D environment with Lambertian surfaces. These sequences are of interest to specialized applications such as augmented reality, remote-controlled robots operating in hazardous environments or remote exploration by drones. This software has been filed at the APP (Agence de Protection des Programmes) under the number IDDN.FR.001.110031.000.S.P.2010.000.31235.

5.4. Depth maps extractor in multi-view (MV2MVD)

Participant: Laurent Guillo [contact person].

This software estimates depth maps from multi-view videos, to provide Multi-View plus Depth (MVD) videos. MVD videos can be used to synthesize virtual views of the scene, or to render a different number of views than captured in the original video, for instance on an auto-stereoscopic display. This software produces depth maps of higher quality than those generated by the Depth Estimation Reference Software from the MPEG-3DV group, in terms of virtual views synthesis quality. This software has been filed at the APP (Agence de Protection des Programmes) under the number IDDN.FR.001.110034.000.S.P.2010.000.31235.

5.5. JPF-Joint Projection Filling

Participant: Fabien Racapé [contact person].

In the context of multi-view videos, this software generates virtual views of the scene from any viewpoint using a proposed method named Joint Projection Filling (JPF). The latter belongs to Depth-Image-Based Rendering (DIBR) methods, relying on warping equations, which project a reference view onto a virtual viewpoint. Each input view is defined by a "color" (or "texture") map and a "depth" map, which associates a depth value to each image pixel. The JPF method performs forward projection on depth map, using connectivity information to fill in disocclusions in a single step. Depth-based inpainting can then be used to fill in color disocclusions.

5.6. LDI builder

Participant: Fabien Racapé [contact person].

This software constructs a Layered Depth Image (LDI) representation of un-rectified Multi-View + Depth (MVD) sequences. The Incremental construction scheme reduces inter-layer correlation. The generated I-LDI is compatible with the M3DPlayer, permitting 3D visualisation and free viewpoint rendering of the 3D scene. The software also implements a virtual-view rendering technique which significantly reduces ghosting artefacts by eliminating untrusted texture boundaries detected in depth maps, as well as cracking artefacts thanks to an epipolar geometry aided inpainting method.

5.7. Visual Fixation Analysis

Participant: Olivier Le Meur [contact person].

From a set of fixation data and a picture, the software called Visual Fixation Analysis extracts from the input data a number of features (fixation duration, saccade length, orientation of saccade...) and computes an human saliency map. The software can also be used to assess the degree of similarity between a ground truth (eye fixation data) and a predicted saliency map. This software is dedicated to people working in cognitive science and computer vision. This software has been registered at the APP (Agence de Protection des Programmes).

5.8. ADT PICOVIN-P

Participants: Laurent Guillo [contact person], Thomas Guionnet.

The ADT Picovin-P is a technological development action, which works closely with the project-team SIROCCO. This development structure is the follow-up of the ADT Picovin. It gives its support to the project-team to integrate new and relevant algorithms into the state-of-the-art video codec and to take part in standardization.

During this year, the ADT first pursued its developments on Intra prediction in the context of the standardization initiative referred to as High Efficiency Video Coding (HEVC) and led by the Joint ITU/MPEG Collaborative Team on Video Coding (JCT-VC). HEVC is implemented as a test model, the HEVC test Model (HM) in which the ADT tools have been integrated. We then followed the standardization activities within the Joint Collaborative Team on 3D Video Coding Extension (JCT-3V). JCT-VC and JCT-3V have been both created by the ITU-T Study Group 16 (VCEG) and the ISO/IEC JTC 1/SC 29/WG 11 (MPEG). While JCT-VC aims at developing a new generation 2D video coding standard, JCT-3V aims at developing 3D extensions for video codecs, which are AVC (ATM) or HEVC (HTM) based.

As part of JCT-3V, we submitted several proposals related to the handling of the merge list of predictor candidates. They were about the re-ordering of the candidates in the list and about the addition of new candidates. Two of them have been accepted in the dedicated core experiment (CE) during the 1st JCT-3V meeting which was held in Stockholm in July 2012. An improved version related to the addition of candidates has been accepted in a CE during the 2nd JCT-3V meeting in Shanghai in October 2012. It will be integrated in the coming test model (HTM) and evaluated during the next meeting in Geneva in January 2013. During 2012, the ADT also took part in cross checks which aims at evaluating and testing tools studied in core experiments. As part of cross checks related to JCT-VC or JCT-3V the ADT ran 9 tests jointly with companies such as Canon, Huawei, HiSilicon and Nokia. The ADT Picovin-P started in October 2011 and lasted one year. During this year, one permanent engineer from the SED Rennes (development and experimentation department of Inria Rennes) and one senior engineer specialized in video compression are involved in the ADT. It is supported by the technological development department of Inria.

SISYPHE Project-Team

4. Software

4.1. The Matlab System Identification ToolBox (SITB)

Participant: Qinghua Zhang.

This development is made in collaboration with Lennart Ljung (Linköping University, Sweden), Anatoli Juditsky (Joseph Fourier University, France) and Peter Lindskog (NIRA Dynamics, Sweden).

The System Identification ToolBox (SITB) is one of the main Matlab toolboxes commercialized by The Mathworks. Inria participates in the development of its extension to the identification of nonlinear systems which is released since 2007. It includes algorithms for both black box and grey box identification of nonlinear dynamic systems. Inria is mainly responsible for the development of black box identification, with nonlinear autoregressive (NLARX) models and block-oriented (Hammerstein-Wiener) models.

4.2. Inverse Scattering for Transmission Lines (ISTL)

Participants: Michel Sorine, Qinghua Zhang.

ISTL is a software for numerical computation of the inverse scattering transform for electrical transmission lines. In addition to the inverse scattering transform, it includes a numerical simulator generating the reflection coefficients of user-specified transmission lines. With the aid of a graphical interface, the user can interactively define the distributed characteristics of a transmission line. This software is mainly for the purpose of demonstrating a numerical solution to the inverse problem of non uniform transmission lines. Its current version is limited to the case of lossless transmission lines. It is registered at Agence pour la Protection des Programmes (APP) under the number IDDN.FR.001.120003.000.S.P.2010.000.30705.

4.3. CGAO: Contrôle Glycémique Assisté par Ordinateur

Participants: Alexandre Guerrini, Michel Sorine.

This development is made in collaboration with Pierre Kalfon (Chartres Hospital) and the small business enterprise LK2.

This software CGAO developed with LK2 and Hospital Louis Pasteur (Chartres) provides efficient monitoring and control tools that will help physicians and nursing staff to avoid hyperglycaemia and hypoglycaemia episodes in Intensive Care Units. It has been used in a large clinical study, **CGAO-REA**. More than 3500 patients have been included in CGAO-REA. Commercialization is done by LK2. CGAO has been sold to the company Fresenius Kabi.

4.4. DYNPEAK: a user-friendly interface for the analysis of LH (Luteinizing Hormone) secretion rhythms

Participants: Frédérique Clément, Arnaud Ferré, Vincent Ladevèze, Claire Médigue, Serge Steer, Mouhamadou-Bachir Syll, Alexandre Vidal, Qinghua Zhang.

DYNPEAK is a Webresource interface dedicated to the analysis of the pulsatile rhythm of secretion of the pituitary hormone LH, that aims at providing the final users (experimentalists and clinicians) with a simple-to-use version of the algorithm developed in [54]. DYNPEAK is developed within the PALOMA project devoted to the development of a prototype for a collaborative platform in Biomathematics.

4.5. LARY_CR: Software package for the Analysis of Cardio Vascular and Respiratory Rhythms

Participants: Claire Médigue, Serge Steer.

LARY_CR is a software package dedicated to the study of cardiovascular and respiratory rhythms [97]. It presents signal processing methods, from events detection on raw signals to the variability analysis of the resulting time series. The events detection concerns the heart beat recognition on the electrocardiogram, defining the RR time series, the maxima and minima on the arterial blood pressure defining the systolic and diastolic time series. These detections are followed by the resampling of the time series then their analyse. This analyse uses temporal and time frequency methods: Fourier Transform, spectral gain between the cardiac and blood pressure series, Smooth Pseudo Wigner_Ville Distribution, Complex DeModulation, temporal method of the cardiovascular Sequences. The objective of this software is to provide some tools for studying the autonomic nervous system, acting in particular in the baroreflex loop; its functioning is reflected by the cardiovascular variabilities and their relationships with the other physiological signals, especially the respiratory activity. Today LARY_CR is used internally, in the framework of our clinical collaborations, and the functions devoted to the rhythms analysis are now freely available through the Scilab Cardiovascular toolbox, as an atom module.

SMIS Project-Team

5. Software

5.1. Introduction

In our research domain, developing software prototypes is mandatory to validate research solutions and is an important vector for publications, demonstrations at conferences and exhibitions as well as for cooperations with industry. This prototyping task is however difficult because it requires specialized hardware platforms (e.g., new generations of smart tokens), themselves sometimes at an early stage of development.

For a decade, we have developed successive prototypes addressing different application domains, introducing different technical challenges and relying on different hardware platforms. PicoDBMS was our first attempt to design a full-fledged DBMS embedded in a smart card [42] [32]. Chip-Secured Data Access (C-SDA) embedded a reduced SQL query engine and access right controller in a secure chip and acted as an incorruptible mediator between a client and an untrusted server hosting encrypted data [37]. Chip-Secured XML Access (C-SXA) was an XML-based access rights controller embedded in a smart card [38]. Prototypes of C-SXA have been the recipient of the e-gate open 2004 Silver Award and SIMagine 2005 Gold award, two renowned international software contests. The next subsections details the two prototypes we are focusing on today.

5.2. PlugDB engine

Participants: Nicolas Anciaux [correspondent], Luc Bouganim, Philippe Pucheral, Shaoyi Yin, Yanli Guo, Lionel Le Folgoc, Alexei Troussov.

More than a stand-alone prototype, PlugDB is part of a complete architecture dedicated to a secure and ubiquitous management of personal data. PlugDB aims at providing an alternative to a systematic centralization of personal data. To meet this objective, the PlugDB architecture lies on a new kind of hardware device called Secure Portable Token (SPT). Roughly speaking, a SPT combines a secure microcontroller (similar to a smart card chip) with a large external Flash memory (Gigabyte sized). The SPT can host data on Flash (e.g., a personal folder) and safely run code embedded in the secure microcontroller. PlugDB engine is the cornerstone of this embedded code. PlugDB engine manages the database on Flash (tackling the peculiarities of NAND Flash storage), enforces the access control policy defined on this database, protects the data at rest against piracy and tampering, executes queries (tackling low RAM constraint) and ensures transaction atomicity. Part of the on-board data can be replicated on a server (then synchronized) and shared among a restricted circle of trusted parties through crypto-protected interactions. PlugDB engine has been registered at APP (Agence de Protection des Programmes) in 2009 [33] and a new version is registered each year. The underlying Flash-based indexing system has also been patented by Inria and Gemalto [43]. It has been demonstrated in a dozen of national and international events including JavaOne and SIGMOD. It is being experimented in the field to implement a secure and portable medical-social folder helping the coordination of medical care and social services provided at home to dependent people. In 2012, we have ported PlugDB-engine on a new hardware platform to 1) become completely independent from Gemalto, 2) have a plug-and-play implementation on Android, 3) serve as a basement to port it on other custom hardware implementation. We have already discussed with hardware companies located in “île de France” to produce new hardware tokens to host future versions of PlugDB-engine. Link: http://www-smis.inria.fr/_DMSP/home.php.

5.3. uFLIP Benchmark

Participants: Luc Bouganim [correspondent], Philippe Bonnet, Bjorn Jónsson, Lionel Le Folgoc.

It is amazingly easy to produce meaningless results when measuring flash devices, partly because of the peculiarity of flash memory, but primarily because their behavior is determined by layers of complex, proprietary, and undocumented software and hardware. uFLIP is a component benchmark for measuring the response time distribution of flash IO patterns, defined as the distribution of IOs in space and time. uFLIP includes a benchmarking methodology which takes into account the particular characteristics of flash devices. The source code of uFLIP, available on the web (700 downloads, 4000 distinct visitors), was registered at APP in 2009 [35]. It has been demonstrated at SIGMOD.

Link: <http://www.uflip.org>.

SOCRATE Team

5. Software

5.1. WSnet

Socrate is an active contributor to WSnet (<http://wsnet.gforge.inria.fr/>) a multi-hop wireless network discrete event simulator. WSnet was created in the ARES team and it is now supported by the D-NET team of Inria Rhône-Alpes.

5.2. Wiplan

Wiplan is a software including an Indoor propagation engine and a wireless LAN optimization suite, which has been registered by INSA-Lyon. The heart of this software is the propagation simulation core relying on an original method, MR-FDPF (multi-resolution frequency domain ParFlow). The discrete ParFlow equations are translated in the Fourier domain providing a wide linear system, solved in two steps taking advantage of a multi-resolution approach. The first step computes a cell-based tree structure referred to as the pyramid. In the second phase, a radiating source is simulated, taking advantage of the pre-processed pyramidal structure. Using of a full-space discrete simulator instead of classical ray-tracing techniques is a challenge due to the inherent high computation requests. However, we have shown that the use of a multi-resolution approach allows the main computation load to be restricted to a pre-processing phase. Extensive works have been done to make predictions more realistic. The network planning and optimization suite is based on a multi-criteria model relying on a Tabu solver. The development of the wiplan software is a part of the european project iPlan (IAPP-FP7 project). See also the web page <http://wiplan.citi.insa-lyon.fr>.

STARS Team

5. Software

5.1. SUP

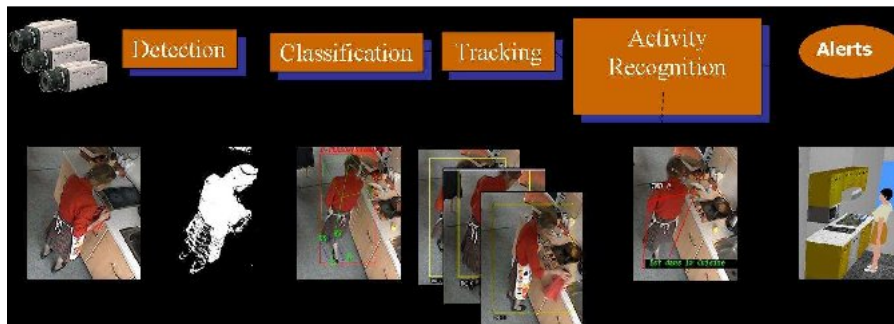


Figure 5. Tasks of the Scene Understanding Platform (SUP).

SUP is a Scene Understanding Software Platform written in C and C++ (see Figure 5). *SUP* is the continuation of the *VSIP* platform. *SUP* is splitting the workflow of a video processing into several modules, such as acquisition, segmentation, etc., up to activity recognition, to achieve the tasks (detection, classification, etc.) the platform supplies. Each module has a specific interface, and different plugins implementing these interfaces can be used for each step of the video processing. This generic architecture is designed to facilitate:

1. integration of new algorithms in *SUP*;
2. sharing of the algorithms among the Stars team.

Currently, 15 plugins are available, covering the whole processing chain. Several plugins are using the Genius platform, an industrial platform based on *VSIP* and exploited by Keeneo.

Goals of *SUP* are twofold:

1. From a video understanding point of view, to allow the Stars researchers sharing the implementation of their work through this platform.
2. From a software engineering point of view, to integrate the results of the dynamic management of vision applications when applied to video analytics.

5.2. ViSEval

ViSEval is a software dedicated to the evaluation and visualization of video processing algorithm outputs. The evaluation of video processing algorithm results is an important step in video analysis research. In video processing, we identify 4 different tasks to evaluate: detection, classification and tracking of physical objects of interest and event recognition.

The proposed evaluation tool (*ViSEval*, visualization and evaluation) respects three important properties:

- To be able to visualize the algorithm results.
- To be able to visualize the metrics and evaluation results.
- For users to easily modify or add new metrics.

The ViSEvAI tool is composed of two parts: a GUI to visualize results of the video processing algorithms and metrics results, and an evaluation program to evaluate automatically algorithm outputs on large amount of data. An XML format is defined for the different input files (detected objects from one or several cameras, ground-truth and events). XSD files and associated classes are used to check, read and write automatically the different XML files. The design of the software is based on a system of interfaces-plugins. This architecture allows the user to develop specific treatments according to her/his application (e.g. metrics). There are 6 interfaces:

1. The video interface defines the way to load the images in the interface. For instance the user can develop her/his plugin based on her/his own video format. The tool is delivered with a plugin to load JPEG image, and ASF video.
2. The object filter selects which objects (e.g. objects far from the camera) are processed for the evaluation. The tool is delivered with 3 filters.
3. The distance interface defines how the detected objects match the ground-truth objects based on their bounding box. The tool is delivered with 3 plugins comparing 2D bounding boxes and 3 plugins comparing 3D bounding boxes.
4. The frame metric interface implements metrics (e.g. detection metric, classification metric, ...) which can be computed on each frame of the video. The tool is delivered with 5 frame metrics.
5. The temporal metric interface implements metrics (e.g. tracking metric,...) which are computed on the whole video sequence. The tool is delivered with 3 temporal metrics.
6. The event metric interface implements metrics to evaluate the recognized events. The tool provides 4 metrics.

The GUI is composed of 3 different parts:

1. The widows dedicated to result visualization (see Figure 6):
 - Window 1: the video window displays the current image and information about the detected and ground-truth objects (bounding-boxes, identifier, type,...).
 - Window 2: the 3D virtual scene displays a 3D view of the scene (3D avatars for the detected and ground-truth objects, context, ...).
 - Window 3: the temporal information about the detected and ground truth objects, and about the recognized and ground-truth events.
 - Window 4: the description part gives detailed information about the objects and the events,
 - Window 5: the metric part shows the evaluation results of the frame metrics.
2. The object window enables the user to choose the object to be displayed (see Figure 7).
3. The multi-view window displays the different points of view of the scene (see Figure 8).

The evaluation program saves, in a text file, the evaluation results of all the metrics for each frame (whenever it is appropriate), globally for all video sequences or for each object of the ground truth.

