



RESEARCH CENTER

FIELD

**Algorithmics, Programming, Software and Architecture**

Activity Report 2014

# Section Application Domains

Edition: 2015-03-24



## ALGORITHMICS, COMPUTER ALGEBRA AND CRYPTOLOGY

|                                      |    |
|--------------------------------------|----|
| 1. ARIC Project-Team                 | 5  |
| 2. CAMEL Project-Team                | 6  |
| 3. CASCADE Project-Team              | 8  |
| 4. CRYPT Team (section vide)         | 10 |
| 5. GALAAD2 Team                      | 11 |
| 6. GEOMETRICA Project-Team           | 12 |
| 7. GRACE Project-Team                | 13 |
| 8. LFANT Project-Team (section vide) | 14 |
| 9. POLSYS Project-Team               | 15 |
| 10. SECRET Project-Team              | 16 |
| 11. SPECFUN Project-Team             | 17 |
| 12. VEGAS Project-Team               | 18 |

## ARCHITECTURE, LANGUAGES AND COMPILATION

|  |    |
|--|----|
| 13. ALF Project-Team                   | 19 |
| 14. ATEAMS Project-Team (section vide) | 20 |
| 15. CAIRN Project-Team                 | 21 |
| 16. CAMUS Team                         | 22 |
| 17. COMPSYS Project-Team               | 23 |
| 18. DREAMPAL Team (section vide)       | 24 |
| 19. GCG Team                           | 25 |
| 20. PAREO Project-Team                 | 26 |
| 21. POSTALE Team (section vide)        | 27 |
| 22. TASC Project-Team                  | 28 |

## EMBEDDED AND REAL-TIME SYSTEMS

|                          |    |
|--------------------------|----|
| 23. AOSTE Project-Team   | 29 |
| 24. CONVECS Project-Team | 31 |
| 25. HYCOMES Team         | 32 |
| 26. MUTANT Project-Team  | 33 |
| 27. PARKAS Project-Team  | 35 |
| 28. SPADES Team          | 36 |
| 29. TEA Project-Team     | 37 |

## PROOFS AND VERIFICATION

|  |    |
|--|----|
| 30. ANTIQUE Team                         | 38 |
| 31. CELTIQUE Project-Team (section vide) | 40 |
| 32. DEDUCTEAM Exploratory Action         | 41 |
| 33. ESTASYS Exploratory Action           | 42 |
| 34. GALLIUM Project-Team                 | 43 |
| 35. MARELLE Project-Team                 | 45 |
| 36. MEXICO Project-Team                  | 46 |
| 37. PARSIFAL Project-Team                | 48 |

|  |    |
|--|----|
| 38. PIR2 Project-Team (section vide) ..... | 50 |
| 39. SUMO Project-Team .....                | 51 |
| 40. TEMPO Team .....                       | 53 |
| 41. TOCCATA Project-Team .....             | 54 |
| 42. VERIDIS Project-Team .....             | 55 |
| SECURITY AND CONFIDENTIALITY               |    |
| 43. CARTE Project-Team .....               | 56 |
| 44. CASSIS Project-Team .....              | 58 |
| 45. COMETE Project-Team .....              | 60 |
| 46. DICE Team (section vide) .....         | 61 |
| 47. PRIVATICS Project-Team .....           | 62 |
| 48. PROSECCO Project-Team .....            | 64 |

## **ARIC Project-Team**

# **4. Application Domains**

## **4.1. Hardware Arithmetic**

The application domains of hardware arithmetic operators are

- digital signal processing;
- image processing;
- embedded applications;
- reconfigurable computing;
- cryptography.

## **4.2. Floating-point and Validated Numerics**

Our expertise on validated numerics is useful to analyze and improve, and guarantee the quality of numerical results in a wide range of applications including:

- scientific simulation;
- global optimization;
- control theory.

Much of our work, in particular the development of correctly rounded elementary functions, is critical to the

- reproducibility of floating-point computations.

## **4.3. Cryptography, Cryptology, Communication Theory**

Lattice reduction algorithms have direct applications in

- public-key cryptography.

Another interesting field of application is

- communications theory.

## CAMEL Project-Team

# 4. Application Domains

## 4.1. Cryptology

The first application domain for our research is cryptology. This includes cryptography (constructive side) and cryptanalysis (breaking systems). For the cryptanalysis part, although it has practical implications, we do not expect any transfer in the classical sense of the term: it is more directed to governmental agencies and the end-users who build their trust, based on the cryptanalysis effort. It is noteworthy that analysis documents from governmental agencies (see e.g., [31]) use cryptanalysis results as their key material.

### 4.1.1. Cryptography

Our cryptographic contributions are related to multiple facets of the large realm of curve-based cryptology. While it is quite clear that enough algorithms exist in order to provide cryptographers with elliptic curves having a suitably hard discrete logarithm (as found in cryptographic standards for instance), one must bear in mind that refinements of the requirements and extensions to curves of higher genus raise several interesting problems. Our work contributes to expanding the cryptographer's capabilities in these areas.

In the context of genus-2 curves, our work aims at two goals. First, improvements on the group law on selected curves yield better speed for the associated cryptosystems. The cryptographic primitives, and then the whole suite of cryptographic protocols built upon such curves would be accelerated. The second goal is the expansion of the set of curves that can be built given a set of desired properties. Using point counting algorithms for arbitrary curves, a curve offering a 128-bit security level, together with nice properties for fast arithmetic, has been computed by CAMEL [4]. Another natural target for construction of curves for cryptography is also the suitability of curves for pairings. We expect to be able to compute such curves.

Important objects related to the structure of genus-2 curves are the isogenies between their Jacobians. Computing such isogenies is a key point in understanding important underlying objects such as the endomorphism ring, and can be useful in various situations, including for cryptographic or cryptanalytic applications. The team has produced important results in this context [6], [2].

### 4.1.2. Cryptanalysis

Our research on cryptanalysis is important for the cryptographic industry: by detecting weak instances, and setting new records we contribute to the definition of recommended families of systems together with their key sizes. The user's confidence in a cryptographic primitive is also related to how well the underlying problem is studied by researchers.

In particular, our involvement in computations with "NFS-like" algorithms encompasses of course the task of assessing the computational limits for integer factorization (as was done by the team by factoring RSA-768 [5]) and discrete-logarithm computations (as was done by the team in 2013 for the field  $\text{GF}(2^{809})$  [15]). The impact of the former is quite clear as it concerns the RSA algorithm; record-sized computations attract broad interest and determine updates on key-length recommendations. The latter are particularly important for pairing-based cryptography, since, in this context, one naturally encounters discrete-logarithm problems in extension fields of large degree. To this regard the breakthrough provided by the new quasi-polynomial discrete logarithm [17] is of course of utmost importance.

## 4.2. Computer Algebra Systems

Some of our software libraries are being used by computer algebra systems. Most of those libraries are free software, with a license that allows proprietary systems to link them. This gives us a maximal visibility, with a large number of users.

### **4.2.1. Magma**

Magma is a very large computational algebra package. It provides a mathematically rigorous environment for computing with algebraic, number-theoretic, combinatoric, and geometric objects. It is developed in Sydney, by the team around John Cannon. It is non-commercial (in the sense that its goal is not to make profit), but is not freely distributed and is not open-source.

Several members of the team have visited Sydney — several years ago — to contribute to the development of Magma, by implementing their algorithms or helping in integrating their software. Our link to Magma exists also via the libraries it uses: it currently links GNU MPFR and GNU MPC for its floating-point calculations, and links GMP-ECM as part of its factorization suite.

### **4.2.2. Pari/GP**

Pari/GP is a computational number theory system that is composed of a C library and an interpreter on top of it. It is developed in Bordeaux, where Karim Belabas from the LFANT project-team is the main maintainer. Its license is GPL. Although we do not directly contribute to this package, we have good contact with the developers.

### **4.2.3. Sage**

Sage is a fairly large scale and open-source computer algebra system written in Python. Sage aggregates a large amount of existing free software, aiming at selecting the fastest free software package for each given task. The motto of Sage is that instead of “reinventing the wheel” all the time, Sage is “building the car”. To date, Sage links GNU MPFR, GMP-ECM, and GNU MPC as standard packages.

## **4.3. Standardization**

### **4.3.1. Floating-point arithmetic**

The IEEE 754 standard for floating-point arithmetic was revised in 2008. The main new features are some new formats for decimal computations, and the recommendation of correctly rounded transcendental functions. The new decimal formats should not have an impact on our work, since we either use integer-only arithmetic, or arbitrary-precision binary floating-point arithmetic through the GNU MPFR library.

A new standard (P1788) is currently under construction for interval arithmetic. We are not officially involved in this standard, but we follow the discussions, to check in particular that the proposed standard will also cover arbitrary precision (interval) arithmetic.

## CASCADE Project-Team

# 4. Application Domains

## 4.1. Privacy for the Cloud

Many companies have already started the migration to the Cloud and many individuals share their personal informations on social networks. While some of the data are public information, many of them are personal and even quite sensitive. Unfortunately, the current access mode is purely right-based: the provider first authenticates the client, and grants him access, or not, according to his rights in the access-control list. Therefore, the provider itself not only has total access to the data, but also knows which data are accessed, by whom, and how: privacy, which includes secrecy of data (confidentiality), identities (anonymity), and requests (obliviousness), should be enforced. Moreover, while high availability can easily be controlled, and thus any defect can immediately be detected, failures in privacy protection can remain hidden for a long time. The industry of the Cloud introduces a new implicit trust requirement: nobody has any idea at all of where and how his data are stored and manipulated, but everybody should blindly trust the providers. The providers will definitely do their best, but this is not enough. Privacy-compliant procedures cannot be left to the responsibility of the provider: however strong the trustfulness of the provider may be, any system or human vulnerability can be exploited against privacy. This presents too huge a threat to tolerate. *The distribution of the data and the secrecy of the actions must be given back to the users. It requires promoting privacy as a global security notion.*

In order to protect the data, one needs to encrypt it. Unfortunately, traditional encryption systems are inadequate for most applications involving big, complex data. Recall that in traditional public key encryption, a party encrypts data to a single known user, which lacks the expressiveness needed for more advanced data sharing. In enterprise settings, a party will want to share data with groups of users based on their credentials. Similarly, individuals want to selectively grant access to their personal data on social networks as well as documents and spreadsheets on Google Docs. Moreover, the access policy may even refer to users who do not exist in the system at the time the data is encrypted. Solving this problem requires an entirely new way of encrypting data.

A first natural approach would be **fully homomorphic encryption** (FHE, see above), but a second one is also **functional encryption**, that is an emerging paradigm for public-key encryption: it enables more fine-grained access control to encrypted data, for instance, the ability to specify a decryption policy in the ciphertext so that only individuals who satisfy the policy can decrypt, or the ability to associate keywords to a secret key so that it can only decrypt documents containing the keyword. Our work on functional encryption centers around two goals:

1. to obtain more efficient pairings-based functional encryption;
2. and to realize new functionalities and more expressive functional encryption schemes.

Another approach is **secure multi-party computation protocols**, where interactivity might provide privacy in a more efficient way. Recent implicit interactive proofs of knowledge can be a starting point. But stronger properties are first expected for improving privacy. They can also be integrated into new ad-hoc broadcast systems, in order to distribute the management among several parties, and eventually remove any trust requirements.

Strong privacy for the Cloud would have a huge societal impact since it would revolutionize the trust model: users would be able to make safe use of outsourced storage, namely for personal, financial and medical data, without having to worry about failures or attacks of the server.



## 4.2. Hardware Security

Cryptography is only one component of information security, but it is a crucial component. Without cryptography, it would be impossible to establish secure communications between users over insecure networks like the Internet. In particular, public-key cryptography (invented by Diffie and Hellman in 1976) enables to establish secure communications between users who have never met physically before. One can argue that companies like E-Bay or Amazon could not exist without public-key cryptography. Since 30 years the theory of cryptography has developed considerably. However cryptography is not only a theoretical science; namely at some point the cryptographic algorithms must be implemented on physical devices, such as PCs, smart cards or RFIDs. Then problems arise: in general smart cards and RFIDs have limited computing power and leak information through power consumption and electro-magnetic radiations. Similarly a PC can be exposed to various computer viruses which can leak private informations to a remote attacker. Such information leakage can be exploited by an attacker; this is called a **side-channel attack**. It is well known that a cryptographic algorithm which is perfectly secure in theory can be completely insecure in practice if improperly implemented.

