



RESEARCH CENTER
Bordeaux - Sud-Ouest

FIELD

Activity Report 2014

Section Software

Edition: 2015-03-24

1. BACCHUS Team	4
2. CAGIRE Team	8
3. CARMEN Team	10
4. CQFD Project-Team	12
5. FLOWERS Project-Team	13
6. GEOSTAT Project-Team	29
7. HIEPACS Project-Team	30
8. LFANT Project-Team	35
9. MAGIQUE-3D Project-Team	38
10. MAGNOME Project-Team	40
11. MANAO Project-Team	42
12. MC2 Team	45
13. MNEMOSYNE Project-Team	48
14. PHOENIX Project-Team	50
15. POTIOC Project-Team	58
16. REALOPT Project-Team	60
17. RUNTIME Team	61
18. SISTM Team	66

BACCHUS Team

4. New Software and Platforms

4.1. AeroSol

Participants: Simon Delmas [Cagire], Damien Genet [Bacchus], Maxime Mogé [Cagire], Yann Moguen [Cagire], Vincent Perrier [Corresponding member], Mario Ricchiuto [Bacchus].

The AeroSol software is jointly developed in teams Bacchus and Cagire. It is a high order finite element library written in C++. The code has been designed so as to allow for efficient computations, with continuous and discontinuous finite elements methods on hybrid and possibly curvilinear meshes. The work of the team Bacchus is focused on continuous finite elements methods, while the team Cagire is focused on discontinuous Galerkin methods. However, everything is done for sharing the largest part of code we can. More precisely, classes concerning IO, finite elements, quadrature, geometry, time iteration, linear solver, models and interface with PaMPA are used by both of the teams. This modularity is achieved by mean of template abstraction for keeping good performances. The distribution of the unknowns is made with the software PaMPA, developed within the team Bacchus and the team Castor.

The work performed this year in the BACCHUSteam has focused on the experimentation of parallelisation solutions on heterogenous machines, and in particular on the study of efficient solution for shared memory parallelism. In particular, in the framework of the PhD of D. Genet, the coupling with runtime systems such as StarPU (Inria team Runtime) and DAGuE (University of Tennessee), has been compared to a more classical OpenMP implementation. This initial work, done for scalar problems, will be now extended to systems of equations.

4.2. COCA

Participants: Mario Ricchiuto [corresponding member], Gregory Perrot.

COCA(CodeOxydationCompositesAutocicatrisants) is a Fortran 90 code for the simulation of the oxidation process in self-healing composites COCA solves the discrete finite element equations relative to the oxidation (chemistry) and flow (potential) models. Time integration is performed with an implicit approach (Backward Euler or second order backward differencing). The linear algebraic systems arising in the discretization are solved with the MUMPS library. Physical outputs of interest can be obtained to use COCA as a numerical closure for continuous mechanics solvers in order to perform numerical strain tests for self-healing composites.

4.3. RealfluidS

Participants: Pietro Marco Congedo, Héloïse Beaugendre [corresponding member], Cécile Dobrzynski, Quentin Viville, Leo Nouveau.

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. The uncertainty quantification library RobUQ has been coupled to the software. A partitioning tool exists in the package, which uses Scotch. Recently, the code has been developed for taking into account real-gas effects, in order to use arbitrarily complex equations of state. Further developments concerning multiphase effects are under way.

4.4. 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/was used in gamma3 team, at EPFL (maths department), Dassault Aviation, Lemma (a french SME), Renault 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/>).

Version 5.0 of MMG3D allows the modification of the surface triangulation based on cubic Bezier patches. A. Froehly, ingenieur in the FUI Rodin, is working on this new version.

More details can be found on <http://www.math.u-bordeaux1.fr/~doj/logiciels/mmg3d.php>.

4.5. ORComp

Participants: Pietro Marco Congedo [Corresponding member], 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 RealfluidScore 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.

4.6. sDEM

Participants: Pietro Marco Congedo [Corresponding member], Maria-Giovanna Rodio.

The sDEM platform is a simulation tool permitting to simulate multiphase flows with transition modelling. In particular, the code relies on the formulation of a DEM method, the use of a complex thermodynamics, the possibility to model cavitating phenomena. Moreover, the method has been generalized in order to take into account directly uncertainty, thus proposing the so-called Stochastic DEM (sDEM) method. This is one of the first stochastic semi-intrusive scheme, permitting to consider uncertainties in multiphase flows including heat and mass transfer terms. This software is developed together with the University of Zurich.

4.7. PaMPA

Participants: Cédric Lachat, François Pellegrini [Corresponding member], Cécile Dobrzynski.

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, in order to handle other types of elements.

Version 1.0 of PaMPA allows users to declare distributed meshes, 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 in parallel the areas of a mesh that need to be emeshed, 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 (team PUMAS) and AeroSol (teams BACCHUS and CAGIRE).

Following expressions of interest from industrial partners, a formal industrialization process of PaMPA has been started, under the auspices of the *Direction du Transfert Technologique* (DTI) at Inria. In this context, much work was directed towards improving the robustness of the code, by including it into a continuous integration framework based on Jenkins.

4.8. RobUQ

Participants: Pietro Marco Congedo [Corresponding member], Maria Giovanna Rodio, Kunkun Tang.

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.

4.9. Scotch

Participant: François Pellegrini [corresponding member].

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

Scotch (<http://www.labri.fr/~pelegri/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.

The Scotch package consists of two libraries: the sequential Scotch library, and the parallel PT-Scotch library (for "*Parallel Threaded Scotch*") that operates according to the distributed memory paradigm, using MPI. Scotch was the first full 64-bit implementation of a general purpose graph partitioner.

Version 6.0, released on December 2012, offers many new features: static mapping with fixed vertices, static remapping, and static remapping with fixed vertices. Several critical algorithms of the formerly strictly sequential Scotch library can now run in a multi-threaded way. All of these features, which exist only in the sequential version, will be available to the parallel PT-Scotch library in the upcoming release 6.1, the development of which has been pursued this year. Also, Scotch has been integrated into the Jenkins continuous integration framework that is used for other projects of the team, such as PaMPA and AeroSol.

Scotch has been integrated in numerous third-party software, which indirectly contribute to its diffusion. It is natively available in several Linux and Unix distributions, as well as on some vendors platforms (SGI, etc).

4.10. SLOWS

Participants: Luca Arpaia, Andrea Filippini, Maria Kazolea, 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. For non-hydrostatic propagation the enhanced Boussinesq equations of Madsen and Sorensen are used. 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. Newton and frozen Newton loops are used to solve the implicit nonlinear equations. The linear algebraic systems arising in the discretization are solved with the MUMPS library. This year implicit and explicit (extrapolated) multistep higher order time integration methods have been implemented, and a mesh adaptation technique based on simple mesh deformation has been also included. The current mid-term objective is to merge SLOWS with the TUCWave code developed by M. Kazolea during her PhD to obtain the first (worldwide) non-hydrostatic dynamically adaptive unstructured mesh Boussinesq code for wave propagation in the near shore region.

4.11. Nomesh

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

Nomesh is a software allowing the generation of third order curved simplicial meshes. Starting from a "classical" mesh with straight elements composed by triangles and/or tetrahedra, we are able to curve the boundary mesh. Starting from a mesh with some curved elements, we can verify if the mesh is valid, that means there is no crossing elements and only positive Jacobian. If the curved mesh is non valid, we modify it using linear elasticity equations until having a valid curved mesh.

CAGIRE Team

5. New Software and Platforms

5.1. AeroSol

Participants: Hamza Belkhayat Zougari [Cagire], Simon Delmas [Cagire], Damien Genet [Bacchus], Francois Pellegrini [Bacchus], Vincent Perrier [Cagire, correspondant], 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 work of the team Bacchus is focused on continuous finite elements methods, while the team Cagire is focused on discontinuous Galerkin methods. However, everything is done for sharing the largest part of code we can. More precisely, classes concerning IO, finite elements, quadrature, geometry, time iteration, linear solver, models and interface with PaMPA are used by both of the teams. This modularity is achieved by means of template abstraction for keeping good performances.

The distribution of the unknowns is made with the software PaMPA, developed within the team Bacchus and the team Castor.

At the end of 2013, Aerosol had the following features

- **development environment** use of CMake for compilation (gcc, icc and xlc), CTest for automatic tests and memory checking, lcov and gcov for code coverage reports. Development of a CDash server for collecting the unitary tests and the memory checking. Beginning of the development of an interface for functional tests. Optional linking with HDF5, PAPI, with dense small matrices libraries (BLAS, Eigen)
- **In/Out** link with the XML library for handling with parameter files. Parallel reader for GMSH, with an embedded geometrical pre-partitioner. Writer on the VTK-ASCII legacy format (cell and point centered). Parallel output in vtu and pvtu (Paraview) for cell-centered visualization, and XDMF/HDF5 format for both cell and point centered visualization.
- **Quadrature formula** up to 11th order for Lines, Quadrangles, Hexaedra, Pyramids, Prisms, up to 14th order for tetrahedron, up to 21st order for triangles. Gauss-Lobatto type quadrature formula for lines, triangles, quadrangles and hexaedra.
- **Finite elements** up to fourth degree for Lagrange finite elements and hierarchical orthogonal finite element basis (with Dubiner transform on simplices) on lines, triangles, quadrangles, tetrahedra, prisms, hexaedra and pyramids. Finite element basis that are interpolation basis on Gauss-Legendre points for lines, quadrangles, and hexaedra, and triangle (only 1st and 2nd order)
- **Geometry** elementary geometrical functions for first order lines, triangles, quadrangles, prisms, tetrahedra, hexaedra and pyramids.
- **Time iteration** explicit Runge-Kutta up to fourth order, explicit Strong Stability Preserving schemes up to third order. Optimized CFL time schemes: SSP(2,3) and SSP(3,4). CFL time stepping. Implicit integration with BDF schemes from 2nd to 6th order
- **Linear Solvers** link with the external linear solver UMFPack, PETSc and MUMPS. Internal solver for diagonal and block-diagonal matrices.
- **Memory handling** discontinuous and continuous, sequential and parallel discretizations based on PaMPA for generic meshes, including hybrid meshes.

- **Models** Perfect gas Euler system, real gas Euler system (template based abstraction for a generic equation of state), scalar advection, Waves equation in first order formulation, generic interface for defining space-time models from space models. Diffusive models: isotropic and anisotropic diffusion, compressible Navier-Stokes.
- **Numerical schemes** continuous Galerkin method for the Laplace problem (up to fifth order) with non consistent time iteration or with direct matrix inversion. Explicit and implicit discontinuous Galerkin methods for hyperbolic systems, diffusive and advection-diffusion problems. Beginning of optimization by stocking the geometry for advection problems. SUPG and Residual distribution schemes.
- **Numerical fluxes** centered fluxes, exact Godunov' flux for linear hyperbolic systems, and Lax-Friedrich flux. Riemann solvers for Low Mach flows.
- **Boundary conditions** Periodic boundary conditions, time-dependent inlet and outlet boundary conditions.
- **Parallel computing** Mesh redistribution, computation of Overlap with PaMPA. collective asynchronous communications (PaMPA based). Asynchronous point to point communications. Tests on the cluster Avakas from MCIA, and on Mésocentre de Marseille, and PlaFRIM. Tuer-1 Turing (Blue-Gene).
- **C++/Fortran interface** Tests for binding fortran with C++.
- **Instrumentation** Aerosol can give some traces on memory consumption/problems with an interfacing with the PAPI library. Tests have also been performed with VTUNE and TAU.

This year, the following features were added

- **In/Out** Ability of saving the high order solution and restarting from it. Computation of volumic and probe statistics. Ability of saving averaged layer data in quad and hexa meshes. Ability of defining user defined output visualization variables.
- **Geometry** handling of high order meshes.
- **Time iteration** Newton method for stationary problems. Implicite unstationary time iterator non consistent in time for stationary problems. Implementation of in house GMRES and conjugate gradient based on Jacobian free iterations.
- **Models** scalar advection-diffusion model
- **Numerical schemes** Optimization of DG schemes for advection-diffusion problems: stocking of the geometry and use of BLAS for all the linear phases of the scheme.
- **Numerical fluxes** Development of a new numerical flux accurate for steady and unsteady computations.
- **Boundary conditions** Adiabatic wall and isothermal wall; Steger-Warming based boundary condition.
- **Instrumentation** Tests with Maqao and Scalasca (VIHPS workshop)
- **Validation** Poiseuille, Taylor-Green vortex. Laplace equation on a ring and Poiseuille flow on a ring. Implementation of volumic forcing based on wall dissipation.

CARMEN Team

5. New Software and Platforms

5.1. CEPS: a Cardiac ElectroPhysiology Simulator

The Carmen team develops a software code to perform high performance numerical simulations in cardiac electrophysiology using unstructured three-dimensional grids. The software, called CEPS (*Cardiac Electrophysiology Simulation*), is developed as a common tool for researchers in the Carmen team and for our partners and colleagues in scientific computing and biomedical engineering. The goal of CEPS is to easily allow the development of new numerical methods and new physical models. Thanks to the ADT, we are now able to use CEPS for the benchmark named *Second N-version Cardiac Electrophysiology Benchmark Specification actual developments*, see ([benchmark](#)) for more details.

As compared to other existing softwares, CEPS aims at providing a more general framework of integration for new methods or models and a better efficiency in parallel. CEPS is designed to run on massively parallel architectures, and to make use of state-of-the-art and well known computing libraries to achieve realistic and complex heart simulations. CEPS also includes software engineering and validation tools. We use the platform GForge ([ceps](#)) based on Subversion. This allows to keep a history of developments for developers and users.

Some of our collaborators actively participate to the testing and discussion for the development of CEPS, namely:

- C. Pierre, LMA University of Pau et des Pays de l'Adour;
- R. Turpault, IMB University of Bordeaux;

5.2. PROPAG

The workhorse for our applied simulation studies of the whole human heart is PROPAG, a code that has its origins at the Université de Montréal in Canada, and has been further developed by the Institute of Computational Science in Lugano, Switzerland. PROPAG is highly configurable and works with arbitrary model geometries. It runs efficiently on high-performance computing systems with many thousands of cores, including a “difficult” system such as the BlueGene/Q “Turing” at IDRIS. It is particularly useful for whole-heart studies, which typically rely on very large model sizes (in the order of 10^8 elements), several different membrane models and cell types in a single simulation run, and several regionally varying parameters.