The ViSEvAI software was tested and validated into the context of the Cofriend project through its partners (Akka,...). The tool is also used by IMRA, Nice hospital, Institute for Infocomm Research (Singapore),... The software version 1.0 was delivered to APP (French Program Protection Agency) on August 2010. ViSEvAI is under GNU Affero General Public License AGPL (<http://www.gnu.org/licenses/>) since July 2011. The tool is available on the web page : http://www-sop.inria.fr/teams/pulsar/EvaluationTool/ViSEvAI_Description.html

5.3. Clem

The *Clem Toolkit* [63](see Figure 9) is a set of tools devoted to design, simulate, verify and generate code for LE [19] [77] programs. LE is a synchronous language supporting a modular compilation. It also supports automata possibly designed with a dedicated graphical editor.

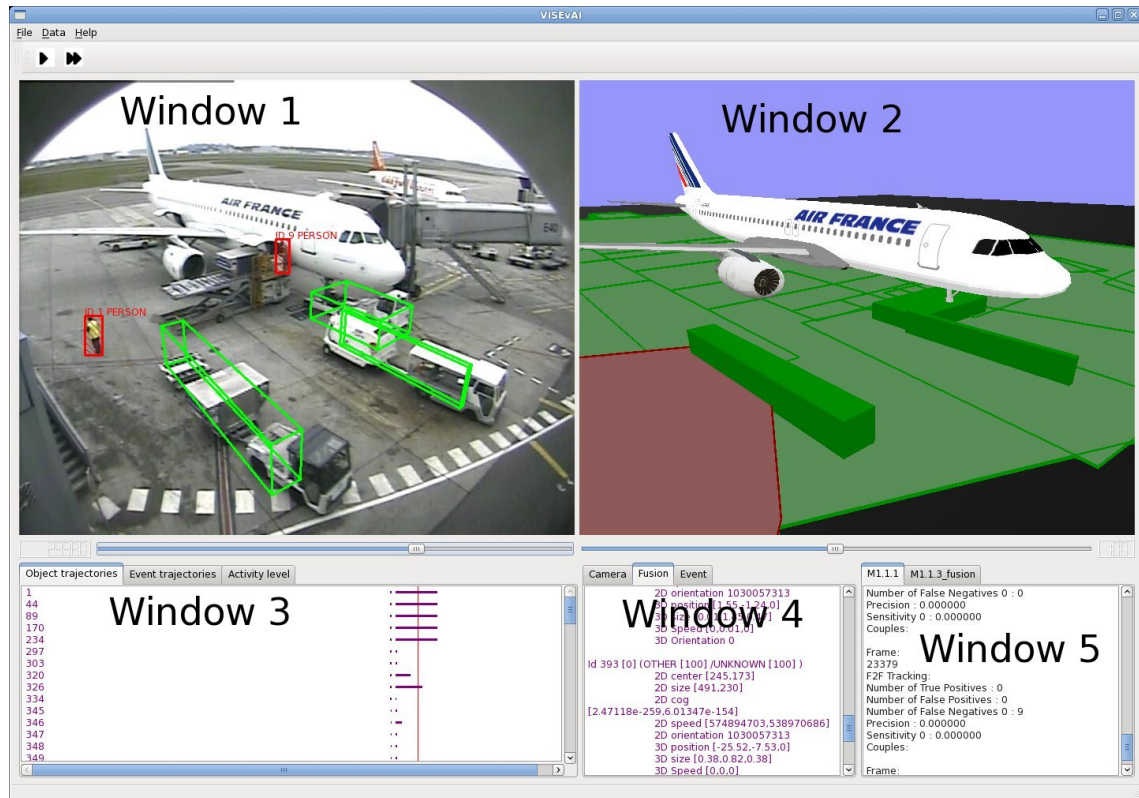


Figure 6. GUI of the ViSEvAI software

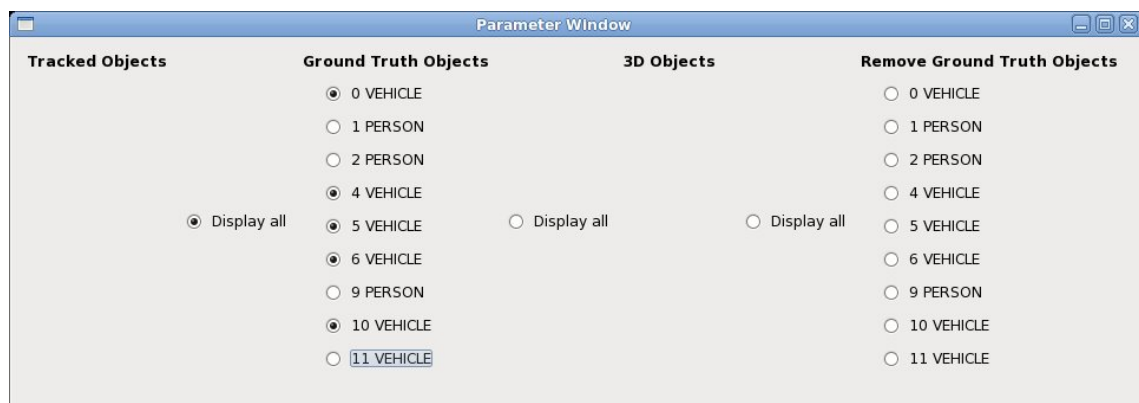


Figure 7. The object window enables users to choose the object to display

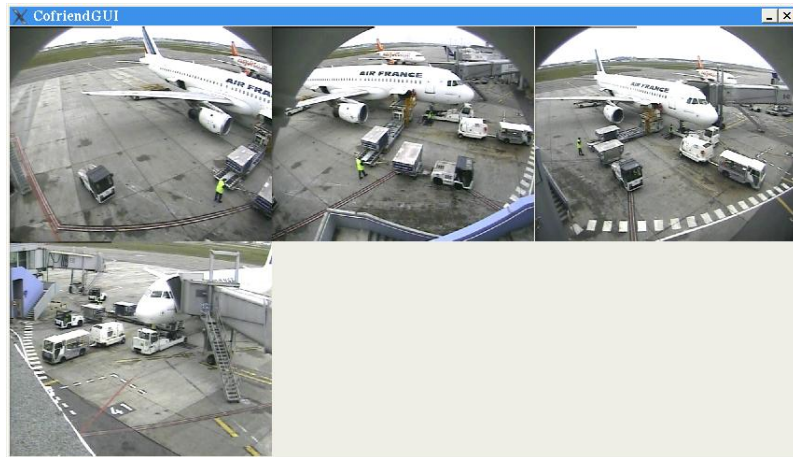


Figure 8. The multi-view window

Each LE program is compiled later into lec and lea files. Then when we want to generate code for different backends, depending on their nature, we can either expand the lec code of programs in order to resolve all abstracted variables and get a single lec file, or we can keep the set of lec files where all the variables of the main program are defined. Then, the *finalization* will simplify the final equations and code is generated for simulation, safety proofs, hardware description or software code. Hardware description (Vhdl) and software code (C) are supplied for LE programs as well as simulation. Moreover, we also generate files to feed the NuSMV model checker [61] in order to perform validation of program behaviors.

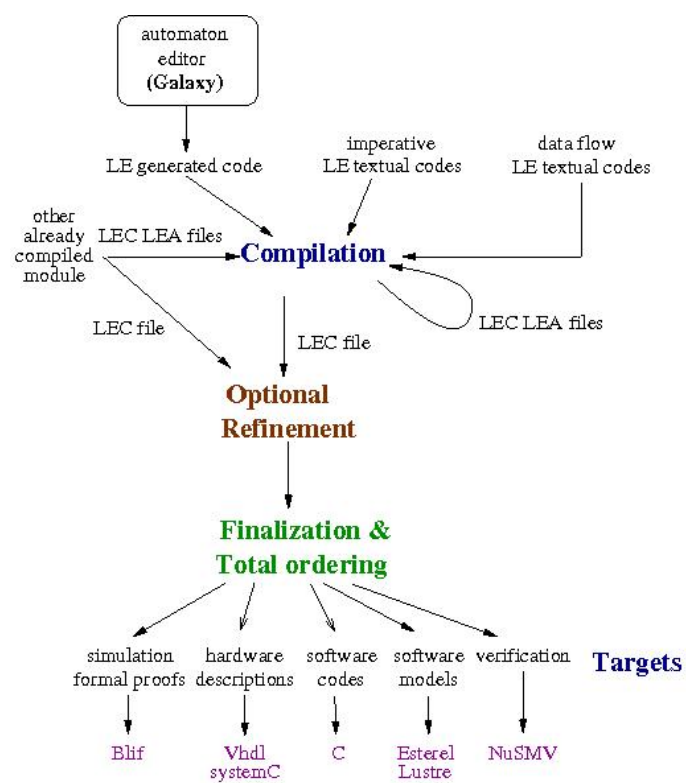


Figure 9. The Clem Toolkit

STEPP Exploratory Action

5. Software

5.1. TEOS: Trans Exploration and Optimization Software

Participants: Anthony Tschirhard, Mathieu Vadon, Elise Arnaud, Emmanuel Prados.

The TEOS software offers a set of tools to help the calibration of the land use and transport integrated model TRANUS. It uses some exploration and optimization procedures of the relevant parameters.

TAO Project-Team

5. Software

5.1. Metis

Participants: Olivier Teytaud [correspondent], Adrien Couëtoux, Jérémie Decock, Jean-Joseph Christophe.

Energy, Optimization, Planning

Many works in Energy Optimization, in particular in the case of high-scale sequential decision making, are based on one software per application, because optimizing the software eventually implies losing generality. Our goal is to develop with Artelys a platform, Metis, which can be used for several applications. In 2012 we interfaced existing codes in Artelys and codes developed in the Tao team; experiments have been performed and test cases have been designed. A main further work is the introduction of generic tools for stochastic dynamic programming into the platform, for comparison and hybridization with other tools from the UCT-SIG.

Our favorite challenge is the hybridization of “classical” tools (based on constraint satisfaction problems, or mixed integer linear programming or mixed integer quadratic programming), which are fast and accurate, with non-linear solvers which can take care of a sophisticated (non-linear) model.

5.2. MoGo

Participants: Olivier Teytaud [correspondent], Hassen Doghmen, Jean-Baptiste Hoock.

Go, Multi-armed bandit

MoGo and its Franco-Taiwanese counterpart MoGoTW is a Monte-Carlo Tree Search program for the game of Go, which made several milestones of computer-Go in the past (first wins against professional players in 19x19; first win with disadvantageous side in 9x9 Go). Recent results include 7 wins out of 12 against professional players (in Brisbane, 2012). However, the work in the UCT-SIG has now shifted to energy management.

5.3. CMA-ES: Covariance Matrix Adaptation Evolution Strategy

Participant: Nikolaus Hansen [correspondent].

Evolutionary Computation, stochastic optimization, real-parameter optimization

The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) is one of the most powerful continuous domain evolutionary algorithms. The CMA-ES is considered state-of-the-art in continuous domain evolutionary computation³ and has been shown to be highly competitive on different problem classes. The algorithm is widely used in research and industry as witnessed by hundreds of published applications. We provide source code for the CMA-ES in C, Java, Matlab, Octave, Python, and Scilab including the latest variants of the algorithm.

Links: http://www.lri.fr/~hansen/cmaes_inmatlab.html

5.4. COmparing Continuous Optimizers

Participants: Nikolaus Hansen [correspondent], Anne Auger, Marc Schoenauer.

Evolutionary Computation, stochastic optimization, real-parameter optimization, benchmarking, derivative free optimization

³H.-G. Beyer (2007). Evolution Strategies, *Scholarpedia*, page 1965.

COCO (COMparing Continuous Optimizers) is a platform for systematic and sound comparisons of real-parameter global optimizers. COCO provides benchmark function testbeds (noiseless and noisy) and tools for processing and visualizing data generated by one or several optimizers. The code for processing experiments is provided in Matlab and C. The post-processing code is provided in Python. The code is under continuous development and has been used for the GECCO 2009, 2010 and 2012 workshops on “Black Box Optimization Benchmarking” (BBOB) (see Section 3.3), and serves as a basis for the test platform in the CSDL project.

Link: <http://coco.gforge.inria.fr/doku.php>

5.5. MultiBoost

Participants: Balázs Kégl [correspondent], Djalel Benbouzid.

multi-class, multi-label classification

The MultiBoost package [4] provides a fast C++ implementation of multi-class/multi-label/multi-task boosting algorithms. It is based on ADABOOST.MH but it also implements popular cascade classifiers, ARC-GV, and FILTERBOOST. The package contains common multi-class base learners (stumps, trees, products, Haar filters). Further base learners and strong learners following the boosting paradigm can be easily implemented in a flexible framework.

Link: <http://multiboost.org>

5.6. Grid Observatory

Participants: Cécile Germain [correspondent], Julien Nauroy, Michèle Sebag.

Autonomic Computing, Green Computing The Grid Observatory software suite collects and publishes traces of the EGI (European Grid Initiative) grid usage. With the release and extensions of its portal, the Grid Observatory has made a database of grid usage traces available to the wider computer science community. These data are stored on the grid, and made accessible through a web portal without the need of grid credentials. More than 140 users are currently registered. The GO is supported by an Inria ADT (Action de Développement Technologique).

The Green Computing Observatory (GCO), part of the GO initiative monitors a large computing center (Laboratoire de l’Accélérateur Linéaire - LAL) within the EGI grid, and publishes the data through the Grid Observatory. A tool has been made available to provide interactive visualization of the site activity (load, power usage and more) at the global and individual level. The GCO is supported by the CNRS PEPS program, and by University Paris-Sud through the MRM (Moyens de Recherche Mutualisés) program. The first barrier to improved energy efficiency is the the lack of overall data collection on the energy consumption of individual components of data centers. The Green Computing Observatory collects monitoring data on energy consumption of a large computing center, and publishes them through the Grid Observatory portal. These data include the detailed monitoring of the processors and motherboards, as well as global site information, such as overall consumption and external temperature, as global optimization is a promising way of research. A second barrier is making the collected data usable. The difficulty is to make the data readily consistent and complete, as well as understandable for further exploitation. For this purpose, the GCO opts for an ontological approach in order to rigorously define the semantics of the data (what is measured) and the context of their production (how are they acquired and/or calculated). The first achievements of these developments have been described in [40], and presented at the ICT-COST meeting and GreenDays@Lyon.

Link: <http://grid-observatory.org>

TASC Project-Team

5. Software

5.1. CHOCO

Participants: Nicolas Beldiceanu, Alexis De Clerq, Sophie Demasse, Jean-Guillaume Fages, Narendra Jussien [correspondant], Arnaud Letort, Xavier Lorca [correspondant], Thierry Petit, Charles Prud'homme [correspondant], Remi Douence.

CHOCO is a Java discrete constraints library integrating within a same system *explanations*, *soft constraints* and *global constraints* (90000 lines of source code). This year developments were focussing on the following aspects:

1. Since September 2011, we are working on a new version of the **CHOCO** solver. This implies a total refactoring of the source code in order to make it simpler to use and maintain. We introduce a new propagation engine framework that directly handle state-of-the-art techniques, such as *advisors*, *propagator groups*, *activity-based search* and *explanations*, to ensure a good level of efficiency, and plug a **MiniZinc** modeling language parser. An alpha release will be available by the beginning of 2013.
2. In the context of the new version of the **CHOCO** solver we design an *adaptive propagation engine* to enhance performance as well as a *solver independent language to write strategies* for controlling the new adaptive propagation engine. The adaptive propagation engine can both deal with variable-oriented propagation engines and constraint-oriented propagation engines. It is usually accepted that there is no best approach in general and modern constraint solvers therefore implement only one.
3. New scalable global constraints were provides both in the context of *graph constraints* (with also graph variables) and in the context of *scheduling constraints*. These constraints respectively allow to handle sparse graphs with up to 10000 vertices, and resource scheduling problems with up to one million tasks.
4. A new global constraint called *focus* for concentrating high cost values motivated by several concrete examples, such as resource constrained scheduling problems with machine rentals, was introduced.
5. The work on providing probability-based constraints to get light propagation filtering algorithm has been pursued. A particular focus has been put on calculating the probabilistic indicator for the bound-consistency propagator of an *alldifferent* constraint.

N. Beldiceanu, A. De Clerq, S. Demasse, J.-G. Fages, N. Jussien, A. Letort, X. Lorca, T. Petit, C. Prud'Homme and R. Douence have contributed in 2012. The link to the system and documentation is <http://choco.emn.fr>.

5.2. IBEX

Participants: Ignacio Araya, Anthony Baire, Gilles Chabert [correspondant], Rémi Douence, Bertrand Neveu, Gilles Trombettoni.

IBEX (Interval-Based EXplorer) is a C++ library for solving nonlinear constraints over real numbers (25000 lines of source code). The main feature of Ibex is its ability to build solver/paver strategies declaratively through the contractor programming paradigm.

Continuing last year work on the redesign of the architecture of **IBEX**, the **IBEX** library has been entirely refactored from scratch to provide a more clean and easy-to-use interface as well as a more powerful engine. The development started in late 2011, the kernel has been completed in mid-2012 and almost all the functionalities of **IBEX** integrated in the new architecture. Global optimization and system solving front-end algorithms have been tested on more than 500 benchmarks. Installation scripts for a deployment on multiple platforms have also been done by an engineer of Inria (Anthony Baire). A first web site has been activated, with an on-line installation documentation, a programming tutorial (still under writing), and an API. An alpha release is now available for download. A first **training course** on **IBEX 2.0** has been organized at **ENSTA Bretagne**, Brest, the 17-18th December with about 25 participants. Similar training courses will also occur in 2013.

An explorative study aimed at showing that the explicit representation of search trees can play a distinguished role in the field of numerical constraints was done this year. The idea was also to define a new high-level language to handle explicit search trees, in the fashion of quadrees (that one can intersects, etc.). We have developed a prototype in Haskell to validate the approach and have illustrated it over different examples.

5.3. CHOCO-IBEX

Participants: Gilles Chabert [correspondant], Charles Prud'homme [correspondant].

Work has been done to provide an interface for connecting the **CHOCO** and the **IBEX** libraries in order to handle problems where we both have continuous and discrete variables. This interface allows to filter continuous domains from **CHOCO** with the **IBEX** engine as well as to check for unsatisfiability or entailment. It also manages reification variables. This interface has been tested on a toy problem and seems to work as expected. Some glue code (on both sides) is still missing to handle reification and should be integrated in a short term. The interface should be ready for usage with the next version of CHOCO (3.0).

5.4. Global Constraint Catalog

Participants: Nicolas Beldiceanu [correspondant], Mats Carlsson, Helmut Simonis.