In general, countermeasures against side-channel attacks are heuristic and can only make a particular implementation resist particular attacks. Instead of relying on ad-hoc security patches, a better approach consists in working in the framework of **provable security**. The goal is to prove that a cryptosystem does not only resist specific attacks but can resist any possible side-channel attack. As already demonstrated with cryptographic protocols, this approach has the potential to significantly increase the security level of cryptographic products. Recently the cryptography research community has developed new security models to take into account these practical implementation attacks; the most promising such model is called the **leakage-resilient model**.

Therefore, our goal is to define new security models that take into account any possible side-channel attack, and then to design new cryptographic schemes and countermeasures with a proven security guarantee against side-channel attacks.

**CRYPT Team (section vide)**

## **GALAAD2 Team**

# **4. Application Domains**

## **4.1. Shape modeling**

Geometric modeling is increasingly familiar for us (synthesized images, structures, vision by computer, Internet, ...). Nowadays, many manufactured objects are entirely designed and built by means of geometric software which describe with accuracy the shape of these objects. The involved mathematical models used to represent these shapes have often an algebraic nature. Their treatment can be very complicated, for example requiring the computations of intersections or isosurfaces (CSG, digital simulations, ...), the detection of singularities, the analysis of the topology, etc. Optimizing these shapes with respect to some physical constraints is another example where the choice of the models and the design process are important to lead to interesting problems in algebraic geometric modeling and computing. We propose the development of methods for shape modeling that take into account the algebraic specificities of these problems. We tackle questions whose answer strongly depends on the context of the application being considered, in direct relationship with the industrial contacts that we are developing in Computer Aided Geometric Design.

## **4.2. Shape processing**

Many problems encountered in the application of computer sciences start from measurement data, from which one wants to recover a curve, a surface, or more generally a shape. This is typically the case in image processing, computer vision or signal processing. This also appears in computer biology where the geometry of distances plays a significant role, for example, in the reconstruction from NMR (Nuclear Magnetic Resonance) experiments, or the analysis of realizable or accessible configurations. In another domain, scanners which tend to be more and more easily used yield large set of data points from which one has to recover a compact geometric model. We are working in collaboration with groups in agronomy on the problem of reconstruction of branching models (which represent trees or plants). We are investigating the application of algebraic techniques to these reconstruction problems. Geometry is also highly involved in the numerical simulation of physical problems such as heat conduction, ship hull design, blades and turbines analysis, mechanical stress analysis. We apply our algebraic-geometric techniques in the isogeometric approach which uses the same (B-spline) formalism to represent both the geometry and the solutions of partial differential equations on this geometry.

## **GEOMETRICA Project-Team**

# **4. Application Domains**

## **4.1. Application Domains**

- Medical Imaging
- Numerical simulation
- Geometric modeling
- Geographic information systems
- Visualization
- Data analysis
- Astrophysics
- Material physics

## GRACE Project-Team

# 4. Application Domains

## 4.1. Cryptography and Cryptanalysis

In the twenty-first century, cryptography plays two essential roles: it is used to ensure *security* and *integrity* of communications and communicating entities. Contemporary cryptographic techniques can be used to hide private data, and to prove that public data has not been modified; to provide anonymity, and to assert and prove public identities. The creation and testing of practical cryptosystems involves

1. The design of provably secure protocols;
2. The design and analysis of compact and efficient algorithms to implement those protocols, and to attack their underlying mathematical and computational problems;
3. The robust implementation of those algorithms in low-level software and hardware, and their deployment in the wild.

While these layers are interdependent, GRACE's cryptographic research is focused heavily on the middle layer: we design, implement, and analyze the most efficient algorithms for fundamental tasks in contemporary cryptography. Our "clients", in a sense, are protocol designers on the one hand, and software and hardware engineers on the other.

F. Morain and B. Smith work primarily on the number-theoretic algorithms that underpin the current state-of-the-art in public-key cryptography (which is used to establish secure connections, and create and verify digital signatures, among other applications). For example, their participation in the ANR CATREL project aims to give a realistic assessment of the security of systems based on the Discrete Logarithm Problem, by creating a free, open, algorithmic package implementing the fastest known algorithms for attacking DLP instances. This will have an extremely important impact on contemporary pairing-based cryptosystems, as well as legacy finite field-based cryptosystems. On a more constructive note, F. Morain's elliptic curve point counting and primality proving algorithms are essential tools in the everyday construction of strong public-key cryptosystems, while B. Smith's recent work on elliptic curves aims to improve the speed of curve-based cryptosystems (such as Elliptic Curve Diffie–Hellman key exchange, a crucial step in establishing secure internet connections) without compromising their security.

D. Augot, F. Levy-dit-Vehel, and A. Couvreur's research on codes has far-reaching applications in *code-based cryptography*. This is a field which is growing rapidly in importance—partly due to the supposed resistance of code-based cryptosystems to attacks from quantum computing, partly due to the range of new techniques on offer, and partly because the fundamental problem of parameter selection is relatively poorly understood. For example, A. Couvreur's work on filtration attacks on codes has an important impact on the design of code-based systems using wild Goppa codes or algebraic geometry codes, and on the choice of parameter sizes for secure implementations.

Coding theory also has important practical applications in the improvement of conventional symmetric cryptosystems. For example, D. Augot's recent work on MDS matrices via BCH codes gives a more efficient construction of optimal diffusion layers in block ciphers. Here we use combinatorial, non-algorithmic properties of codes, in the internals of designs of block ciphers.

While coding theory brings tools as above for the classical problems of encryption, authentication, and so on, it can also provide solutions to new cryptographic problems. This is classically illustrated by the use of Reed-Solomon codes in secret sharing schemes. Grace is involved in the study, construction and implementation of locally decodable codes, which have applications in quite a few cryptographic protocols : *Private Information Retrieval*, *Proofs of Retrievability*, *Proofs of Ownership*, etc.

**LFANT Project-Team (section vide)**

## **POLSYS Project-Team**

### **4. Application Domains**

#### **4.1. Cryptology**

We propose to develop a systematic use of structured systems in Algebraic Cryptanalysis. We want to improve the efficiency and to predict the theoretical complexity of such attacks. We plan to demonstrate the power of algebraic techniques in new areas of cryptography such as Algebraic Number Theory (typically, in curve based cryptography).

#### **4.2. Engineering sciences**

Solving polynomial systems over the reals arise as a critical issue in wide range of problems coming from engineering sciences (biology, physics, control theory, etc.). We will focus on developing general enough software that may impact on these domains with a particular focus on control theory

## **SECRET Project-Team**

# **4. Application Domains**

## **4.1. Domain**

Our main application domains are:

- cryptology, including classical cryptology and quantum cryptography,
- error-correcting codes, especially codes for quantum communications and fault-tolerant quantum computing,
- reverse-engineering of communication systems.



## **SPECFUN Project-Team**

# **4. Application Domains**

## **4.1. Experimental mathematics with special functions**

Applications in combinatorics and mathematical physics frequently involve equations of so high orders and so large sizes, that computing or even storing all their coefficients is impossible on existing computers. Making this tractable is another challenge of our project. The approach we believe in is to design algorithms of good, ideally quasi-optimal, complexity in order to extract precisely the required data from the equations, while avoiding the computationally intractable task of completely expanding them into an explicit representation.

Typical applications with expected high impact are the automatic discovery and proof of results in combinatorics and mathematical physics for which human proofs are currently unattainable.

## **VEGAS Project-Team**

### **3. Application Domains**

#### **3.1. Computer graphics**

We are interested in the application of our work to virtual prototyping, which refers to the many steps required for the creation of a realistic virtual representation from a CAD/CAM model.

When designing an automobile, detailed physical mockups of the interior are built to study the design and evaluate human factors and ergonomic issues. These hand-made prototypes are costly, time consuming, and difficult to modify. To shorten the design cycle and improve interactivity and reliability, realistic rendering and immersive virtual reality provide an effective alternative. A virtual prototype can replace a physical mockup for the analysis of such design aspects as visibility of instruments and mirrors, reachability and accessibility, and aesthetics and appeal.

Virtual prototyping encompasses most of our work on effective geometric computing. In particular, our work on 3D visibility should have fruitful applications in this domain. As already explained, meshing objects of the scene along the main discontinuities of the visibility function can have a dramatic impact on the realism of the simulations.

#### **3.2. Solid modeling**

Solid modeling, i.e., the computer representation and manipulation of 3D shapes, has historically developed somewhat in parallel to computational geometry. Both communities are concerned with geometric algorithms and deal with many of the same issues. But while the computational geometry community has been mathematically inclined and essentially concerned with linear objects, solid modeling has traditionally had closer ties to industry and has been more concerned with curved surfaces.

Clearly, there is considerable potential for interaction between the two fields. Standing somewhere in the middle, our project has a lot to offer. Among the geometric questions related to solid modeling that are of interest to us, let us mention: the description of geometric shapes, the representation of solids, the conversion between different representations, data structures for graphical rendering of models and robustness of geometric computations.

## **ALF Project-Team**

# **4. Application Domains**

## **4.1. Any computer usage**

The ALF team is working on the fundamental technologies for computer science: processor architecture and performance-oriented compilation. The research results have impacts on any application domain that requires high performance executions (telecommunication, multimedia, biology, health, engineering, environment ...), but also on many embedded applications that exhibit other constraints such as power consumption, code size and guaranteed response time. Our research activity implies the development of software prototypes.

**ATEAMS Project-Team (section vide)**

## CAIRN Project-Team

# 4. Application Domains

## 4.1. Panorama

**keywords:** telecommunications, wireless communications, wireless sensor networks, content-based image retrieval, video coding, intelligent transportation systems, automotive, security

Our research is based on realistic applications, in order to both discover the main needs created by these applications and to invent realistic and interesting solutions.

The high complexity of the **Next-Generation (4G) Wireless Communication Systems** leads to the design of real-time high-performance specific architectures. The study of these techniques is one of the main field of applications for our research, based on our experience on WCDMA for 3G implementation.

In **Wireless Sensor Networks (WSN)**, where each wireless node has to operate without battery replacement for a long time, energy consumption is the most important constraint. In this domain, we mainly study energy-efficient architectures and wireless cooperative techniques for WSN.

**Intelligent Transportation Systems (ITS)**, and especially Automotive Systems, more and more take advantage of information technology advances. While wireless transmissions allow a car to communicate with another one or even with road infrastructure, **automotive industry** can also propose driver assistance and more secure vehicles thanks to improvements in computation accuracy for embedded systems.

Other important fields will also be considered: hardware cryptographic and security modules, specialized hardware systems for the filtering of the network traffic at high-speed, high-speed true-random number generation for security, content-based image retrieval and video processing.

## 4.2. 4G Wireless Communication Systems

With the advent of the next generation (4G) broadband wireless communications, the combination of MIMO (Multiple-Input Multiple-Output) wireless technology with Multi-Carrier CDMA (MC-CDMA) has been recognized as one of the most promising techniques to support high data rate and high performance. Moreover, future mobile devices will have to propose interoperability between wireless communication standards (4G, WiMax ...) and then implement MIMO pre-coding, already used by WiMax standard. Finally, in order to maximize mobile devices lifetime and guarantee quality of services to consumers, 4G systems will certainly use cooperative MIMO schemes or MIMO relays. Our research activity focuses on MIMO pre-coding and MIMO cooperative communications with the aim of algorithmic optimization and implementation prototyping.

## 4.3. Wireless Sensor Networks

Sensor networks are a very dynamic domain of research due, on the one hand, to the opportunity to develop innovative applications that are linked to a specific environment, and on the other hand to the challenge of designing totally autonomous communicating objects. Cross-layer optimizations lead to energy-efficient architectures and cooperative techniques dedicated to sensor networks applications. In particular, cooperative MIMO techniques are used to decrease the energy consumption of the communications.

## 4.4. Multimedia processing

In multimedia applications, audio and video processing is the major challenge embedded systems have to face. It is computationally intensive with power requirements to meet. Video or image processing at pixel level, like image filtering, edge detection and pixel correlation or at block-level such as transforms, quantization, entropy coding and motion estimation have to be accelerated. We investigate the potential of reconfigurable architectures for the design of efficient and flexible accelerators in the context of multimedia applications.

## **CAMUS Team**

# **4. Application Domains**

## **4.1. Application domains**

Performance being our main objective, our developments' target applications are characterized by intensive computation phases. Such applications are numerous in the domains of scientific computations, optimization, data mining and multimedia.