PROPAG is presently used in our group to study the relation between the substrate, complexity, and electrocardiographic features of atrial fibrillation and of cardiomyopathy-related ventricular arrhythmia, providing the efficiency and flexibility that is required to handle the complex anatomical structures that are involved.

5.3. YAPI: A new project for the development of a platform for the simulation of the electrophysiology cardiac with CEPS

Many of our projects rely on realistic or even patient-tailored meshes to represent the anatomy of the human heart and torso. The construction of such meshes provides challenges on many levels, from the delineation of the anatomical structures in medical images to the construction of high-quality meshes. The construction of such meshes provides challenges on many levels, from the delineation of the anatomical structures in medical images to the construction of high-quality meshes. We presently use a variety of in-house and public software packages to perform this work and are able to produce meshes of sufficient quality, but we strive for an important streamlining of this work. We have initiated a discussion with several groups inside and outside Inria who have similar needs or can offer solutions. We specifically investigate the possibility to build a common software which combines and complements our present solutions. The new code should make various methods

easily accessible and automate the work as much as possible. Because accuracy and mesh quality are important requirements, the new code should also provide convenient options for human intervention where algorithms fall short. For example, manual segmentation and mesh editing should be as easy and efficient as they are in medical-imaging tools and 3D-editing software, respectively, but well integrated into the workflow.

CQFD Project-Team

5. New Software and Platforms

5.1. Package PCAmixdata

Mixed data type arise when observations are described by a mixture of numerical and categorical variables. The R package PCAmixdata extends standard multivariate analysis methods to incorporate this type of data. The key techniques included in the package are PCAmix (PCA of a mixture of numerical and categorical variables), PCArot (rotation in PCAmix) and MFAMix (multiple factor analysis with mixed data within a dataset). The MFAMix procedure handles a mixture of numerical and categorical variables within a group - something which was not possible in the standard MFA procedure. We also included techniques to project new observations onto the principal components of the three methods in the new version of the package.

5.2. QuantifQuantile: an R package to estimate conditional quantiles using optimal quantization

QuantifQuantile is an R package that allows to perform quantization-based quantile regression. The different functions of the package allow the user to construct an optimal grid of N quantizers and to estimate conditional quantiles. This estimation requires a data driven selection of the size N of the grid that is implemented in the functions. Illustration of the selection of N is available, and graphical output of the resulting estimated curves or surfaces (depending on the dimension of the covariate) is directly provided via the plot function.

5.3. Biips: Software for Bayesian Inference with Interacting Particle Systems

Biips is a software platform for automatic Bayesian inference with interacting particle systems. Biips allows users to define their statistical model in the probabilistic programming BUGS language, as well as to add custom functions or samplers within this language. Then it runs sequential Monte Carlo based algorithms (particle filters, particle independent Metropolis-Hastings, particle marginal Metropolis-Hastings) in a black-box manner so that to approximate the posterior distribution of interest as well as the marginal likelihood. The software is developed in C++ with interfaces with the softwares R, Matlab and Octave [53].

FLOWERS Project-Team

5. New Software and Platforms

5.1. Perception Tools

Participants: David Filliat [correspondant], Louis-Charles Caron, Alexander Gepperth.

5.1.1. Of 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 and the movement of a wheeled mobile robot. The floor plane coefficients are known a priori and are used to eliminate from the point cloud all points that are close to this plane and have a normal perpendicular to it. The floor plane coefficients also allow the detection of walls. Successive RANSACs are run to find planes that are perpendicular to the floor plane, and contain a large number of points. With these large structural regions removed, the only points remaining in the point cloud are the objects in the scene. These objects are separated by clustering the points based on a distance criteria. Close-by points are considered to form a single object.

Objects are characterized by their shape, texture. The texture information is encoded as a histogram that approximates the form of the distribution of color values in the object. A separate histogram is built for the red, green and blue channels. The shape of an object is encoded by computing thousands of randomly chosen Surflet-pair relation features and comping them into a histogram of occurrence.

The classification is done by a 3-layer feed-forward neural network. The network is trained on a dataset of point clouds of 53 objects. After training, the neural network is run on the features computed from each object detected in the segmentation stage [86].

5.1.2. 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.

5.1.3. A Python OptiTrack client

Participant: Pierre Rouanet [correspondant].

This python library allows you to connect to an OptiTrack from NaturalPoint (<http://www.naturalpoint.com/optitrack/>). This camera permits the tracking of 3D markers efficiently and robustly. With this library, you can connect to the Motive software used by the OptiTrack and retrieve the 3D position and orientation of all your tracked markers directly from python.

5.2. Datasets

5.2.1. Assemblies of objects for the 3rd hand project

Participants: Yoan Mollard [correspondant], Thibaut Munzer, Manuel Lopes.

The 3rd hand project aims to develop a semi-autonomous robot assistant that acts as a third hand of a human worker. Especially, both should be able to undertake assembly tasks together, in a cooperative way. In order to analyse assembly tasks we recorded 6 datasets of two objects being assembled by a human. The experiment setup has the form of a single user assembling simple furnitures (a chair and a bench) composed by several distinct parts (seating, back, legs). Each part is tracked thanks to an Optitrack system and a set of reflective markers during the whole assembly. The experimental setup records the absolute poses of each part (position and orientation) and relative poses of each couple of objects.

5.3. Learning algorithms

5.3.1. KidLearn

Participants: Manuel Lopes [correspondant], Benjamin Clement, Pierre-Yves Oudeyer, Didier Roy.

The KidLearn software provides an Intelligent Tutoring System that optimizes teaching sequences based on the estimated level of each particular student. Two algorithms, RiARiT and ZPDES have been developed and are described in [37], [39] and [38]. We updated the Game of Money that we developed last year which allows students between 7-8 years to learn how to use money. It still includes 3 main components: i) a webserver that handles the requests and stores the experiments in a database; ii) a GUI that provides the interface for the game; and iii) the optimization software.

Graphical interfaces in ITS can have unwanted side effects. For this reason, the interface was entirely designed with the help of a didactician, with several specific design choices motivated by pedagogical, motivational and attention requirements. For example, the interface, shown in Figure 1, is such that:

- display is as clear and simple as possible;
- there is no chronometer, so that students are not put under time pressure;
- coins and banknotes have realistic visual appearance, and their relative sizes are respected;
- customer and merchant are represented to indicate clearly the role of the student;
- text quantity is kept to minimum;

Four principal regions are defined in the graphical interface, as shown in Figure 1, on the left picture. The first is the wallet location where users can pick and drag the money items and drop them on the repository location to compose the correct price. The object and the price are present in the object location.



Figure 1. Interface with two exemple of type of exercises, Left: customer/one object, Right : merchant/two objects

We performed a more developed and complete user study than last year, considering 5 different schools in the Bordeaux metropolitan area. We had a total of 400 students between 7 and 8 years old. We divided them into 4 groups, with one control group where student does not use the software and 3 groups where exercises are proposed using : a) a predefined sequence; b) ZPDES; c) RiARiT. To measure student learning, students pass pre-test few days before using the interface, and a post test fews days after using the interface. The control group pass the pre and post test at the same time that others but without using the interface between. The results of this study have been presented in [69].

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). In the context of the DIGITEO-funded project “PrActIx”, CEA LIST has now started using this library. In 2014, parts have been made real-time safe for use on the Meka Humanoid robot. This has been fundamental in achieving the results for [65], [64].

5.3.3. *Self-calibration BCI - Matlab library*

Participants: Jonathan Grizou [correspondant], Iñaki Iturrate, Luis Montesano, Manuel Lopes, Pierre-Yves Oudeyer.

The Matlab software implements the algorithms described in [45]. Downloadable from <https://github.com/jgrizou/lfui>.

It allows a robot to be instructed a new task by a human using communicative signals initially totally unknown to the robot. It is was extended and improved in the context of EEG-based brain-machine interfaces (BMIs) [44].

It results in a BCI based control of sequential tasks with feedback signals that do not require any calibration process. As a by-product, the method provides an unsupervised way to train a decoder with the same performance than state-of-the-art supervised classifiers, while keeping the system operational and solving, with a lower performance during the first steps, the unknown task. The algorithm has been tested with online experiments (fig. 2), showing that the users were able to guide from scratch an agent to a desired position.

To improve the efficiency of the algorithm, we introduced a new planning method that uses the uncertainty in the signal-target estimation. This planner is inspired by exploration methods with exploration bonuses that allow guiding to reduce the uncertainty in an efficient way. We showed that trying to follow the best hypothesis does not explore the space significantly to reduce uncertainty and thus identify the correct task. Only through an approach that plans how to reduce the uncertainty multiple steps ahead are we sure that the agent will reach states that can only be explained by the correct hypothesis.

5.3.4. *DyNAMoS: parallel multi-process simulation of distributed neural architectures*

Participants: Alexander Gepperth [correspondant], Mathieu Lefort.

This simulation software comes in the form of a PYTHON module and allows a user to define and simulate complex neural architectures while making use of the parallelism inherent to modern multi-core processors. A special focus lies on on-line learning, processing inputs one by one, in contrast to batch processing of whole databases at a time.

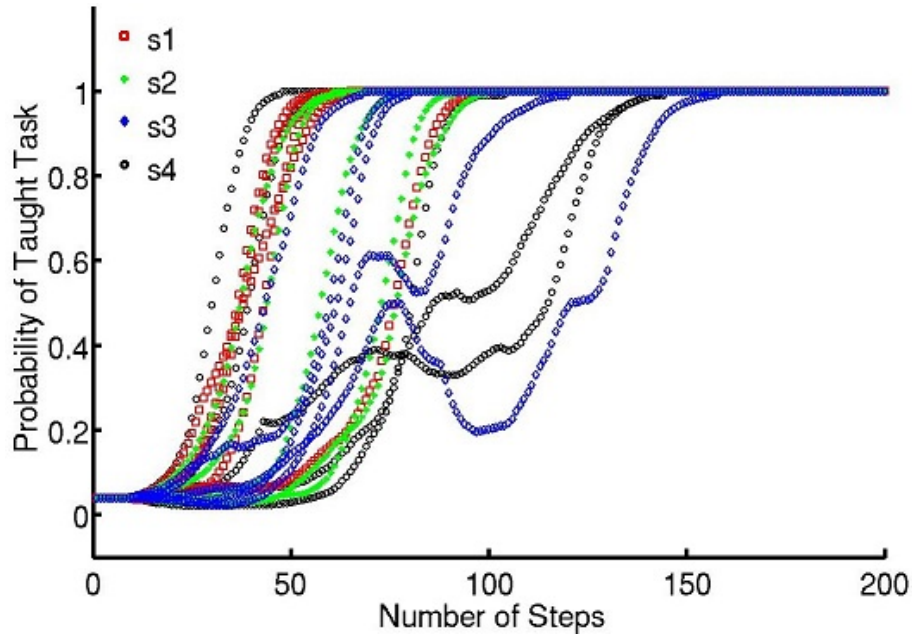


Figure 2. Results from the online BCI experiment for identifying the task. Evolution of the probability of the taught task for each subject and run

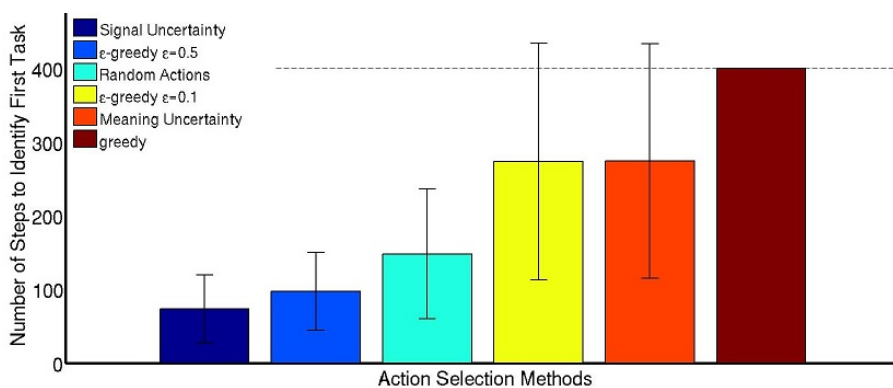


Figure 3. Comparison between different exploration methods. Planning wrt. uncertainty in both task and signal space is the most efficient method

The connectivity of an architecture, as well as neural dynamics and learning rules, are defined by editing simple text-based configuration files. A simple instantiation of a pre-defined simulator class together with the name of the configuration file launches the simulation. Users can provide continuous input to the architecture, as well as inspect and visualize all elements of the simulation, by subclassing the simulator class and redefining the appropriate methods in a clean and Pythonic way. DyNAMoS can be, and is in fact meant to be, extended by user-defined learning methods and dynamics models, which is possible through a well-documented interface all such functions must respect. DyNAMoS distributes computation across multiple processes that are spawned dynamically, possibly on multiple computers, which communicate by TCP/IP or Linux interprocess communication depending on whether they are on the same computer. All aspects of multi-process handling and communication are completely hidden from the user who may merely specify which neural map is executed on which physical process if he wishes to.

This software has been used to speed up computations and provides a common platform for implementing online and incremental learning algorithms. Up to now, we have included linear and logistic regression, various versions of self-organizing maps, MLP and LWPR. It will be made available on GitHub in 2015 after final tests have been concluded.

5.3.5. *pyStreamPlayer: synchronized replay of multiple sensor recordings and supplementary data*

Participant: Alexander Gepperth [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.

5.3.6. *Multimodal: framework around the NMF algorithm for multimodal learning*

Participant: Olivier Mangin [correspondant].

The python code provides a minimum set of tools and associated libraries to reproduce the experiments in [98], together with the choreography datasets. The code, publicly available at <https://github.com/omangin/multimodal>, under the new BSD license, is primarily intended for reproduction of the multimodal learning experiment mentioned above. It has already been reused in several experimentations by other member of the team and is expected to play an important role in further collaborations. It is also expected that the public availability of the code encourages further experimentation by other scientists with data coming from other domains, thus increasing both the impact of the aforementioned publication and the knowledge on the algorithm behaviors. The nonnegative matrix factorization algorithm used in the experiments is also available as a third party extension to <http://scikit-learn.org>.