The global constraint catalog presents and classifies global constraints and describes different aspects with meta data. It consist of

1. a **pdf** version that can be downloaded from <http://www.emn.fr/z-info/sdemasse/gccat/> (at item *last working version*) containing 406 constraints, 3397 pages and 758 figures,
2. an on line version accessible from the previous address,
3. meta data describing the constraints (buton *PL* for each constraint, e.g., [alldifferent.pl](#)),
4. an online service (i.e, a *constraint seeker*) which provides a web interface to search for global constraints, given positive and negative ground examples.

This year developments were focussing on:

1. maintaining the catalogue,
2. making the *core global constraints* (10 constraints) more accessible to a wider audience:
 - for this purpose examples with their corresponding pictures have been systematically provided for showing all solutions for an example of each core global constraint.
 - in addition a set of about 30 exercises with their corrections have been done for half of the core global constraints.
3. a redesign of all the 758 figures of the catalog has been undertaken in autumn 2012 using **TikZ** (in December 2012 312 figures were redesigned).
4. adding constraints related to sequences that we found relevant for learning constraints from electricity production curves.

N. Beldiceanu, **M. Carlsson** (SICS, Sweden) and **H. Simonis** (4C, Ireland) have contributed in 2012. The link to the global constraint catalog is <http://www.emn.fr/z-info/sdemasse/gccat/>.

TEXMEX Project-Team

5. Software

5.1. Software

When applicable, we provide the IDDN is the official number, which is obtained when registering the software at the APP (Agence de Protection des Programmes).

5.1.1. *New Software*

5.1.1.1. *Aabot*

Participant: Jonathan Delhumeau.

AABOT is a tool to facilitate annotation of large video databases. Its primary design focus has been for the annotation on commercials in two 6-month long TV databases. The software keeps a database of already annotated commercials and suggests when it finds a new probable instance. It also validates user annotations by suggesting similar existing commercials if it finds any which are similar by name or content. The user can then confirm the creation of new commercials or accept the correction if he was mistaken.

AABOT is accessed via a web-browser. It is mostly used by uploading and downloading an annotation file. An interactive HTML5 interface is also available when some user feedback is needed (during validation). It uses Peyote as an description / indexation engine.

First APP deposit: IDDN.FR.001.4200010.000.S.P.2012.000.20900.

5.1.1.2. *Peyote*

Participants: Sébastien Champion, Jonathan Delhumeau [correspondent], Hervé Jégou.

Peyote is a framework for Video and Image description, indexation and nearest neighbor search. It can be used as-is by a video-search or image-search front-end with the implemented descriptors and search modules. It can also be used via scripting for large-scale experimentation. Finally, thanks to its modularity, it can be used for scientific experimentation on new descriptors or indexation methods. Peyote is used in the AABOT software and was used for the Mediaeval Placing task [68] and the Trecvid Instance Search task.

First APP deposit: IDDN.FR.001.4200008.000.S.P.2012.000.20900.

5.1.1.3. *Watermarking Effective Key Length Evaluation*

Participant: Teddy Furon [correspondent].

This software was developed in collaboration with Patrick BAS (CNRS, Ecole Centrale de Lille)

Weckle is a software suite in Matlab and R for the numerical evaluation of the effective key length of watermarking schemes based on Spread Spectrum, a concept which was proposed in [22], [23].

5.1.2. *Most active software started before 2012*

5.1.2.1. *Babaz*

Participants: Jonathan Delhumeau, Guillaume Gravier, Hervé Jégou [correspondent].

Babaz (<http://babaz.gforge.inria.fr/>) is a audio database management system with an audio-based search function, which is intended for audio-based search in video archives. First APP deposit: IDDN.FR.001.010006.000.S.P.2012.000.10000. It is licensed under the terms of the GNU General Public License v3.0.

5.1.2.2. *Bigimbaz*

Participant: Hervé Jégou [correspondent].

Bigimbaz is a platform originally developed in the LEAR project-team, and now co-maintained by TEXMEX. It integrates several contributions on image description and large-scale indexing: detectors, descriptors, retrieval using bag-of-words and inverted files, and geometric verification.

5.1.2.3. *BonzaiBoost*

Participant: Christian Raymond [correspondent].

The software homepage is available at <http://bonzaiboost.gforge.inria.fr/>.

BonzaiBoost stands for boosting over small decisions trees. BonzaiBoost is a general purpose machine-learning program based on decision tree and boosting for building a classifier from text and/or attribute-value data. Currently one configuration of BonzaiBoost is ranked first on <http://mlcomp.org> a website which propose to compare several classification algorithms on many different datasets

5.1.2.4. *Irisa_Ne*

Participant: Christian Raymond [correspondent].

IRISA_NE is a couple of Named Entity tagger, one of them is based on CRF and the other HMM. It is dedicated to automatic transcriptions of speech. It does not take into account uppercase or punctuation and has no concept of sentences. However, they also manage texts with punctuation and capitalization.

5.1.2.5. *Nero*

Participant: Sébastien Campion [correspondent].

The service is available at <https://nero.irisa.fr>.

NERO is an online Named Entities Recognition system. It is implemented within a web service that allows other member of the community to evaluate our results online without any client side setup. An HTTP Rest API, Shell and Python client are provided. The protocol used is HTTPS to secure the transactions between the user and the server. A user account is needed, which allow a fine monitoring. Usage are also limited to 100 thousand characters per account.

5.1.2.6. *SURVA*

Participants: Sébastien Campion [correspondent], Jonathan Delhumeau.

Speed Up Robust Video Alignement enables to quickly and efficiently synchronize the same video with two coding and quality formats (i.e. without the same number of frame). First APP deposit: IDDN.FR.001.420009.000.S.P.2012.000.20900.

5.1.2.7. *PimPy*

Participant: Sébastien Campion [correspondent].

PIMPY provides a convenient and high level API to manage common multimedia indexing tasks. It includes severals features. It is used, in particular

- to retrieve video features, such as histogram, binarized DCT descriptor, SIFT, SURF, etc ;
- to detect video cuts and dissolve (GoodShotDetector) ;
- for fast video frame access (pyffas) ;
- for raw frame extraction, or video segment extraction and re-encoding ;
- to search a video segment in another video (content based retrieval) ;
- to perform scene clustering.

First APP deposit: IDDN.FR.001.260038.000.S.P.2011.000.40000

5.1.2.8. *Pqcodes*

Participant: Hervé Jégou [correspondent].

This software is jointly maintained by Matthijs Douze, from Inria Grenoble.

Pqcodes is a library which implements the approximate k nearest neighbor search method of [83] based on product quantization. This software has been transferred to two companies (in August 2011 and May 2012, respectively).

The current version registered at the APP is IDDN.FR.001.220012.001.S.P.2010.000.10000.

5.1.2.9. Yael

Participant: Hervé Jégou [correspondent].

This software is jointly maintained by Matthijs Douze, from Inria Grenoble.

Yael is a C/python/Matlab library providing (multi-threaded, Blas/Lapack, low level optimization) implementations of computationally demanding functions. In particular, it provides very optimized functions for k-means clustering and exact nearest neighbor search. The library has been downloaded about 1000 times in 2012.

The current version registered at APP is IDDN.FR.001.220014.001.S.P.2010.000.10000.

5.1.2.10. IRISA News Topic Segmenter (irints)

Participants: Guillaume Gravier [correspondent], Camille Guinaudeau, Pascale Sébillot, Anca-Roxana Simon.

This software is dedicated to unsupervised topic segmentation of texts and transcripts. The software implements several of our research methods and is particularly adapted for automatic transcripts. It provides topic segmentation capabilities virtually for any word-based language, with presets for French, English and German. The software has been licensed to several of our industrial partners.

5.1.3. Other softwares

- BAG-OF-COLORS, implements a technique to describe the images based on color.
- I-DESCRIPTION. IDDN.FR.001.270047.000.S.P.2003.000.21000.
- ASARES is a symbolic machine learning system that automatically infers, from descriptions of pairs of linguistic elements found in a corpus in which the components are linked by a given semantic relation, corpus-specific morpho-syntactic and semantic patterns that convey the target relation. IDDN.FR.001.0032.000.S.C.2005.000.20900.
- ANAMORPHO detects morphological relations between words in many languages IDDN.FR.001.050022.000.S.P.2008.000.20900.
- DIVATEX is a audio/video frame server. IDDN.FR.001.320006.000.S.P.2006.000.40000,
- NAVITEX is a video annotation tool. IDDN.FR.001.190034.000.S.P.2007.000.40000,
- TELEMEX is a web service that enables TV and radio stream recording.
- VIDSIG computes a small and robust video signature (64 bits per image).
- VIDSEG computes segmentation features such as cuts, dissolves, silences in audio track, changes of ratio aspect, monochrome images. IDDN.FR.001.250009.000.S.P.2009.000.40000 ,
- ISEC, web application used as graphical interface for image searching engines based on retrieval by content.
- GPU-KMEANS, implementation of k-means algorithm on graphical process unit (graphic cards)
- CORRESPONDENCE ANALYSIS computes a factorial correspondence analysis (FCA) for image retrieval.
- GPU CORRESPONDENCE ANALYSIS is an implementation of the previous software Correspondence Analysis on graphical processing unit (graphical card).
- CAVIZ is an interactive graphical tool that allows to display and to extract knowledge from the results of a Correspondence Analysis on images.
- KIWI (standing for Keywords Extractor) is mostly dedicated to indexing and keyword extraction purposes.

- TOPIC SEGMENTER, is a software dedicated to topic segmentation of texts and (automatic) transcripts.
- S2E (Structuring Events Extractor) is a module which allows the automatic discovery of audiovisual structuring events in videos.
- 2PAC builds classes of words of similar meanings (“semantic classes“) specific to the use that is made of them in that given topic. IDDN.FR.001.470028.000.S.P.2006.000.40000
- FAESTOS (Fully Automatic Extraction of Sets of keywords for TOPic characterization and Spotting) is a tool composed of a sequence of statistical treatments that extracts from a morpho-syntactically tagged corpus sets of keywords that characterize the main topics that corpus deals with. IDDN.FR.001.470029.000.S.P.2006.000.40000
- FISHNET is an automatic web pages grabber associated with a specific theme.
- MATCH MAKER, semantic relation extraction by statistical methods.
- IRISAPHON produces phonetic words.
- PYTHON-GEOHASH is an implementation of the Geometric Hashing algorithm of [90] to check if geometrical consistency between pairs of images.
- AVSST is an Automatic Video Stream Structuring Tool. First, it allows the detection of repetitions in a TV stream. Second, a machine learning method allows the classification of programs and inter-programs such as advertisements, trailers, etc. Finally, the electronic program guide is synchronized with the right timestamps based on dynamic time warping. A graphical user interface is provided to manage the complete workflow.
- TVSEARCH is a content based retrieval search engine used to search and propagate manual annotation such as advertisement in a TV corpora.
- SAMUSA detects speech and/or musical segment in multimedia content.
- KERTRACK is a visual graphical interface for tracking visual targets based on particle filter tracking or based on mean-shift.
- MOZAIC2D creates of spatio-temporal mosaic based on dominant motion compensation.

5.2. Demonstration: Texmix

Participants: Morgan Bréhinier, Sébastien Campion [correspondent], Guillaume Gravier.

The gradual migration of television from broadcast diffusion to Internet diffusion offers tremendous possibilities for the generation of rich navigable contents. However, it also raises numerous scientific issues regarding de-linearization of TV streams and content enrichment. In this Texmix demonstration, we illustrate how speech in TV news shows can be exploited for de-linearization of the TV stream. In this context, de-linearization consists in automatically converting a collection of video files extracted from the TV stream into a navigable portal on the Internet where users can directly access specific stories or follow their evolution in an intuitive manner.

Structuring a collection of news shows requires some level of semantic understanding of the content in order to segment shows into their successive stories and to create links between stories in the collection, or between stories and related resources on the Web. Spoken material embedded in videos, accessible by means of automatic speech recognition, is a key feature to semantic description of video contents. We have developed multimedia content analysis technology combining automatic speech recognition, natural language processing and information retrieval to automatically create a fully navigable news portal from a collection of video files.

The demonstration was presented in several workshops (Futur en Seine - Paris, Futur TV - Berlin, ICMR - Hong Kong, French Minister for higher education and research - Rennes, RFIA - Lyon) and a video has been made available online on the portal of the EIT ICT Labs OpenSEM project. An article about this demonstrator was also published in 'Emergences' <http://emergences.inria.fr/2012/newsletter-n22/L22-TEXMIX>.

See the demo at <http://texmix.irisa.fr>.

5.3. Experimental platform

Participants: Laurent Amsaleg, Sébastien Campion [correspondent], Patrick Gros, Pascale Sébillot.

Until 2005, we used various computers to store our data and to carry out our experiments. In 2005, we began some work to specify and set-up dedicated equipment to experiment on very large collections of data. During 2006 and 2007, we specified, bought and installed our first complete platform. It is organized around a very large storage capacity (155TB), and contains 4 acquisition devices (for Digital Terrestrial TV), 3 video servers, and 15 computing servers partially included in the local cluster architecture (IGRIDA).

In 2008, we build up a corpus of multimedia data. It consists in a continuous recording (6 months) of two TV channels and three radios. It also includes web pages related to these contents captured on broadcaster's website. This corpus is to be used for different studies like the treatment of news along the time and to provide sub-corpus like TV news within the Quaero project (see below). The manual annotation of all the TV programs is under progress. A dedicated website has been developed in 2009 to provide a user support. It contains useful information such as references of available and ready to use software on the cluster, list of corpus stored on the platform, pages for monitoring disk space consumption and cluster loading, tutorials for best practices and cookbooks for treatments of large datasets. In 2010, we have acquired a new large memory server with 144GB of RAM which is used for memory demanding tasks, in particular to improve the speed of building index or language model. The previous server dedicated to this kind of jobs (acquired in 2008) has been upgraded to 96GB of RAM.

This year, we extended our storage capacity to 215TB and expanded our computing resources with two new large memory servers with 256GB of RAM for each of them.

This platform is funded by a joint effort of Inria, INSA Rennes and University of Rennes 1.

TOCCATA Team

5. Software

5.1. The CiME rewrite toolbox

Participants: Évelyne Contejean [contact], Claude Marché, Andrei Paskevich.

Keywords: Equational reasoning, Rewriting, Termination, Confluence, Completion

CiME is a rewriting toolbox. Distributed since 1996 as open source, at URL <http://cime.lri.fr>. Beyond a few dozens of users, CiME is used as back-end for other tools such as the TALP tool developed by Enno Ohlebusch at Bielefeld university for termination of logic programs; the MU-TERM tool (<http://www.dsic.upv.es/~lucas/csr/termination/muterm/>) for termination of context-sensitive rewriting; the CARIBOO tool (developed at Inria Nancy Grand-Est) for termination of rewriting under strategies; and the MTT tool (<http://www.lcc.uma.es/~duran/MTT/>) for termination of Maude programs. CiME2 is no longer maintained, and the currently developed version is CiME3, available at <http://a3pat.ensiie.fr/pub>. The main new feature of CiME3 is the production of traces for *Coq*. CiME3 is also developed by the participants of the A3PAT project at the CNAM, and is distributed under the Cecill-C license.

5.2. The Why platform

Participants: Claude Marché [contact], François Bobot, Jean-Christophe Filliâtre, Guillaume Melquiond, Andrei Paskevich.

Keywords: Deductive verification, Java programming language, Java modeling language, Java Card, ANSI C programming language.

Criteria for Software Self-Assessment ⁵: A-3, SO-4, SM-3, EM-2, SDL-5-down, OC-4.

The *Why* platform is a set of tools for deductive verification of Java and C source code. In both cases, the requirements are specified as annotations in the source, in a special style of comments. For Java (and Java Card), these specifications are given in JML and are interpreted by the *Krakatoa* tool. Analysis of C code must be done using the external *Frama-C* environment, and its *Jessie* plugin which is distributed in *Why*.

The platform is distributed as open source, under GPL license, at <http://why.lri.fr/>. The internal VC generator and the translators to external provers are no longer under active development, as superseded by the *Why3* system described below.

The *Krakatoa* and *Jessie* front-ends are still maintained, although using now by default the *Why3* VC generator. These front-ends are described in a specific web page <http://krakatoa.lri.fr/>. They are used for teaching (University of Evry, Ecole Polytechnique, etc.), used by several research groups in the world, e.g at Fraunhofer Institute in Berlin [86], at Universidade do Minho in Portugal [49], at Moscow State University, Russia (<http://journal.ub.tu-berlin.de/eceasst/article/view/255>).

5.3. The Why3 system

Participants: Jean-Christophe Filliâtre [contact], François Bobot, Claude Marché, Guillaume Melquiond, Andrei Paskevich.

Keywords: Deductive verification

Criteria for Software Self-Assessment: A-3-up, SO-4, SM-4, EM-4, SDL-4, OC-4.

⁵Self-evaluation following the guidelines (<http://www.inria.fr/content/download/11783/409665/version/4/file/SoftwareCriteria-V2-CE.pdf>) of the Software Working Group of Inria Evaluation Committee (<http://www.inria.fr/institut/organisation/instances/commission-d-evaluation>)

Why3 is the next generation of *Why*. *Why3* clearly separates the purely logical specification part from generation of verification conditions for programs. It features a rich library of proof task transformations that can be chained to produce a suitable input for a large set of theorem provers, including SMT solvers, TPTP provers, as well as interactive proof assistants.

It is distributed as open source, under GPL license, at <http://why3.lri.fr/>.

Why3 is used as back-end of our own tools *Krakatoa* and *Jessie*, but also as back-end of the GNATprove tool (Adacore company), and in a near future of the WP plugin of Frama-C. *Why3* has been used to develop and prove a significant part of the programs of our team gallery <http://proval.lri.fr/gallery/index.en.html>, and used for teaching (Master Parisien de Recherche en Informatique).

Why3 is used by other academic research groups, e.g. within the CertiCrypt/EasyCrypt project (<http://easycrypt.gforge.inria.fr/>) for certifying cryptographic programs.

5.4. The Alt-Ergo theorem prover

Participants: Sylvain Conchon [contact], Évelyne Contejean, Alain Mebsout, Mohamed Iguernelala.

Keywords: Automated theorem proving, Combination of decision procedures, Satisfiability modulo theories
Criteria for Software Self-Assessment: A-3-up, SO-4, SM-4-up, EM-4, SDL-5, OC-4.