Applications involving intensive computations are necessarily high energy consumers. However this consumption can be significantly reduced thanks to optimization and parallelization. Although this issue is not our prior objective, we can expect some positive effects for the following reasons:

- Program parallelization tries to distribute the workload equally among the cores. Thus an equivalent performance, or even a better performance, to a sequential higher frequency execution on one single core, can be obtained.
- Memory and memory accesses are high energy consumers. Lowering the memory consumption, lowering the number of memory accesses and maximizing the number of accesses in the low levels of the memory hierarchy (registers, cache memories) have a positive consequence on execution speed, but also on energy consumption.

## COMPSYS Project-Team

# 4. Application Domains

## 4.1. Compilers for Embedded Computing Systems

The previous sections described our main activities in terms of research directions, but also places Compsys within the embedded computing systems domain, especially in Europe. We will therefore not come back here to the importance, for industry, of compilation and embedded computing systems design.

In terms of application domain, the embedded computing systems we consider are mostly used for multimedia: phones, TV sets, game platforms, etc. But, more than the final applications developed as programs, our main application is the computer itself: how the system is organized (architecture) and designed, how it is programmed (software), how programs are mapped to it (compilation and high-level synthesis).

The industry that can be impacted by our research is thus all the companies that develop embedded processors, hardware accelerators (programmable or not), embedded systems, and those (the same plus other) that need software tools to map applications to these platforms, i.e., that need to use or even develop programming languages, program optimization techniques, compilers, operating systems. Compsys do not focus on all these critical parts, but our activities are connected to them.

## 4.2. Users of HPC Platforms and Scientific Computing

The convergence between embedded computing systems and high-performance computing (HPC) technologies offers new computing platforms and tools for the users of scientific computing (e.g., people working in numerical analysis, in simulation, modeling, etc.). The proliferation of “cheap” hardware accelerators and multicores makes the “small HPC” (as opposed to computing centers with more powerful computers, grid computing, and exascale computing) accessible to a larger number of users, even though it is still difficult to exploit, due to the complexity of parallel programming, code tuning, interaction with compilers, which result from the multiple levels of parallelism and of memories in the recent architectures. The link between compiler and code optimization research (as in Compsys) and such users are still to be reinforced, both to guarantee the relevance of compiler research efforts with respect to application needs, and to help users better interact with compiler choices and understand performance issues.

The support of Labex MILYON (through its thematic quarters, such as the thematic quarter on compilation we organized in 2013 <sup>0</sup>, or the upcoming 2016 thematic quarter on high-performance computing) and the activities of the LyonCalcul initiative <sup>0</sup> are means to get closer to users of scientific computing, even if it is too early to know if Compsys will indeed be directly helpful to them.

---

<sup>0</sup>Thematic quarter on compilation: <http://labexcompilation.ens-lyon.fr/>

<sup>0</sup>Lyon Calcul federation: <http://lyoncalcul.univ-lyon1.fr>

**DREAMPAL Team (section vide)**



## **GCG Team**

# **4. Application Domains**

## **4.1. Transfer**

The main industrial sector related to the research activities of GCG is the one of semi-conductor (programmable architectures spanning from embedded systems to servers). Obviously any computing application which has the objective of exploiting as much as possible the resources (in terms of high-performance but also low energy consumption) of the host architecture is intended to take advantage of advances in compiler and runtime technology. These applications are based over numerical kernels (linear algebra, FFT, convolution...) that can be adapted on a large spectrum of architectures. Members of GCG already maintain fruitful and strong collaborations with several companies such as STMicroelectronics, Kalray, plus a recent (not yet formal) collaboration with Intel.

## PAREO Project-Team

# 4. Application Domains

## 4.1. Application Domains

Beside the theoretical transfer that can be performed via the cooperations or the scientific publications, an important part of the research done in the *Pareo* project-team is published within software. *Tom* is our flagship implementation. It is available via the Inria Gforge (<http://gforge.inria.fr>) and is one of the most visited and downloaded projects. The integration of high-level constructs in a widely used programming language such as Java may have an impact in the following areas:

- Teaching: when (for good or bad reasons) functional programming is not taught nor used, *Tom* is an interesting alternative to exemplify the notions of abstract data type and pattern-matching in a Java object oriented course.
- Software quality: it is now well established that functional languages such as Caml are very successful to produce high-assurance software as well as tools used for software certification. In the same vein, *Tom* is very well suited to develop, in Java, tools such as provers, model checkers, or static analyzers.
- Symbolic transformation: the use of formal anchors makes possible the transformation of low-level data structures such as C structures or arrays, using a high-level formalism, namely pattern matching, including associative matching. *Tom* is therefore a natural choice each time a symbolic transformation has to be implemented in C or Java for instance. *Tom* has been successfully used to implement the Rodin simplifier, for the B formal method.
- Prototyping: by providing abstract data types, private types, pattern matching, rules and strategies, *Tom* allows the development of quite complex prototypes in a short time. When using Java as the host-language, the full runtime library can be used. Combined with the constructs provided by *Tom*, such as strategies, this procures a tremendous advantage.

One of the most successful transfer is certainly the use of *Tom* made by Business Objects/SAP. Indeed, after benchmarking several other rule based languages, they decided to choose *Tom* to implement a part of their software. *Tom* is used in Paris, Toulouse and Vancouver. The standard representation provided by *Tom* is used as an exchange format by the teams of these sites.

**POSTALE Team (section vide)**

## TASC Project-Team

# 4. Application Domains

## 4.1. Introduction

Constraint programming deals with the resolution of decision problems by means of rational, logical and computational techniques. Above all, constraint programming is founded on a clear distinction between, on the one hand the description of the constraints intervening in a problem, and on the other hand the techniques used for the resolution. The ability of constraint programming to handle in a flexible way heterogeneous constraints has raised the commercial interest for this paradigm in the early eighties. Among his fields of predilection, one finds traditional applications such as computer aided decision-making, scheduling, planning, placement, logistics or finance, as well as applications such as electronic circuits design (simulation, checking and test), DNA sequencing and phylogeny in biology, configuration of manufacturing products or web sites, formal verification of code.

## 4.2. Panorama

In 2014 the **TASC** team was involved in the following application domains:

- *Replanning* in industrial timetabling problems in a Labcom project with **Eurodécision**.
- *Planning and replanning* in Data Centres taking into account energy consumption in the EPOC (Energy Proportional and Opportunistic Computing system) project.
- *Packing complex shapes* in the context of a warehouse (NetWMS2 project).
- Building decision support system for *city development planning with evaluation of energy impacts* (**SUSTAINS** project).
- *Optimizing electricity production* in the context of the **Gaspard Monge call program for Optimisation and Operation Research**. We extract global constraints from daily energy production temporal series issued from all productions plants of **EDF**.

## AOSTE Project-Team

# 4. Application Domains

## 4.1. Multicore System-on-Chip design

Synchronous formalisms and GALS or multiclock extensions are natural model representations of hardware circuits at various abstraction levels. They may compete with HDLs (Hardware Description Languages) at RTL and even TLM levels. The main originality of languages built upon these models is to be based on formal *synthesis* semantics, rather than mere simulation forms.

The flexibility in formal Models of Computation and Communication allows specification of modular Latency-Insensitive Designs, where the interconnect structure is built up and optimized around existing IP components, respecting some mandatory computation and communication latencies prescribed by the system architect. This allows a real platform view development, with component reuse and timing-closure analysis. The design and optimization of interconnect fabric around IP blocks transform at modeling level an (untimed) asynchronous versions into a (scheduled) multiclock timed one.

Also, Network on Chip (NoC) design may call for computable switching patterns, just like computable scheduling patterns were used in (predictable) Latency-Insensitive Design. Here again formal models, such as Cyclo-static dataflow graphs and extended Kahn networks with explicit routing schemes, are modeling elements of choice for a real synthesis/optimization approach to the design of systems. New parallel architecture paradigms, such as GPU co-processors or Massively Parallel Processor Arrays (MPPA) form natural targets as NoC-based platforms.

Multicore embedded architecture platform may be represented as Marte UML component diagrams. The semantics of concurrent applications may also be represented as Marte behavior diagrams embodying precise MoCCs. Optimized compilations/syntheses rely on specific algorithms, and are represented as model transformations and allocation (of application onto architecture).

Our current work aims thus primarily at providing Theoretical Computer Science foundations to this domain of multicore embedded SoCs, with possibly efficient application in modeling, analysis and compilation wherever possible due to some natural assumptions. We also deal with a comparative view of Esterel and SystemC TLM for more practical modeling, and the relation between the Spirit IP-Xact interface standard in SoC domain with its Marte counterpart.

## 4.2. Automotive and avionic embedded systems

Model-Driven Engineering is in general well accepted in the transportation domains, where design of digital software and electronic parts is usually tightly coupled with larger aspects of system design, where models from physics are being used already. The formalisms **AADL** (for avionics) and **AutoSar** [55] (for automotive) are providing support for this, unfortunately not always with a clean and formal semantics. Thus there is a strong need here for approaches that bring closer together formal methods and tools on the one hand, engineering best practices on the other hand.

From a structural point of view AUTOSAR succeeded in establishing a framework that provides significant confidence in the proper integration of software components from a variety of distinct suppliers. But beyond those structural (interface) aspects, dynamic and temporal views are becoming more of a concern, so that AUTOSAR has introduced the AUTOSAR Specification of Timing Extension. AUTOSAR (discrete) timing models consist of timing descriptions, expressed by events and event chains, and timing constraints that are imposed on these events and event chains.

An important issue in all such formalisms is to mix in a single design framework heterogeneous time models and tasks: based on different timebases, with different triggering policy (event-triggered and time-triggered), and periodic and/or aperiodic tasks, with distinct periodicity if ever. Adequate modeling is a prerequisite to the process of scheduling and allocating such tasks onto complex embedded architectural platforms (see AAA approach in foundation section 3.3 ). Only then can one devise powerful synthesis/analysis/verification techniques to guide designers towards optimized solutions.

Traceability is also an important concern, to close the gap between early requirements and constraints modelling on the one hand, verification and correct implementation of these constraints at the different levels of the development on the other hand.

## CONVECS Project-Team

# 4. Application Domains

## 4.1. Application Domains

The theoretical framework we use (automata, process algebras, bisimulations, temporal logics, etc.) and the software tools we develop are general enough to fit the needs of many application domains. They are applicable to virtually any system or protocol that consists of distributed agents communicating by asynchronous messages. The list of recent case studies performed with the CADP toolbox (see in particular § 6.5) illustrates the diversity of applications:

- *Bioinformatics*: genetic regulatory networks, nutritional stress response, metabolic pathways,
- *Component-based systems*: Web services, peer-to-peer networks,
- *Databases*: transaction protocols, distributed knowledge bases, stock management,
- *Distributed systems*: virtual shared memory, dynamic reconfiguration algorithms, fault tolerance algorithms, cloud computing,
- *Embedded systems*: air traffic control, avionic systems, medical devices,
- *Hardware architectures*: multiprocessor architectures, systems on chip, cache coherency protocols, hardware/software codesign,
- *Human-machine interaction*: graphical interfaces, biomedical data visualization, plasticity,
- *Security protocols*: authentication, electronic transactions, cryptographic key distribution,
- *Telecommunications*: high-speed networks, network management, mobile telephony, feature interaction detection.

## HYCOMES Team

# 4. Application Domains

## 4.1. Modeling cyber-physical systems

Hybrid systems modeling plays a particular role in the design of cyber-physical systems, eg. systems mixing physical devices, computing platforms, communication buses and control and diagnosis software. A faithful modeling of the physical environment is a key element in a successful design of a cyber-physical system.

Several types of physical components can be found in a system, for example: mechanical, hydraulic or electrical. Component models should cover several viewpoints. For instance, the three viewpoints of an electronic device would be its electrical, thermal and reliability models. All these viewpoints interact, and it is not possible to analyze any of them in isolation. Let alone these complex cross-viewpoint interactions, modeling physics requires refined mathematics. For instance, it is a misconception to assume that physical laws result in smooth dynamics that can be captured by systems of ordinary differential equations. On the contrary, physics is often nonsmooth, meaning that trajectories may be discontinuous — consider the example of colliding billiard balls. Physical systems are networks of elementary components. The dynamics of each component can often be captured by a simple (differential) equation. However, these (differential) equations are coupled by network equations (Kirchhoff laws, mechanical couplings, ...) resulting from the structure of the system. The end result, is a system mixing differential equations with linear or algebraic constraints: a system of differential algebraic equations (DAE).