5.3.7. *Explauto: an autonomous exploration library*

Participants: Clément Moulin-Frier [correspondant], Pierre Rouanet.

Explauto is a framework developed to study, model and simulate curiosity-driven learning and exploration in virtual and robotic agents. The code repository is available at: <https://github.com/flowersteam/explauto>.

This library provides high-level API for an easy definition of:

- Virtual and robotics setups (Environment level)
- Sensorimotor learning iterative models (Sensorimotor level)
- Active choice of sensorimotor experiments (Interest level)

It is cross-platform and has been tested on Linux, Windows and Mac OS. It has been released under the GPLv3 license.

Explauto's scientific roots trace back from Intelligent Adaptive Curiosity algorithmic architecture [15], which has been extended to a more general family of autonomous exploration architecture by [3] and recently expressed as a compact and unified formalism [102]. The library is detailed in [60].

This library has been used in many experiments including:

- the control of a 2D simulated arm
- the exploration of the inverse kinematics of a poppy humanoid (both on the real robot and on the simulated version)
- acoustic model of a vocal tract

5.3.8. *Explorers Framework*

Participants: Benureau Fabien [correspondant], Pierre-Yves Oudeyer.

The Explorers framework is aimed at creating, testing and comparing autonomous exploration strategies for sensorimotor spaces in robots. The framework is largely strategy-agnostic, and is aimed at expressing motor babbling, goal babbling and intrinsically motivated exploration algorithms, among other. It is also able to express strategies that feature transfer learning, such as the reuse algorithm we introduce in [34].

At the center of the framework, an explorer receives observations and provides motor commands for the environment to execute.

We can then easily express a typical goal babbling architecture (the feedback update is not pictured).

Here, the explorer interacts with the environment, rather than the inverse model. Such an architecture allows to filter motor commands that are proposed by the inverse model, and eventually to select another goal if the motor command is not satisfactory or possible to execute. The framework is organized in a modular way. This allows to create flexible hierarchical architectures made of several, atomic or themselves composite, exploration strategies.

The framework has been released this year under the *OpenScience* license (<http://fabien.benureau.com/openscience.html>), and made available on github (<https://github.com/humm/explorers>). Using provided examples, users can easily modify the exploration parameters and investigate for instance the differences between motor and goal babbling exploration strategies.

5.3.9. *PyQMC: Python library for Quasi-Metric Control*

Participant: Steve Nguyen [correspondant].

PyQMC (<https://github.com/SteveNguyen/pyqmc>) is a python library implementing the control method described in <http://dx.doi.org/10.1371/journal.pone.0083411>. It allows to solve discrete markovian decision processes by computing a Quasi-Metric on the state space. This model based method has the advantage to be goal independent and thus can produce a policy for any goal with relatively few recomputation. New addition to this method is the possibility of online learning of the transition model and the Quasi-Metric.

5.4. Software Platforms

5.4.1. *Meka robot platform enhancement and maintenance*

Participants: Antoine Hoarau, Freek Stulp, David Filliat.

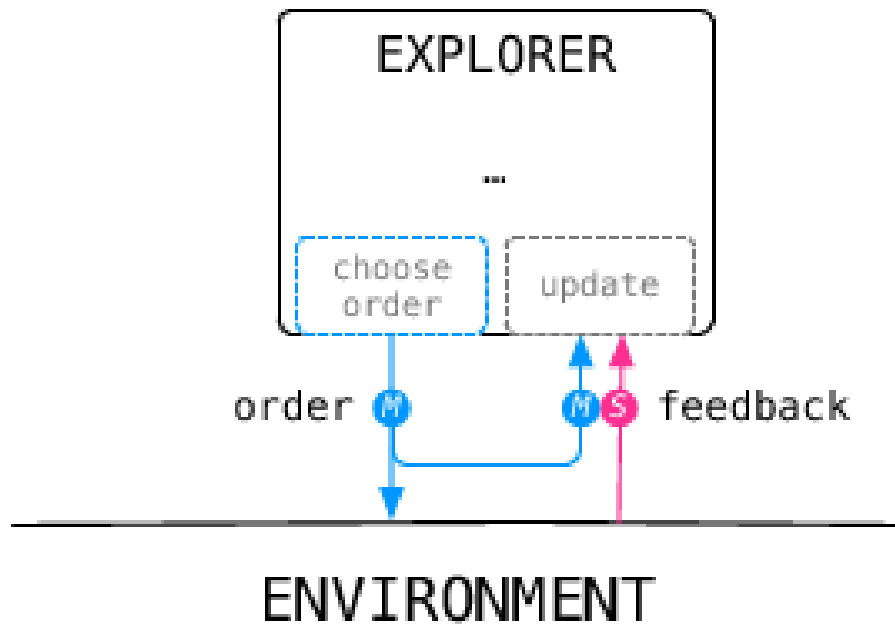


Figure 4.

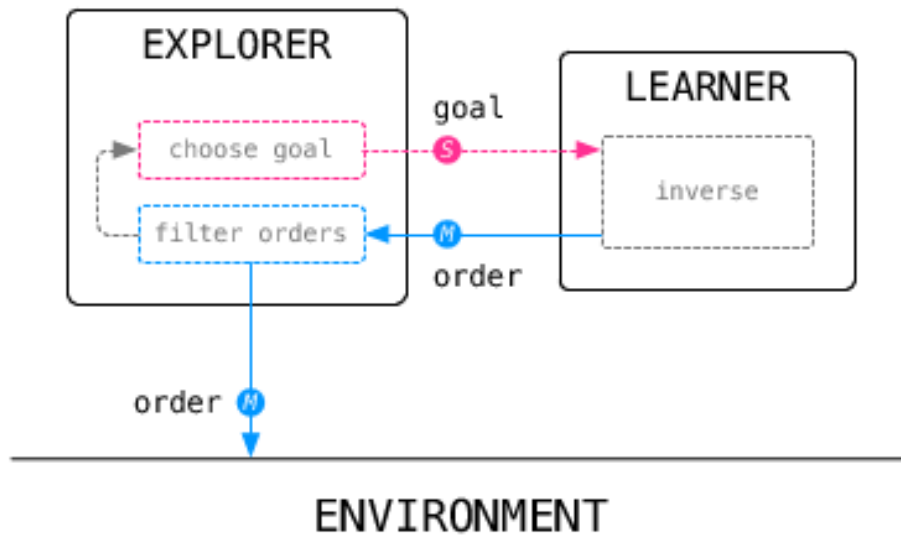


Figure 5.

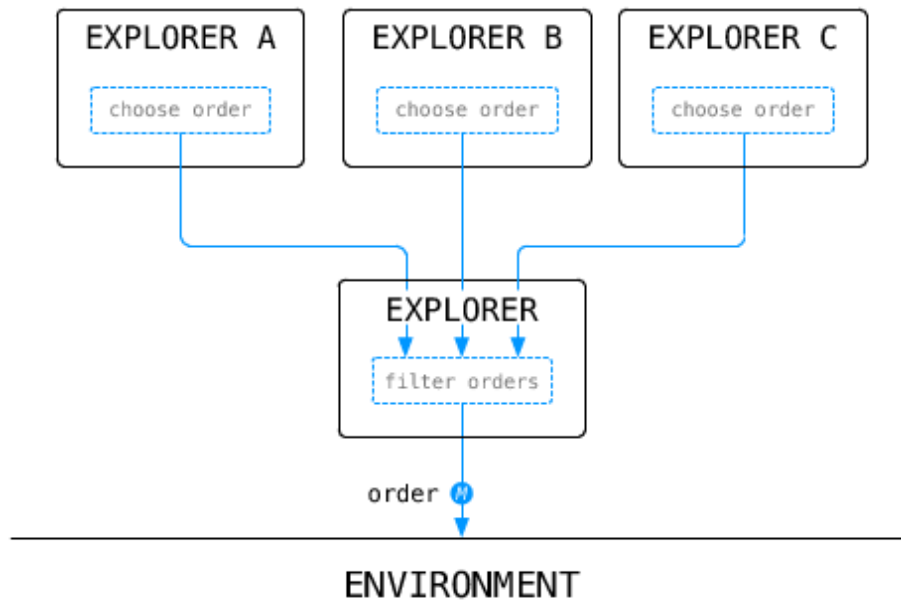


Figure 6.

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. 7) 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 start their experiments.

The platform is evolving as the software (Ubuntu, ROS, our code) is constantly updated and requires some maintenance so less is needed for later. A few demos were added, as the hand shaking demo, in which the robot detects people via kinect and initiates a hand shake with facial expressions. This demo has been used to setup a bigger human robot interaction experiment, currently tested on subjects at Ensta (cf. 8). Finally, we've seen that the robot itself also needs some maintenance; some components broke (a finger tendon), a welding got cold (in the arm) and a few cables experienced fatigue (led matrix and cameras) (cf. 9).

5.4.2. Teaching concepts to the Meka robot

Participants: Fabio Pardo [Correspondant], Olivier Mangin, Anna-Lisa Vollmer, Yuxin Chen, David Filliat.

This platform was developed during Fabio Pardo's internship, in the dual context of the study of Anna-Lisa Vollmer's research on human robot interaction protocole during a learning task, and Olivier Mangin's research on mechanism for word learning and multimodal concept acquisition. The platform is centered around an interaction zone where objects are presented to a Meka robot augmented with a kinect camera placed on top of the interaction zone. Several colorful objects are available to be presented and described to the robot. Several

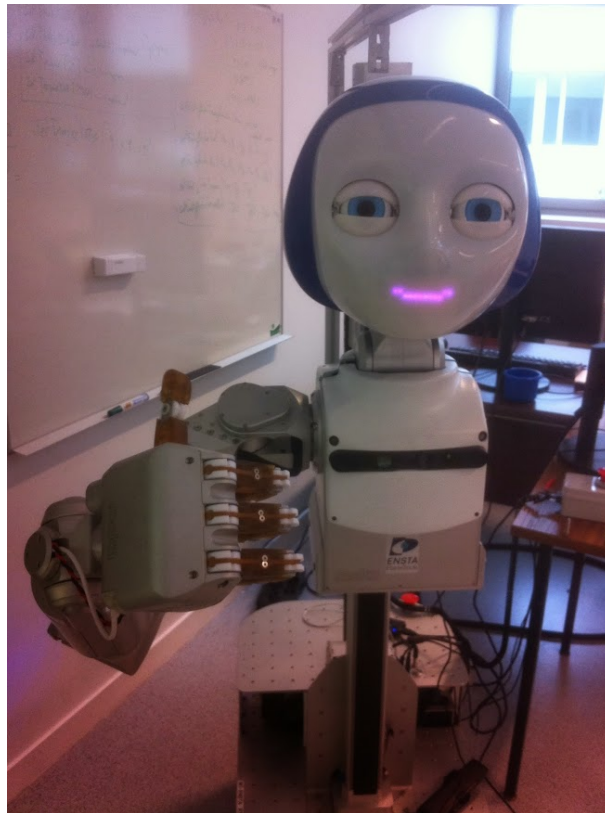


Figure 7. The Meka robot platform acquired by ENSTA ParisTech

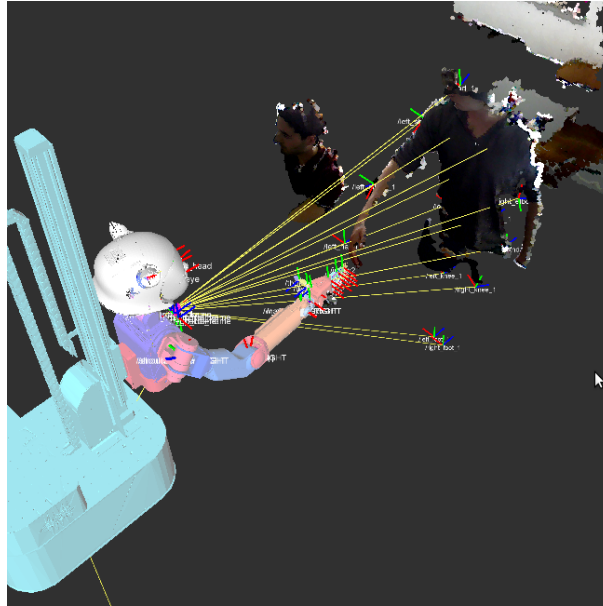


Figure 8. Hand shake demo visualized on Rviz (ROS)

object may be present at the same time on the table. Typical objects are easily characterized by their colors and shapes, such as the *red ball*, the *yellow cup*, or the *blue wagon with red wheels*.

The robot software is capable of abstracting the visual and acoustic perception in the following way. The camera image is segmented into objects; from each object, a set of descriptors is extracted, typically SIFT or shape descriptors and color histograms. An incremental clustering algorithm transforms the continuous descriptors into a histogram of discrete visual descriptors that is provided to the learning algorithm. The acoustic stream is segmented into sentences by a silence detection process and each sentence is fed to Google's text-to-speech API. Finally, each sentence is represented as a histogram of the words recognized in the sentence.

The robot is capable of learning multimodal concepts, spanning words and visual concepts, through the nonnegative matrix factorization framework introduced by Olivier Mangin (see). In addition, several behaviors are programmed in the robot, such as gaze following objects or understanding a few interaction questions.

The framework is illustrated on the following video <https://www.youtube.com/watch?v=Ym5aYfzoQX8>. It enables to modify the interaction as well as the learning mechanisms in order to study the interaction between the teacher and the learning robot.

5.4.3. Experiment platform for multiparameters simulations

Participants: Fabien Benureau, Paul Fudal.

Simulations in robotics have many shortcomings. At the same time, they offer high customizability, rapidity of deployment, absence of failure, consistency across time and scalability. In the context of the PhD work of Fabien Benureau, it was decided to investigate hypothesis first in simulation before moving to real hardware. In order to be able to test a high number of different hypotheses, we developed a software platform that would scale to the computing resource available.