Alt-Ergo is an automatic, little engine of proof dedicated to program verification, whose development started in 2006. It is fully integrated in the program verification tool chain developed in our team. It solves goals that are directly written in the *Why*'s annotation language; this means that *Alt-Ergo* fully supports first order polymorphic logic with quantifiers. *Alt-Ergo* also supports the standard [103] defined by the SMT-lib initiative.

It is currently used in our team to prove correctness of C and Java programs as part of the *Why* platform and the new *Why3* system. *Alt-Ergo* is also called as an external prover by the Pangolin tool developed by Y. Regis Ganas, Inria project-team Gallium <http://code.google.com/p/pangolin-programming-language/>. *Alt-Ergo* is usable as a back-end prover in the SPARK verifier for ADA programs, since Oct 2010. It is planed to be integrated in next generation of Airbus development process.

Alt-Ergo is distributed as open source, under the CeCILL-C license, at URL <http://alt-ergo.lri.fr/>.

5.5. The Cubicle model checker modulo theories

Participants: Sylvain Conchon [contact], Alain Mebsout.

Partners: A. Goel, S. Krstić (Intel Strategic Cad Labs in Hillsboro, OR, USA), F. Zaïdi (LRI, Université Paris-sud)

Keywords: Satisfiability modulo theories, model checking, array-based systems

Cubicle is an open source model checker for verifying safety properties of array-based systems. This is a syntactically restricted class of parametrized transition systems with states represented as arrays indexed by an arbitrary number of processes. Cache coherence protocols and mutual exclusion algorithms are typical examples of such systems.

Cubicle model-checks by a symbolic backward reachability analysis on infinite sets of states represented by specific simple formulas, called cubes. Cubicle is based on ideas introduced by MCMT (<http://users.mat.unimi.it/users/ghilardi/mcmt/>) from which, in addition to revealing the implementation details, it differs in a more friendly input language and a concurrent architecture. Cubicle is written in OCaml. Its SMT solver is a tightly integrated, lightweight and enhanced version of *Alt-Ergo*; and its parallel implementation relies on the Functor library.

5.6. Bibtex2html

Participants: Jean-Christophe Filliâtre [contact], Claude Marché.

Keywords: Bibliography, Bibtex format, HTML, World Wide Web.

Criteria for Software Self-Assessment: A-5, SO-3, SM-3, EM-3, SDL-5, OC-4.

Bibtex2html is a generator of HTML pages of bibliographic references. Distributed as open source since 1997, under the GPL license, at <http://www.lri.fr/~filliatr/bibtex2html/>. We estimate that between 10000 and 100000 web pages have been generated using Bibtex2html.

Bibtex2html is also distributed as a package in most Linux distributions. Package popularity contests show that it is among the 20% most often installed packages.

5.7. OCamlgraph

Participants: Jean-Christophe Filliâtre [contact], Sylvain Conchon.

Keywords: Graph, Library, *OCaml*.

OCamlgraph is a graph library for *OCaml*. It features many graph data structures, together with many graph algorithms. Data structures and algorithms are provided independently of each other, thanks to *OCaml* module system. OCamlgraph is distributed as open source, under the LGPL license, at <http://OCamlgraph.lri.fr/>. It is also distributed as a package in several Linux distributions. OCamlgraph is now widely spread among the community of *OCaml* developers.

5.8. Mlpost

Participants: Jean-Christophe Filliâtre [contact], François Bobot.

Keywords: Library, *OCaml*.

Mlpost is a tool to draw scientific figures to be integrated in LaTeX documents. Contrary to other tools such as TikZ or MetaPost, it does not introduce a new programming language; it is instead designed as a library of an existing programming language, namely *OCaml*. Yet it is based on MetaPost internally and thus provides high-quality PostScript figures and powerful features such as intersection points or clipping. Mlpost is distributed as open source, under the LGPL license, at <http://mlpost.lri.fr/>. Mlpost was presented at JFLA'09 [51].

5.9. Functory

Participant: Jean-Christophe Filliâtre [contact].

Keywords: Library, *OCaml*.

Functory is a distributed computing library for *OCaml*. The main features of this library include (1) a polymorphic API, (2) several implementations to adapt to different deployment scenarios such as sequential, multi-core or network, and (3) a reliable fault-tolerance mechanism. Functory was presented at JFLA 2011 [84] and at TFP 2011 [83].

5.10. The Pff library

Participant: Sylvie Boldo [contact].

Keywords: Interactive theorem proving, floating-point arithmetic. Criteria for Software Self-Assessment: A-2, SO-3, SM-3, EM-3, SDL-5, OC-4.

The Pff library for the *Coq* proof assistant is a formalization of floating-point arithmetic with high-level definitions and high-level properties [58].

It is distributed as open source, under a LGPL license, at <http://lipforge.ens-lyon.fr/www/pff/>, and is packaged in Debian and Ubuntu as “coq-float”.

It was initiated by M. Dumas, L. Rideau and L. Théry in 2001, and then developed and maintained by S. Boldo since 2004. It is now only maintained by S. Boldo. The development has ended as this library is now subsumed by the Flocq library (see below).

5.11. The Flocq library

Participants: Sylvie Boldo [contact], Guillaume Melquiond.

Keywords: Interactive theorem proving, floating-point arithmetic.

Criteria for Software Self-Assessment: A-2, SO-3, SM-3, EM-3, SDL-4, OC-4.

The Flocq library for the *Coq* proof assistant is a comprehensive formalization of floating-point arithmetic: core definitions, axiomatic and computational rounding operations, high-level properties [5]. It provides a framework for developers to formally certify numerical applications.

It is distributed as open source, under a LGPL license, at <http://flocq.gforge.inria.fr/>. It was first released in 2010.

5.12. The Gappa tool

Participant: Guillaume Melquiond [contact].

Keywords: Automated theorem proving, floating-point arithmetic, fixed-point arithmetic.

Criteria for Software Self-Assessment: A-3, SO-4, SM-4, EM-3, SDL-4, OC-4.

Given a logical property involving interval enclosures of mathematical expressions, Gappa tries to verify this property and generates a formal proof of its validity. This formal proof can be machine-checked by an independent tool like the *Coq* proof-checker, so as to reach a high level of confidence in the certification [79] [109].

Since these mathematical expressions can contain rounding operators in addition to usual arithmetic operators, Gappa is especially well suited to prove properties that arise when certifying a numerical application, be it floating-point or fixed-point. Gappa makes it easy to compute ranges of variables and bounds on absolute or relative roundoff errors.

Gappa is being used to certify parts of the mathematical libraries of several projects, including CRLibm, FLIP, and CGAL. It is distributed as open source, under a Cecill-B / GPL dual-license, at <http://gappa.gforge.inria.fr/>. Part of the work on this tool was done while in the Arénaire team (Inria Rhône-Alpes), until 2008.

5.13. The Interval package for Coq

Participant: Guillaume Melquiond [contact].

Keywords: Interactive theorem proving, interval arithmetic, floating-point arithmetic.

Criteria for Software Self-Assessment: A-3, SO-4, SM-3, EM-3, SDL-4, OC-4.

The Interval package provides several tactics for helping a *Coq* user to prove theorems on enclosures of real-valued expressions. The proofs are performed by an interval kernel which relies on a computable formalization of floating-point arithmetic in *Coq*.

It is distributed as open source, under a LGPL license, at <http://www.lri.fr/~melquion/soft/coq-interval/>. Part of the work on this library was done while in the Mathematical Components team (Microsoft Research–Inria Joint Research Center).

In 2010, the Flocq library was used to straighten and fill the floating-point proofs of the Interval package.

5.14. The Alea library for randomized algorithms

Participants: Christine Paulin-Mohring [contact], Pierre Courtieu.

Keywords: Interactive theorem proving, randomized algorithms, probability

Criteria for Software Self-Assessment: A-2, SO-3, SM-2, EM-3, SDL-4, OC-4.

The ALEA library is a *Coq* development for modeling randomized functional programs as distributions using a monadic transformation. It contains an axiomatisation of the real interval $[0, 1]$ and its extension to positive real numbers. It introduces definition of distributions and general rules for approximating the probability that a program satisfies a given property.

It is distributed as open source, at <http://www.lri.fr/~paulin/ALEA>. It is currently used as a basis of the Certicrypt environment (MSR-Inria joint research center, Imdea Madrid, Inria Sophia-Antipolis) for formal proofs for computational cryptography [54]. It is also experimented in LABRI as a basis to study formal proofs of probabilistic distributed algorithms.

5.15. The Coccinelle library for term rewriting

Participant: Évelyne Contejean [contact].

Keywords: Interactive theorem proving, Coq, rewriting, termination certificate Coccinelle is a *Coq* library for term rewriting. Besides the usual definitions and theorems of term algebras, term rewriting and term ordering, it also models some of the algorithms implemented in the CiME toolbox, such a matching, matching modulo associativity-commutativity, computation of the one-step reducts of a term, RPO comparison between two terms, etc. The RPO algorithm can effectively be run inside *Coq*, and is used in the Color development (<http://color.inria.fr/>) as well as for certifying Spike implicit induction theorems in *Coq*(Sorin Stratulat).

Coccinelle is developed by Évelyne Contejean, available at <http://www.lri.fr/~contejea/Coccinelle>, and is distributed under the Cecill-C license.

5.16. The Coquelicot library for real analysis

Participants: Sylvie Boldo [contact], Catherine Lelay, Guillaume Melquiond.

Keywords: Interactive theorem proving, real analysis

Criteria for Software Self-Assessment: A-2, SO-4, SM-2, EM-3, SDL-1, OC-4.

Coquelicot is a *Coq* library dedicated to real analysis: differentiation, integration, and so on. It is a conservative extension of the standard library of *Coq*, but with a strong focus on usability.

Coquelicot is available at <http://coquelicot.saclay.inria.fr/>.

5.17. CFML

Participant: Arthur Charguéraud [contact].

Keywords: Program verification, Interactive theorem proving, *OCaml*

Criteria for Software Self-Assessment: A-2, SO-4, SM-2, EM-3, SDL-1, OC-4. The *CFML* tool supports the verification of *OCaml* programs through interactive *Coq* proofs. The tool is made of two parts: on the one hand, a characteristic formula generator implemented as a *OCaml* program that parses *OCaml* code and produces Coq formulae; and, on the other hand, a Coq library that provides notation and tactics for manipulating characteristic formulae interactively in Coq.

CFML is distributed under the LGPL license, and is available at <http://arthur.chargueraud.org/softs/cfml/>. The tool has been initially developed by A. Charguéraud in 2010, and has been maintained and improved since by the author.

TOSCA Project-Team

5. Software

5.1. CarbonQuant

Participant: Mireille Bossy [correspondant].

CarbonQuant is a simulator project of CO2 allowances prices on a EU-ETS type market, by an indifference price approach.

It aims to demonstrate the high potentiality of stochastic control solvers, to quantify sensibilities of a carbon market with respect to its design.

Starting in September 2011, CarbonQuant is an ADT ¹ Inria.

See also the web page <http://carbonvalue.gforge.inria.fr>.

- Version: 0.1

¹Technology Development Action

TREC Project-Team

5. Software

5.1. Gibbs' Sampler

Participants: François Baccelli, Chung Shue Chen.

The work on the self optimization of cellular networks based on Gibbs' sampler (see Section 6.1.1.5) carried out with Chung Shue Chen (ALU) in the joint laboratory with Alcatel-Lucent, led to the development of a software prototype that was presented by L. Roullet at the Inria Alcatel-Lucent joint laboratory seminar in November 2012.

5.2. PSI2

Participant: Ana Bušić.

The work on perfect sampling (see Section 6.2.2) has been partially implemented in a software tool PSI2, in collaboration with MESCAL team [Inria Grenoble - Rhône-Alpes]; <https://gforge.inria.fr/projects/psi>.

TRIO Project-Team

5. Software

5.1. ANR Open-PEOPLE platform

Participants: Fabrice Vergnaud, Jérôme Vatrinet, Kévin Roussel, Olivier Zendra.

The aim of Open-PEOPLE is to provide a platform for estimating and optimizing the power and energy consumption of systems. The Open-PEOPLE project formally started in April 2009. Two systems administrator and software developers had been hired initially: Sophie Alexandre and Kévin Roussel. Another system administrator and software developer, Jonathan Ponroy, joined them in 2010 when he finished his work on the ANR MORE project where he worked previously. Sophie Alexandre contract ended in February 2011.

Since the beginning of the Open-PEOPLE project, we had made significant progress in setting up the infrastructure for the software part of the platform, for which Inria Nancy Grand Est is responsible. We had included new features to be able to fully integrate and test software developed as Eclipse plugins, relying on the Buckminster tool. We had also created a specific extension set for SVN and Hudson, called OPCIM (Open-PEOPLE Continuous Integration Mechanism). OPCIM had been registered at APP on 13/04/2010 with number IDDN.FR.001.150008.000.S.P.2010.000.10000.

Concerning the Open-PEOPLE platform itself, we had first tackled the high-level work, working with our partners on the definition of the requirements of the platform according to the needs of industry. We had then realized the specification work to define the global perimeter of our platform, according to the previous requirements. As part of this work had also been designed exchanges formats between the various tools. We had also designed at Inria Nancy Grand Est a Tools integration Protocol, which specified requirements for external tools to be integrated in our platform. All this design work had been materialized in several reports which were deliveries provided to ANR.

We had also designed and developed an authentication component (Eclipse plugin) for the platform, so as to be able to provide a unique, secured access gate to the platform to all the tools that are or shall be integrated into it.

We had also started and almost finished developing an Internet portal giving access and control to the Open-PEOPLE Hardware Platform, located at our partner's UBS in Lorient. Our portal features included user account management facilities, on the admin side, and on the user side, the ability to create, save, edit, reuse and of course submit jobs, make reservations for the hardware platform resources and get back tests results.

Finally, we had started working on two important parts of the software platform.

First, a way to unify the user experience despite the fact the platform federates several tools which were not developed to interact together. This implied an important and in-depth study of the wanted ergonomics for the platform, which involved taking into account both user needs and habits and the features of the available software tools.

The second work which had begun in 2011 was the design (then implementation) of the communications of between the various tools of the platform. This skeleton is a key part of our platform, and the quality of its design has a tremendous impact on its maintainability and its extensibility.

Note that the Open-PEOPLE project had been successfully evaluated on 14/09/2010 by ANR. Developments done during the first two years in the project are detailed in the 2009 and 2010 activity reports. In 2011, these developments had gone on.

We had continued the work to solidify our development platform supporting our work and that of our partners. We had produced a finer grained definition of the software platform functionalities, and a more precise definition of the tools integration protocol. We had worked towards the corresponding implementation documents, adding two new deliverables about the architecture of the software platform and the ergonomics of the software platform. For the latter, we had extensively interviewed user about ergonomics and designed several GUI mockups. We had progressed on the implementation of the software platform, especially with respect to the internet portal to remote-control the hardware platform. We had participated to the definition of the hardware platform and its functionalities, and participated actively to the work on the Specification document for HW / SW interfacing. We had provided the first concrete design and implementation of the HW/SW platform interfacing, with our implementation of the remote control portal for the HW platform. This remote control module had been completed in Fall 2011.

We had also participated to the work pertaining to basic components model homogenization, by reviewing this in the context of the software platform architecture and implementation, which had resulted in several incremental improvements of the underlying models. Finally, progressing towards the first release of the software part of the Open-PEOPLE platform, we had realized an ergonomic study for the consumption laws editors, with mockups and user interviews and validation. We had worked on the implementation of the editors for the consumption laws, which had required learning new environments and development tools (related to the EMF framework and the AADL, QUDV and MathML models). As a consequence, we had completed the implementation of the GUI and engine to create units and quantities. We had finalized the architecture needed to integrate external modules in the platform.

In 2012, this work went on. Basically, 2012 was the year of the concrete Open-PEOPLE platform, where all our efforts finally came to maturity. We thus completed global GUI of the Open-PEOPLE Software Platform. We performed the integration of various external tools and modules and the . We provided several improvements to the Remote Control Module providing access to the Hardware Platform. We finalized the implementation of consumption laws editors. We implemented the export and import functionality of Open-PEOPLE models. We created a new community-based website to allow sharing of Open-PEOPLE models.

We overall progressed as forecast in an iterative development and release schedule.

Version 1.0 (2012-04-06) was the first embodiment and public release of the Open-PEOPLE platform.

Version 1.1 (2012-06-12) added a default environment with pre-set Units, Quantities and AADL Property associations, asynchronous file uploads and downloads in the Remote Control Module, and better handling of big files (file size limit is now 4GB), and several bug fixes.

Version 1.3 (2012-09-27) changed internal mechanism of QUDV serialization (quantities, units and property associations), added version information to QUDV and Weaving meta-models, added internal builders to automatically generates QUDV and Weaving configuration files, added support of OSATE 2, improved UI reactivity (especially during file transfers), added progress bars for the remote control, and several bug fixes.

Version 1.4 (2012-10-25) added the Adele Graphical Editor, new OSATE 2 Snapshot, and several bugfixes.

Version 2.0 (2012-12-13) added RDALTE, AADL2SystemC, and a Standard environment with models and model sharing implemented (including a community sharing website), a new snapshot of OSATE2.

5.2. VITRIL

Participants: Frédéric Diss, Pierre Caserta, Olivier Zendra.

The aim of the VITRIL operation is to provide tools for the advanced and immersive visualization of programs. It partners with the University of Montréal, University of Montpellier and Pareo team of Inria Nancy Grand Est.

Last years, in VITRAIL, we had developed software to instrument and trace Java programs at the bytecode level. We then had developed an analysis tool able to exploit these traces to compute relevant software metrics. We had hired Damien Bodenes as software developer, and had begun the work on a prototype able to render a 3D world, symbolizing software, onto various visualization hardware, with the possibility to change the display metaphor. The main part of our development work had been in 2009 the choice and validation of the technology, and a first architecture. In 2010, the development had go on at a good pace, building on chosen technologies and architecture. This had brought new experience, and with the first actual runs of our platform, we had realized that with the Irrlicht platform we had chosen, we could reach unforeseeable problem when scaling up. We had thus decided to reverse our choice to the Ogre3D 3D engine at the beginning of 2010. Our development had then progressed steadily.