The Hycomes team is focusing on the design of hybrid systems modeling languages with DAE and nonsmooth dynamics (Fillipov differential inclusions, or complementarity systems), with applications in the energy industry (power plants, smart grids), and in the railway, automotive and aeronautic industries — see section 3.1 for a deeper insight on the research program.

## 4.2. Supporting requirements engineering for aeronautic systems design

The design of embedded systems onboard certified civil aircrafts, for instance navigation, fly-by-wire and FADEC (Full Authority Digital Engine Control) applications, has to follow a stringent discipline imposed by civil aviation authorities. Designers have to provide evidence that both the design process they used and the system under design meet several industry standards, including the well-known ED-79/ARP-4754A and the DO-178 A/B regarding hardware and software artifacts.

These standards prescribe that every feature of a design can be traced back to one or several system-level requirements. Conversely, evidence shall be provided that every requirement has been accounted for. Correctness, consistency, compatibility and completeness of requirements are four key properties described in the ED-79/ARP-4754A standard that should also be assessed every time requirements are transformed. This puts a high burden on designers, especially on the system architect: requirements capture and analysis is by large a heuristic and manual process.

Formal requirements engineering is in its infancy: mathematical models, formal analysis techniques and links to system implementation must be developed. We advocate the use of contract-based reasoning techniques (see section 3.3 ) to support requirements engineering activities, during the early stages of the design process [4].



## MUTANT Project-Team

### 4. Application Domains

#### 4.1. Authoring and Performing Interactive Music

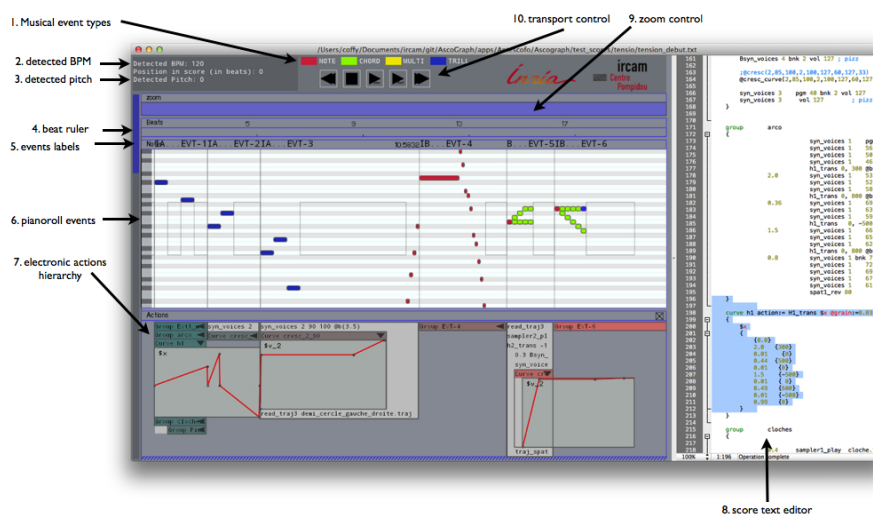


Figure 2. Screenshot of Ascograph, the Antescofo graphical score editor

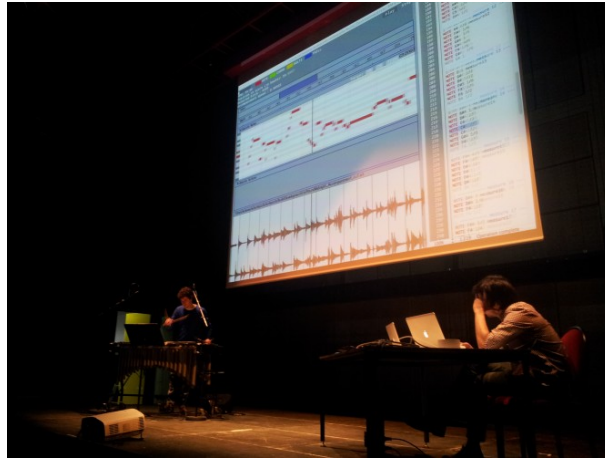
The combination of both realtime machine listening systems and reactive programming paradigms has enabled the *authoring* of interactive music systems as well as their realtime performance within a coherent synchronous framework called *Antescofo*. The module, developed since 2008 by the team members, has gained increasing attention within the user community worldwide with more than 40 prestigious public performances yearly. The outcomes of the teams's research will enhance the interactive and reactive aspects of this emerging paradigm as well as creating novel authoring tool for such purposes.

The *AscoGraph* authoring environment, started in 2013 and shown in Figure 2, is the first step towards such authoring environments. The outcome of the *ANR Project INEDIT* (with LABRI and GRAME and coordinated by team leader), will further extend the use-cases of *Antescofo* for interactive multimedia pieces with more complex temporal structures and computational paradigms.

#### 4.2. Realtime Music Information Retrieval

Realtime Music Information Retrieval is used as front-end for various applications requiring sonic interaction between software/hardware and the physical worlds. MuTant has focused on realtime machine listening since its inception and holds state-of-the-art algorithms for realtime alignment of audio to symbolic score, realtime tempo detection, realtime multiple-pitch extraction. Recent results have pushed our application to more generalised listening schemes beyond music signals as reported in [26].

### 4.3. Automatic Accompaniment/Creative Tools for Entertainment Industry



*Figure 3. Automatic Accompaniment Session with Antescofo during ACM CHI 2013 Conference*

Technologies developed by MuTant can find their way with general public (besides professional musicians) and within the entertainment industry. Recent trends in music industry show signs of tendencies towards more intelligent and interactive interfaces for music applications. Among them is reactive and adaptive automatic accompaniment and performance assessment as commercialized by companies such as *MakeMusic*. Technologies developed around *Antescofo* can enhance interaction between user and the computer for such large public applications.

Highlights in 2014 include collaborations with Orchestre de Paris Archives that resulted in prototype **demonstrated to public** in June 2014.

We will pursue this by licensing our technologies to third-party companies.

## **PARKAS Project-Team**

# **4. Application Domains**

## **4.1. Provably safe and efficient computing systems**

The project addresses the design, semantics and implementation of programming languages together with compilation techniques to develop provably safe and efficient computing systems. Traditional applications can be found in safety critical embedded systems with hard real-time constraints such as avionics (e.g., fly-by-wire command), railways (e.g., on board control, engine control), nuclear plants (e.g., emergency control of the plant). While embedded applications have been centralized, they are now massively parallel and physically distributed (e.g., sensor networks, train tracking, distributed simulation of factories) and they integrate computationally intensive algorithms (e.g., video processing) with a mix of hard and soft real-time constraints. Finally, systems are heterogeneous with discrete devices communicating with physical ones (e.g., interface between analog and digital circuits). Programming and simulating a whole system from a unique source code, with static guarantees on the reproducibility of simulations together with a compiler to generate target embedded code is a scientific and industrial challenge of great importance.

## SPADES Team

# 4. Application Domains

## 4.1. Industrial Applications

Our applications are in the embedded system area, typically: transportation, energy production, robotics, telecommunications, systems on chip (SoC). In some areas, safety is critical, and motivates the investment in formal methods and techniques for design. But even in less critical contexts, like telecommunications and multimedia, these techniques can be beneficial in improving the efficiency and the quality of designs, as well as the cost of the programming and the validation processes.

Industrial acceptance of formal techniques, as well as their deployment, goes necessarily through their usability by specialists of the application domain, rather than of the formal techniques themselves. Hence, we are looking to propose domain-specific (but generic) realistic models, validated through experience (*e.g.*, control tasks systems), based on formal techniques with a high degree of automation (*e.g.*, synchronous models), and tailored for concrete functionalities (*e.g.*, code generation).

## 4.2. Industrial Design Tools

The commercially available design tools (such as UML with real-time extensions, MATLAB/ SIMULINK/ dSPACE<sup>0</sup>) and execution platforms (OS such as VXWORKS, QNX, real-time versions of LINUX ...) start now to provide besides their core functionalities design or verification methods. Some of them, founded on models of reactive systems, come close to tools with a formal basis, such as for example STATEMATE by iLOGIX.

Regarding the synchronous approach, commercial tools are available: SCADE<sup>0</sup> (based on LUSTRE), CONTROLBUILD and RT-BUILDER (based on SIGNAL) from GEENSY<sup>0</sup> (part of DASSAULTSYSTEMES), specialized environments like CELLCONTROL for industrial automatism (by the INRIA spin-off ATHYS– now part of DASSAULTSYSTEMES). One can observe that behind the variety of actors, there is a real consistency of the synchronous technology, which makes sure that the results of our work related to the synchronous approach are not restricted to some language due to compatibility issues.

## 4.3. Current Industrial Cooperations

Regarding applications and case studies with industrial end-users of our techniques, we cooperate with STMicroelectronics on dynamic data-flow models of computation for streaming applications, dedicated to high definition video applications for their new STHORM manycore chip.

---

<sup>0</sup><http://www.dspaceinc.com>

<sup>0</sup><http://www.esterel-technologies.com>

<sup>0</sup><http://www.geensoft.com>

## TEA Project-Team

# 4. Application Domains

## 4.1. Application Domains

From our continuous collaboration with major academic and industrial partners through projects TOPCASED, OPENEMBEDD, SPACIFY, CESAR, OPEES, P and CORAIL, our experience has primarily focused on the aerospace domain. The topics of time and architecture of team TEA extend to both avionics and automotive, as demonstrated from this section to section 8. Yet, the research focus on time in team TEA is central in any aspect of, cyber-physical, embedded system design in automotive, music synthesis, signal processing, software radio, circuit and system on a chip design; many application domains which, should more collaborators join the team, would definitely be worth investigating.

Nonetheless, the application domains of our two direct collaborations with industry, avionics with Thales and automotive Toyota, are perfectly in line with the research objectives of team TEA and will allow us to quickly stream our theoretical results onto software and standards, which we will continue to distribute in open-source.

Multi-scale, multi-aspect time modelling, analysis and software synthesis will greatly contribute to architecture modelling in these domains, with applications to optimised (distributed, parallel, multi-core) code generation for avionics (our project with Thales avionics, section 8) as well as modelling standards, real-time simulation and virtual integration in automotive (our project with Toyota, section 8).

Together with the importance of open-source software, one of these project, the FUI Project P, demonstrated that a centralised model for system design could not just be a domain-specific programming language, such as discrete Simulink data-flows or a synchronous language. Synchronous languages implement a fixed model of time using logical clocks that are abstraction of time as sensed by software. They correspond to a fixed viewpoint in system design, and in a fixed hardware location in the system, which is not adequate to our purpose and must be extended.

In project P, we first tried to define a centralised model for importing discrete-continuous models onto a simplified implementation of SIMULINK: P models. Certified code generators would then be developed from that format. Because this does not encompass all aspects being translated to P, the P meta-model is now being extended to architecture description concepts (of the AADL) in order to become better suited for the purpose of system design. Another example is the development of System Modeller on top of SCADE, which uses the more model-engineering flavoured formalism SysML to try to unambiguously represent architectures around SCADE modules.

An abstract specification formalism, capable of representing time, timing relations, with which heterogeneous models can be abstracted, from which programs can be synthesised, naturally appears better suited for the purpose of virtual prototyping. RT-Builder, developed by TNI, was industrially proven and deployed for that purpose at Peugeot. It served to develop the virtual platform simulating all onboard electronics of PSA cars. This ‘hardware in the loop’ simulator was used to test equipments supplied by other manufacturers with respect to virtual cars. In the advent of the related automotive standard, RT-Builder then became AUTOSAR-Builder.

RT-Builder is the commercial implementation of Signal, whose industrial transfer with TNI was realised in the 90s by Paul Le Guernic and Albert Benveniste. As its actual industry usage has demonstrated, it is clear that the synchronous multi-clocked, or polychronous MoCC of Signal is an appropriate semantic core for the design of embedded software architectures.

## ANTIQUÉ Team

# 4. Application Domains

## 4.1. Verification of safety critical embedded software

The verification of safety critical embedded software is a very important application domain for our group. First, this field requires a high confidence in software, as a bug may cause disastrous events. Thus, it offers an obvious opportunity for a strong impact. Second, such software usually have better specifications and a better design than many other families of software, hence are an easier target for developing new static analysis techniques (which can later be extended for more general, harder to cope with families of programs). This includes avionics, automotive and other transportation systems, medical systems...

For instance, the verification of avionics systems represent a very high percentage of the cost of an airplane (about 30 % of the overall airplane design cost). The state of the art development processes mainly resort to testing in order to improve the quality of software. Depending on the level of criticality of a software (at highest levels, any software failure would endanger the flight) a set of software requirements are checked with test suites. This approach is both costly (due to the sheer amount of testing that needs to be performed) and unsound (as errors may go unnoticed, if they do not arise on the test suite).

