RESEARCH CENTER

FIELD
Algorithmics, Programming, Software and Architecture

Activity Report 2013

Section Application Domains

Edition: 2014-03-20

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.

CARAMEL 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 document from governmental agencies (see e.g. [29]) 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 CARAMEL [5]. 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 [7], [3]

Implementations of curve-based cryptography, both in hardware and software, are a necessary step on the way to assessing cryptographic speed. We plan to provide such implementations. In particular, on the hardware side, one of our goals is the design of a complete cryptographic coprocessor, including all the primitives for curve-based and pairing-based cryptography, providing optimized and configurable efficiency vs area trade-off. Such work has been proposed in [1].

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 [6]) and discrete-logarithm computations (as was done by the team in 2013 for the field $\mathrm{GF}(2^{809})$ [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 [22] 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 — a few 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 the goal of 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.

4.3. Copyright Protection

Similarly to the privacy concern, the digital world makes easy the large-scale diffusion of information. But in some cases, this can be used in violation of some copyrights. Cryptography should help at solving this problem, which is actually two-fold: one can either mark the original document in order to be able to follow the distribution (and possibly trace the traitor who illegally made it public) or one can publish information in an encrypted way, so that authorized people only can access it.

CRYPT Team

4. Application Domains

4.1. Security Estimates for Cryptography

An important application of cryptanalysis is to evaluate the concrete security of a given cryptosystem, so that key sizes and parameters are chosen appropriately. In some sense, cryptanalysis is the crash test of cryptography. When one uses cryptography, the first thing that one does is to select parameters and key sizes: in the real world, several well-known cryptographic failures happened due to inappropriate key sizes. Cryptanalysis analyzes the best attacks known: it assesses their cost (depending on the platform) and their performances (such as success probability). Sometimes the exact cost of an attack cannot be evaluated accurately nor rigorously, but fortunately, it is often possible to give an order of magnitude, which allows to select key sizes with a reasonable security margin.

On the other hand, it must be stressed that cryptanalysis depends on the state of the art: today's best attack may be completely different from tomorrow's best attack. The case of MD5 is a good reminder of this well-known fact.

4.2. Algorithmic Number Theory

Algorithms developed for cryptanalysis have sometimes applications outside cryptanalysis, especially in algorithmic number theory. This has happened for lattices and elliptic curves, and is not surprising, considering that some of the problems studied by cryptanalysis are very basic (like integer factoring), and therefore ubiquitous. Cryptanalysis motivates the search of truly-efficient algorithms, and experiments are common in public-key cryptanalysis, which allows to really verify improvements.

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 rôles: 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; and
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.

François Morain and Benjamin 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, Morain's elliptic curve point counting and primality proving algorithms are essential tools in the everyday construction of strong public-key cryptosystems, while 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.

Daniel Augot, Françoise Levy-dit-Vehel, and Alain 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, Couvreur's work on distinguishing codes has an important impact on the design of code-based systems built over algebraic geometry codes, and on the choice of parameter sizes for secure implementations. But coding theory also has important practical applications in the improvement of conventional symmetric cryptosystems. For example, Augot's recent work on MDS matrices via BCH codes gives a more efficient construction of optimal diffusion layers in block ciphers.

4.2. Privacy

While cryptography classically aims to provide confidentiality for messages during their transmission between a sender and a recipient, privacy is a broader, more subtle, and sometimes less technical issue.

Daniel Augot with other groups from Inria (Comete, SMIS) started discussions with lawyers and economists, fostered by IDEX Paris-Saclay's *Institut de la société du numérique*, to understand the privacy concerns of ordinary citizens. On a more technical side, privacy can be protected with cryptographic protocols other than encryption. In this direction, Grace is engaged since April 2013 in a collaboration with Alcatel–Lucent on private data storage and retrieval in the cloud.

LFANT Project-Team

4. Application Domains

4.1. Number theory

Being able to compute quickly and reliably algebraic invariants is an invaluable aid to mathematicians: It fosters new conjectures, and often shoots down the too optimistic ones. Moreover, a large body of theoretical results in algebraic number theory has an asymptotic nature and only applies for large enough inputs; mechanised computations (preferably producing independently verifiable certificates) are often necessary to finish proofs.

For instance, many Diophantine problems reduce to a set of Thue equations of the form $P(x, y) = a$ for an irreducible, homogeneous $P \in \mathbb{Z}[x, y]$, $a \in \mathbb{Z}$, in unknown integers x, y. In principle, there is an algorithm to solve the latter, provided the class group and units of a rupture field of P are known. Since there is no other way to prove that the full set of solutions is obtained, these algebraic invariants must be computed and certified, preferably without using the GRH.

Deeper invariants such as the Euclidean spectrum are related to more theoretical concerns, e.g., determining new examples of principal, but not norm-Euclidean number fields, but could also yield practical new algorithms: Even if a number field has class number larger than 1 (in particular, it is not norm-Euclidean), knowing the upper part of the spectrum should give a *partial* gcd algorithm, succeeding for almost all pairs of elements of $\mathcal{O}_K$. As a matter of fact, every number field which is not a complex multiplication field and whose unit group has rank strictly greater than 1 is almost norm-Euclidean [34], [35].

Algorithms developed by the team are implemented in the free PARI/GP system for number theory maintained by K. Belabas, which is a reference and the tool of choice for the worldwide number theory community.