We had released in 2010 a first prototype of our platform, with all the underlying architecture, able to provide navigation features and interaction capacities limited to the driving of the navigation, as per our plans. This had included dual screen management.

Our first prototype, using 2 large 2D screens, with a city metaphor, had been demonstrated during the "Fête de la Science" in November 2010 and had received a lot of attention and enthusiasm from the general public. About 55 persons per day had visited our booth and got demonstrations.

We had also progressed significantly in our Java bytecode tracer, by improving its granularity, the completeness of the traced information, and its performance as well. We have a unique tool which is able to trace both program classes and JDK classes, at basic block level. In addition, it does so with a dynamic instrumentation of classes, which means there is no need to have an instrumented version of the class files on disk. This is very convenient, especially when changing machine of JVM, or when upgrading either the JDK or the program itself. In addition, the performance is good enough that the instrumented programs are still fully usable in an interactive way, without bothering the user. To the best of our knowledge, this is the only Java bytecode tracer that offers these features nowadays.

Our software development had lead to several registrations with APP:

- VITRAIL - Visualizer had been first registered on 29/12/2009 under number IDDN.FR.001.530021.000.S.P.2009.000.10000.
- VITRAIL - Tracer, was registered at APP on 20/09/2010 with number IDDN.FR.001.380001.000.S.P.2010.000.10000.

In 2011, we acquired a workstation and three 30 inches computer screens, to be able to set up a "boxed 3D workstation", that would provide display in front and on both sides of the operator. This would constitute the next step in our experiments, by improving immersion with a larger field of vision (on the sides). The software developments to do this are ongoing. We also integrated a WiiMote interaction device to our system, but our experiments found that its spacial resolution was too poor for our needs.

We finally improved significantly our VITRAIL prototype in 2011, especially by designing and implementing a new representation for the relations between software (hence visual) elements, with limited clutter and the possibility to regroup links and see their direction.

In 2012, we continued working on the analysis of software, gathering statistics about polymorphism in Java programs, aiming at comparing various type analyses made statically (CHA, RTA, VTA) and the dynamic trace provided by (a) real execution(s). This work is going on and has not been published yet.

We also developed a public website for the VITRAIL project, which is going live these days.

TRISKELL Project-Team

5. Software

5.1. Kermeta

Participants: Didier Vojtisek [correspondant], Olivier Barais, Arnaud Blouin, Benoit Combemale, Jacques Falcou, François Fouquet, Marie Gouyette, Clément Guy, Jean-Marc Jézéquel, Jonathan Marchand.

Nowadays, object-oriented meta-languages such as MOF (meta-object Facility) are increasingly used to specify domain-specific languages in the model-driven engineering community. However, these meta-languages focus on structural specifications and have no built-in support for specifications of operational semantics. Integrated with the industrial standard Ecore and aligned with the OMG standard EMOF 2.0, the Kermeta language consists in an extension to these meta languages to support behavior definition. The language adds precise action specifications with static type checking and genericity at the meta level. Based on object-orientation and aspect orientation concepts, the Kermeta language adds model specific concepts. It is used in several use cases:

- to give a precise semantic of the behavior of a metamodel which then can be simulated.
- to act as a model transformation language.
- to act as a constraint language.

The development environment built for the Kermeta language provides an integrated workbench based on Eclipse. It offers services such as : model execution, text editor (with syntax highlighting, code autocompletion), additional views and various import/export transformations.

Thanks to Kermeta it is possible to build various frameworks dedicated to domain specific metamodels. Those frameworks are organised into MDKs (Model Development Kits). For example, Triskell proposes MDKs to work with metamodels such as Java5, UML2, RDL (requirements), Ecore, Traceability,...

In 2011, Kermeta tooling has been refactored into a version 2.0.x in order to ease the integration of various MOF related languages in the tool chain. This new version also focuses on a fully compiled mode that allows to deploy Kermeta programs in production environments.

See also the web page <http://www.kermeta.org>.

- APP: IDDN.FR.001.420009.000.S.P.2005.000.10400
- Version: 2.0.1
- Programming language: Java, Scala, Kermeta

Main competitors:

- XMF-Mosaic is developed by Ceteva and is now open-source since 2008.
- GME is a large scale Meta-Modeling Environment developed at Vanderbilt University (ISIS project) since 2002.
- MOFLON is a Metamodeling Framework with Graph Transformations, developed by A. Schuerr's group (TU-Darmstadt) since 2008.
- XCore is a recent (2011) Eclipse project supported by Itemis/Macro Modelling that provides a single operational surface syntax for Ecore.
- Many QVT inspired model transformation tools focused on model transformations.

Main innovative features:

Kermeta was one of the first solutions to offer an operational semantics on top of EMOF. It still proposes several unique features that cannot be found in the tools presented above, such as:

- aspect weaving at the metamodel level allows fast prototyping of a wide variety of tools;
- model typing allows a safe reuse of algorithms and transformations across different metamodels.

5.2. Kevoree

Participants: Olivier Barais [correspondant], François Fouquet, Erwan Daubert, Jean-Émile Dartois, Johann Bourcier, Antonio Mattos, Noël Plouzeau.

Kevoree is an open-source models@runtime platform ¹ to properly support the dynamic adaptation of distributed systems. Models@runtime basically pushes the idea of reflection [82] one step further by considering the reflection layer as a real model that can be uncoupled from the running architecture (e.g. for reasoning, validation, and simulation purposes) and later automatically resynchronized with its running instance.

Kevoree has been influenced by previous work that we carried out in the DiVA project [82] and the Entimid project [83]. With Kevoree we push our vision of models@runtime [81] farther. In particular, Kevoree provides a proper support for distributed models@runtime. To this aim we introduced the *Node* concept to model the infrastructure topology and the *Group* concept to model semantics of inter node communication during synchronization of the reflection model among nodes. Kevoree includes a *Channel* concept to allow for multiple communication semantics between remote *Components* deployed on heterogeneous nodes. All Kevoree concepts (Component, Channel, Node, Group) obey the object type design pattern to separate deployment artifacts from running artifacts. Kevoree supports multiple kinds of very different execution node technology (e.g. Java, Android, MiniCloud, FreeBSD, Arduino, ...).

Kevoree is distributed under the terms of the LGPL open source license.

Main competitors:

- the Fractal/Frascati eco-system ².
- SpringSource Dynamic Module ³
- GCM-Proactive ⁴
- OSGi ⁵
- Chef ⁶
- Vagran ⁷

Main innovative features:

- distributed models@runtime platform (with a distributed reflection model and an extensible models@runtime dissemination set of strategies).
- Support for heterogeneous node type (from Cyber Physical System with few resources until cloud computing infrastructure).
- Fully automated provisioning model to correctly deploy software modules and their dependencies.
- Communication and concurrency access between software modules expressed at the model level (not in the module implementation).

Impact: Several European projects leveraging the Kevoree platform have recently been accepted. Besides we are currently developing a testbed named DAUM. This testbed is developed since mid 2011 to experiment with Kevoree in real life situations. More precisely, DAUM is a highly dynamic pervasive system that mixes wireless smart sensors, user interaction devices such as digital pads, and distributed data servers in a cloud. The current specialization of DAUM is a distributed tactical information and decision system for firefighters. This application includes individual sensors in the personal protective equipment of firefighters, embedded computation nodes that are fully reconfigurable in real time and over the air, distributed monitoring servers

¹<http://www.kevoree.org>

²<http://frascati.ow2.org>

³<http://www.springsource.org/osgi/>

⁴<http://proactive.inria.fr/>

⁵<http://www.osgi.org>

⁶<http://wiki.opscode.com/display/chef/Deploy+Resource>

⁷<http://vagrantup.com/>

in trucks, and personal computers for information access and decision making. The DAUM platform is used internally to try research results on distributed *models@runtime*. DAUM is used externally to prepare and support cooperation activities with other research teams (the Myriads Inria team is a partner of DAUM) and with potential industrial partners.

See also the web page <http://www.kevoree.org>.

- Version: 1.0
- Programming language: Java, Scala, Kermeta

TYPICAL Project-Team

4. Software

4.1. Coq

Participants: Bruno Barras [Contact], Jean-Marc Notin, Enrico Tassi.

Coq is a major proof system and the primary object and / or tool of our research. Its development is now mainly coordinated by the π^2 Inria Paris-Rocquencourt project-team, and some members of the TypiCal team are active developers of the system.

4.2. Coqfinitgroup

Participants: Cyril Cohen, Assia Mahboubi [Contact], Enrico Tassi.

Coqfinitgroup is the development corresponding to the full formalization of the proof of the Feit-Thompson theorem. It is probably the most advanced formal development of group theory today. Its current size is about 80.000 lines of (compact) **Coq** code. Assia Mahboubi and Cyril Cohen have been actively participating to this long term formalization project.

4.3. Ssreflect

Participants: Assia Mahboubi [Contact], Enrico Tassi.

SSReflect is a proof language extension of **Coq** developed under Georges Gonthier (Microsoft Research). It was originally designed to make the formalization of the Four Color Theorem possible and has been evolving since. It is important to note that it is shipped with redesigned basic proof libraries. Enrico Tassi has worked on an extended language of patterns for term selection now included in the distribution of this extension. Members of the Typical are in charge of the documentation and distribution of this extension.

URBANET Team

5. Software

5.1. WSNet.

UrbaNet is an active contributor to WSnet (<http://wsnet.gforge.inria.fr/>), a discrete event simulator dedicated to large scale wireless networks developed and maintained by members of Inria and CITI lab. A major part of this contribution is represented by the implementation of state of the art protocols for medium access control and routing.

The WSNet simulation results obtained following this process are sometimes used as an input for another part of our development effort, which consists in prototype software based on the combination of CPLEX and AMPL for solving mixed integer linear programming problems with column generation.

5.2. TAPASCologne vehicular mobility dataset.

Based on the data made available by the Institute of Transportation Systems at the German Aerospace Center (ITS-DLR), the dataset aims at reproducing, with a high level of realism, car traffic in the greater urban area of the city of Cologne, Germany. To that end, different state-of-art data sources and simulation tools are brought together, so to cover all of the specific aspects required for a proper characterization of vehicular traffic:

- The street layout of the Cologne urban area is obtained from the OpenStreetMap (OSM) database;
- The microscopic mobility of vehicles is simulated with the Simulation of Urban Mobility (SUMO) software;
- The traffic demand information on the macroscopic traffic flows across the Cologne urban area (i.e., the O/D matrix) is derived through the Travel and Activity PATterns Simulation (TAPAS) methodology;
- The traffic assignment of the vehicular flows described by the TAPASCologne O/D matrix over the road topology is performed by means of Gawron's dynamic user assignment algorithm.

The resulting synthetic trace of the car traffic in a the city of Cologne covers a region of 400 square kilometers for a period of 24 hours, comprising more than 700.000 individual car trips. More information is available on the project website at <http://kolntrace.project.citi-lab.fr/> .

VEGAS Project-Team

4. Software

4.1. QI: Quadrics Intersection

QI stands for “Quadrics Intersection”. QI is the first exact, robust, efficient and usable implementation of an algorithm for parameterizing the intersection of two arbitrary quadrics, given in implicit form, with integer coefficients. This implementation is based on the parameterization method described in [10], [29], [30], [31] and represents the first complete and robust solution to what is perhaps the most basic problem of solid modeling by implicit curved surfaces.

QI is written in C++ and builds upon the LiDIA computational number theory library [24] bundled with the GMP multi-precision integer arithmetic [23]. QI can routinely compute parameterizations of quadrics having coefficients with up to 50 digits in less than 100 milliseconds on an average PC; see [10] for detailed benchmarks.

Our implementation consists of roughly 18,000 lines of source code. QI has being registered at the Agence pour la Protection des Programmes (APP). It is distributed under the free for non-commercial use Inria license and will be distributed under the QPL license in the next release. The implementation can also be queried via a web interface [25].

Since its official first release in June 2004, QI has been downloaded six times a month on average and it has been included in the geometric library EXACUS developed at the Max-Planck-Institut für Informatik (Saarbrücken, Germany). QI is also used in a broad range of applications; for instance, it is used in photochemistry for studying the interactions between potential energy surfaces, in computer vision for computing the image of conics seen by a catadioptric camera with a paraboloidal mirror, and in mathematics for computing flows of hypersurfaces of revolution based on constant-volume average curvature.

4.2. Isotop: Topology and Geometry of Planar Algebraic Curves

ISOTOP is a Maple software for computing the topology of an algebraic plane curve, that is, for computing an arrangement of polylines isotopic to the input curve. This problem is a necessary key step for computing arrangements of algebraic curves and has also applications for curve plotting. This software has been developed since 2007 in collaboration with F. Rouillier from Inria Paris - Rocquencourt. It is based on the method described in [28] which incorporates several improvements over previous methods. In particular, our approach does not require generic position.

Isotop is registered at the APP (June 15th 2011) with reference IDDN.FR.001.240007.000.S.P.2011.000.10000. This version is competitive with other implementations (such as ALCIX and INSULATE developed at MPII Saarbrücken, Germany and TOP developed at Santander Univ., Spain). It performs similarly for small-degree curves and performs significantly better for higher degrees, in particular when the curves are not in generic position.

We are currently working on an improved version integrating our new bivariate polynomial solver [27].

4.3. CGAL: Computational Geometry Algorithms Library

Born as a European project, CGAL (<http://www.cgal.org>) has become the standard library for computational geometry. It offers easy access to efficient and reliable geometric algorithms in the form of a C++ library. CGAL is used in various areas needing geometric computation, such as: computer graphics, scientific visualization, computer aided design and modeling, geographic information systems, molecular biology, medical imaging, robotics and motion planning, mesh generation, numerical methods...

In computational geometry, many problems lead to standard, though difficult, algebraic questions such as computing the real roots of a system of equations, computing the sign of a polynomial at the roots of a system, or determining the dimension of a set of solutions. We want to make state-of-the-art algebraic software more accessible to the computational geometry community, in particular, through the computational geometric library CGAL. On this line, we contributed a model of the *Univariate Algebraic Kernel* concept for algebraic computations [26] (see Sections 8.2.2 and 8.4). This CGAL package improves, for instance, the efficiency of the computation of arrangements of polynomial functions in CGAL [32]. We are currently developing a model of the *Bivariate Algebraic Kernel* based on our new bivariate polynomial solver [27]. This work is done in collaboration with F. Rouillier at Inria Paris - Rocquencourt and L. Peñaranda at the university of Athens.

4.4. Fast_polynomial: fast polynomial evaluation software

The library *fast_polynomial*¹ provides fast evaluation and composition of polynomials over several types of data. It is interfaced for the computer algebra system *sage*. This software is meant to be a first step toward a certified numerical software to compute the topology of algebraic curves and surfaces. It can also be useful as is and is submitted for integration in the computer algebra system *Sage*.

This software is focused on *fast online computation*, *multivariate evaluation*, *modularity*, and *efficiency*.

Fast online computation. The library is optimized for the evaluation of a polynomial on several point arguments given one after the other. The main motivation is numerical path tracking of algebraic curves, where a given polynomial criterion must be evaluated several thousands of times on different values arising along the path.

Multivariate evaluation. The library provides specialized fast evaluation of multivariate polynomials with several schemes, specialized for different types such as *mpz* big ints, *boost* intervals with hardware precision, *mpfi* intervals with any given precision, etc.

Modularity. The evaluation scheme can be easily changed and adapted to the user needs. Moreover, the code is designed to easily extend the library with specialization over new C++ objects.

Efficiency. The library uses several tools and methods to provide high efficiency. First, the code uses templates, such that after the compilation of a polynomial for a specific type, the evaluation performance is equivalent to low-level evaluation. Locality is also taken into account: the memory footprint is minimized, such that an evaluation using the classical Hörner scheme will use $O(1)$ temporary objects and divide and conquer schemes will use $O(\log(n))$ temporary objects, where n is the degree of the polynomial. Finally, divide and conquer schemes can be evaluated in parallel, using a number of threads provided by the user.

¹http://trac.sagemath.org/sage_trac/ticket/13358

VERIDIS Project-Team

5. Software

5.1. The veriT solver

Participants: Rodrigo Castaño, David Déharbe, Pablo Federico Dobal, Pascal Fontaine [correspondent].

The veriT solver is an SMT (Satisfiability Modulo Theories) solver developed in cooperation with David Déharbe from the Federal University of Rio Grande do Norte in Natal, Brazil. The solver can handle large quantifier-free formulas containing uninterpreted predicates and functions, and arithmetic on integers and reals. It features a very efficient decision procedure for difference logic, as well as a simplex-based reasoner for full linear arithmetic. It also has some support for user-defined theories, quantifiers, and lambda-expressions. This allows users to easily express properties about concepts involving sets, relations, etc. The prover can produce an explicit proof trace when it is used as a decision procedure for quantifier-free formulas with uninterpreted symbols and arithmetic. To support the development of the tool, a regression platform using Inria's grid infrastructure is used; it allows us to extensively test the solver on thousands of benchmarks in a few minutes. The veriT solver is available as open source under the BSD license, and distributed through the web site <http://www.veriT-solver.org>.

Efforts in 2012 have been focused on efficiency, with various improvements and the redesign of the core solver. A preliminary prototype integrating **Redlog** for handling non-linear arithmetic showed encouraging results. Short term future works include improving the design, adding full support for non-linear arithmetic, and increasing efficiency.

We target applications where validation of formulas is crucial, such as the validation of TLA^+ and B specifications, and work together with the developers of the respective verification platforms to make veriT even more useful in practice. In 2012, we presented at ABZ [16] a plugin for Rodin using SMT solvers (and notably veriT) to discharge B proof obligations: on a large repository of industrial and academic cases, this SMT-based plugin decreased by 75% the number of proof obligations requiring human interactions, compared to the original B prover. See also section 8.1 for our work within the DeCert project.

For helping development within and around veriT, Pablo Federico Dobal has been hired for two years starting September 2012 as a young engineer supported by the Inria ADT program.

5.2. The TLA^+ proof system

Participants: Stephan Merz [correspondent], Hernán-Pablo Vanzetto.

TLAPS, the TLA^+ proof system, is a platform for developing and mechanically verifying TLA^+ proofs. It is developed at the Joint MSR-Inria Centre. The TLA^+ proof language is declarative and based on standard mathematical logic; it supports hierarchical and non-linear proof construction and verification. TLAPS consists of a *proof manager* that interprets the proof language and generates a collection of proof obligations that are sent to *backend verifiers* that include theorem provers, proof assistants, SMT solvers, and decision procedures.