By contrast, static analysis can ensure higher software quality at a lower cost. Indeed, a static analyzer will catch all bugs of a certain kind. Moreover, a static analysis run typically lasts a few hours, and can be integrated in the development cycle in a seamless manner. For instance, **ASTRÉE** successfully verified the absence of runtime error in several families of safety critical fly-by-wire avionic software, in at most a day of computation, on standard hardware. Other kinds of synchronous embedded software have also been analyzed with good results.

In the future, we plan to greatly extend this work so as to verify *other families of embedded software* (such as communication, navigation and monitoring software) and *other families of properties* (such as security and liveness properties).

Embedded software in charge of communication, navigation, monitoring typically rely on a *parallel* structure, where several threads are executed in parallel, and manage different features (input, output, user interface, internal computation, logging...). This structure is also often found in automotive software. An even more complex case is that of *distributed* systems, where several separate computers are run in parallel and take care of several sub-tasks of a same feature, such as braking. Such a logical structure is not only more complex than the synchronous one, but it also introduces new risks and new families of errors (deadlocks, data-races...). Moreover, such less well designed, and more complex embedded software often utilizes more complex data-structures than synchronous programs (which typically only use arrays to store previous states) and may use dynamic memory allocation, or build dynamic structures inside static memory regions, which are actually even harder to verify than conventional dynamically allocated data structures. Complex data-structures also introduce new kinds of risks (the failure to maintain structural invariants may lead to runtime errors, non termination, or other software failures). To verify such programs, we will design additional abstract domains, and develop new static analysis techniques, in order to support the analysis of more complex programming language features such as parallel and concurrent programming with threads and manipulations of complex data structures. Due to their size and complexity, the verification of such families of embedded software is a major challenge for the research community.

Furthermore, embedded systems also give rise to novel security concerns. It is in particular the case for some aircraft-embedded computer systems, which communicate with the ground through untrusted communication media. Besides, the increasing demand for new capabilities, such as enhanced on-board connectivity, e.g. using mobile devices, together with the need for cost reduction, leads to more integrated and interconnected systems. For instance, modern aircrafts embed a large number of computer systems, from safety-critical cockpit avionics to passenger entertainment. Some systems meet both safety and security requirements.

Despite thorough segregation of subsystems and networks, some shared communication resources raise the concern of possible intrusions. Because of the size of such systems, and considering that they are evolving entities, the only economically viable alternative is to perform automatic analyses. Such analyses of security and confidentiality properties have never been achieved on large-scale systems where security properties interact with other software properties, and even the mapping between high-level models of the systems and the large software base implementing them has never been done and represents a great challenge. Our goal is to prove empirically that the security of such large scale systems can be proved formally, thanks to the design of dedicated abstract interpreters.

The long term goal is to make static analysis more widely applicable to the verification of industrial software.

## 4.2. Static analysis of software components and libraries

An important goal of our work is to make static analysis techniques easier to apply to wider families of software. Then, in the longer term, we hope to be able to verify less critical, yet very commonly used pieces of software. Those are typically harder to analyze than critical software, as their development process tends to be less rigorous. In particular, we will target operating systems components and libraries. As of today, the verification of such programs is considered a major challenge to the static analysis community.

As an example, most programming languages offer Application Programming Interfaces (API) providing ready-to-use abstract data structures (e.g., sets, maps, stacks, queues, etc.). These APIs, are known under the name of containers or collections, and provide off-the-shelf libraries of high level operations, such as insertion, deletion and membership checks. These container libraries give software developers a way of abstracting from low-level implementation details related to memory management, such as dynamic allocation, deletion and pointer handling or concurrency aspects, such as thread synchronization. Libraries implementing data structures are important building bricks of a huge number of applications, therefore their verification is paramount. We are interested in developing static analysis techniques that will prove automatically the correctness of large audience libraries such as Glib and Threading Building Blocks.

## 4.3. Biological systems

Computer Science takes a more and more important role in the design and the understanding of biological systems such as signaling pathways, self assembly systems, DNA repair mechanisms. Biology has gathered large data-bases of facts about mechanistic interactions between proteins, but struggles to draw an overall picture of how these systems work as a whole. High level languages designed in Computer Science allow to collect these interactions in integrative models, and provide formal definitions (i.e., semantics) for the behavior of these models. This way, modelers can encode their knowledge, following a bottom-up discipline, without simplifying *a priori* the models at the risk of damaging the key properties of the system. Yet, the systems that are obtained this way suffer from combinatorial explosion (in particular, in the number of different kinds of molecular components, which can arise at run-time), which prevents from a naive computation of their behavior.

We develop various abstract interpretation-based analyses, tailored to different phases of the modeling process. We propose automatic static analyses in order to detect inconsistencies in the early phases of the modeling process. These analyses are similar to the analysis of classical safety properties of programs. They involve both forward and backward reachability analyses as well as causality analyses, and can be tuned at different levels of abstraction. We also develop automatic static analyses so as to identify the key elements in the dynamics of these models. The results of these analyses are sent to another tool, which is used to automatically simplify the models. The correctness of this simplification process is proved by the means of abstract interpretation: this ensures formally that the simplification preserves the quantitative properties that have been specified beforehand by the modeler. The whole pipeline is parameterized by a large choice of abstract domains which exploits different features of the high level description of models.

**CELTIQUE Project-Team (section vide)**



## DEDUCTEAM Exploratory Action

### 4. Application Domains

#### 4.1. Safety of aerospace systems

In parallel with this effort in logic and in the development of proof checkers and automated theorem proving systems, we always have been interested in using such tools. One of our favorite application domain is the safety of aerospace systems. Together with César Muñoz' team in Nasa-Langley, we have proved the correctness of several geometric algorithms used in air traffic control.

This has led us sometimes to develop such algorithms ourselves, and sometimes to develop tools for automating these proofs.

#### 4.2. B-set theory

Set theory appears to be an appropriate theory for automated theorem provers based on Deduction modulo, in particular the several extensions of *Zenon* (*Super Zenon* and *Zenon Modulo*). Modeling techniques using set theory are therefore good candidates to assess these tools. This is what we have done with the *B* method whose formalism relies on set theory. A collaboration with *Siemens* has been developed to automatically verify the *B* proof rules of *Atelier B* [42]. From this work presented in the Doctoral dissertation of Mélanie Jacquél, the *Super Zenon* tool [13] [35] has been designed in order to be able to reason modulo the *B* set theory. As a sequel of this work, we contribute to the *BWare* project whose aim is to provide a mechanized framework to support the automated verification of *B* proof obligations coming from the development of industrial applications. In this context, we have recently designed *Zenon Modulo* [33], [34] (Pierre Halmagrand's PhD thesis, which has started on October 2013) to deal with the *B* set theory. In this work, the idea is to manually transform the *B* set theory into a theory modulo and provide it to *Zenon Modulo* in order to verify the proof obligations of the *BWare* project.

#### 4.3. Termination certificate verification

Termination is an important property to verify, especially in critical applications. Automated termination provers use more and more complex theoretical results and external tools (e.g. sophisticated SAT solvers) that make their results not fully trustable and very difficult to check. To overcome this problem, a language for termination certificates, called **CPF**, has been developed since several years now. Deducteam develops a formally certified tool, **Rainbow**, based on the Coq library **CoLoR**, that is able to automatically verify the correctness of such termination certificates.

## **ESTASYS Exploratory Action**

### **4. Application Domains**

#### **4.1. Application Domains**

In two years, ESTASYS should lead to the creation of a top class research team at Inria as well as to an interdisciplinary community of researchers and practitioners at the world level.

**ESTASYS sets the foundations for an engineering domain dedicated to SoS that will benefit the European software industry.** This is achieved by creating mathematical models that capture the computational power, autonomous decisions and complex stochastic and real-time dynamics of SoS. ESTASYS produces new decidability and complexity results, simulation-based techniques, and algorithms with correctness arguments. All aim at efficient reasoning about SoS and are traced back to case studies. **Our strategy to work in close collaboration with contact in industry will guarantee their wider adoption by the European software industry.**

In the near future, The ESTASYS-PLASMA toolset will be distributed as open source whenever possible, but will create a new market of tools for SoS.

## **GALLIUM Project-Team**

# **4. Application Domains**

## **4.1. High-assurance software**

A large part of our work on programming languages and tools focuses on improving the reliability of software. Functional programming, program proof, and static type-checking contribute significantly to this goal.

Because of its proximity with mathematical specifications, pure functional programming is well suited to program proof. Moreover, functional programming languages such as Caml are eminently suitable to develop the code generators and verification tools that participate in the construction and qualification of high-assurance software. Examples include Esterel Technologies's KCG 6 code generator, the Astrée static analyzer, the Caduceus/Jessie program prover, and the Frama-C platform. Our own work on compiler verification combines these two aspects of functional programming: writing a compiler in a pure functional language and mechanically proving its correctness.

Static typing detects programming errors early, prevents a number of common sources of program crashes (null references, out-of bound array accesses, etc), and helps tremendously to enforce the integrity of data structures. Judicious uses of generalized abstract data types (GADTs), phantom types, type abstraction and other encapsulation mechanisms also allow static type checking to enforce program invariants.

## **4.2. Software security**

Static typing is also highly effective at preventing a number of common security attacks, such as buffer overflows, stack smashing, and executing network data as if it were code. Applications developed in a language such as Caml are therefore inherently more secure than those developed in unsafe languages such as C.

The methods used in designing type systems and establishing their soundness can also deliver static analyses that automatically verify some security policies. Two examples from our past work include Java bytecode verification [62] and enforcement of data confidentiality through type-based inference of information flows and noninterference properties [67].

## **4.3. Processing of complex structured data**

Like most functional languages, Caml is very well suited to expressing processing and transformations of complex, structured data. It provides concise, high-level declarations for data structures; a very expressive pattern-matching mechanism to destructure data; and compile-time exhaustiveness tests. Therefore, Caml is an excellent match for applications involving significant amounts of symbolic processing: compilers, program analyzers and theorem provers, but also (and less obviously) distributed collaborative applications, advanced Web applications, financial modeling tools, etc.

## **4.4. Rapid development**

Static typing is often criticized as being verbose (due to the additional type declarations required) and inflexible (due to, for instance, class hierarchies that must be fixed in advance). Its combination with type inference, as in the Caml language, substantially diminishes the importance of these problems: type inference allows programs to be initially written with few or no type declarations; moreover, the OCaml approach to object-oriented programming completely separates the class inheritance hierarchy from the type compatibility relation. Therefore, the Caml language is highly suitable for fast prototyping and the gradual evolution of software prototypes into final applications, as advocated by the popular "extreme programming" methodology.

## 4.5. Teaching programming

Our work on the Caml language has an impact on the teaching of programming. Caml Light is one of the programming languages selected by the French Ministry of Education for teaching Computer Science in *classes préparatoires scientifiques*. OCaml is also widely used for teaching advanced programming in engineering schools, colleges and universities in France, the USA, and Japan.

## **MARELLE Project-Team**

### **4. Application Domains**

#### **4.1. Reliability of embedded software**

Software embedded in physical devices performs computations where the inputs are provided by measures and the outputs are transformed into actions performed by actuators. To improve the quality of these devices, we expect that all the computations performed in this kind of software will need to be made more and more reliable. We claim that formal methods can serve this purpose and we develop the libraries and techniques to support this claim. This implies that we take a serious look at how mathematics can be included in formal methods, especially concerning geometry and calculus.

#### **4.2. Security and Cryptography**

The modern economy relies on the possibility for every actor to trust the communications they perform with their colleagues, customers, or providers. We claim that this trust can only be built by a careful scrutiny of the claims made by all public protocols and software that are reproduced in all portable devices, computers, and internet infrastructure systems. We advocate the use of formal methods in these domains and we provide easy-to-use tools for cryptographers so that the formal verification of cryptographic algorithms can become routine and amenable to public scrutiny.

#### **4.3. Mathematics and Education**

As libraries for theorem provers evolve, they tend to cover an ever increasing proportion of the mathematical background expected from engineers and scientists of all domains. Because the content of a formally verified library is extremely precise and explicit, we claim that this will provide a new kind of material for teaching mathematics, especially useful in remote education.