We designed simple continuous simulations around a of-the-shelf 2D physics engine and wrote a highly modular platform that would automatically deploy experiments on cluster environments, with proper handling

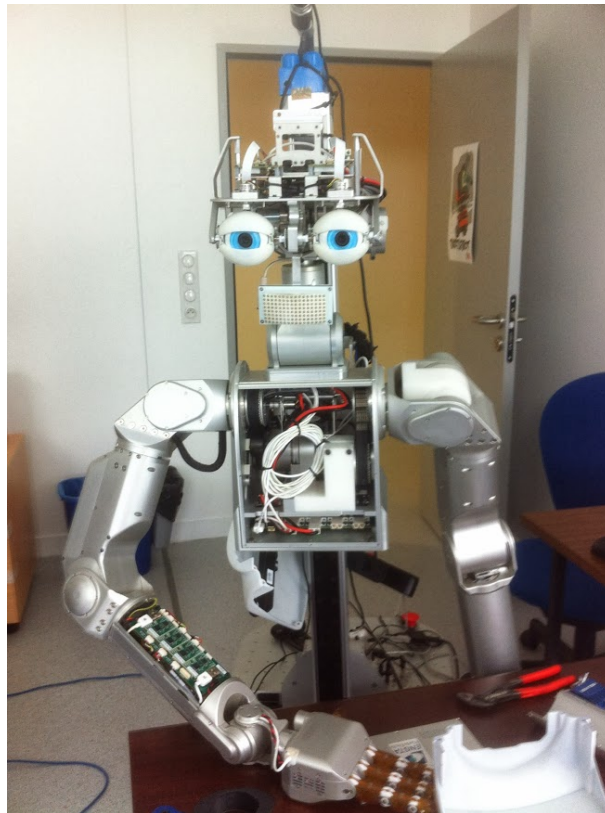


Figure 9. Maintenance is required on the robot

of dependencies; our work investigate transfer learning, and some experiments's input data is dependent of the results of another.

So far, this platform and the university cluster has allowed to conduct thousands of simulations in parallel, totaling more than 10 years of simulation time. It has led us to present many diverse experiments in our published work [34], each repeated numerous times. It has allowed us to conduct a multi-parameter analysis on the setup, which led to new insights, which are being presented in a journal article to be submitted in the beginning of this year.

Because of its high modularity, this platform is proving to be highly flexible. We are currently adapting it to a modified, cluster-ready, version of the V-REP simulator. Those simulations will serve to back ones on similar real-world hardware that are currently setup.

5.4.4. *pypot*

Participants: Pierre Rouanet [correspondant], Steve N'Guyen, Matthieu Lapeyre.

Pypot is a framework developed to make it easy and fast to control custom robots based on dynamixel motors. This framework provides different levels of abstraction corresponding to different types of use. More precisely, you can use pypot to:

1. directly control robotis motors through a USB2serial device,
2. define the structure of your particular robot and control it through high-level commands,
3. define primitives and easily combine them to create complex behavior.

Pypot has been entirely written in Python to allow for fast development, easy deployment and quick scripting by non-necessary expert developers. It can also benefits from the scientific and machine learning libraries existing in Python. The serial communication is handled through the standard library and thus allows for rather high performance (10ms sensorimotor loop). It is crossed-platform and has been tested on Linux, Windows and Mac OS.

Pypot is also compatible with the V-REP simulator (<http://www.coppeliarobotics.com>). This allows the transparent switch from a real robot to its simulated equivalent without having to modify the code.

Pypot also defined a REST API permitting the development of web apps such as a web control interface facilitating the use of a robotic platform.

Pypot is part of the Poppy project (<http://www.poppy-project.org>) and has been released under an open source license GPL V3. More details are available on pypot website: <https://github.com/poppy-project/pypot>

5.5. Experimental Setups

5.5.1. *Experimental Platform for User Study of Curiosity-driven Exploration*

Participants: Pierre Rouanet [correspondant], Jonathan Grizou, Brice Miard, Julie Golliot.

This platform has been developed to investigate curiosity-driven behaviors and more precisely how humans explore new sensori-motor spaces. It consists in several simple games where users control a 2D/3D shape with the movements of their body. They have to discover the mapping between their movements and a shape displayed on the screen and learn how to make the controlled shape match the target one (fig 10).

The software is entirely written in Python. It includes a Kinect wrapper allowing the access of 3D position of tracked skeleton joints. It provides a framework for creating new games based on the 2D drawing library (pygame). It also includes a web server used to display game instructions, cut-scene videos and questionnaire.

The presentation of the platform and the preliminary results of a user's study have been reported in [58].

5.5.2. *Learning and representing object assembly tasks*

Participants: Yoan Mollard [correspondant], Thibaut Munzer, Pierre Rouanet, Manuel Lopes.

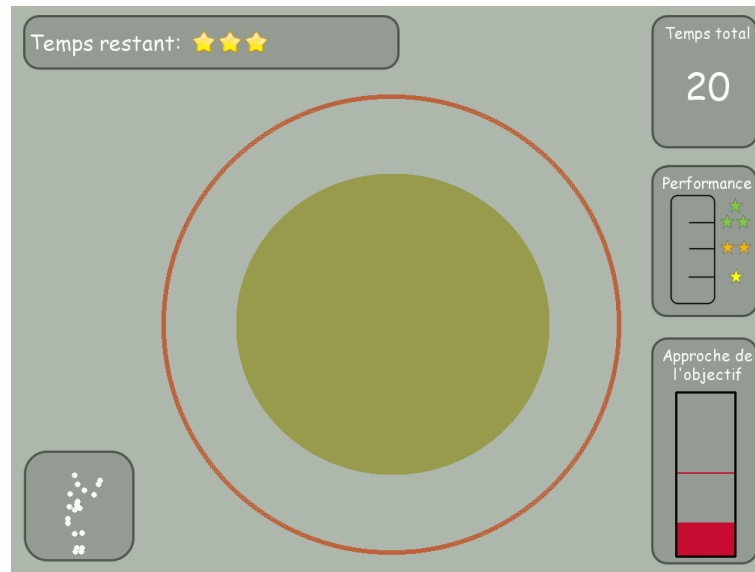


Figure 10. A screenshot representing the game interface as seen by the user.

In the context of the 3rd hand project [8.3.1.1](#) we created a framework for learning assembly tasks from demonstration. In this work we showed how a complex assembly task could be automatically decomposed in components allowing to learn constraints between different objects and their assembly plan. We created also a Graphical User Interface (GUI) allowing to present the learned data in a intuitive way, so that the user can be aware of what the computer has learned. This awareness is crucial for Human-Robot cooperation since the robot will base its decisions on the learned data. Making them clear to the user also allow to rely on him to find potential errors and correct the noise. Thus, the user can program the robot by combining demonstrations and manual corrections minimizing the overall programming phase. Our experimental setup consists in several sequential phases:

1. **Demonstrations:** User provides several demonstrations of an assembly. All parts of the objects are individually tracked by an Optitrack tracking system
2. **Constraint extraction:** Trajectories are analysed to extract rigid constraints
3. **Segmentation:** Constraints on all demonstrations are segmented to find one constraint per object
4. **Plan computation:** We deduce relational MDP trajectories from raw data, creating one assembly step per constraint
5. **Presentation and correction:** The raw constraints and assembly plan are presented to the user in a friendly way through a 3D GUI so that he is able to visualize and correct them [12](#)
6. **Execution:** The corrected informations are then sent to the robot for actual execution. The execution system only receives constraints and the plan, all motions are computed by a motion planner to reach the goals, but motions could also been extracted from the demonstrations using dynamic motor primitives (DMP). We used a simulated Baxter robot that we acquired during the year.

The framework is written in C++ (GUI) and Python (tracking system, data analysis and execution system), and is completely integrated into ROS. The main steps of the workflow are shown on figure [11](#) .

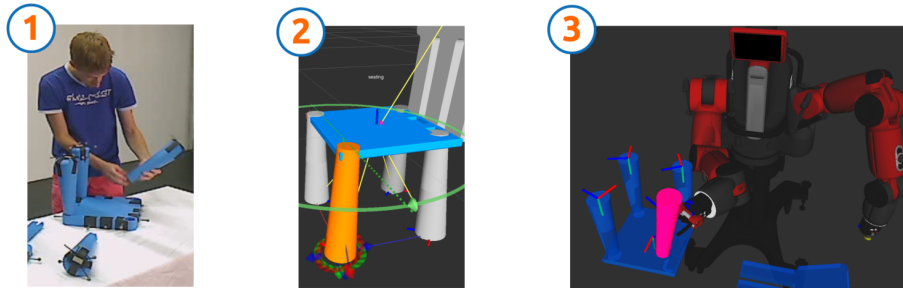


Figure 11. Demonstration, correction and execution of an assembly

The GUI itself 12 represents rigid constraints visually, and provides all the controls necessary to correct them using a graphical procedure. It shows the learned assembly plan as a list of sequential steps that the user can browse like any assembly manual. Also the GUI introduces degrees of freedom in the form of standard mechanical joints (rotational, prismatic, cylindrical joints ...) that the robot can use during execution to simplify the motions and decrease failures during motion planning. The GUI draw graphical cues to represent them and is also able to animate them to make them even clearer.

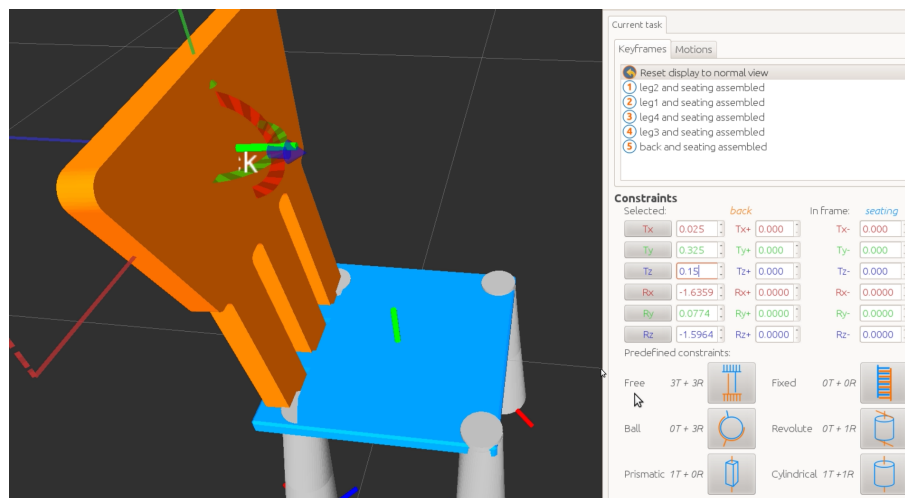


Figure 12. Detailed view of the GUI showing the learned constraints and assembly plan

5.6. Hardware

5.6.1. Poppy Platform

Participants: Matthieu Lapeyre [correspondant], Pierre Rouanet, Jonathan Grizou, Pierre-Yves Oudeyer [supervisor].

The Poppy Project [54], [53], [23] (see Figure 13 , <http://www.matthieu-lapeyre.com/thesis.pdf>) develops an open-source 3D printed humanoid platform based on robust, flexible, easy-to-use and reproduce hardware and software. In particular, the use of 3D printing and rapid prototyping technologies is a central aspect of this project, and makes it easy and fast not only to reproduce the platform, but also to explore morphological variants. Poppy targets three domains of use: science, education and art (see <http://www.poppy-project.org>).

Poppy was initially designed with a scientific objective, aiming to be a new experimental platform opening the possibility to systematically study the role of morphology in sensorimotor control, in human-robot interaction and in cognitive development. Indeed, a suitable design of a robot morphology can greatly simplify control problems, increase robustness, and open new modes of interaction with the physical and social world. Thus, being able to study the body as an experimental variable, something which can be systematically changed and experimented, is of paramount importance. Yet, until recently it was complicated because building a robot relied on heavy and costly manufacturing techniques. 3D printing has changed the landscape of what is possible: Poppy Project transposed it to humanoid robotics, and it is now possible to explore new body shapes in just a few days. It enables and simplifies the experimentation, the reproduction and the modification of the morphology in research laboratories. It also allows collaborative working, sharing and replication of the results on these issues between laboratories. The ambition is to become a reference platform for benchmarking and dissemination of scientific results.

Thanks to the fact that it integrates advanced and yet easily accessible techniques in an embodiment that motivates students and the wider public, this platform also meets a growing societal need: education and training in technologies combining computer science, electronics and mechanics, as well as a training tool to the emergent revolutionary 3D printing process. With its openness, its design and its rather low-cost, Poppy provides a unique context for experimentation and learning of these technologies in a Do-It-Yourself (DIY) approach. Several experiences with Poppy in secondary, high schools, science museums and Fablabs in France and abroad are underway and will be discussed in the incoming sections. Finally, the possibility to easily modify both the hardware and the software also makes Poppy a useful tool for artistic projects working with interactive computerized installations.

5.6.1.1. Open-Source Robotic Platform

Poppy is the first complete 3D printed open-source and open-hardware humanoid robot. Its 3D printed skeleton and its Arduino-based electronics are open-hardware (Creative Commons). Its software is open-source (GPL V3), and allows programming beginners as well as advanced roboticists to control the robot in Python thanks to the PyPot library (<https://github.com/poppy-project/pypot>). Its motors are common off-the-shell Robotis actuators (http://www.robotis.com/xe/dynamixel_en), and allow for compliant control and soft physical human-robot interaction. Poppy presents an original mechanical structure which permits to obtain a light structure with 3.5kg for 84cm height. Before the arrival of 3D printing techniques, this kind of complex structure was either impossible to produce or extremely expensive. Now, anyone can produce and modify such robot in their home using affordable personal 3D printers.

Several web tools support collaboration and sharing among members of the Poppy community: a portal web site (www.poppy-project.org), GitHub repositories for the hardware and software with associated wikis for documentation (www.github.com/poppy-project/), and a forum based on Discourse⁰ technology (forum.poppy-project.org).

⁰www.discourse.org

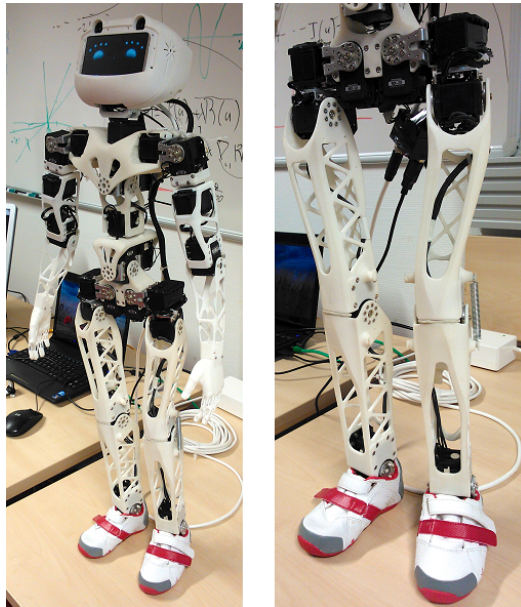


Figure 13. a. Global view of the Poppy platform. b. Zoom on legs design

GEOSTAT Project-Team

5. New Software and Platforms

5.1. Fluex

Participants: Rémi Paties [correspondent], Hussein Yahia, Joel Sudre.