TLAPS is publically available at <http://msr-inria.inria.fr/~doligez/tlaps/>, it is distributed under a BSD-like license. It handles the non-temporal part of TLA^+ and can currently be used to prove safety, but not liveness properties. Its backends include a tableau prover for first-order logic, an encoding of TLA^+ in the proof assistant Isabelle, and a backend for interfacing with SMT solvers. The SMT backend has been improved significantly in 2012 and is now considered by users as the most useful backend prover for system verification. Version 1.0 of TLAPS was released in January 2012, followed by version 1.1 in November, and the system was presented at the conference FM 2012 [15].

VERTECS Project-Team

5. Software

5.1. STG

Participant: Thierry Jéron.

STG (Symbolic Test Generation) is a prototype tool for the generation and execution of test cases using symbolic techniques. It takes as input a specification and a test purpose described as IOSTS, and generates a test case program also in the form of IOSTS. Test generation in STG is based on a syntactic product of the specification and test purpose IOSTS, an extraction of the subgraph corresponding to the test purpose, elimination of internal actions, determinisation, and simplification. The simplification phase now relies on NBAC, which approximates reachable and coreachable states using abstract interpretation. It is used to eliminate unreachable states, and to strengthen the guards of system inputs in order to eliminate some *Inconclusive* verdicts. After a translation into C++ or Java, test cases can be executed on an implementation in the corresponding language. Constraints on system input parameters are solved on-the-fly (i.e. during execution) using a constraint solver. The first version of STG was developed in C++, using Omega as constraint solver during execution. This version has been deposited at APP under number IDDN.FR.001.510006.000.S.P.2004.000.10600.

A new version in OCaml has been developed in the last years. This version is more generic and will serve as a library for symbolic operations on IOSTS. Most functionalities of the C++ version have been re-implemented. Also a new translation of abstract test cases into Java executable tests has been developed, in which the constraint solver is LUCKYDRAW (VERIMAG). This version has also been deposited at APP and is available for download on the web as well as its documentation and some examples.

Finally, in collaboration with ULB, we implemented a prototype SMACS, derived from STG, devoted to the control of infinite systems modeled by STS.

5.2. SIGALI

Participant: Hervé Marchand.

SIGALI is a model-checking tool that operates on ILTS (Implicit Labeled Transition Systems, an equational representation of an automaton), an intermediate model for discrete event systems. It offers functionalities for verification of reactive systems and discrete controller synthesis. It is developed jointly by the ESPRESSO and VERTECS teams. The techniques used consist in manipulating the system of equations instead of the set of solutions, which avoids the enumeration of the state space. Each set of states is uniquely characterized by a predicate and the operations on sets can be equivalently performed on the associated predicates. Therefore, a wide spectrum of properties, such as liveness, invariance, reachability and attractivity, can be checked. Algorithms for the computation of predicates on states are also available [27] [23]. SIGALI is connected with the Polychrony environment (ESPRESSO project-team) as well as the Matou environment (VERIMAG), thus allowing the modeling of reactive systems by means of Signal Specification or Mode Automata and the visualization of the synthesized controller by an interactive simulation of the controlled system. SIGALI is registered at APP.

Sigali is also integrated as part of the compiler of the language BZR.

VIRTUAL PLANTS Project-Team

4. Software

4.1. V-Plants

Participants: Frédéric Boudon, Christophe Godin [coordinator], Yann Guédon, Christophe Pradal [software architect], Jean-Baptiste Durand, Pascal Ferraro.

Computer algorithms and tools developed by the Virtual Plants team are integrated in a common software suite *V-Plants*, dedicated to the modeling and analysis of plant development at different scales (e.g. cellular tissue, whole plant, stand). The VPlants packages are integrated in OpenAlea as Python components. Several components are distributed and usable through the visual programming environment (see figure 2):

- Multi-scale geometric modeling and visualization. VPlants.PlantGL is a geometric library which provides a set of graphical tools and algorithms for 3D plant modeling at different scales [9]. It is used by many other components to represent the geometry of biological shapes from 3D meristems, plant architectures to plant populations. VPlants.PlantGL is built around a scene-graph data structure and provides efficient algorithms and original geometrical shapes (parametric surfaces, dedicated envelops), that are useful for plant modeling.
- Statistical sequence and tree analysis. Different statistical packages (i.e. VPlants.StafTool, VPlants.SequenceAnalysis, VPlants.TreeMatching and VPlants.TreeAnalysis) are now available in OpenAlea. They provide different models and algorithms for plant architecture analysis and simulation.
- Meristem functioning and development. A first set of components has been created in the last 4-years period to model meristem development in OpenAlea. These tools are currently being integrated thoroughly in the platform so that modelers and biologists can use them, and reuse components easily (for meristem 3D reconstruction, cell tracking, statistical analysis of tissues, creating and manipulating atlases, creating or loading models of growth that can further be run on digitized structures, etc).
- Standard data structure for plants. A new implementation of the MTG formalism for representing and manipulating multiscale plant architecture has been developed. It provides a central data-structure to represent plants in a generic way in OpenAlea. This implementation is available through the packages OpenAlea.MTG. These components make it possible to share plant representations between users and fosters the interoperability of new models.
- Simulation system. The study of plant development requires increasingly powerful modeling tools to help understand and simulate the growth and functioning of plants. In the last decade, the formalism of L-systems has emerged as a major paradigm for modeling plant development. Previous implementations of this formalism were made based on static languages, i.e. languages that require explicit definition of variable types before using them. These languages are often efficient but involve quite a lot of syntactic overhead, thus restricting the flexibility of use for modelers. We developed L-Py an adaptation of L-systems to the Python language (basis of OpenAlea). Thanks to its dynamic typing property, syntax is simple, code execution is made easy and introspection property of the language makes it possible to parameterize and manipulate simply complex models. Independent L-systems can be composed to build-up more complex modular models. MTG structures (that are a common way to represent plants at several scales) can be translated back and forth into L-system data-structure and thus make it easy to reuse in L-systems tools for the analysis of plant architecture based on MTGs. Extensions to integrate multiscale dynamic models are currently being developed in collaboration with P. Prusinkiewicz and his team from the University of Calgary. A paper presenting L-Py [14] has been published to *Frontiers in Technical Advances in Plant Sciences*.

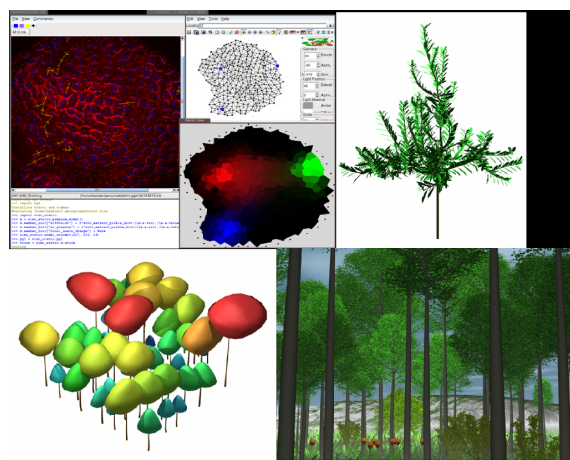


Figure 1. V-Plants components of the OpenAlea platform: simulating plant processes at different scales. Top Left: Reconstruction of a virtual meristem, analysis and simulation of the auxin fluxes inside the meristem. Top Right: Reconstruction of a virtual apple tree from digitized data. Bottom: Simulation of an ecosystem (A beech "Fagus Sylvatica L." trees forest) with a multi level approaches. On the left, explicit representation of the crown volumes that serves as input to generate the detailed representation, on the right.

4.2. OpenAlea

Participants: Frédéric Boudon, Christophe Godin, Yann Guédon, Christophe Pradal [coordinator], Christian Fournier, Julien Coste.

This research theme is supported by the Inria ADT Grant OpenAlea 2.0 and by a Agropolis RTRA Grant named OpenAlea.

OpenAlea [10] is an open source and collaborative software project primarily dedicated to the plant research community. It is designed as a component framework to dynamically glue together models from different plant research labs, and to enhance re-usability of existing models in the plant research community.

The architecture of OpenAlea is based on a component architecture. It provides a set of standard components (OpenAlea.Stdlib), a package manager to dynamically add and retrieve new components, and a port graph data-structure to compose models by interconnecting components into a data-flow.

Visualea provides a visual programming environment, used by scientists to build new model interactively by connecting available components together through an easy-to-use graphical user interface.

In 2012, one major release was done : Openalea 1.0. The following progresses were accomplished:

1. Develop and extend OpenAlea and Visualea:
 - The standard library of components has been extended with useful scientific packages such as a flexible data plotting package (Openalea.Pylab), 2D and 3D image manipulation (Openalea.Image) and linear algebra operations (Openalea.Numpy).
 - Several models of computation have been implemented on the data-flow data-structure to enable discrete event simulation and control flow inside OpenAlea.
2. Animation and diffusion
 - The first OpenAlea Workshop have been held in Montpellier and has been attended by more than 60 scientists. A scientific board has been defined to manage the development and diffusion of OpenAlea. It is composed by 12 scientists.

- Ecomeristem is a crop growth, eco-physiological model that was designed for rice (model plant for cereals) to account for plant morphogenesis and its plasticity depending on genetic potential and sensitivity to the environment (water, temperature, radiation).
- Alinea.Nema is a module used for modeling of nitrogen dynamics between leaves.
- MAppleT is a FSPM model of an apple tree taking into account stochastic models for the topological development, a biomechanical model for branch bending, physiological laws as well as light interception.
- M2A3PC is a generic model to simulate spread of a pathogen on a growing plant like vine/powdery mildew and apple tree/apple scab.

In 2012, a 3D model of gramineous leave has been developed and presented at the PMA conference [28]. This dynamic leaf model is used to simulate different species of annual plants such as rice, wheat and maize.

VISAGES Project-Team

5. Software

5.1. Vistal

Participants: Olivier Commowick, Clément Philipot.

VistaL is a software platform of 3D and 3D+t image analysis allowing the development of generic algorithms used in different contexts (rigid and non-rigid registration, segmentation, statistical modelling, calibration of free-hand 3D ultrasound system and so on, diffusion tensor image processing, tractography). This software platform is composed of generic C++ template classes (Image3D, Image4D, Lattice and so on) and a set of 3D/3D+t image processing libraries. VistaL is a multi-operating system environment (Windows, Linux/Unix...). A web site presenting the project has been developed, precompiled packages and the SDK are now available. VistaL APP registration number is:IDDN.FR.001.200014.S.P.2000.000.21000.

See also the web page <http://vital.gforge.inria.fr>.

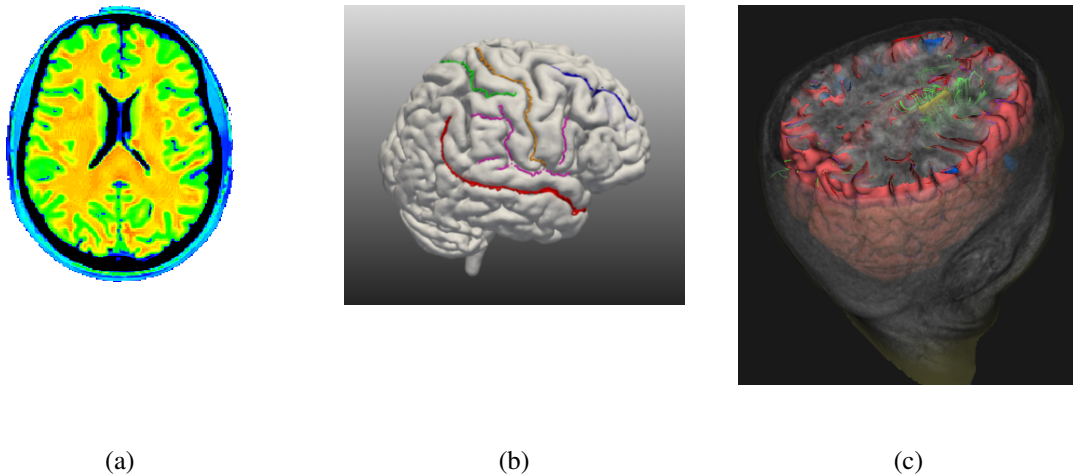


Figure 2. Some ViSTAL results screenshots: a) The ViSTAL Logo, b) ViSTAL Brain surface and sulci modelisation, c) The ROI3D Extraction view

- Keywords: medical image processing, image analysis, registration, segmentation, denoising
- Software benefit: New methodological image processing, some GPU based algorithms, easy to use C++ library
- APP: IDDN.FR.001.200014.S.P.2000.000.21000
- License: Licence Propriétaire
- Type of human computer interaction: C++ API and less complete Python API
- OS/Middleware: Windows, Mac et Linux.
- Required library or software: CMake (GPL) - ITK (BSD) - VTK (BSD) - Boost (BSD) - Libxml++ (LGPL) - CppUnit (LGPL)
- Programming language: C/C++, Python
- Documentation: Documentation Doxygen, documentation utilisateur.

5.2. CLARCS: C++ Library for Automated Registration and Comparison of Surfaces

Participants: Juan Francisco Garamendi, Sylvain Prima.

In collaboration with Benoit Combès (Géosciences Rennes, UMR 6118) and Alexandre Abadie (Inria Saclay Île-de-France), within the 3D-MORPHINE ARC project (<http://3dmorphine.inria.fr>), we conceived and implemented a C++ library (named CLARCS) for the automated analysis and comparison of surfaces. One of the primary goal of this library is to allow the assessment and quantification of morphological differences of free-form surfaces from medical or paleoanthropological data.

- APP: IDDN.FR.001.130002.000.S.P.2011.000.21000
- Programming language: CC++

CLARCS was presented at the MeshMed MICCAI workshop (<http://www2.imm.dtu.dk/projects/MeshMed/2011/index.html>) [49] and is to be distributed through a dedicated website (<http://clarcs.inria.fr>).

We also developed a surface viewer (named 'Surface').

- APP: IDDN.FR.001.110019.000.S.P.2011.000.21000
- Programming language: C++, Python

5.3. SUBANA: SURFACE-BASED NEURONAVIGATION ON ATLAS FOR TMS

Participant: Sylvain Prima.

In collaboration with Charles Garraud (Syneika), Benoit Combès (Géosciences Rennes, UMR 6118) and Pierre Hellier (Technicolor), we developed a software for i) the automated surface reconstruction of the face and skull cap from sparsely acquired points and ii) the automated nonlinear registration of free-form surfaces. The latter step is implemented using the CLARCS library (<http://clarcs.inria.fr>). The primary goal of this software is the surface-based neuronavigation for transcranial magnetic stimulation. The method was presented at the MeshMed MICCAI workshop (<http://www2.imm.dtu.dk/projects/MeshMed/2011/index.html>) [50].

- APP: IDDN.FR.001.440010.000.S.P.2010.000.31230
- Patent: was granted, but the reference number is unknown
- Programming language: C++

5.4. Shanoir

Participants: Guillaume Renard, Justine Guillaumont, Christian Barillot.

Shanoir (Sharing NeuroImaging Resources) is an open source neuroinformatics platform designed to share, archive, search and visualize neuroimaging data. It provides a user-friendly secure web access and offers an intuitive workflow to facilitate the collecting and retrieving of neuroimaging data from multiple sources and a wizard to make the completion of metadata easy. Shanoir comes along many features such as anonymization of data, support for multi-centres clinical studies on subjects or group of subjects.

Shanoir APP registration number is : IDDN.FR.001.520021.000.S.P.2008.000.31230

See also the web page <http://www.shanoir.org>

- Keywords: neuroimaging, ontology, sharing neuroimage
- Software benefit: full featured neuroimaging management system with additionnal web services
- APP: IDDN.FR.001.200014.S.P.2000.000.21000
- License: Licence QPL
- Type of human computer interaction: Online web application, web service (SOAP messages based)
- OS/Middleware: Windows, Mac et Linux.
- Required library or software : Java 1.6, JBoss server, JBoss Seam, JSF, JPA Hibernate, EJB, Richfaces, Faceless, Ajax4JSF, DcmTk, Dcm4chee.
- Programming language: Java
- Documentation : see the website

5.5. QtShanoir

Participants: Olivier Commowick, Guillaume Renard.

QtShanoir is a C++ Qt based library for querying data from a Shanoir server. For those who don't know what is shanoir, see the shanoir website at <http://shanoir.org>. QtShanoir uses the soap based webservice provided by a shanoir server to get and display studies, patients, data with their associated metadata. In QtShanoir, you will find a set of Qt widgets (inherited from a QWidget object) that you can embed in your Qt application.

An APP registration is in progress and the library has been release in october under the LGPL license. See <http://qtshanoir.gforge.inria.fr>.

- Keywords : medical imaging, dicom
- Software benefit: offers a great solution to query a Shanoir server. Can be easily re used in larger Qt applications
- License: no defined licence for the moment
- Type of human computer interaction: C++ library
- OS/Middelware: Linux, Windows and Mac
- Required library or software : Qt
- Programming language: C++
- Documentation : <http://qtshanoir.gforge.inria.fr>

5.6. AutoMRI

Participants: Camille Maumet, Isabelle Corouge, Elise Banner.

AutoMRI is an SPM-based set of tools to study structural and functional MRI data. This software is currently made up of 9 modules : autofMRI, autoVBM, automorpho, autoASL, autoFASL, autoROI, autoasltemplate, autofmricontrario and autoNCEMRA. AutofMRI produces statistical maps of activations and deactivations at the group or the subject level based on functional MRI data. It can deal with block or event-related designs and is highly configurable in order to fit to a wide range of needs. autoVBM performs between-group voxel-based morphometric analysis in order to outline regions of grey (or white) matter volume reduction and increase. To further study a morphometric or a functional analysis, regions of interest analysis can be performed with autoROI. This module also provides the user with laterality indexes. Automorpho performs one-versus-many group analysis on anatomical data in order to outline pathological dysplasia or heterotopia. AutoFASL (collaboration with Rémi Dubujet) produces statistical maps of activations and deactivations at the group or the subject level based on functional Arterial Spin Labeling data. AutoASL performs between-group voxel-based morphometric analysis in order to outline regions of reduced (or increased) perfusion. Autoasltemplate focus on patient-specific detection of perfusion abnormalities with a standard massively univariate General Linear Model or with an a contrario approach. Autofmricontrario provides an alternative to autofmri to produce statistical maps of activations and deactivations at the subject level using an a contrario approach. autoNCEMRA enables automatic processing of 4D MRA data to remove unwanted signal from the skull, using a mask based on 3D T1w segmentation of grey matter, white matter and CSF. Thus, denoised maximum intensity projections in axial, coronal and sagittal planes can be calculated to enable accurate assessment of hemodynamic patterns, from arterial input to venous drainage (in particular in patients presenting arteriovenous malformations).