## MEXICO Project-Team

### 4. Application Domains

#### 4.1. Telecommunications

**Participants:** Stefan Haar, Serge Haddad.

*MExICO*'s research is motivated by problems on system management in several domains:

- In the domain of service oriented computing, it is often necessary to insert some Web service into an existing orchestrated business process, e.g. to replace another component after failures. This requires to ensure, often actively, conformance to the interaction protocol. One therefore needs to synthesize *adaptators* for every component in order to steer its interaction with the surrounding processes.
- Still in the domain of telecommunications, the supervision of a network tends to move from out-of-band technology, with a fixed dedicated supervision infrastructure, to in-band supervision where the supervision process uses the supervised network itself. This new setting requires to revisit the existing supervision techniques using control and diagnosis tools.

We have participated in the Univerself Project (see below) on self-aware networks, and will be searching new cooperations.

#### 4.2. Transport Systems

**Participants:** Stefan Haar, Simon Theissing.

We participate in the project MIC on multi-modal transport systems with in the IRT *System X*, with academic partners UPMC, IFSTTAR and CEA, and several industrial partners including Alstom (project leader), COSMO and Renault. Transportation operators in an urban area need to plan, supervise and steer different means of transportation with respect to several criteria:

- Maximize capacity;
- guarantee punctuality and robustness of service;
- minimize energy consumption.

The systems must achieve these objectives not only under ideal conditions, but also be robust to perturbations (such as a major cultural or sport event creating additional traffic), modifications of routes (roadwork, accidents, demonstrations, ... ) and tolerant to technical failures. Therefore, systems must be enabled to raise appropriate alarms upon detection of anomalies, diagnose the type of anomaly and select the appropriate response.

While the above challenges belong already to the tasks of individual operators in the unimodal setting, the rise of and increasing demand for *multi-modal* transports forces to achieve these planning, optimization and control goals not in isolation, but in a cooperative manner, across several operators. The research task here is first to analyze the transportation system regarding the available means, capacities and structures, and so as to identify the impacting factors and interdependencies of the system variables. Based on this analysis, the task is to derive and implement robust planning, with tolerance to technical faults; diagnosis and control strategies that are optimal under several, possibly different, criteria (average case vs worst case performance, energy efficiency, etc.) and allow to adapt to changes e.g. from nominal mode to reduced mode, sensor failures, etc.

#### 4.3. Biological Systems

**Participants:** Stefan Haar, Serge Haddad, Stefan Schwoon, Thomas Chatain, Loïc Jezequel.

We have begun in 2014 to examine concurrency issues in systems biology, and are currently enlarging the scope of our research's applications in this direction. To see the context, note that in recent years, a considerable shift of biologists' interest can be observed, from the mapping of *static* genotypes to *gene expression*, i.e. the processes in which genetic information is used in producing functional products. These processes are far from being uniquely determined by the gene itself, or even jointly with static properties of the environment; rather, *regulation* occurs throughout the expression processes, with specific mechanisms increasing or decreasing the production of various products, and thus modulating the outcome. These regulations are central in understanding cell fate (how does the cell differentiate ? Do mutations occur ? etc), and progress there hinges on our capacity to analyse, predict, monitor and control complex and variegated processes. Our first step in this domain is related in the conference contribution [33], where we apply Petri net unfolding techniques for the efficient computation of *attractors* in a regulatory network; that is, to identify strongly connected reachability components that correspond to stable evolutions, e.g. of a cell that differentiates into a specific functionality (or mutation). This constitutes the starting point of a broader research with Petri net unfolding techniques in regulation. In fact, the use of *ordinary* Petri nets for capturing regulatory network (RN) dynamics overcomes the limitations of traditional RN models : those impose e.g. Monotonicity properties in the influence that one factor had upon another, i.e. always increasing or always decreasing, and were thus unable to cover all actual behaviours (see [76]). Rather, we follow the more refined model of boolean networks of automata, where the local states of the different factors jointly determine which state transitions are possible. For these connectors, ordinary PNs constitute a first approximation, improving greatly over the literature but leaving room for improvement in terms of introducing more refined logical connectors. Future work thus involves transcending this class of PN models. Via unfoldings, one has access – provided efficient techniques are available – to all behaviours of the model, rather than over-or under-approximations as previously. This opens the way to efficiently searching in particular for determinants of the cell fate : which attractors are reachable from a given stage, and what are the factors that decide in favor of one or the other attractor, etc. The list of potential applications in biology and medicine of such a methodology would be too long to reproduce here.

## PARSIFAL Project-Team

# 4. Application Domains

## 4.1. Integrating a model checker and a theorem prover

The goal of combining model checking with inductive and co-inductive theorem in a rather appealing one. The strengths of systems in these two different systems are strikingly different. A model checker is capable of exploring a finite space automatically: such a tool can repeatedly explores all possible cases for how a computational space can be explored. On the other hand, a theorem prover might be able to prove clever things about a search space. For example, a model checker could attempt to discover whether or not there exists a winning strategy for, say, tic-tac-toe while an inductive theorem prover might be able to prove that if there is a winning strategy from one board then there is a winning strategy from any symmetric version of that board. Of course, being about to combine proofs from these system could drastically reduce the state exploration and proof certificate that needs to be produced to prove the existence of winning strategies.

Our first step to providing an integration of model checking and (inductive) theorem proving was to develop a strong logic, we call  $\mathcal{G}$ , that extends intuitionistic logic with notions of least and greatest fixed points. We have developed the proof theory of this logic in earlier papers [3] [52]. We have now recently converted the Bedwyr system so that it formally accepts almost all definitions and statements of theorems that are accepted by the inductive theorem prover Abella. Thus, these two systems are proving theorems in the same logic and their theorems can now be shared.

The tabling mechanism of Bedwyr has been extended so that its it can make use of previously proved lemmas. Thus, when a goal to prove that some board position has a winning strategy, the lemma can to conclude yes if some symmetric board position is already in the table.

For more about recent progress on providing checkable proof certificates for model checking, see the web site for Bedwyr <http://slimmer.gforge.inria.fr/bedwyr/>.

## 4.2. Implementing trusted proof checkers

Traditionally, theorem provers—whether interactive or automatic—are usually monolithic: if any part of a formal development was to be done in a particular theorem prover, then all parts of it would need to be done in that prover. Increasingly, however, formal systems are being developed to integrate the results returned from several, independent and high-performing, specialized provers: see, for example, the integration of Isabelle with an SMT solver [51] as well as the Why3 and ESC/Java systems.

Within the Parsifal team, we have been working on foundational aspects of this problem of integrating different provers. As we have described above, we have been developing a formal framework for defining the semantics of proof evidence. We have also been working on building prototype checkers of proof evidence which are capable to executing such formal definitions. The proof definition language described in the papers [47], [46] is currently given an implementation in the  $\lambda$ Prolog programming language [69]. This initial implementation will be able to serve as a “reference” proof checker: others developing proof evidence definitions will be able to use this reference checker to make sure that they are getting their definitions to do what they expect.

Using  $\lambda$ Prolog as an implementation language has both good and bad points. The good points are that it is rather simple to confirm that the checker is, in fact, sound. The language also supports a rich set of abstracts which make it impossible to interfere with the code of the checker (no injection attacks are possible). On the negative side, however, the performance of our  $\lambda$ Prolog interpreters is lower than specially written checkers and kernels.



### 4.3. Trustworthy implementations of theorem proving techniques

Instead of integrating different provers by exchanging proof evidence and relying on a back-end proof-checker, another approach to integration consists in re-implementing the theorem proving techniques as proof-search strategies, on an architecture that guarantees correctness. Focused systems can serve as the basis of such an architecture, identifying points of choice and backtrack and providing primitives for the exploration of the search space. These form a trusted *Application Programming Interface* that can be used to program and experiment various proof-search heuristics without worrying about correctness. No proof-checking is needed if one trusts the implementation of the API.

Following the description, in this framework, of quantifier-free techniques such as DPLL(T) [2], we are now exploring how the architecture can be adapted to accommodate techniques that handle quantifiers. In particular, unification-based or triggers-based techniques [37], [49].

This approach has led to the development of the Psyche engine.

**PL.R2 Project-Team (section vide)**

## SUMO Project-Team

# 4. Application Domains

## 4.1. Telecommunication network management

The domain of autonomic network management, will remain an important playground for SUMO. It covers a wide variety of problems, ranging from distributed (optimal) control to distributed diagnosis, optimization, re-configuration, provisioning, etc. We have a long experience in model-based diagnosis, in particular distributed (active) diagnosis, and have recently proposed promising techniques for self-modeling. It consists in building the model of the managed network on the fly, guided by the needs of the diagnosis algorithm. This approach allows one to deal with potentially huge models, that are only described by their construction grammar, and discovered at runtime. Another important research direction concerns the management of “multi-resolution” models, that can be considered at different granularity levels. This feature is central to network design, but has no appropriate modeling formalism nor management approaches. This is a typical investigation field for abstraction techniques. Technology is ahead of theory in this domain since networks are already driven or programmed through management policies, that assign high level objectives to an abstract view of the network, leaving open the question of their optimal implementation. As a last topic of investigation, today management issues are no longer isolated within one operator, but range across several of them, up to the supported services, which brings game theory aspects into the picture.

## 4.2. Control of data centers

Data centers are another example of a large scale reconfigurable and distributed system: they are composed of thousands of servers on which Virtual Machines (VM) can be (de)activated, migrated, etc. depending on the requests of the customers, on the load of the servers and on the power consumption. Autonomic management functionalities already exist to deploy and configure applications in such a distributed environment. They can also monitor the environment and react to events such as failures or overloads and reconfigure applications and/or infrastructures accordingly and autonomously. To supervise these systems, Autonomic Managers (AM) can be deployed in order to apply administration policies of specific aspects to the different entities of a data center (servers, VM, web services, power supply, etc). These AMs may be implemented in different layers: the hardware level, the operating system level or the middleware level. Therefore several control loops may coexist, and they have to take globally consistent decisions to manage the trade-off between availability, performance, scalability, security and energy consumption. This leads to multi-criteria optimization and control problems in order to automatically derive controllers in charge of the coordination of the different AMs. We are relatively new on this topic, that will require more technical investment from us. But we are driven to it by both the convergence of IT and networking, by virtualization techniques that reach networks (see the growing research effort about network operating systems), and by the call for more automation in the management of clouds. We believe our experience in network management can help. Some members of SUMO are already involved in the ANR Ctrl-Green, which addresses the controller coordination problem. We are also in contact with the Myriads team, which research interests moved from OS for grids/clouds to autonomic methods. This is supported as well by the activities of b<>com, the local IRT, where some projects in cloud management and in networking may start joint activities.

## 4.3. Web services and distributed active documents

Data centric systems are already deployed, and our goal is not to design new languages, architectures, or standards for them, but rather to propose techniques for the verification and monitoring of the existing systems. A bottleneck is the complexity and heterogeneity of web-based systems, that make them difficult to model and analyze. However, one can still hope for some lightweight verification or monitoring techniques for some specific aspects, for example to check the absence of conflict of interest in a transaction system, to verify

(off line) and maintain (on line) the QoS, to prevent security breaches, etc. Safety aspects of WS are little addressed; any progress in that area would be useful. Besides, modeling issues are central for some applications of data centric systems. Collaborative work environments with shared active documents can be found in many domains ranging from banking, maintenance of critical systems, webstores... We consider that models for data driven systems can find applications in most of these application areas. Our approach will be to favor purely declarative approaches for the specification of such collaborative environments. We have contacts with Centre Pasteur in Yaoundé on the design of diseases monitoring systems in developing countries. Diseases monitoring systems can be seen as a collaborative edition work, where each actor in the system reports and aggregates information about cases he or she is aware of. This collaboration is an opportunity to confront our models to real situations and real users needs. Formally modeling such a large distributed system can be seen as a way to ensure its correctness. We also envision to promote this approach as a support for maintenance operations in complex environments (train transportation, aeronautics,...). We believe this framework can be useful both for the specification of distributed maintenance procedures, for circulating information and sharing processes across teams, but also for the analysis of the correctness of procedures, possibly for their optimization or redesign, and finally to automatically elaborate logs of maintenance operations. We are in contact with several major companies on these topics, for the maintenance application side. Other industrial contacts need to be built: we have preliminary contact with IBM (leader in business artifacts), and would like to establish relations with SAP (leader in service architectures).

## **TEMPO Team**

# **4. Application Domains**

## **4.1. Cyber Physical Systems**

The overall project is geared towards the development of complex heterogeneous cyber physical systems that require high reliability such as nuclear power, energy distribution, industry automation and transportation, where formal verification methods are necessary.