- Previous software engineer Denis Arrivault has delivered the first Fluex package in December 2013, consisting of a core implementation under Gforge of the Microcanonical Multiscale Formalism in the form of C++ classes, for 1D, 2D 3D and 3D+t general signals. The Fluex project is carried on in 2014 by Rémi Paties. Contact: remi.paties@inria.fr.
- The Fluex project has been supported by the FLUEX ADT.

5.2. Platforms

5.2.1. Plafrim

GEOSTAT has participated financially in the acquisition of a computing server SGI UV2000 for **PLAFRIM** through funding with the OPTAD project (Conseil Région Aquitaine).

HIEPACS Project-Team

5. New Software and Platforms

5.1. Introduction

We describe in this section the software that we are developing. The first list 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 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

Participant: Emmanuel Agullo [corresponding member].

MaPHyS (Massively Parallel Hybrid Solver) is a software package that implements a parallel linear solver coupling 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 foresee 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 .

MaPHyS can be found at <http://maphys.gforge.inria.fr>.

5.3. PaStiX

Participant: Pierre Ramet [corresponding member].

Complete and incomplete supernodal sparse parallel factorizations.

PaStiX (Parallel Sparse matriX package) is a scientific library that provides a high performance parallel solver for very large sparse linear systems based on block direct and block ILU(k) iterative methods. Numerical algorithms are implemented in single or double precision (real or complex): LLt (Cholesky), LDLt (Crout) and LU with static pivoting (for non symmetric matrices having a symmetric pattern).

The **PaStiX** library uses the graph partitioning and sparse matrix block ordering package **Scotch**. **PaStiX** is based on an efficient static scheduling and memory manager, in order to solve 3D problems with more than 50 million of unknowns. The mapping and scheduling algorithm handles a combination of 1D and 2D block distributions. This algorithm computes an efficient static scheduling of the block computations for our supernodal parallel solver which uses a local aggregation of contribution blocks. This can be done by taking into account very precisely the computational costs of the BLAS 3 primitives, the communication costs and the cost of local aggregations. We also improved this static computation and communication scheduling algorithm to anticipate the sending of partially aggregated blocks, in order to free memory dynamically. By doing this, we are able to reduce the aggregated memory overhead, while keeping good performance.

Another important point is that our study is suitable for any heterogeneous parallel/distributed architecture when its performance is predictable, such as clusters of multicore nodes. In particular, we now offer a high performance version with a low memory overhead for multicore node architectures, which fully exploits the advantage of shared memory by using an hybrid MPI-thread implementation.

Direct methods are numerically robust methods, but the very large three dimensional problems may lead to systems that would require a huge amount of memory despite any memory optimization. A studied approach consists in defining an adaptive blockwise incomplete factorization that is much more accurate (and numerically more robust) than the scalar incomplete factorizations commonly used to precondition iterative solvers. Such incomplete factorization can take advantage of the latest breakthroughs in sparse direct methods and particularly should be very competitive in CPU time (effective power used from processors and good scalability) while avoiding the memory limitation encountered by direct methods.

PaStiX is publicly available at <http://pastix.gforge.inria.fr> under the Inria CeCILL licence.

5.4. HIPS

Participant: Pierre Ramet [corresponding member].

Multilevel method, domain decomposition, Schur complement, parallel iterative solver.

HIPS (Hierarchical Iterative Parallel Solver) is a scientific library that provides an efficient parallel iterative solver for very large sparse linear systems.

The key point of the methods implemented in **HIPS** is to define an ordering and a partition of the unknowns that relies on a form of nested dissection ordering in which cross points in the separators play a special role (Hierarchical Interface Decomposition ordering). The subgraphs obtained by nested dissection correspond to the unknowns that are eliminated using a direct method and the Schur complement system on the remaining of the unknowns (that correspond to the interface between the sub-graphs viewed as sub-domains) is solved using an iterative method (GMRES or Conjugate Gradient at the time being). This special ordering and partitioning allows for the use of dense block algorithms both in the direct and iterative part of the solver and provides a high degree of parallelism to these algorithms. The code provides a hybrid method which blends direct and iterative solvers. **HIPS** exploits the partitioning and multistage ILU techniques to enable a highly parallel scheme where several subdomains can be assigned to the same process. It also provides a scalar preconditioner based on the multistage ILUT factorization.

HIPS can be used as a standalone program that reads a sparse linear system from a file ; it also provides an interface to be called from any C, C++ or Fortran code. It handles symmetric, unsymmetric, real or complex matrices. Thus, **HIPS** is a software library that provides several methods to build an efficient preconditioner in almost all situations.

HIPS is publicly available at <http://hips.gforge.inria.fr> under the Inria CeCILL licence.

5.5. MetaPart

Participant: Aurélien Esnard [corresponding member].

MetaPart is a library that addresses the challenge of (dynamic) load balancing for emerging complex parallel simulations, such as multi-physics or multi-scale coupling applications. First, it offers a uniform API over state-of-the-art (hyper-) graph partitioning software packages such as **Scotch**, PaToH, METIS, Zoltan, Mondriaan, etc. etc. Based upon this API, it provides a framework that facilitates the development and the evaluation of high-level partitioning methods, such as MxN repartitioning or coupling-aware partitioning (co-partitioning).

The framework is publicly available at Inria Gforge: <http://metapart.gforge.inria.fr>.

5.6. MPICPL

Participant: Aurélien Esnard [corresponding member].

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.

MPICPL was developed by HIEPACS for the purpose of the ANR NOSSI. It uses advanced features of MPI2 standard. The framework is publicly available at Inria Gforge: <http://mpicpl.gforge.inria.fr>.

5.7. ScalFMM

Participant: Olivier Coulaud [corresponding member].

ScalFMM (Parallel Fast Multipole Library for Large Scale Simulations) is a software library to simulate N-body interactions using the Fast Multipole Method.

ScalFMM 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 are two main parts: 1) the management of the octree and the parallelization of the method ; 2) the kernels. This new architecture allows us to easily add new FMM algorithm or kernels and new paradigm of parallelization. The code is extremely documented and the naming convention fully respected. Driven by its user-oriented philosophy, **ScalFMM** is using CMAKE as a compiler/installer tool. Even if **ScalFMM** is written in C++ it will support a C and fortran API soon.

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 at 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.

The **ScalFMM** package is available at <http://scalform.gforge.inria.fr>

5.8. ViTE

Participant: Mathieu Faverge [corresponding member].

Visualization, Execution trace

ViTE is a trace explorer. It is a tool made to visualize execution traces of large parallel programs. It supports Pajé, a trace format created by Inria Grenoble, and OTF and OTF2 formats, developed by the University of Dresden and allows the programmer a simpler way to analyse, debug and/or profile large parallel applications. It is an open source software licenced under CeCILL-A.

The **ViTE** software is available at <http://vite.gforge.inria.fr> and has been developed in collaboration with the Inria Bordeaux - Sud-Ouest SED team, Telecom SudParis and Inria Grenoble.

In the same context we also contribute to the EZtrace and GTG libraries in collaboration with F. Trahay from Telecom SudParis. EZTrace (<http://eztrace.gforge.inria.fr>) is a tool that aims at generating automatically execution trace from HPC programs. It generates execution trace files thanks to the GTG library (<http://gtg.gforge.inria.fr>) that can be later interpreted by visualization tools such as **ViTE**.

5.9. Other software

For the materials physics applications, a lot of development will be done in the context of ANR projects (**NOSSI** and **OPTIDIS**, see Section 4.1) in collaboration with LaBRI, CPMOH, IPREM, EPFL and with CEA Saclay and Bruyère-le-Châtel.

- **FAST**

Participant: Olivier Coulaud [corresponding member].

FAST is a linear response time dependent density functional program for computing the electronic absorption spectrum of molecular systems. It uses an $O(N^3)$ linear response method based on finite numerical atomic orbitals and deflation of linear dependence in atomic orbital product space. This version is designed to work with data produced by the SIESTA DFT code. The code produces as principal output a numerical absorption spectrum (complex part of the polarisability, loosely called the polarisability below) and a list of transition energies and oscillator strengths deduced from fitting Lorentzians to the numerical spectrum. Considering the absence of hybrid functionals in SIESTA and that concerning calculation of spectra, generalized gradient Hamiltonians are not usually considered to be notably better than the local density approximation, the present release of **FAST** works only with LDA, which despite its limitations, has provided useful results on the systems to which the present authors have applied it. The **FAST** library is available at <http://people.bordeaux.inria.fr/coulaud/Softwares/FAST/index.html>.

- **OptiDis**

Participant: Olivier Coulaud [corresponding member].

OptiDis is a new code for large scale dislocation dynamics simulations. Its aim is to simulate real life dislocation densities (up until $5 \cdot 10^{22}$ dislocations/ m^{-2}) in order to understand plastic deformation and study strain hardening. The main application is to observe and understand plastic deformation on irradiated zirconium. Zirconium alloys is the first containment barrier against the dissemination of radioactive elements. More precisely, with neutron irradiated zirconium alloys we are talking of channeling mechanism, which means to stick with the reality, more than tens of thousands of induced loops so 10^8 degrees of freedom in the simulation.

The code is based on Numodis code developed at CEA Saclay and the **ScalFMM** library developed in our Inria project. The code is written in C++ language and using the last features of C++11. One of the main aspects is the hybrid parallelism MPI/OpenMP that gives the software the ability to scale on large cluster while the computation load rises. In order to achieve that, we use different levels of parallelism. First of all, the simulation box is spread over MPI processes, we then use a thinner level for threads, dividing the domain using an Octree representation. All these parts are driven by the **ScalFMM** library. On the last level our data are stored in an adaptive structure absorbing dynamic of this kind of simulation and handling well task parallelism.

The two following packages are mainly designed and developed in the context of a US initiative led by ICL and to which we closely collaborate through the associate team **MORSE**.

- **PLASMA**

Participant: Mathieu Favergé [corresponding member].

The **PLASMA** (Parallel Linear Algebra for Scalable Multi-core Architectures) project aims at addressing the critical and highly disruptive situation that is facing the Linear Algebra and High Performance Computing community due to the introduction of multi-core architectures.

The **PLASMA** ultimate goal is to create software frameworks that enable programmers to simplify the process of developing applications that can achieve both high performance and portability across a range of new architectures.

The development of programming models that enforce asynchronous, out of order scheduling of operations is the concept used as the basis for the definition of a scalable yet highly efficient software framework for Computational Linear Algebra applications.

The **PLASMA** library is available at <http://icl.cs.utk.edu/plasma>.

- **PaRSEC/DPLASMA**

Participant: Mathieu Faverge [corresponding member].

PaRSEC Parallel Runtime Scheduling and Execution Controller, is a generic framework for architecture aware scheduling and management of micro-tasks on distributed many-core heterogeneous architectures. Applications we consider can be expressed as a Direct Acyclic Graph of tasks with labeled edges designating data dependencies. DAGs are represented in a compact problem-size independent format that can be queried on-demand to discover data dependencies in a totally distributed fashion. **PaRSEC** assigns computation threads to the cores, overlaps communications and computations and uses a dynamic, fully-distributed scheduler based on architectural features such as NUMA nodes and algorithmic features such as data reuse.

The framework includes libraries, a runtime system, and development tools to help application developers tackle the difficult task of porting their applications to highly heterogeneous and diverse environments.

DPLASMA (Distributed Parallel Linear Algebra Software for Multicore Architectures) is the leading implementation of a dense linear algebra package for distributed heterogeneous systems. It is designed to deliver sustained performance for distributed systems where each node featuring multiple sockets of multicore processors, and if available, accelerators like GPUs or Intel Xeon Phi. **DPLASMA** achieves this objective through the state of the art **PaRSEC** runtime, porting the **PLASMA** algorithms to the distributed memory realm.

The **PaRSEC** runtime and the **DPLASMA** library are available at <http://icl.cs.utk.edu/parsec>.

5.10. Platforms

5.10.1. *PlaFRIM: an experimental parallel computing platform*

PLAFRIM is an experimental platform for research in modeling, simulations and high performance computing. This platform has been set up from 2009 under the leadership of Inria Bordeaux Sud-Ouest in collaboration with computer science and mathematics laboratories, respectively Labri and IMB with a strong support in the region Aquitaine.

It aggregates different kinds of computational resources for research and development purposes. The latest technologies in terms of processors, memories and architecture are added when they are available on the market. It is now more than 1,000 cores (excluding GPU and Xeon Phi) that are available for all research teams of Inria Bordeaux, Labri and IMB. This computer is in particular used by all the engineers who work in HiePACS and are advised by F. Rue from the SED.

The PlaFRIM platform initiative is coordinated by O. Coulaud.

LFANT Project-Team

4. New Software and Platforms

4.1. Pari/Gp

Participants: Karim Belabas [correspondent], Bill Allombert, Henri Cohen, Andreas Enge, Hamish Ivey-Law.

<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: 2.7.2
- Version of gp2c: 0.0.9
- License: GPL v2+
- Programming language: C

4.2. GNU MPC

Participants: Andreas Enge [correspondent], Mickaël Gastineau [CNRS], Philippe Théveny [INRIA project-team ARIC], 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 GNUMPFR.

It is a prerequisite for the GNU compiler collection GCC since version 4.5, where it is used in the C and Fortran front ends for constant folding, the evaluation of constant mathematical expressions during the compilation of a program. Since 2011, it is an official GNU project.

2012 has seen the first release of the major version 1.0.

- Version: 1.0.2 *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

4.3. 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: 0.4.2 *Cassava*
- License: LGPL v2.1+
- Programming language: C

4.4. 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 [8].

- Version: 0.2 *Blindhühnchen*
- License: GPL v2+
- Programming language: C

4.5. AVIsogenies

Participants: Damien Robert [correspondent], 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

4.6. APIP

Participant: Jérôme Milan.