- Keywords : fMRI, MRI, ASL, fASL, SPM, automation
- Software benefit: Automatic MRI data analysis based on SPM. Once the parameters are set, the analysis is performed without human interaction.
- APP: Part in IDDN.FR.001.130017.000.S.A.2012.000.31230
- Type of human computer interaction: Matlab function (script, no GUI)
- OS/Middlewares: Linux/Windows
- Required library or software : Matlab, SPM, SPM toolboxes : Marsbar, LI-toolbox, NS
- Programming language: Matlab
- Documentation : Available

5.7. Medinria

Participants: René-Paul Debroize, Clément Philipot, Guillaume Pasquier, Olivier Commowick.

Medinria is a national Inria project shared between 4 Inria teams (Asclepios, Athena, Parietal and Visages). It aims at creating an easily extensible platform for the distribution of research algorithms developed at Inria for medical image processing. This project has been funded by the D2T (ADT MedInria-NT) in 2010 and renewed for two years in 2012. The Visages team leads this Inria national project and participates in the development of the common core architecture and features of the software as well as in the development of specific plugins for the team's algorithm. Medinria 2.0.1 has been released in April 2012 for the main distribution platforms. Development of an SDK and of a new versions is underway and should be released in June 2013.

See also the web page <http://med.inria.fr>

- Keywords: medical imaging, diffusion imaging, registration, filtering, user-friendly interface
- Software benefit: user-friendly interface to cutting-edge research tools for research clinicians. Straightforward to add functionalities through plugins.
- License: core: BSD, plugins: choice of each team.
- Type of human computer interaction: Qt-based GUI
- OS/Middleware: Windows, Mac et Linux.
- Required library or software : Qt, DTK, ITK, VTK.
- Programming language: C++

5.8. USGraphCut

Participant: Christian Barillot.

This software has been developed in collaboration with Jan Petr and Alexandre Krupa during the ANR USComp project. It concerns the segmentation of echographic data by using the graph cut algorithm. It allows the segmentation and the tracking of evolving objects in 2D/3G echographic data in real time thanks to a specific CUDA framework.

5.9. CtrlQ - MR Quality Assurance

Participants: René-Paul Debroize, Isabelle Corouge, Elise Banner.

As part of the monitoring of the 3Tesla MR equipment, a quality control consistent with the one recommended by the American College Of Radiology (ACR) is performed weekly. As part of its MRI accreditation program, the ACR standardized a procedure for monitoring quality consisting of a series of measurements performed on a standardized imaging protocol on a test object with known geometry. A robust and intuitive software was developed, with a graphical interface, to ensure the automation of the measurements necessary to the control. The application was developed in C++ using the Qt, ITK, and DCMTK libraries.

5.10. SimuBloch

Participants: Fang Cao, Olivier Commowick, Elise Banner, Christian Barillot.

We developed a simulator package *SimuBloch*, which is made for a fast simulation of image sequences based on Bloch equations, which can be run directly from VIP Portal: <http://vip.creatis.insa-lyon.fr>. The current version is v0.3. The simulator allows to construct 6 different MR pulse sequences:

1. *SimuBlochSE*: Simulation of spin echo sequences.
2. *SimuBlochGRE*: Simulation of gradient echo sequences.
3. *SimuBlochIR-SE*: Simulation of inversion recovery spin echo sequences.
4. *SimuBlochIR-GRE*: Simulation of inversion recovery gradient echo sequences.
5. *SimuBlochSP-GRE*: Simulation of spoiled gradient echo sequences.
6. *SimuBlochCoherentGRE*: Simulation of coherent gradient echo sequences.

VR4I Team

5. Software

5.1. OpenMASK: Open-Source platform for Virtual Reality

Participants: Alain Chauffaut [contact], Ronan Gagne [contact], Georges Dumont, Thierry Duval, Marwan Badawi.

OPENMASK (Open Modular Animation and Simulation Kit) is a federative platform for research developments in the VR4i team. Technology transfer is a significant goal of our team so this platform is available as OpenSource software (<http://www.openmask.org>). OpenMASK is a C++ software platform for the development and execution of modular applications in the fields of animation, simulation and virtual reality. The main unit of modularity is the simulated object (OSO) which can be viewed as frequential or reactive motors. It can be used to describe the behavior or motion control of a virtual object as well as input devices control like haptic interfaces. Two OSO communicate with synchronous data flows or with asynchronous events. OpenMASK is well suited to develop applications in our new immersive room as ergonomics studies, including immersion, interaction, physic and haptic.

5.2. GVT : Generic Virtual Training

Participants: Bruno Arnaldi, Valérie Gouranton [contact], Florian Nouviale, Thomas Lopez, Andrés Saraos Luna.

The aim of GVT software is to offer personalized VR training sessions for industrial equipments. The most important features are the human and equipment security in the VR training (in opposition to the real training), the optimization of the learning process, the creation of dedicated scenarios, multiple hardware configurations: laptop computer, immersion room, distribution on network, etc.

The actual kernel of GVT platform is divided into two main elements that rely on innovative models we have proposed: LORA and STORM models.

- A Behavior Engine. The virtual world is composed of behavioral objects modelled with STORM (Simulation and Training Object-Relation Model).
- A Scenario Engine. This engine is used to determine the next steps of the procedure for a trainee, and its state evolves as the trainee achieves actions. The scenario is written in the LORA language (Language for Object-Relation Application).

A commercialized version of GVT, which includes a pedagogical engine developed in CERV laboratory, proposes training on individual procedures. A prototype is also available that enables users to train on collaborative procedures with one another or with virtual humans.

In the ANR Corvette 7.1.3 and in the FUI SIFORAS 7.1.1, new features of GVT Software are proposed.

5.3. OpenViBE Software

Participants: Anatole Lécuyer [contact], Laurent Bonnet, Jozef Legény, Yann Renard.

OpenViBE is a free and open-source software devoted to the design, test and use of **Brain-Computer Interfaces**.

The OpenViBE platform consists of a set of software modules that can be integrated easily and efficiently to design BCI applications. Key features of the platform are its modularity, its high-performance, its portability, its multiple-users facilities and its connection with high-end/VR displays. The "designer" of the platform enables to build complete scenarios based on existing software modules using a dedicated graphical language and a simple Graphical User Interface (GUI).

This software is available on the Inria Forge under the terms of the LGPL-V2 licence, and it was officially released in June 2009. Since then, the OpenViBE software has already been downloaded more than 300 times, and it is used by numerous entities worldwide.

Our first international tutorial about OpenViBE was held at the International BCI Meeting in June 2010 (Monterey, US), with around 30 participants.

More information, downloads, tutorials, documentation, videos are available on OpenViBE website : <http://openvibe.inria.fr>

WAM Project-Team

5. Software

5.1. XML Reasoning Solver

Participants: Pierre Genevès, Nabil Layaïda, Nils Gesbert, Manh-Toan Nguyen.

The **XML Reasoning Solver** is a tool for the static analysis of XPath queries and XML schemas based on the latest theoretical advances. It allows automated verification of properties that are expressed as logical formulas over trees. A logical formula may for instance express structural constraints or navigation properties (like e.g. path existence and node selection) in finite trees.

The tool can solve many fundamental XML problems such as satisfiability of XPath expressions in the presence of XML schemas, containment and equivalence of XPath expressions, and many other problems that can be formulated with XPath expressions and schemas (DTDs, XML Schemas, Relax-NG).

The system is implemented in Java and uses symbolic techniques (binary decision diagrams) in order to enhance its performance. It is capable of comparing path expressions in the presence of real-world DTDs (such as the W3C SMIL and XHTML language recommendations, for instance). The cost ranges from several milliseconds for comparison of XPath queries without tree types, to several seconds for queries under very large, heavily recursive, type constraints, such as the XHTML DTD. These measurements shed light for the first time on the cost of solving static analysis problems in practice. Furthermore, the analyzer generates XML counter-examples that allow program defects to be reproduced independently from the analyzer.

5.1.1. Extensions for CSS

We have introduced the first system capable of statically verifying properties of a given cascading style sheet (CSS) over the whole set of documents to which this stylesheet applies [5]. The system is composed of a set of parsers for reading the CSS and schema files (XML Schema, Relax NG, or DTD) together with a text file corresponding to problem description as a logical formula. We have developed a compiler that translates CSS files into their logical representations. Then, the solver takes the overall problem formulation and checks it for satisfiability.

5.1.2. XQuery IDE

We have started the development of an XQuery IDE with a web interface. This prototype integrates static analyses performed by the solver inside a development environment suited for XQuery programmers.

5.2. Timesheets Library

Participants: Nicolas Hairon, Cécile Roisin.

The goal of the **Timesheets library** is to synchronize HTML5 content using declarative synchronization languages defined by W3C standards (namely **SMIL Timing and Synchronization** and **SMIL Timesheets**).

With the raise of HTML5 which natively supports continuous content (audio, video) there is a dramatic need for handling synchronization, animation and user interaction in an efficient and homogeneous way. As web browsers do not support SMIL, except for SVG Animation (which is based on the SMIL BasicAnimation module), multimedia web authoring remains difficult and relies on code-based, non-standard solutions.

Therefore we are developing a generic, cross-browser JavaScript implementation for scheduling the dynamic behavior of HTML5 content that can be described with declarative SMIL markup. Using a declarative language makes sense for the most common tasks, which currently require JavaScript programming:

- it is much easier for web authors and for web authoring tool developers;
- it is a much better way to achieve good accessibility and indexability;
- it is easier to maintain, since no specific JavaScript code is used.

This open source library is now deployed and used by external users. As far as we know, ENS Lyon was the first user: its site html5.ens-lyon contains several dozens of scientific conferences where the video capture of each conference is synchronized with the slides, a structured timeline and a table of contents. This web site was demonstrated in May at the WWW 2012 conference. University of Evry makes also a important use of the Timesheets library as a tool for teaching multimedia concepts at master level.

5.3. Mobile Audio Language

Participants: Yohan Lasorsa, Jacques Lemordant.

5.3.1. MAUDL library

The MAUDL library (Mobile AUDio Language) is an evolution of the ARIA library whose primary target was games on mobile devices.

Augmented Reality Audio applications use sound objects to create a soundscape. A sound object is a time structure of audio chunks whose duration is on the time scale of 100 ms to several seconds. These sound objects have heterogeneous and time-varying properties. In order to describe Interactive Audio (IA) contents, we created MAUDL, an XML language inspired by iXMF that is well adapted to the design of dynamic soundtracks for navigation systems.

MAUDL prevents audio information overwhelming through categorization at the declarative level and the use of priority queues at the execution level. This allows to take account of speed when walking, and rapid hand gestures when interrogating the environment for example. MAUDL can be used as an authoring time interchange file format for interactive mobile applications or as a runtime file format that is actually loaded through the web and played directly in the mobile. MAUDL is a cue-oriented interactive audio system, audio services being requested using named events and the systems response to each event being determined by the audio artist. The current version of the API supports iOS and further support for other mobile platforms (Android) is planned.

5.3.2. 3D Audio Pointer

A virtual 3D audio pointer provides an intuitive guide to the user of a mobile navigation application, reducing the need for cognitive work when compared to vocal instructions. We have built such a pointer using the MAUDL language. It gives the user the azimuth using HRTF spatialized audio cues, with additional hints taking the form of variations in the sound used. It allows to superpose other kinds of audio contents, such as voice while the pointer is active, to indicate distance for example. This audio object is suitable for different sorts of navigation systems, such as POIs browsers, self-guided audio tours, or applications for following predefined routes.

5.4. Mixed Reality Browser (MRB)

Participants: Yohan Lasorsa, Jacques Lemordant, David Liodenot, Thibaud Michel, Mathieu Razafimahazo.

The concept of Mixed Reality comes from the fact that the real/virtual dichotomy is not sharp, but interpolatively smooth over a virtuality continuum. Idealized notions of reality and virtuality can be thought of as endpoints on a continuum, an instance of the former approach corresponding for example to a see-through display with natural sounds, an instance of the latter to texture-mapped image-based rendering (panoramas) with synthetic sound objects.

Augmented Reality (AR) mode refers to all cases in which the auditory or visual display of an otherwise real environment is augmented by means of virtual sound or graphic objects. The converse case on the virtuality continuum is Augmented Virtuality (AV), where a virtual world, one that is generated primarily by computer, like with synthetic 3D graphic or synthetic panoramic, is being augmented with the audio-visual content of points of interest (POIs).

The **Mixed Reality Browser** (MRB) is a geolocalized web browser running on mobile devices. It uses standard and open XML formats for content authoring (HTML5, OSM and MAUDL) to allow anyone to create an augmented or virtual reality city tour that can be used with this application.

The introduction of mobile augmented reality browsers has forced a rethink on what kind of reality should be offered. Mobility induces a need for telepresence and simulation to free the user or the developer of the necessity to go every time in the real world. Mobility is the main reason behind the concept of the Mixed Reality Browser. By its intrinsic characteristics, MRB supports advance MR applications like mobile remote maintenance and assisted navigation.

5.5. Interactive eXtensible Engine (IXE)

Participants: Yohan Lasorsa, Jacques Lemordant, David Liodenot, Thibaud Michel, Mathieu Razafimahazo.

GPS navigation systems when used in an urban environment are limited in precision and can only give instructions at the level of the street and not of the sidewalk. GPS is limited to outdoor navigation and requires some delicate transitioning system when switching to another positioning system to perform indoor navigation.

IXE is an open source urban pedestrian navigation system based on Inertial Measurement Unit (IMU) and running on mobile phones with onboard geographic data and a routing engine. With IXE, the distinction between indoor and outdoor is blurred as an IMU-based location engine can run indoor and outdoor. IXE allows augmented reality queries on customized embedded geographical data. Queries on route nodes or POIs, on ways and relations are predefined for efficiency and quality of information.

Following the web paradigm, IXE is a browser for XML documents describing navigation networks: by using the micro-format concept, one can define inside OpenStreetMap a complex format for pedestrian navigation networks allowing navigation at the level of sidewalks or corridors. The big advantage of doing this instead of defining new XML languages is that we can use the standard OpenStreetMap editor JOSM to create navigation networks in a short amount of time.

The purpose of the IXE browser is to read these OSM documents and to generate from them visible or audible navigation information. IXE works on any mobile phone running under iOS or Android. Its heart is composed of three engines, one for dead-reckoning navigation, one for interactive audio and the last one for Augmented Reality visual information, allowing quick reconfiguration for extremely varied applications.

IXE can be used for accessible navigation allowing independent living for people with disabilities.

WILLOW Project-Team

5. Software

5.1. Patch-based Multi-view Stereo Software (PMVS)

PMVS is a multi-view stereo software that takes a set of images and camera parameters, then reconstructs 3D structure of an object or a scene visible in the images. Only rigid structure is reconstructed. The software outputs a set of oriented points instead of a polygonal (or a mesh) model, where both the 3D coordinate and the surface normal are estimated at each oriented point. The software and its documentation are available at <http://www.di.ens.fr/pmvs/>. The software is distributed under GPL. A US patent corresponding to this software “Match, Expand, and Filter Technique for Multi-View Stereopsis” was issued on December 11, 2012 and assigned patent number 8,331,615.

5.2. SParse Modeling Software (SPAMS)

SPAMS v2.3 was released as open-source software in May 2012 (v1.0 was released in September 2009 and v2.0 in November 2010). It is an optimization toolbox implementing algorithms to address various machine learning and signal processing problems involving

- Dictionary learning and matrix factorization (NMF, sparse PCA, ...)
- Solving sparse decomposition problems with LARS, coordinate descent, OMP, SOMP, proximal methods
- Solving structured sparse decomposition problems (ℓ_1/ℓ_2 , ℓ_1/ℓ_∞ , sparse group lasso, tree-structured regularization, structured sparsity with overlapping groups,...).

The software and its documentation are available at <http://www.di.ens.fr/willow/SPAMS/>.

5.3. Local dense and sparse space-time features

This is a package with Linux binaries implementing extraction of local space-time features in video. We are preparing a new release of the code implementing highly-efficient video descriptors described in Section 6.4.5. Previous version of the package was released in January 2011. The code supports feature extraction at Harris3D points, on a dense space-time grid as well as at user-supplied space-time locations. The package is publicly available at <http://www.di.ens.fr/~laptev/download/stip-2.0-linux.zip>.

5.4. Automatic Mining of Visual Architectural Elements

The code on automatic mining of visual architectural elements (v4.3) described in (Doersch *et al.* SIGGRAPH [6]) has been publicly released online in December 2012 (earlier version v3.0 was released in September 2012) at http://graphics.cs.cmu.edu/projects/whatMakesParis/paris_sigg_release.tar.gz.

5.5. Automatic Alignment of Paintings

The code for automatic alignment of paintings to a 3D model (Russell *et al.* 2011) was made publicly available in October 2012 at <http://www.di.ens.fr/willow/research/paintingalignment/index.html>.

5.6. Multi-Class Image Cosegmentation

This is a package of Matlab code implementing multi-class cosegmentation (Joulin *et al.* CVPR 2012 [13] and unsupervised discriminative clustering for image co-segmenting (Joulin *et al.* CVPR 2010) and (Joulin *et al.* NIPS 2010). The aim is to segment a given set of images containing objects from the same category, simultaneously and without prior information. The package was last updated in September 2012 and is available at <http://www.di.ens.fr/~joulin/code/DALCIM.zip>.

5.7. Convex Relaxation of Weakly Supervised Models

This is a package of Matlab code implementing a general multi-class approach to weakly supervised classification described in (Joulin and Bach ICML 2012 [12]). The goal is to avoid local minima typically occurring expectation-maximization like algorithms and to optimize a cost function based on a convex relaxation of the soft-max loss. The package was last updated in September 2012 and is available at http://www.di.ens.fr/~joulin/code/ICML12_Joulin.zip.