## **4.2. Simulation**

Simulation is relevant to most areas where complex embedded systems are used, not only to the semiconductor industry for System-on-Chip modeling, but also to any application where a complex hardware platform must be assembled to run the application software. It has applications for example in industry automation, digital TV, telecommunications and transportation.

## **4.3. Automated Test Generation**

Manual testing is tedious. Automated testing makes it possible to increase test coverage while also minimizing the amount of redundancy created by manual testing.

## TOCCATA Project-Team

# 4. Application Domains

## 4.1. Mission-Critical Software

The application domains we target involve safety-critical software, that is where a high-level guarantee of soundness of functional execution of the software is wanted. The domains of application include the following. For each of them we refer to our past or current actions, in particular in relations with projects, contracts and industrial partners. Currently our industrial collaborations mainly belong to the first of these domains, transportation.

- **Transportation** including aeronautics, railroad, space flight, automotive.

These domains were considered in the context of the ANR U3CAT project, led by CEA, in partnership with Airbus France, Dassault Aviation, Sagem Défense et Sécurité. It included proof of C programs via *Frama-C/Jessie/Why*, proof of floating-point programs, the use of the *Alt-Ergo* prover via CAVEAT tool (CEA) or *Frama-C/WP*. This action is continued in the new project Soprano.

Aeronautics is the main target of the Verasco project, led by Verimag, on the development of certified static analyzers, in partnership with Airbus.

The former FUI project Hi-Lite, led by Adacore company, uses *Why3* and *Alt-Ergo* as back-end to SPARK2014, an environment for verification of Ada programs. This is applied to the domain of aerospace (Thales, EADS Astrium). This action is continued in the new joint laboratory ProofInUse. A recent paper [71] provides an extensive list of applications of SPARK, a major one being the British air control management.

In the current ANR project BWare, we investigate the use of *Why3* and *Alt-Ergo* as an alternative back-end for checking proof obligations generated by *Atelier B*, whose main applications are railroad-related software ([http://www.methode-b.com/documentation\\_b/ClearSy-Industrial\\_Use\\_of\\_B.pdf](http://www.methode-b.com/documentation_b/ClearSy-Industrial_Use_of_B.pdf)), a collaboration with Mitsubishi Electric R&D Centre Europe (Rennes) and ClearSy (Aix-en-Provence).

- **Energy** is naturally an application in particular with our long-term partner CEA, in the context of U3CAT and Soprano projects.
- **Communications and Data** in particular in contexts with a particular need for security or confidentiality: smart phones, Web applications, health records, electronic voting, etc.

Part of the applications of SPARK [71] include verification of security-related properties, including cryptographic algorithms.

Our new AJACS project addresses issues related to security and privacy in web applications written in Javascript, also including correctness properties.

The Cubicle model checker modulo theories based on the *Alt-Ergo* SMT prover, in collaboration with Intel Strategic Cad Labs (Hillsboro, OR, USA) is particularly targeted to the verification of concurrent programs and protocols (<http://cubicle.lri.fr/>).

- **Medicine**, including diagnostic devices, computer-assisted surgery

Such applications involve techniques for control and command close to what is done in transportation. Moreover, in this context, there is a need for modeling using differential equations, finite elements, hybrid systems, which are considered in other projects of us: FastRelax, ELFIC, Cafein.

- **Financial applications, banking**

We add projects in the past about safety and security of smart cards, in collaboration with Gemalto (European project VerifiCard, two CIFRE theses). Banking is naturally a domain of application of techniques dealing with security and confidentiality already mentioned above.

## **VERIDIS Project-Team**

# **4. Application Domains**

## **4.1. Application Domains**

Distributed algorithms and protocols are found at all levels of computing infrastructure, from many-core processors and systems-on-chip to wide-area networks. We are particularly interested in novel paradigms, for example ad-hoc networks that underly mobile and low-power computing or overlay networks and peer-to-peer networking that provide services for telecommunication or cloud computing services. Computing infrastructure must be highly available and is ideally invisible to the end user, therefore correctness is crucial. One should note that standard problems of distributed computing such as consensus, group membership or leader election have to be reformulated for the dynamic context of these modern systems. We are not ourselves experts in the design of distributed algorithms, but work together with domain experts on the modeling and verification of these protocols. These cooperations help us focus on concrete algorithms and ensure that our work is relevant to the distributed algorithm community.

Formal verification techniques that we study can contribute to certify the correctness of systems. In particular, they help assert under which assumptions an algorithm or system functions as required. For example, the highest levels of the Common Criteria for Information Technology Security Evaluation require code analysis, based on mathematically precise foundations. While initially the requirements of certified development have mostly been restricted to safety-critical systems, the cost of unavailable services due to malfunctioning system components and software provides wider incentives for verification. For example, we are working on modeling and verifying medical devices that require closed-loop models of both the system and its environment.

## CARTE Project-Team

# 4. Application Domains

## 4.1. Computer Virology

### 4.1.1. *The theoretical track.*

It is rightful to wonder why there are only a few fundamental studies on computer viruses while it is one of the important flaws in software engineering. The lack of theoretical studies explains maybe the weakness in the anticipation of computer diseases and the difficulty to improve defenses. For these reasons, we do think that it is worth exploring fundamental aspects, and in particular self-reproducing behaviors.

### 4.1.2. *The virus detection track*

The crucial question is how to detect viruses or self-replicating malwares. Cohen demonstrated that this question is undecidable. The anti-virus heuristics are based on two methods. The first one consists in searching for virus signatures. A signature is a regular expression, which identifies a family of viruses. There are obvious defects. For example, an unknown virus will not be detected, like ones related to a 0-day exploit. We strongly suggest to have a look at the independent audit [45] in order to understand the limits of this method. The second one consists in analysing the behavior of a program by monitoring it. Following [47], this kind of methods is not yet really implemented. Moreover, the large number of false-positive implies this is barely usable. To end this short survey, intrusion detection encompasses virus detection. However, unlike computer virology, which has a solid scientific foundation as we have seen, the IDS notion of “malwares” with respect to some security policy is not well defined. The interested reader may consult [67].

### 4.1.3. *The virus protection track*

The aim is to define security policies in order to prevent malware propagation. For this, we need (i) to define what is a computer in different programming languages and setting, (ii) to take into consideration resources like time and space. We think that formal methods like rewriting, type theory, logic, or formal languages, should help to define the notion of a formal immune system, which defines a certified protection.

### 4.1.4. *The experimentation track*

This study on computer virology leads us to propose and construct a “high security lab” in which experiments can be done in respect with the French law.

## 4.2. Computations and Dynamical Systems

### 4.2.1. *Continuous computation theories*

Understanding computation theories for continuous systems leads to studying hardness of verification and control of these systems. This has been used to discuss problems in fields as diverse as verification (see e.g., [29]), control theory (see e.g., [37]), neural networks (see e.g., [68]), and so on. We are interested in the formal decidability of properties of dynamical systems, such as reachability [58], the Skolem-Pisot problem [33], the computability of the  $\omega$ -limit set [57]. Those problems are analogous to verification of safety properties.

Contrary to computability theory, complexity theory over continuous spaces is underdeveloped and not well understood. A central issue is the choice of the representation of objects by discrete data and its effects on the induced complexity notions. As for computability, it is well known that a representation is gauged by the topology it induces. However more structure is needed to capture the complexity notions: topologically equivalent representations may induce different classes of polynomial-time computable objects, e.g., developing a sound complexity theory over continuous structures would enable us to make abstract computability results more applicable by analysing the corresponding complexity issues. We think that the preliminary step towards such a theory is the development of higher-order complexity, which we are currently carrying out.



In contrast with the discrete setting, it is of utmost importance to compare the various models of computation over the reals, as well as their associated complexity theories. In particular, we focus on the General Purpose Analog Computer of Claude Shannon [69], on recursive analysis [74], on the algebraic approach [65] and on Markov computability [60]. A crucial point for future investigations is to fill the gap between continuous and discrete computational models. This is one deep motivation of our work on computation theories for continuous systems.

#### **4.2.2. Analysis and verification of adversary systems**

The other research direction on dynamical systems we are interested in is the study of properties of adversary systems or programs, i.e., of systems whose behavior is unknown or indistinct, or which do not have classical expected properties. We would like to offer proof and verification tools, to guarantee the correctness of such systems. On one hand, we are interested in continuous and hybrid systems. In a mathematical sense, a hybrid system can be seen as a dynamical system, whose transition function does not satisfy the classical regularity hypotheses, like continuity, or continuity of its derivative. The properties to be verified are often expressed as reachability properties. For example, a safety property is often equivalent to (non-)reachability of a subset of unsure states from an initial configuration, or to stability (with its numerous variants like asymptotic stability, local stability, mortality, etc ...). Thus we will essentially focus on verification of these properties in various classes of dynamical systems.

We are also interested in rewriting techniques, used to describe dynamic systems, in particular in the adversary context. As they were initially developed in the context of automated deduction, the rewriting proof techniques, although now numerous, are not yet adapted to the complex framework of modelization and programming. An important stake in the domain is then to enrich them to provide realistic validation tools, both in providing finer rewriting formalisms and their associated proof techniques, and in developing new validation concepts in the adversary case, i.e., when usual properties of the systems like, for example, termination are not verified. For several years, we have been developing specific procedures for property proofs of rewriting, for the sake of programming, in particular with an inductive technique, already applied with success to termination under strategies [48], [49], [50], to weak termination [51], sufficient completeness [53] and probabilistic termination [55]. The last three results take place in the context of adversary computations, since they allow for proving that even a divergent program, in the sense where it does not terminate, can give the expected results. A common mechanism has been extracted from the above works, providing a generic inductive proof framework for properties of reduction relations, which can be parametrized by the property to be proved [54], [56]. Provided program code can be translated into rule-based specifications, this approach can be applied to correctness proof of software in a larger context. A crucial element of safety and security of software systems is the problem of resources. We are working in the field of Implicit Computational Complexity. Interpretation based methods like Quasi-interpretations (QI) or sup-interpretations, are the approach we have been developing these last years [62], [63], [64]. Implicit complexity is an approach to the analysis of the resources that are used by a program. Its tools come essentially from proof theory. The aim is to compile a program while certifying its complexity.

## CASSIS Project-Team

# 4. Application Domains

## 4.1. Verification of Security Protocols

Security protocols such as SET, TLS and Kerberos, are designed for establishing the confidence of electronic transactions. They rely on cryptographic primitives, the purpose of which is to ensure integrity of data, authentication or anonymity of participants, confidentiality of transactions, etc.

Experience has shown that the design of those protocols is often erroneous, even when assuming that cryptographic primitives are perfect, i.e., that an encoded message cannot be decrypted without the appropriate key. An intruder can intercept, analyze and modify the exchanged messages with very few computations and therefore, for example, generate important economic damage.

Analyzing cryptographic protocols is complex because the set of configurations to consider is very large, and can even be *infinite*: one has to consider any number of sessions, any size of messages, sessions interleaving, some algebraic properties of encryption or data structures.

Our objective is to automatize as much as possible the analysis of protocols starting from their specification. This consists in designing a tool that is easy to use, enables the specification of a large number of protocols thanks to a standard high-level language, and can either look for flaws in a given protocol or check whether it satisfies a given property. Such a tool is essential for verifying existing protocols, but also for helping in designing new ones. For our tool to be easy to use, it has to provide a graphical interface allowing a user to easily perform push-button verification.

Our tools for verifying security protocols are available as components of the AVISPA and AVANTSSAR platforms.

## 4.2. Automated Boundary Testing from Formal Specifications

We have introduced a new approach for test generation from set-oriented formal specifications: the BZ-TT method. This method is based on Constraint Logic Programming (CLP) techniques. The goal is to test every operation of the system at every boundary state using all input boundary values of that operation. It has been validated in several industry case studies for smart card OS and application validation (GSM 11-11 standard [75] and Java Card Virtual Machine Transaction mechanism [77]), information system and for embedded software [85].

This test generation method can be summed up as follows: from the formal model, the system computes boundary values to create boundary states; test cases are generated by traversal of the state space with a preamble part (sequences of operations from the initial state to a boundary state), a body part (critical invocations), an identification part (observation and oracle state computation) and a post-amble part (return path to initial or boundary state). Then, an executable test script file is generated using a test pattern and a table of correspondence between abstract operations (from the model) and concrete ones. This approach differs in several main points from previous works (e.g., [81]): first, using boundary goals as test objectives avoids the complete construction of the reachability graph; second, this process is fully automated and the test engineer could just drive it at the boundary value computation level or for the path computation.