<http://www.lix.polytechnique.fr/~milanj/apip/apip.xhtml>

APIP, Another Pairing Implementation in PARI, is a library for computing standard and optimised variants of most cryptographic pairings.

The following pairings are available: Weil, Tate, ate and twisted ate, optimised versions (à la Vercauteren–Hess) of ate and twisted ate for selected curve families.

The following methods to compute the Miller part are implemented: standard Miller double-and-add method, standard Miller using a non-adjacent form, Boxall et al. version, Boxall et al. version using a non-adjacent form.

The final exponentiation part can be computed using one of the following variants: naive exponentiation, interleaved method, Avanzi–Mihailescu’s method, Kato et al.’s method, Scott et al.’s method.

Part of the library has been included into PARI/GP proper.

- Version: 2012-10-17
- License: GPL v2+
- Programming language: C with libpari

4.7. CMH

Participants: Andreas Enge, Emmanuel Thomé [INRIA project-team CAMEL].

<http://cmh.gforge.inria.fr/>

CMH computes Igusa class polynomials, parameterising two-dimensional abelian varieties (or, equivalently, Jacobians of hyperelliptic curves of genus 2) with given complex multiplication.

- Version: 1.0
- License: GPL v3+
- Programming language: C

4.8. Cubic

Participant: Karim Belabas.

<http://www.math.u-bordeaux1.fr/~belabas/research/software/cubic-1.2.tgz>

CUBIC is a stand-alone 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

4.9. Euclid

Participant: Pierre Lezowski.

<http://www.math.u-bordeaux1.fr/~plezowsk/euclid/index.php>.

Euclid is a program to compute the Euclidean minimum of a number field. It is the practical implementation of the algorithm described in [38]. Some corresponding tables built with the algorithm are also available. Euclid is a stand-alone program depending on the PARI library.

- Version: 1.2
- License: LGPL v2+
- Programming language: C

4.10. 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

MAGIQUE-3D Project-Team

5. New Software and Platforms

5.1. Hou10ni

Participant: Julien Diaz [correspondant].

This software, written in FORTRAN 90, simulates the propagation of waves in heterogeneous 2D and 3D media in time-domain and in frequency domain. It is based on an Interior Penalty Discontinuous Galerkin Method (IPDGM) and allows for the use of meshes composed of cells of various order (p -adaptivity in space).

This year, we have implemented the 3D version for the simulation of elastodynamic waves. This version handles polynomials of arbitrary order while the previous one was only able to deal with polynomials of degree up to three.

We have also improved the parallelism by coupling the code to a mesh partitioner and we have totally rewritten the code to handle MPI parallelism both for the construction of the matrices and for the time scheme.

5.2. Montjoie

Participant: Marc Duruflé [correspondant].

Montjoie is a code developed by Marc Duruflé with contributions of students, including Juliette Chabassier during her PhD. It provides a C++ framework for solving partial differential equations on unstructured meshes with finite element-like methods (continuous finite element, discontinuous Galerkin formulation, edge elements and facet elements). The handling of mixed elements (tetrahedra, prisms, pyramids and hexahedra) has been implemented for these different types of finite elements methods in the context of Morgane Bergot's PhD. Several applications are currently available : wave equation, elastodynamics, aeroacoustics, Maxwell's equations. In 2014, the implementation of linearized Euler equations and Galbrun's equation has been improved and extended to the axisymmetric case. Raman effect has been implemented in the 1-D non-linear Schrödinger equation.

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

5.3. Elasticus

Participant: Simon Ettouati.

Within the framework of the strategic action DIP, Magique-3D collaborates with Total to develop a computing platform, DIVA, meant to produce accurate images of the subsurface. To achieve this, approximate solutions of the first-order wave problem are computed thanks to a Discontinuous Galerkin (DG) Method. It is increasingly difficult to include new numerical schemes developed in the team in the industrial and highly parallel environment of Total.

Elasticus is a sequential library, independent of DIVA and developed in Fortran, to simulate wave propagation in geophysical environment, based on a DG method. It is meant to help PhD students and post-doctoral fellows to easily implement their algorithms in the library. Thus, readability of the code is privileged to optimization of its performances. Developed features should be easily transferred in the computing platform of Total. Contrary to DIVA which only computes approximate solutions with P1, P2 and P3 elements, Elasticus manages arbitrary orders for the spatial discretization with DG method. Matrices on the reference element for arbitrary orders are computed thanks to a library developed by J. Diaz.

5.4. DIVA-DG

Participants: Lionel Boillot, Marie Bonnasse-Gahot, Théophile Chaumont-Frelet, Jérôme Luquel.

DIVA-DG is the simulation code that we develop in collaboration with our partner Total. This year we have implemented

- 2D/3D anisotropic elastic Absorbing Boundary Conditions for time-domain.
- 2D elastic imaging conditions.
- 2D multiscale strategy to take into account fine scale heterogeneities on coarse meshes in frequency domain.
- Hybridized Discontinuous Galerkin method for 2D elastodynamic in frequency domain.

MAGNOME Project-Team

5. New Software and Platforms

5.1. Magus: Genome exploration and analysis

Participants: David James Sherman [correspondant], Pascal Durrens, Florian Lajus, Xavier Calcas.

The MAGUS genome annotation system integrates genome sequences and sequences features, *in silico* analyses, and views of external data resources into a familiar user interface requiring only a Web navigator. MAGUS implements 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 results in an n -fold increase in curation speed, compared to curation of individual genes. This allows us to maintain standards of high-quality manual annotation while efficiently using the time of volunteer curators. MAGUS can be used on small installations with a web server and a relational database on a single machine, or scaled out in clusters or elastic clouds using Apache Cassandra for NoSQL data storage and Apache Hadoop for Map-Reduce (figure 1). For more information see the MAGUS Gforge web site.⁰ MAGUS 2.0 was developed in an Inria Technology Development Action (ADT) and is distributed with an open-source license.

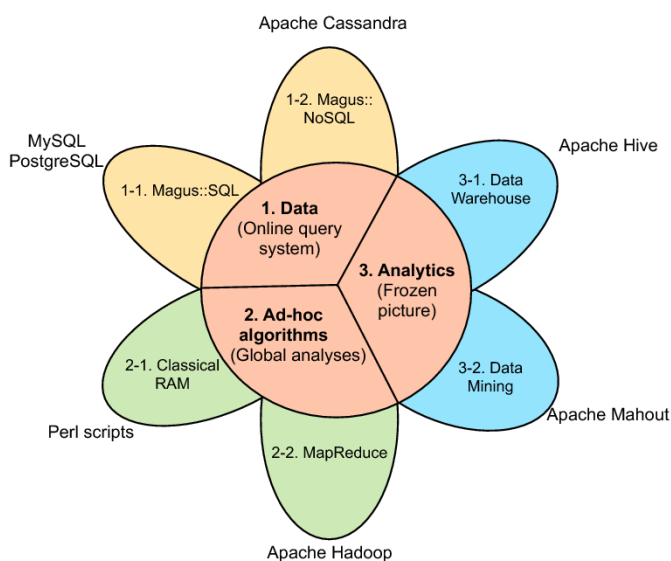


Figure 1. General architecture of the Tsvetok system implemented in MAGUS, showing the role of the NoSQL (Apache Cassandra) and Map-Reduce (Apache Hadoop) paradigms

5.2. Pantograph: Inference of metabolic networks

Participants: David James Sherman [correspondant], Pascal Durrens, Anna Zhukova.

⁰<http://magus.gforge.inria.fr>

Pantograph is a software tool developed by Nicolás Loira for his thesis, that infers whole-genome metabolic models for eukaryote cell factories from reference models and genome comparison. A novel feature of Pantograph is that it uses expert knowledge implicitly encoded in the scaffold's gene associations, and explicitly transfers this knowledge to the new model. Pantograph is available under an open-source license. For more information see the Pantograph Gforge web site.⁰

5.3. MIMOZA: Generalizing and Visualizing Metabolic Models

Participants: David James Sherman [correspondant], Anna Zhukova.

MIMOZA uses metabolic model generalization and cartographic paradigms to allow human experts to explore a metabolic model in a hierarchical manner. The software creates a zoomable representation of a model submitted by the user in SBML⁰ format. The most general view represents the compartments of the model; the next view shows the visualization of generalized versions of reactions and metabolites in each compartment (see section 6.3); and the most detailed view visualizes the initial model with the generalization-based layout (where similar metabolites and reactions are placed next to each other). The zoomable representation is implemented using the Leaflet⁰ JavaScript library for mobile-friendly interactive maps. Users can click on reactions and compounds to see the information about their annotations. The resulting map can be explored on-line, or downloaded in a COMBINE archive. The software and examples are available at <http://mimoza.bordeaux.inria.fr>.

5.4. Génolevures On Line: Comparative Genomics of Yeasts

Participants: Pascal Durrens [correspondant], David James Sherman.

The Génolevures online database provides archival 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. Génolevures online uses our open-source MAGUS system for genome navigation, with project-specific extensions developed by MAGNOME. For more information see the Génolevures web site.⁰

⁰<http://pathtastic.gforge.inria.fr>

⁰<http://sbml.org>

⁰<http://leafletjs.com>

⁰<http://www.genolevures.org/>

MANAO Project-Team

4. New Software and Platforms

4.1. Software

4.1.1. ALTA Library

Participants: X. Granier & R. Pacanowski & L. Belcour & P. Barla

Keywords: BRDF fitting and analysis

ALTA is a multi-platform software library to analyze, fit and understand BRDFs. It provides a set of command line software to fit measured data to analytical forms, and tools to understand models and data. The targeted audience is composed of all the researchers and professionals who are working on BRDFs, and who want to benchmark new BRDF models and easily compare them with state-of-the-art BRDF models and data. It is also suitable for researchers and professionals who are working on optical measurements, and who want to experiment different fitting procedures and models, or just to perform statistical analysis on their data. The major features in the ALTA library are:

- Open common BRDF data formats (MERL, ASTM)
- Non-linear fitting of BRDF (using third party packages)
- Rational interpolation of BRDF
- Analytic BRDF models
- Scripting mechanism to automatize fitting

ALTA has been supported by the ANR ALTA (ANR-11-BS02-006).

Facts:

- Web: <http://alta.gforge.inria.fr/>
- License: MPLv2

4.1.2. Eigen

Participants: G. Guennebaud

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 but numerous 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 with iterative and direct 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 30k 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 3.2 version in July 2013. Eigen has been supported by Inria through an ADT started in January 2012, and that ended in September 2013. Eigen received the “**high-quality software in geometry processing award**” from the Symposium on Geometry Processing 2013. Eigen is continuously and actively developed with this year an important refactoring of the expression evaluation mechanism, a divide & conquer SVD algorithm, support for AVX in collaboration with Google, and many other features.

Facts:

- Web: <http://eigen.tuxfamily.org/>
- License: MPLv2

4.1.3. *PatateLib*

Participants: N. Mellado, G. Ciaudo, S. Boyé, G. Guennebaud, P. Barla

Keywords: multi-scale analysis, material appearance, vector graphics, expressive rendering, 2D animation
Patate is a header only C++/CUDA library for graphics applications released under the MPL license.

It provides a collection of Computer Graphics techniques that incorporate the latest innovations from Inria research teams working in the field. It strives for efficiency and ease-of-use by focusing on low-level core operators and key algorithms, organized in modules, each tackling a specific set of issues. The central goal of the library is to drastically reduce the time and efforts required to turn a research paper into a ready-to-use solution, for both commercial and academic purposes.

Each module is initially developed by a few persons, usually those who have authored the corresponding research papers. An engineer, Gautier Ciaudo, has been recruited via the ADT program to perform unit tests, bug tracking, and make examples. Our first module provides efficient methods for the fitting and analysis of point-clouds in arbitrary dimensions. It may be used for varied purposes such as curvature computation, surface reconstruction, scale-space analysis, image processing, and sketch vectorization. More modules will be developed in 2015 by Simon Boyé.

Facts:

- Web: <http://patate.gforge.inria.fr>
- License: MPLv2

4.1.4. *PFSTools*

Participant: I. Ihrke

Keywords: high dynamic range image processing, merging, calibration and tone-mapping

The `pfstools` package is a set of command line programs for reading, writing, manipulating and viewing high-dynamic range (HDR) images and video frames. All programs in the package exchange data using a simple generic high dynamic range image format, `pfs`, and they use unix pipes to pass data between programs and to construct complex image processing operations.

`pfstools` come with a library for reading and writing `pfs` files. The library can be used for writing custom applications that can integrate with the existing `pfstools` programs. It also offers a good integration with high-level mathematical programming languages, such as MATLAB or GNU Octave. `pfstools` can be used as an extension for MATLAB or Octave for reading and writing HDR images or simply to effectively store large matrices. The `pfstools` package integrates existing high dynamic range image formats by providing a simple data format that can be used to exchange data between applications. It is accompanied by the `pfscalibration` and `pfstmo` packages.

The `pfscalibration` package provides algorithms for the photometric calibration of cameras and for the recovery of high dynamic range (HDR) images from a set of low dynamic range (LDR) exposures. Maintenance of the `pfscalibration` package is performed by Ivo Ihrke since January 2011. A major update to make the software compatible with current digital SLR cameras and their raw file formats, especially for measurement purposes, has been performed. A new set of MATLAB scripts has been developed for improved calibration performance. It is intended to merge these new procedures into the existing software.

The `pfstmo` package contains the implementation of seven state-of-the-art tone mapping operators suitable for convenient processing of both static images and animations.

The software received wider interest in the Open Source community and third party contributors prepared installation packages which are included in several Linux distributions including Debian, Fedora and Suse.

Facts:

- Web: <http://pfstools.sourceforge.net/>
- License: GPL

4.2. Platforms

4.2.1. COEL - Computational Optics Experimentation Laboratory

We are setting a dedicated experimentation facility up to validate our theoretical tools to design hybrid (optics & computer sciences) systems by creating real setups. Such a facility is unique thanks to the close collaboration between optics and computer science in Bordeaux. Now located in the LP2N, this laboratory consists in a set of on-the-shell elements to design optical systems combined with controllable large-band lighting systems (from pure white sources, to tunable lasers and video-projectors), with a fabrication laboratory to build non-conventional components, with large-scale mechanical elements, with display technologies, and high-performance processing resources.

After initial delays, the lab has now found its final location in LP2N. The basic equipment is in place and first experiments are being performed. We still have to work on the illumination conditions in the room, as well as on the construction of a light-sealed control area inside the experimentation room for independent experiments.