5.8. Non-uniform Deblurring for Shaken Images

An online demo of non-uniform deblurring for shaken images implementing the algorithm described in [8] and (Whyte et al. CPCV 2011) was made available in 2012 at <http://www.di.ens.fr/willow/research/deblurring/>. The demo takes as an input an image uploaded by the user, automatically estimates the blur, and outputs the deblurred image.

WIMMICS Team

5. Software

5.1. Corese

Participants: Olivier Corby [correspondant], Fabien Gandon.

Corese ² (COnceptual REsource Search Engine) is a Semantic Web Factory. It enables users to load and process RDFS schemas, RDF metadata and to query the base of annotations thus created, by using the SPARQL Query Language.

Corese implements RDF, RDFS and SPARQL 1.1 Query Language & Update. Furthermore, Corese query language integrates original features such as approximate search, SQL or XPath. Approximate search consists of searching the best approximate answers to a query according to the ontology types. Corese also integrates a SPARQL-based Rule Language for RDF.

Corese is a Semantic Web Factory that enables us to design and develop Semantic Web applications; it is available for download. In the past, Corese benefited from Inria software development support (ADT) with two software engineers. Corese is registered at the APP and in 2007 we decided to distribute it as open source software under license CeCILL-C.

Corese is used and has been used in more than 50 applications, 24 PhD Thesis and is used for education by several institutions. It has been used as a Semantic Web Factory in such projects as Ontorule, Palette, SevenPro and SeaLife european projects, in e-WOK Hub, Neurolog, ISICIL and Kolflow ANR projects, BioMarker and KmP projects, Semantic Web Import Plugin for Gephi visualization and ECCO ontology editor. The work on Corese was published in [95], [96], [97], [94], [1], [5], [3], [2], [4].

This year we completed the KGRAM SPARQL 1.1 Query & Update interpreter.

Web page: <http://wimmics.inria.fr/corese>

5.2. Semantic Web Import Plugin for Gephi visualization

Participants: Erwan Demairy, Fabien Gandon, Olivier Corby.

The SemanticWebImport ³ plugin is intended to allow the import of semantic data into Gephi open graph visualization platform. Gephi is an interactive visualization and exploration platform for all kinds of networks and complex systems, dynamic and hierarchical graphs. The imported data are obtained by processing a SPARQL request on the semantic data. The data can be accessed following three manners:

1. by accessing local RDF & RDFS files and using the embedded Corese engine to apply the SPARQL request;
2. by accessing a remote REST SPARQL endpoint. In that case, the SPARQL request is applied remotely and the graph is built locally by analyzing the result sent by the endpoint;
3. by accessing a remote SOAP SPARQL endpoint. As for the REST endpoint, the resulting graph is built from the result returned by the endpoint.

The software is released under version 1.0. It has received a development grant (ADT) from Inria.

Web pages:

<http://wiki.gephi.org/index.php/SemanticWebImport>

<https://gforge.inria.fr/projects/segviz-public>

²<http://wimmics.inria.fr/corese>

³<http://wiki.gephi.org/index.php/SemanticWebImport>

5.3. ISICIL

Participants: Nicolas Delaforge, Fabien Gandon [resp.].

The ISICIL software platform is made of several software components:

- XUL (XML-based User interface Language) extensions for the Firefox browser to assist the technology watch and business intelligence tasks by collecting relevant metadata according to the navigation context of the user.
- An application server based on Tomcat publishes services using the REST protocol to process requests of the users' applications and in particular the navigation extensions.

This architecture is summarized in Figure 1. Its major interest lies in the flexibility introduced by the loose coupling between REST services and navigators extensions or other applications.

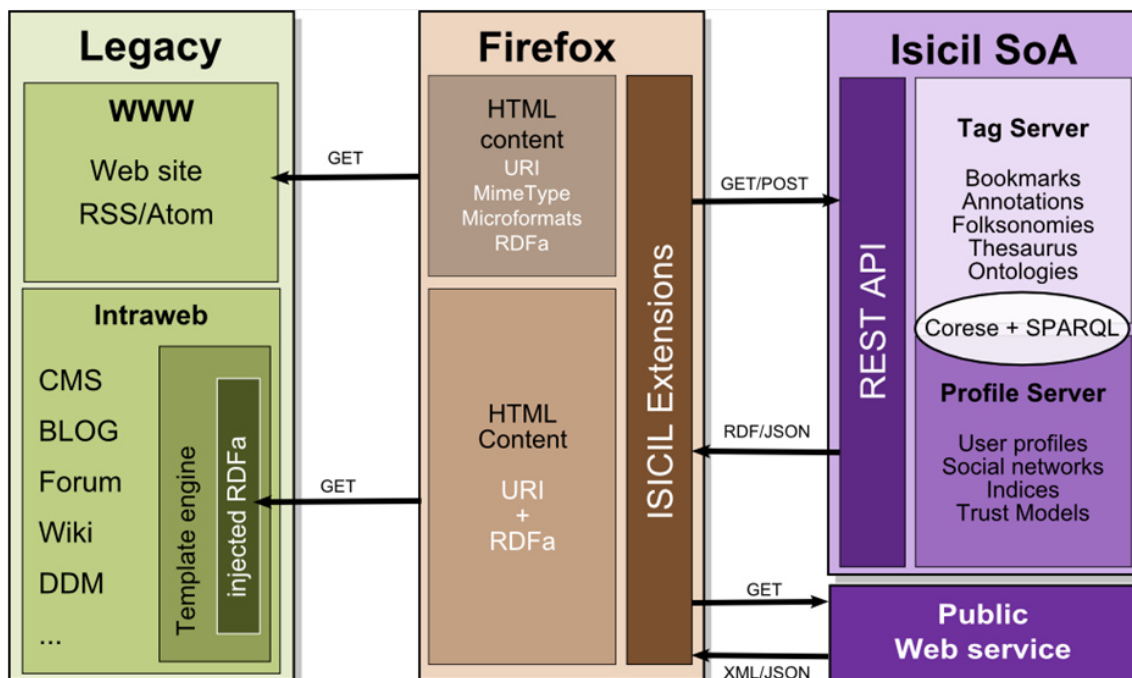


Figure 1. ISICIL Platform Architecture

In the context of the ISICIL ANR project, we have developed a Semantic Web server which provides core services to manage simple tagging of resources (internal or from the Web) and to assist the semantic enrichment of the folksonomy of our communities of users. This server's implementation is based on the ISICIL main framework. The tagging model combines already existing ontologies such as SIOC⁴, SCOT, and Newman's Tag Ontology⁵ as shown in Figure 2. SRTag, the model of folksonomy enrichment, is based on a named graph mechanism in order to maintain diverging statements made between tags using SKOS (for thesaurus like relation between tags) or SCOT (for spelling variant relations), and is shown in Figure 3.

⁴<http://sioc-project.org>

⁵<http://www.holygoat.co.uk/owl/redwood/0.1/tags>

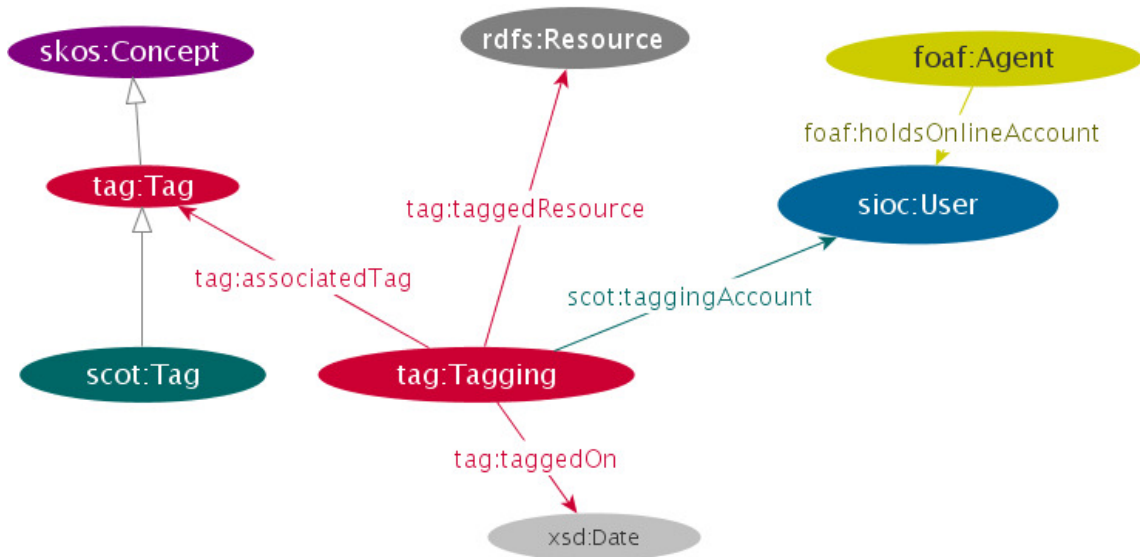


Figure 2. Model of tagging used in the Semantic Tag Server

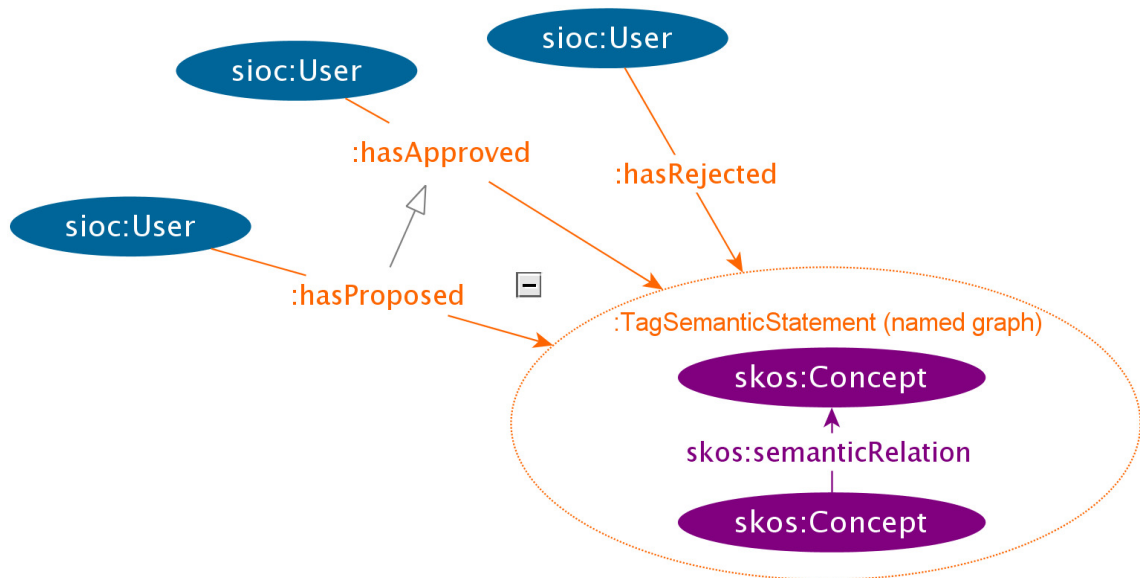


Figure 3. Folksonomy enrichment model

The functionalities of this server can be divided into three categories:

1. Tagging : creating a tag; get tag suggestions based on the input characters; create a tagging, i.e. a link between a resource, a user, and a tag.
2. Computing: an external library (exported as a java jar file) has been developed to perform computations on the tagging data. Two types of computations have been implemented:
 1. Spelling Variant detection.
 2. Related tag detection based on the computation of the similarity between tags [101].
3. Managing Semantic relations between tags: get semantically related tags, reject or propose new semantic relations.

We developed a Firefox extension to help users navigate within a folksonomy and organize semantically the tags. The main idea behind this tool is to combine organization tasks with everyday tasks in the least intrusive way, that is to say, without forcing the user in any way, and by providing a user friendly graphical interface. This tool, developed using the XUL framework ⁶, is supported by the SRTag model and the Semantic Tag Server. Users are provided with search bar for navigating the folksonomy. When available, other tags are suggested and ordered according to their semantic relation with the searched tag (broader, narrower, related, spelling variant). Each suggestion can be either :

- clicked to search content tagged with this tag;
- rejected by clicking a checkbox;
- modified thanks to a drag-and-drop mechanism where a tag can be dropped in another category of semantic relation.

Web page: <https://gforge.inria.fr/projects/isicil/>

⁶<http://developer.mozilla.org/en/XUL>

ZENITH Project-Team

5. Software

5.1. WebSmatch (Web Schema Matching)

Participants: Zohra Bellahsène, Emmanuel Castanier, Rémi Coletta, Duy Hoa Ngo, Patrick Valduriez [contact].

URL: <http://websmatch.gforge.inria.fr/>

In the context of the Action de Développement Technologique (ADT) started in october 2010, WebSmatch is a flexible, open environment for discovering and matching complex schemas from many heterogeneous data sources over the Web. It provides three basic functions: (1) metadata extraction from data sources; (2) schema matching (both 2-way and n-way schema matching), (3) schema clustering to group similar schemas together. WebSmatch is being delivered through Web services, to be used directly by data integrators or other tools, with RIA clients. Implemented in Java, delivered as Open Source Software (under LGPL) and protected by a deposit at APP (Agence de Protection des Programmes). WebSmatch is being used by Datapublica and CIRAD to integrate public data sources.

5.2. YAM++ ((not) Yet Another Matcher)

Participants: Zohra Bellahsène [contact], Duy Hoa Ngo, Konstantin Todorov.

URL: <http://www2.lirmm.fr/~dngo/>

YAM++ is a tool for discovering semantic correspondences between ontologies. YAM++ supports several matching strategies: machine learning; generic methods when learning data are not available; discovering alignment of ontologies represented in different languages. Furthermore, since this year YAM++ is able to deal with large scale ontology matching.

5.3. SON (Shared-data Overlay Network)

Participants: Ayoub Ait Lahcen, Fady Draidi, Esther Pacitti, Didier Parigot [contact], Patrick Valduriez, Guillaume Verger.

URL: <http://www-sop.inria.fr/teams/zenith/SON>

SON is an open source development platform for P2P networks using web services, JXTA and OSGi. SON combines three powerful paradigms: components, SOA and P2P. Components communicate by asynchronous message passing to provide weak coupling between system entities. To scale up and ease deployment, we rely on a decentralized organization based on a DHT for publishing and discovering services or data. In terms of communication, the infrastructure is based on JXTA virtual communication pipes, a technology that has been extensively used within the Grid community. Using SON, the development of a P2P application is done through the design and implementation of a set of components. Each component includes a technical code that provides the component services and a code component that provides the component logic (in Java). The complex aspects of asynchronous distributed programming (technical code) are separated from code components and automatically generated from an abstract description of services (provided or required) for each component by the component generator.

5.4. P2Prec (P2P recommendation service)

Participants: Fady Draidi, Esther Pacitti [contact], Didier Parigot, Guillaume Verger.

URL: <http://p2prec.gforge.inria.fr>

P2Prec is a recommendation service for P2P content sharing systems that exploits users social data. To manage users social data, we rely on Friend-Of-A-Friend (FOAF) descriptions. P2Prec has a hybrid P2P architecture to work on top of any P2P content sharing system. It combines efficient DHT indexing to manage the users FOAF files with gossip robustness to disseminate the topics of expertise between friends. P2Prec is implemented in java using the Data-Shared Overlay Network (SON) infrastructure which is the basis for the ANR DataRing project.

5.5. ProbDB (Probabilistic Database)

Participants: Reza Akbarinia [contact], Patrick Valduriez, Guillaume Verger.

URL: <http://probdb.gforge.inria.fr>

ProbDB is a probabilistic data management system to manage uncertain data on top of relational DBMSs. One of the main features of the prototype is its portability; that means with a minimum effort it can be implemented over any DBMS. In ProbDB, we take advantage of the functionalities provided by almost all DBMSs, particularly the query processing functions. It is implemented in Java on top of PostgreSQL.

5.6. Pl@ntNet-Identify

Participants: Mathias Chouet, Hervé Goëau, Alexis Joly [contact].

URL: <http://identify.plantnet-project.org>

Pl@ntNet-Identify is a web application dedicated to the image-based identification of plants. It has been developed jointly by ZENITH, the AMAP UMR team (CIRAD) and the Inria team IMEDIA. It allows submitting one or several query pictures of a plant and browse the matching species in a large collection of social image data, i.e. plant images collected by the members of a social network. It also allows users to enrich the knowledge of the application by uploading their own pictures in the reference collection. Nowadays, the dataset includes more than 17K images posted by about 100 members of Telabotanica¹ social network. In 2012, about 5000 identification sessions have been recorded. The client side of the application is implemented in Javascript whereas the server side (visual search engine) is mostly implemented in C++.

5.7. Pl@ntNet-DataManager

Participants: Mathias Chouet [contact], Alexis Joly.

Pl@ntNet-DataManager is a software dedicated to managing and sharing distributed heterogeneous botanical data. It is developed jointly by ZENITH, the AMAP UMR team (CIRAD) and Telabotanica non profit organization. It allows scientists to define data structures dedicated to their own datasets, and share parts of their structures and data with collaborators in a decentralized way. Pl@ntNet DataManager offers innovative features like partial or complete P2P synchronization between distant databases (master-master), and a user friendly data structure editor. It also provides full text search, querying, CSV import/export, SQL export, image management, and geolocation. DataManager is built on NoSQL technology (CouchDB database), Javascript (Node.js), HTML5 and CSS3, and may be deployed on a server or run on a local machine (standalone version for Linux, Windows, Mac). It is being used by researchers and engineers of the Pl@ntNet Project (CIRAD, INRA, Inria, IRD, Tela-Botanica) to manage taxonomical referentials, herbarium data and geolocated plant observations.

5.8. SnoopIm

Participants: Julien Champ [contact], Alexis Joly, Pierre Letessier.

URL: <http://otmedia.lirmm.fr/OTmedia/>

¹<http://www.tela-botanica.org/>

SnoopIm is a content-based search engine allowing to discover and retrieve small visual patterns or objects in large collections of pictures (such as logos on clothes, road signs in the background, paintings on walls, etc.) and to derive statistics from them (frequency, visual cover, size variations, etc.). Query objects to be searched can be either selected from the indexed collection of photos, or selected from an external picture (by simply providing its URL). The web application allows online search of multiple users and has a cache feature to speed-up the processing of seen queries. It is implemented in Javascript on top of a C++ library developed in collaboration with INA ². The software is used at INA by archivists and sociologists in the context of the Transmedia Observatory project.

²<http://www.ina-sup.com/>