The BZ-TT method is fully supported by the BZ-Testing-Tools tool-set. This environment is a set of tools dedicated to animation and test cases generation from B, Z or State-Chart formal specifications. It is based on the CLPS constraint solver, able to simulate the execution of the specification. By execution, we mean that the solver computes a so-called constrained state by applying the pre- and post-condition of operations. A constrained state is a constraint store where state variables and also input and output variables support constraints.

One orientation of the current work is to go beyond the finiteness assumption limitations by using symbolic constraint propagation during the test generation process. Second orientation is to extend the result to object oriented specifications as UML/OCL. Third orientation is to adapt the method to security aspect.

### 4.3. Program Debugging and Verification

Catching bugs in programs is difficult and time-consuming. The effort of debugging and proving correct even small units of code can surpass the effort of programming. Bugs inserted while “programming in the small” can have dramatic consequences for the consistency of a whole software system as shown, e.g., by viruses which can spread by exploiting buffer overflows, a bug which typically arises while coding a small portion of code. To detect this kind of errors, many verification techniques have been put forward such as static analysis and software model checking.

Recently, in the program verification community, there seems to be a growing demand for more declarative approaches in order to make the results of the analysis readily available to the end user. To meet this requirement, a growing number of program verification tools integrate some form of theorem proving.

The goals of our research are twofold. First, we perform theoretical investigations of various combinations of propositional and first-order satisfiability checking in order to automate the theorem proving activity required to solve a large class of program analysis problems which can be encoded as first-order formulae. Second, we experimentally investigate how our techniques behave on real problems in order to make program analysis more precise and scalable. Building tools capable of providing a good balance between precision and scalability is one of the crucial challenges to transfer theorem proving technology to the industrial domains.

### 4.4. Verification of Web Services

Driven by rapidly changing requirements and business needs, IT systems and applications are undergoing a paradigm shift: components are replaced by services, distributed over the network, and composed and reconfigured dynamically in a demand-driven way into service-oriented architectures. Exposing services in future network infrastructures means a wide range of trust and security issues need to be addressed. Solving them is extremely hard since making the service components trustworthy is not sufficient: composing services leads to new subtle and dangerous vulnerabilities due to interference between component services and policies, the shared communication layer, and application functionality. Thus, one needs validation of both the service components and their composition into secure service architectures. In this context, there is an obvious need of applying formal methods. Our project aims at applying our proof and constraint solving techniques to reason on web services. More precisely, we focus on the composition problem in the presence of security policies.

### 4.5. Model-Checking of Collaborative Systems

Collaborative systems constitute a class of distributed systems where real human interactions are predominant. In these systems, users at geographically distributed sites interact by simultaneously manipulating shared objects like text documents, XML trees, filesystems, etc. To improve data availability, the shared objects are replicated so that the users update their local replicas and exchange their updates between them. One of the main challenges here is how to ensure the data consistency when the updates are executed in arbitrary orders at different replicas. Operational Transformation (OT) is an optimistic technique which has been proposed to overcome the consistency problem. This technique consists of an application-dependent protocol to enforce the out-of-order execution of updates even though these updates do not naturally commute. The data consistency relies crucially on the correctness of OT protocols whose proof is extremely hard. Indeed, possibly infinitely many cases should be tested. Our research work aims at applying symbolic model-checking techniques to automatically verify OT protocols. Most importantly, we are interested in finding under which conditions the model-checking problem can be reduced to a finite-state model.

## COMETE Project-Team

# 4. Application Domains

## 4.1. Security and privacy

**Participants:** Nicolas Bordenabe, Konstantinos Chatzikokolakis, Catuscia Palamidessi, Marco Stronati.

The aim of our research is the specification and verification of protocols used in mobile distributed systems, in particular security protocols. We are especially interested in protocols for *information hiding*.

Information hiding is a generic term which we use here to refer to the problem of preventing the disclosure of information which is supposed to be secret or confidential. The most prominent research areas which are concerned with this problem are those of *secure information flow* and of *privacy*.

Secure information flow refers to the problem of avoiding the so-called *propagation* of secret data due to their processing. It was initially considered as related to software, and the research focussed on type systems and other kind of static analysis to prevent dangerous operations, Nowadays the setting is more general, and a large part of the research effort is directed towards the investigation of probabilistic scenarios and treaths.

Privacy denotes the issue of preventing certain information to become publicly known. It may refer to the protection of *private data* (credit card number, personal info etc.), of the agent's identity (*anonymity*), of the link between information and user (*unlinkability*), of its activities (*unobservability*), and of its *mobility* (*untraceability*).

The common denominator of this class of problems is that an adversary can try to infer the private information (*secrets*) from the information that he can access (*observables*). The solution is then to obfuscate the link between secrets and observables as much as possible, and often the use randomization, i.e. the introduction of *noise*, can help to achieve this purpose. The system can then be seen as a *noisy channel*, in the information-theoretic sense, between the secrets and the observables.

We intend to explore the rich set of concepts and techniques in the fields of information theory and hypothesis testing to establish the foundations of quantitative information flow and of privacy, and to develop heuristics and methods to improve mechanisms for the protection of secret information. Our approach will be based on the specification of protocols in the probabilistic asynchronous  $\pi$ -calculus, and the application of model-checking to compute the matrices associated to the corresponding channels.

**DICE Team (section vide)**

## PRIVATICS Project-Team

### 3. Application Domains

#### 3.1. Domain 1: Privacy in smart environments.

One illustrative example is our latest work on privacy-preserving smart-metering [2]. Several countries throughout the world are planning to deploy smart meters in house-holds in the very near future. Traditional electrical meters only measure total consumption on a given period of time (i.e., one month or one year). As such, they do not provide accurate information of when the energy was consumed. Smart meters, instead, monitor and report consumption in intervals of few minutes. They allow the utility provider to monitor, almost in real-time, consumption and possibly adjust generation and prices according to the demand. Billing customers by how much is consumed and at what time of day will probably change consumption habits to help matching energy consumption with production. In the longer term, with the advent of smart appliances, it is expected that the smart grid will remotely control selected appliances to reduce demand. Although smart metering might help improving energy management, it creates many new privacy problems. Smart-meters provide very accurate consumption data to electricity providers. As the interval of data collected by smart meters decreases, the ability to disaggregate low-resolution data increases. Analysing high-resolution consumption data, Non-intrusive Appliance Load Monitoring (NALM) can be used to identify a remarkable number of electric appliances (e.g., water heaters, well pumps, furnace blowers, refrigerators, and air conditioners) employing exhaustive appliance signature libraries. We developed DREAM, Differentially privatE smArt Metering, a scheme that is private under the differential privacy model and therefore provides strong and provable guarantees. With our scheme, an (electricity) supplier can periodically collect data from smart-meters and derive aggregated statistics while learning only limited information about the activities of individual households. For example, a supplier cannot tell from a user's trace when he watched TV or turned on heating.

#### 3.2. Domain 2: Big Data and Privacy

We believe that another important problem will be related to privacy issues in big data. Public datasets are used in a variety of applications spanning from genome and web usage analysis to location-based and recommendation systems. Publishing such datasets is important since they can help us analyzing and understanding interesting patterns. For example, mobility trajectories have become widely collected in recent years and have opened the possibility to improve our understanding of large-scale social networks by investigating how people exchange information, interact, and develop social interactions. With billion of handsets in use worldwide, the quantity of mobility data is gigantic. When aggregated, they can help understand complex processes, such as the spread of viruses, and build better transportation systems. While the benefits provided by these datasets are indisputable, they unfortunately pose a considerable threat to individual privacy. In fact, mobility trajectories might be used by a malicious attacker to discover potential sensitive information about a user, such as his habits, religion or relationships. Because privacy is so important to people, companies and researchers are reluctant to publish datasets by fear of being held responsible for potential privacy breaches. As a result, only very few of them are actually released and available. This limits our ability to analyze such data to derive information that could benefit the general public. It is now an urgent need to develop Privacy-Preserving Data Analytics (PPDA) systems that collect and transform raw data into a version that is immunized against privacy attacks but that still preserves useful information for data analysis. This is one of the objectives of Privatics. There exists two classes of PPDA according to whether the entity that is collecting and anonymizing the data is trusted or not. In the trusted model, that we refer to as Privacy-Preserving Data Publishing (PPDP), individuals trust the publisher to which they disclose their data. In the untrusted model, that we refer to as Privacy-Preserving Data Collection (PPDC), individuals do not trust the data publisher. They may add some noise to their data to protect sensitive information from the data publisher.

*Privacy-Preserving Data Publishing:* In the trusted model, individuals trust the data publisher and disclose all their data to it. For example, in a medical scenario, patients give their true information to hospitals to receive proper treatment. It is then the responsibility of the data publisher to protect privacy of the individuals' personal data. To prevent potential data leakage, datasets must be sanitized before possible release. Several proposals have been recently proposed to release private data under the Differential Privacy model [25, 56, 26, 57, 50]. However most of these schemes release a "snapshot" of the datasets at a given period of time. This release often consists of histograms. They can, for example, show the distributions of some pathologies (such as cancer, flu, HIV, hepatitis, etc.) in a given population. For many analytics applications, "snapshots" of data are not enough, and sequential data are required. Furthermore, current work focusses on rather simple data structures, such as numerical data. Release of more complex data, such as graphs, are often also very useful. For example, recommendation systems need the sequences of visited websites or bought items. They also need to analyse people connection graphs to identify the best products to recommend. Network trace analytics also rely on sequences of events to detect anomalies or intrusions. Similarly, traffic analytics applications typically need sequences of visited places of each user. In fact, it is often essential for these applications to know that user A moved from position 1 to position 2, or at least to learn the probability of a move from position 1 to position 2. Histograms would typically represent the number of users in position 1 and position 2, but would not provide the number of users that moved from position 1 to position 2. Due to the inherent sequentiality and high-dimensionality of sequential data, one major challenge of applying current data sanitization solutions on sequential data comes from the uniqueness of sequences (e.g., very few sequences are identical). This fact makes existing techniques result in poor utility. Schemes to privately release data with complex data structures, such as sequential, relational and graph data, are required. This is one the goals of Privatics. In our current work, we address this challenge by employing a variable-length n-gram model, which extracts the essential information of a sequential database in terms of a set of variable-length n-grams [15]. We then intend to extend this approach to more complex data structures.

*Privacy-Preserving Data Collection:* In the untrusted model, individuals do not trust their data publisher. For example, websites commonly use third party web analytics services, such as Google Analytics to obtain aggregate traffic statistics such as most visited pages, visitors' countries, etc. Similarly, other applications, such as smart metering or targeted advertising applications, are also tracking users in order to derive aggregated information about a particular class of users. Unfortunately, to obtain this aggregate information, services need to track users, resulting in a violation of user privacy. One of our goals is to develop Privacy-Preserving Data Collection solutions. We propose to study whether it is possible to provide efficient collection/aggregation solutions without tracking users, i.e. without getting or learning individual contributions.

## PROSECCO Project-Team

# 4. Application Domains

## 4.1. Cryptographic Protocol Libraries

Cryptographic protocols such as TLS, SSH, IPsec, and Kerberos are the trusted base on which the security of modern distributed systems is built. Our work enables the analysis and verification of such protocols, both in their design and implementation. Hence, for example, we build and verify models and reference implementations for well-known protocols such as TLS and SSH, as well as analyze their popular implementations such as OpenSSL.

## 4.2. Hardware-based security APIs

Cryptographic devices such as Hardware Security Modules (HSMs) and smartcards are used to protect long-term secrets in tamper-proof hardware, so that even attackers who gain physical access to the device cannot obtain its secrets. These devices are used in a variety of scenarios ranging from bank servers to transportation cards (e.g. Navigo). Our work investigates the security of commercial cryptographic hardware and evaluates the APIs they seek to implement.

## 4.3. Web application security

Web applications use a variety of cryptographic techniques to securely store and exchange sensitive data for their users. For example, a website may serve pages over HTTPS, authenticate users with a single sign-on protocol such as OAuth, encrypt user files on the server-side using XML encryption, and deploy client-side cryptographic mechanisms using a JavaScript cryptographic library. The security of these applications depends on the public key infrastructure (X.509 certificates), web browsers' implementation of HTTPS and the same origin policy (SOP), the semantics of JavaScript, HTML5, and their various associated security standards, as well as the correctness of the specific web application code of interest. We build analysis tools to find bugs in all these artifacts and verification tools that can analyze commercial web applications and evaluate their security against sophisticated web-based attacks.