The construction and equipment is financed by a special regional grant of the "Conseil Régional d'Acquitaine" (Carer xD) in conjunction with project-specific funds.

MC2 Team

5. New Software and Platforms

5.1. eLYSe

Participant: Olivier Saut.

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: 0.7
- 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

Participant: Olivier Saut.

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: 0.2
- Keywords: Modélisation and numerical simulations
- APP: En cours
- Type of human computer interaction: console
- OS/Middleware: 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/Middleware: unix, linux, mac os
- Required library or software: PETSc item Programming language: C
- Documentation: in progress

5.4. NS-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/Middleware: 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
 - Elmo. C++ Code of Finite Differences on cartesian grid parallelized with PETSC to compute the electropermeabilisation of cells in 2D and 3D.

MNEMOSYNE Project-Team

5. New Software and Platforms

5.1. Positioning

Our previous works in the domain of well-defined distributed asynchronous adaptive computations [43], [40], [45] have already made us define a library (DANA [39]), 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 networks.

We also gather in the middleware EnaS (that stands for *Event Neural Assembly Simulation*; cf. <http://forge.inria.fr/projects/enas>) our numerical and theoretical developments, allowing to simulate and analyze so called "event neural assemblies".

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 effect 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 [30], 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 [39] 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 evolutions 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. 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 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 consists of a platform associated to a scenario that is the closest possible to a survival situation (foraging, predator-prey relationship, partner approach to reproduction) and in which it is 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 can be slowed down enough to be able to run non trivial brainy-bot implementations. This platform [13] has already being used by two students of the team and is now a new deliverable of the KEOpS project.

PHOENIX Project-Team

5. New Software and Platforms

5.1. DiaSuite: a Development Environment for Sense/Compute/Control

Applications

Participants: Charles Consel [correspondent], Milan Kabac, Paul Van Der Walt, Adrien Carteron, Alexandre Spriet.

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 [35]. This layer is dedicated to an architectural pattern commonly used in the pervasive computing domain [31]. 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 2 .

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

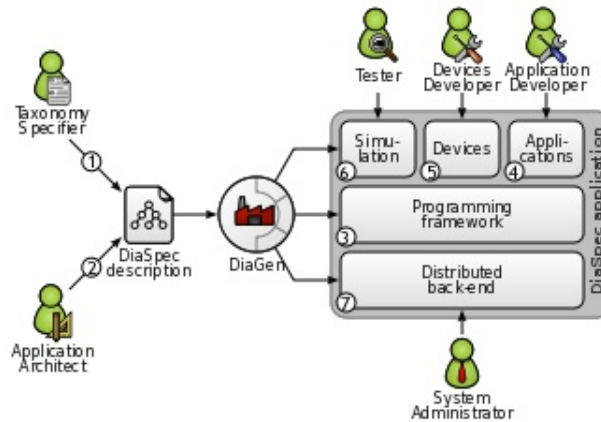


Figure 2. DIASUITE Development Cycle

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 [31]. 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

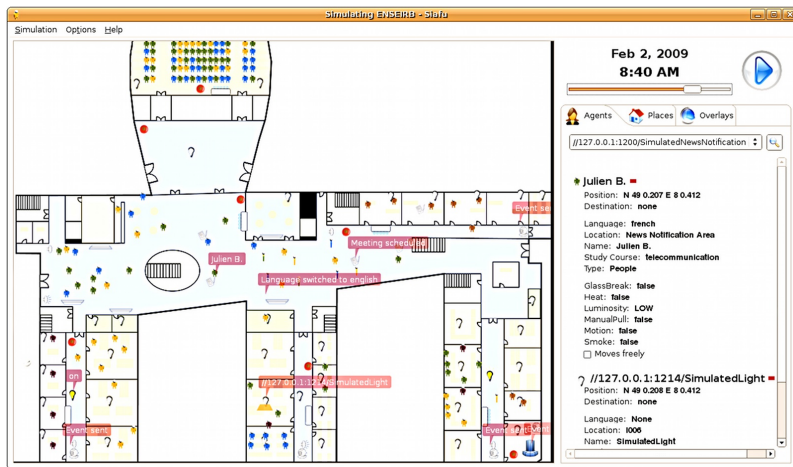


Figure 3. A screenshot of the DIASIM simulator

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 3.

5.2. DiaSuiteBox: an Open Orchestration Platform

Participants: Charles Consel, Adrien Carteron, Alexandre Spriet, Milan Kabac.

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.

5.2.1. DiaSuiteBox platform architecture

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 the behavior of its applications are innocuous and correct with respect to the provided information. Finally, DiaSuiteBox provides an extensible software architecture. This allows the easily connect new device technologies to the platform. For example, the support for new wireless communication technologies such as Zigbee, Z-Wave or Sigfox can be easily added to the DiaSuiteBox platform.

More details can be found on the web page <http://diasuitebox.inria.fr>.

The iQSpot startup uses DiaSuiteBox as a software platform to ease the management of Smart Buildings. In this project, the DiaSuiteBox platform is first used to host building management functionalities such as lighting management, heating/ventilating/air conditioning management, energy efficiency monitoring. It is also used to host software drivers that allow the building management functionalities to interact with the connected devices deployed in buildings. These devices can use wired communication technologies such as LonWorks, BACNet or KNX, as well as wireless communication technologies such as Z-Wave or Zigbee.

5.3. School+ Apps: Assistive tablet applications for school Inclusion

Participants: Charles Consel [correspondent], H el ene Sauz eon, Charles Fage, C ecile Magnier.

School+ is a package of 7 applications. Three applications are assistive applications, guiding the child doing specific tasks. Three others are training applications made as serious games, addressing specific skills. The last application is a meta-application, comprising a link to the three training applications, with an access to statistics of their usage. For each application, data are separated from the design, meaning that every element of each application (pictures, texts, settings, etc.) can be changed at any time. Each application records a log file containing all the interactions performed by the child.

5.3.1. Assistive applications:

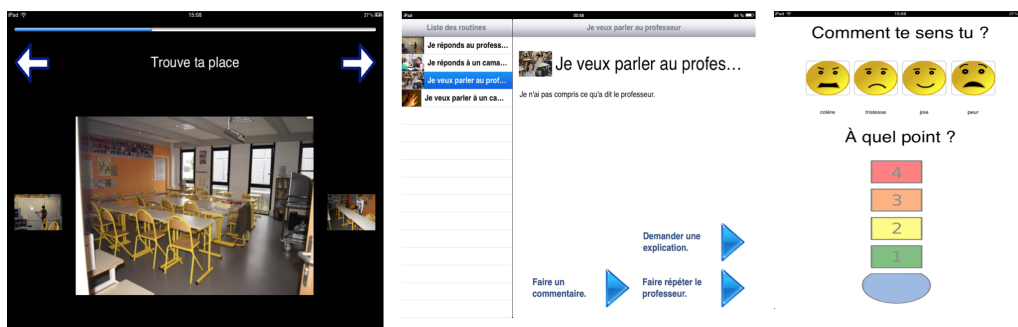


Figure 4. Assistive applications

- **Routines application.** This application shows a list of tasks, with a short description. After clicking the starting button, a specific slideshow is shown; it decomposes a task into steps. For each step, a text and a picture can be displayed. Thumbnail of previous and next steps are also displayed. This application guides the child through classroom situations: entering classroom, taking school materials out of a backpack, writing notes, handling agenda, leaving the classroom.
- **Communication application.** With the same design, the assistance provided by this application targets to communicating situations inside the classroom. The application covers four scenarios addressing two interaction situations (initiating and answering the interaction) and two types of interlocutors (professor and classmate). For each scenario, different slideshows guide the child, depending on the goal of the interaction.
- **Emotion Regulation application.** This application aims to assist the child to self-regulate his/her emotions. Four simplified emoticons are proposed to the child to choose from: anger, sadness, joy and fear. Then, (s)he selects a level of intensity via a thermometer with a scale from 1 to 4. In response, the application delivers different multimedia contents according to the level selected to help the child regulate his/her emotions. Typically, a text (breathing instructions) are shown at level 1, pictures at level 2, a video at level 3 and another text at level 4.

5.3.2. Training applications:

These three applications are serious games with increasing levels of difficulties, reachable after a ratio of good answers has been attained.

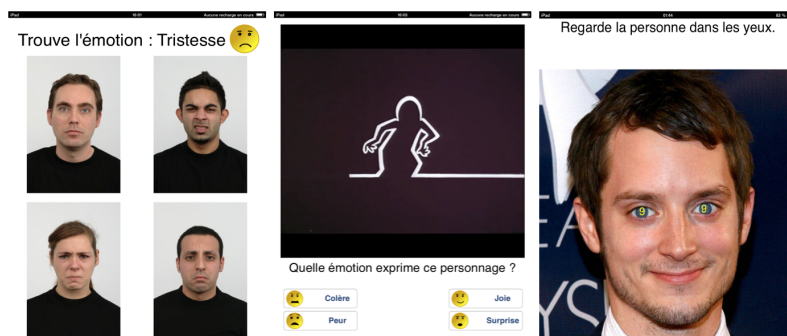


Figure 5. Training applications

- *Emotion Recognition application: pictures.* In this application, the child is instructed to identify a specific emotion among 4 pictures showing different people exhibiting an emotion. Seven emotions are involved in this application: joy, sadness, fear, anger, surprise, disgust and neutral. The emotion to be recognized is displayed together with its simplified emoticon. The type of pictures changes with the difficulty level: level 1 contains pictures of unfamiliar people and level 2 contains pictures of friends and relatives of the child.
- *Emotion Recognition application: videos.* In this application, the child is presented with a fragment of an animated cartoon. At some point, the video stops and the child is asked to identify the emotion of the character. Four emotions are involved in this application: joy, sadness, fear and anger. Videos are slowed down, with a speed percentage that can be changed at each level. Videos change with difficulty level: level 1 contains videos of a very basic cartoon (only one cartoon character drawn by basic form un-textured), level 2 contains a video of more sophisticated cartoons and level 3 contains movies with actors.
- *Attention Training.* In this application, the child is presented a picture of a face and asked to make eye contact with it. Second, a symbol appears briefly in the eyes of the character. Third, the child is asked to identify the symbol shown in the previously displayed picture, to make sure he kept eye contact. The speed at which the symbol appears and disappears is changed according to the difficulty level. Types of pictures also change with the level : level 1 contains pictures of faces and level 2 contains pictures of classroom situations.

5.4. HomeAssist: A Platform for Assistive Living

Participants: Charles Consel, Loïc Caroux [correspondent], Thomas Freslon, Adrien Carteron, David Daney, Lucile Dupuy, Geoffrey Escojido, Bernard N’Kaoua, Hélène Sauzéon, Alexandre Spriet.

The HomeAssist platform proposes a systemic approach to introducing an assistive technological platform for older people. To do so, we formed a trans-disciplinary team that allows (1) to identify the user needs from a gerontological and psychological viewpoint; (2) to propose assistive applications designed by human factors and HCI experts, in collaboration with caregivers and users; (3) to develop and test applications by software engineers; (4) to conduct a field study for assessing the benefits of the platform and assistive applications, in collaboration with caregivers, by deploying the system at the actual home of elders.

The HomeAssist platform is implemented on top of the DiaSuiteBox platform, using a suite of tools, namely DiaSuite, that have been designed, developed and tested by our research group at Inria. The DiaSuite tools include a dedicated integrated development environment that enables applications to be developed quickly and safely. This technology has been successfully applied to a variety of domains where environments consist of networked objects that need to be orchestrated.

5.4.1. Applications

HomeAssist offers an online catalog of applications. Using this catalog, the user and the caregiver determine what and how activities should be assisted by selecting the appropriate assistive applications and configuring them with respect to the user's requirements and preferences. The resulting set of applications forms a personalized assistive support. Additionally, to respond to evolving needs, our platform allows to stop/remove applications easily and to install new ones from the online catalog.

This platform proposes many applications in three domains of everyday life:

- Daily activities: including activity monitoring, light path, and a reminder.
- Home or personal safety: including entrance monitoring, stove monitoring, and warning if no movements are detected after a certain amount of time.
- Communications and social activities: including collaborative games, videoconference, information about local events, TV programming, *etc.*

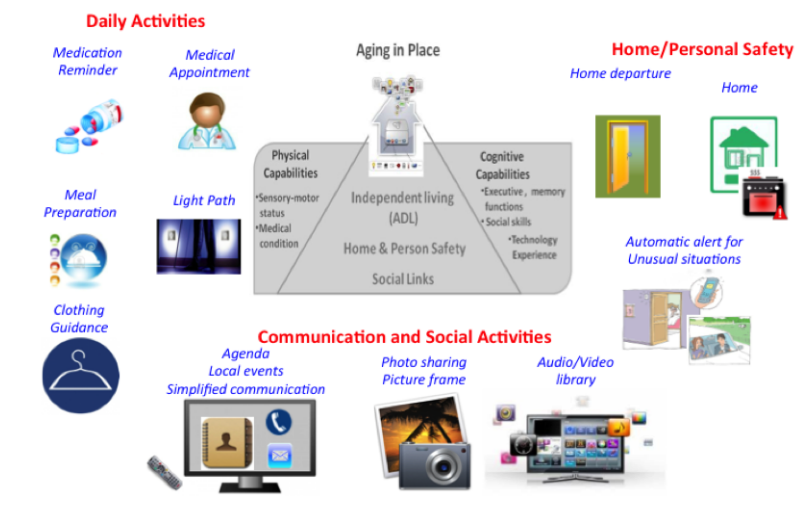


Figure 6. The HomeAssist platform and applications

For video presentations of HomeAssist, see the following:

- <http://videotheque.inria.fr/videotheque/media/23705>. Title: “Dia Suite Box”. Produced in 2013.
- <http://videotheque.inria.fr/videotheque/media/29998>. Title: “DomAssist : L’assistance numérique à la personne”. Produced in 2014.

5.4.2. Devices

Several entities have been identified to deliver an assistive support. These entities include (1) technological devices: wireless sensors (motion detectors, contact sensors and smart electric switches), and two tablets, and (2) software services (agenda, address book, mail agent, and photo agent) to monitor everyday activities and propose assistive applications. Sensors are placed in relevant rooms in the house: kitchen, bedroom, bathroom, and around the entrance.



Figure 7. HomeAssist devices

5.4.3. Experimental validation

A field study is currently being conducted with elderly people. The major purpose of this study is to identify the benefits of using HomeAssist for this population in an ecological framework. We selected 24 elderly people with different levels of autonomy (GIR scores). The HomeAssist technology has been installed in their house during 9 months. Twenty-four non-equipped elders were also selected to participate to the study, as control participants.

The expected impact of HomeAssist reflects the trans-disciplinary nature of the project. We aim to deliver results in the domain of (1) elderly care, (2) ergonomics and human factors, and (3) pervasive computing.

The major expected results are that HomeAssist (1) prolongs ageing in place, improves well-being of the users, and improves the efficiency of the caregiving environment; (2) is a cognitively low-cost assistive technology, and is well accepted and perceived as useful and usable by the users; (3) is technologically robust, and is a validated assistive platform

First results are expected in January 2015.

5.4.4. External Partners

The HomeAssist platform is being developed with support from the following partners:

- Équipe “Handicap et Système Nerveux” (EA 4136), Bordeaux University
- Chaire TSA, Université du Québec Trois-Rivières
- CRIUGM, Université de Montréal
- UDCCAS Gironde
- CARSAT
- Conseil Général 33
- Conseil Régional d’Aquitaine

5.5. DiaSwarm: a Development Environment for orchestrating smart objects at a large scale

Participants: Charles Consel [correspondent], Milan Kabac, Adrien Carteron, Eugène Volanschi.

The development of orchestrating applications which are responsible for large numbers of smart objects raises a number of challenges. We have addressed these by introducing a new design language called DiaSwarm, which is an extended version of the DiaSpec language.

5.5.1. Service discovery

Standard service discovery at the individual object level does not address the needs of applications orchestrating large numbers of smart objects. Instead, a high-level approach which provides constructs to specifying subsets of interest is needed. Our approach allows developers to introduce application-specific concepts (e.g., regrouping parking spaces into lots or districts) at the design time and then these can be used to express discovery operations. Following our design-driven development approach, these concepts are used to generate code to support and guide the programming phase.

5.5.2. Data gathering

Applications need to acquire data from a large number of objects through a variety of delivery models. For instance, air pollution sensors across a city may only push data to the relevant applications when pollution levels exceed tolerated levels. Tracking sensors, however, might determine the location of vehicles and send the acquired measurements to applications periodically (e.g., 10 min. intervals). Data delivery models need to be introduced at design time since they have a direct impact on the application's program structure. In doing so, the delivery models used by an application can be checked against sensor features early in the development process.

5.5.3. Data processing

Data that is generated from hundreds of thousands of objects and accumulated over a period of time calls for efficient processing strategies to ensure the required performance is attained. Our approach allows for an efficient implementation of the data processing stage by providing the developer with a framework based on the MapReduce [30] programming model which is intended for the processing of large data sets.

More details on this software platform can be found in the special issue on Smart Cities of the journal ERCIM News [18], 2014.

POTIOC Project-Team

5. New Software and Platforms

5.1. PapARt

Participant: J r my Laviolle [Main developer].

As part of his thesis work, J r my Laviolle has developed a software suite for PapARt : Paper Augmented Reality Toolkit. This work is being extended to become a state-of-the-art library for projection mapping (spatial augmented reality) and tangible interfaces.

Papart is a Processing⁰ library, and follows the Processing philosophy of openness and ease of use. The main features are :

1. Augmented reality rendering that enables rendering for cameras and projectors.
2. Tracking for Augmented Reality : tracking from ARToolkitPlus⁰, and OpenCV SURF⁰. Extensions with other libraries are planned.
3. Camera support : in addition to the Processing Video library, PapARt support for video from OpenCV, OpenKinect, FFMPEG, FlyCapture and more is available from JavaCV⁰. It provides a wide support on Mac, Windows, Linux and possibly Android.
4. "Tactile" input on planar surface : Touch and hovering can be detected by a depth camera such as Kinect⁰, the current support is Kinect XBOX 360 with openKinect drivers. Extensions are planned for Kinect 2, Kinect for Windows, Microsoft SDK (on Windows), DepthSense and most consumer market depth cameras. It scales elegantly from touch input from finger on small surfaces (such as A3 size) to large surfaces (2m x 3m size).
5. Software infrastructure to create "paper touch screens", following Processing's methods to create drawings and interactive experiences.

Technical challenges for the next few years :

- Color camera, depth camera and projector calibration made easy and more automated.
- Software and hardware installation of such cameras documented with tutorials and technical advices.

Research questions and challenges :

- Creation of tangible interfaces, tangibles elements can be tracked from cameras and depth cameras.
- Capture of part of pieces of paper for image analysis. E.g. to analyse and monitor drawings.
- Interactive projection mapping is an active research field, and such tools could power new research projects.

website: <http://papart.gforge.inria.fr>

5.2. OpenViBE

Participants: Fabien Lotte [local correspondent], Alison Cellard [engineer].

⁰<http://www.processing.org>

⁰<https://launchpad.net/artoolkitplus>

⁰<http://opencv.org/>

⁰<http://bytedeco.org/>

⁰<http://www.microsoft.com/en-us/kinectforwindows/>



Figure 4. Example of a 3D user interface where a child manipulates a 3D scene projected on a sheet of paper to prepare a drawing. Palais de la découverte, Paris, Nov. 2011.

As part of our research work on BCI, we contribute to the development of the OpenViBE⁰ 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 is one of the Inria team contributing to its evolution. Moreover, Potioc is involved in the Inria ADT (Technological Development Action) OpenViBE-NT and OpenViBE-X that is dedicated to the development of OpenViBE together with 3 other Inria teams (Hybrid, Athena, Neurosys). In 2014, we developed new EEG signal processing modules for the OpenViBE software (connectivity measured, wavelets, signal denoising, etc.) and new EEG visualization tools. We also organized demonstrations and workshops about OpenViBE at international conferences (PhyCS 2014, International BCI conference 2014).

⁰<http://openvibe.inria.fr>

REALOPT Project-Team

5. New Software and Platforms

5.1. BaPCod – a generic Branch-and-Price Code

Participants: Issam Tahiri [Software Engineer], François Clautiaux, Boris Detienne, Pierre Pesneau, Ruslan Sadykov, François Vanderbeck [correspondant].

BaPCod is a prototype code that solves Mixed Integer Programs (MIP) by application of decomposition and reformulation approach (relying mostly on Dantzig-Wolfe reformulation techniques). The reformulated problem is solved using a branch-and-price-and-cut (column and cut 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) A modeling language to express a compact integer linear programming model of the application on hand.
- (ii) the automation of the Dantzig-Wolfe reformulation process. The user 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.
- (iii) a default column generation procedure with standard initialization and stabilization [1], [56] [88] [87] [77] and
- (iv) a default branching scheme that is generic to all applications [7],
- (v) default primal heuristics specially developed for use in a decomposition framework [61], [76], [89].

The prototype software was/is used as background solver in our application studies and local PhD thesis. It also serves as the framework for our comparative study in a Inria associated team project and our transfert projects (the prototype 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>.

RUNTIME Team

5. New Software and Platforms

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] [20], 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 30 000 lines of C.
- <http://www.open-mpi.org/projects/hwloc>

5.3. Network Locality

Participant: Brice Goglin.

- *Netloc Locality* (NETLOC) is a library that extends hwloc to network topology information by assembling hwloc knowledge of server internals within graphs of inter-node fabrics such as Ethernet or Infiniband.
- NETLOC targets the same challenges as hwloc but focuses on a wider spectrum by enabling cluster-wide solutions such process placement [21].
- NETLOC is developed in collaboration with University of Wisconsin in LaCrosse and Cisco, within the OPEN MPI project.
- NETLOC is composed of 15 000 lines of C. It was recently merged in the HWLOC repository was better integration.
- <http://netloc.org>

5.4. 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 8 000 lines of C. Its main contributor is Brice GOGLIN.
- <http://runtime.bordeaux.inria.fr/knem/>

5.5. 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 [45] or the recent iWarp [44] 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.gforge.inria.fr/>

5.6. StarPU

Participants: Olivier Aumage, Andra Hugo, Nathalie Furmento, Raymond Namyst, Marc Sergent, Samuel Thibault, Pierre-André Wacrenier.

- STARPU permits high performance libraries or compiler environments to exploit heterogeneous multicore machines possibly equipped with GPGPUs or Xeon Phi 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 architectures (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 *scheduling contexts* which can be used to partition computing resources. Scheduling contexts can be dynamically resized to optimize the allocation of computing resources among concurrently running libraries.
- STARPU provides integration in MPI clusters through a lightweight DSM over MPI.
- STARPU provides a scheduling platform, which makes it easy to implement and experiment with scheduling heuristics
- 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.
- STARPU has support for simulating the execution, by using the simgrid simulator, which allows to reproduce experiments on a remote system, or to even virtually modify the platform used to run the application
- STARPU has been extended to provide runtime support for the KLANG-OMP OpenMP compiler. StarPU's OpenMP runtime support is compliant with most OpenMP 3.0 constructs, and also supports new dependent tasks and accelerated targets constructs introduced by OpenMP 4.0.
- <http://runtime.bordeaux.inria.fr/StarPU/>

5.7. Klang-OMP

Participants: Olivier Aumage, Nathalie Furmento, Samuel Pitoiset, Samuel Thibault.

- The KLANG-OMP software is a source-to-source OpenMP compiler for languages C and C++. It is developed as part of the Inria development action "ADT K'STAR " jointly managed by Inria teams MOAIS (Inria Montbonnot) and RUNTIME (Inria Bordeaux - Sud-Ouest). The KLANG-OMP compiler translates OpenMP directives and constructs into API calls from the StarPU runtime system or the XKaapi runtime system (XKaapi is developed by the MOAIS team).
- The KLANG-OMP compiler is virtually fully compliant with OpenMP 3.0 constructs.
- The KLANG-OMP compiler supports OpenMP 4.0 dependent tasks and accelerated targets.

5.8. Kastors

Participants: Olivier Aumage, Nathalie Furmento, Samuel Pitoiset, Samuel Thibault.

- The KASTORS software is a suite of benchmarks for testing the performance of OpenMP compilers on codes making use of the new dependent tasks OpenMP 4.0 constructs. It is ported and maintained as part of the Inria development action "ADT K'STAR " jointly managed by Inria teams MOAIS (Inria Montbonnot) and RUNTIME (Inria Bordeaux - Sud-Ouest). It is constituted of well known computing kernels that have been ported on the OpenMP 4.0 dependent tasks model. The KASTORS suite has been introduced to the OpenMP community during the IWOMP 2014 conference [32].

5.9. NewMadeleine and PIOMan

Participant: Alexandre Denis.

- NEWMADELEINE is a 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, multiframe 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.
- PIOMAN is a generic framework to be used by communication libraries, that brings seamless asynchronous progression of communication by opportunistically using available cores. It uses system threads and thus is composable with any runtime system used for multithreading.
- PIOMAN is closely integrated with the NEWMADELEINE communication library and PadicoTM.
- The reference software development branch of the NEWMADELEINE software consists in 60 000 lines of code. NEWMADELEINE is available on various networking technologies: Myrinet, Infiniband, Quadrics and ETHERNET. It is developed and maintained by Alexandre DENIS.
- <http://pm2.gforge.inria.fr/newmadeleine/>

5.10. 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://pm2.gforge.inria.fr/PadicoTM/>

5.11. MAQAO

Participants: Denis Barthou, Olivier Aumage, Christopher Haine, James Tombi A Mba.

- 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 for Intel x86 and ARM architectures. For x86 architecture, it can insert 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 and ARM architectures. This model analyzes performance of the codes on the architectures and provides some feed-back hints on how to improve these codes, in particular for vector 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 European FP7 project "MontBlanc".

5.12. QIRAL

Participants: Denis Barthou, Olivier Aumage.

- 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.

5.13. 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.
- Some core algorithms are parallel to speed-up the execution.
- 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>.

SISTM Team

5. New Software and Platforms

5.1. New Software

5.1.1. *TcGSA*

An *R* package for the gene set analysis of longitudinal gene expression data sets. Under development, and soon to be available on the CRAN website, this package implements a Time-course Gene Set Analysis method and provides useful plotting functions facilitating the interpretation of the results.

5.2. Upgraded Software

5.2.1. *NIMROD*

We have written a specific program called NIMROD for estimating parameter of ODE based population models. It has been regularly updated. For instance, we have adapted the program for parallel computing, in collaboration with the MCIA (Mésocentre de calcul intensif Aquitain) facility, which makes available a large computer with more than 3000 cores. This program is described in [43]. Although the program is available on the ISPED website ⁰, it is not user-friendly and needs further improvement to be more widely used. By now, the users are the current or previous (Jérémy Guedj, Julia Drylewicz, Mélanie Prague) members of the team and close collaborators (Andrew Yates). Furthermore, as a validation step, it would need a head-to-head comparison with other available softwares. We bet that our program can be very competitive for parameter identification in ODE models with more than two compartments.

5.2.2. *marqLevAlg*

An *R* package for function optimization. Available on CRAN, this package performs a minimization of function based on the Marquardt-Levenberg algorithm. This package is really useful when the surface to optimize is non-strictly convex or far from a quadratic function. A new convergence criterion, the relative distance to maximum (RDM), allows the user to have a better confidence in the stopping points, other than basic algorithm stabilization.

5.2.3. *VSURF*

An *R* package for Variable Selection Using Random Forests. Available on CRAN, this package performs an automatic (meaning completely data-driven) variable selection procedure. Originally designed to deal with high dimensional data, it can also be applied to standard datasets.

5.2.4. *R2GUESS*

R2GUESS package is a wrapper of the GUESS (Graphical processing Unit Evolutionary Stochastic Search) program. GUESS is a computationally optimised C++ implementation of a fully Bayesian variable selection approach that can analyse, in a genome-wide context, single and multiple responses in an integrated way. The program uses packages from the GNU Scientific Library (GSL) and offers the possibility to re-route computationally intensive linear algebra operations towards the Graphical Processing Unit (GPU) through the use of proprietary CULA-dense library.

⁰<http://etudes.isped.u-bordeaux2.fr/BIOSTATISTIQUE/NIMROD/documentation/html/index.html>.