4.2. Cryptology

Public key cryptology has become a major application domain for algorithmic number theory. This is already true for the ubiquitous RSA system, but even more so for cryptosystems relying on the discrete logarithm problem in algebraic curves over finite fields [6]. For the same level of security, the latter require smaller key lengths than RSA, which results in a gain of bandwidth and (depending on the precise application) processing time. Especially in environments that are constrained with respect to space and computing power such as smart cards and embedded devices, algebraic curve cryptography has become the technology of choice. Most of the research topics of the LFANT team concern directly problems relevant for curve-based cryptology: The difficulty of the discrete logarithm problem in algebraic curves determines the security of the corresponding cryptosystems. Complex multiplication, point counting and isogenies provide, on one hand, the tools needed to create secure instances of curves. On the other hand, isogenies have been found to have direct cryptographic applications to hash functions [36] and encryption [43]. Pairings in algebraic curves have proved to be a rich source for novel cryptographic primitives. Class groups of number fields also enter the game as candidates for algebraic groups in which cryptosystems can be implemented. However, breaking these systems by computing discrete logarithms has proved to be easier than in algebraic curves; we intend to pursue this cryptanalytic strand of research.

Apart from solving specific problems related to cryptology, number theoretic expertise is vital to provide cryptologic advice to industrial partners in joint projects. It is to be expected that continuing pervasiveness and ubiquity of very low power computing devices will render the need for algebraic curve cryptography more pressing in coming years.

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.

We also investigate some cross-disciplinary domains, which require a scientific competence coming from other areas, mainly social aspects of cryptology, cryptology for large databases.

Specfun 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.

3.3. Fast prototyping

We work in collaboration with CIRTES on rapid prototyping. CIRTES, a company based in Saint-Dié-des-Vosges, has designed a technique called Stratoconception® where a prototype of a 3D computer model is constructed by first decomposing the model into layers and then manufacturing separately each layer, typically out of wood of standard thickness (e.g. 1 cm), with a three-axis CNC (Computer Numerical Controls) milling machine. The layers are then assembled together to form the object. The Stratoconception® technique is cheap and allows fast prototyping of large models.

When the model is complex, for example an art sculpture, some parts of the models may be inaccessible to the milling machine. These inaccessible regions are sanded out by hand in a post-processing phase. This phase is very consuming in time and resources. We work on minimizing the amount of work to be done in this last phase by improving the algorithmic techniques for decomposing the model into layers, that is, finding a direction of slicing and a position of the first layer.

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 apply technology advances. While wireless transmissions allow a car to communicate with another 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 systems and processors, 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.

CONTRAINTES Project-Team

4. Application Domains

4.1. Combinatorial optimization

The number and economic impact of combinatorial optimization problems found in the industrial world are constantly increasing. They cover:

- resource allocation;
- placement, bin packing;
- scheduling;
- planning;
- transport;
- etc.

The last fifty years have brought many improvements in Operations Research resolution techniques. In this context, Constraint Programming can be seen as providing, on the one hand, constraint propagation algorithms that can be applied to various numerical or symbolic constraints, and on the other hand, declarative languages to model real-life problems and express complex resolution strategies. The latter point is crucial for designing new algorithms that cannot be defined without a sufficiently high-level language to express them. It allowed for better results than traditional methods, for instance in scheduling, and is promised to an even better future when thinking about the cooperation of global resolution, local consistency techniques and search methods.

The European FP6 Strep project Net-WMS that we have coordinated, has shown the benefit of combining discrete geometry constraints with rules to express physical, common sense and packing business constraints to solve packing problems in the context of warehouse management systems for the automotive industry. In this context, we have developed a rule-based modeling language, called Rules2CP, to express requirements in a declarative and flexible manner, and compile them to efficient constraint programs using reified constraints and a global constraint dedicated to geometrical placement problems in high dimension.

4.2. Computational Systems Biology

In partnership with biologists, we develop and experiment our modeling methods in five main leading applications:

- **Cancer chronotherapy optimization.** This research initiated in 2004 in partnership with Jean Clairambault, EPI BANG, and Francis Lévi INSERM, Hopital Paul Brousse, Villejuif, aims at understanding fundamental mechanisms involved in cancer and chronotherapies through mathematical modeling. Following the EU STREP project TEMPO (2006-2009) on "temporal genomics for patient tailored chronotherapeutics", coordinated by Francis Lévi, and in the framework of the Era-Net SysBio C5Sys project (2010-2013) coordinated by Francis Lévi and David Rand, University of Warwick, UK, we develop coupled models of the cell cycle, the circadian clock, the DNA repair system, irinotecan metabolism and drug injection optimization, focussing on the interactions between the cell cycle and the circadian clock in mammalian cells.
- **Mammalian cell cycle regulation.** This theme that is closely related to the previous one has lead to a formal collaboration in the framework of the ANR Syscomm project CALAMAR, started in 2009 on the "Compositional modeling and Analysis of LArge MoleculAr Regulatory networks". In partnership with Claudine Chaouiya, TAGC INSERM, Marseille, and Laurence Calzone, Institut Curie, Paris, this project aims at applying our computational techniques – both qualitative and quantitative – to the analysis of the large scale RB/E2F network, in order to elucidate various features of the human cell proliferation, especially in the case of healthy and bladder-tumor cells of different aggressiveness.

- **Real-time control of gene expression in yeast.** This research lead in the team by Grégory Batt investigates the possibilities to control gene expression in living cells. In collaboration with Pascal Hersen and Samuel Bottani, biophysicists at the Matière and Systèmes Complexes lab, CNRS/Paris Diderot University, we develop a microfluidic platform and control software for the real-time control of gene expression in yeast. In a larger initiative, we consider a similar problem but in mammalian cells, where the stochasticity of gene expression makes the control problem particularly challenging. The Iceberg Investissement d'Avenir project, coordinated by Grégory Batt, involves the MSC, BM2A, LIFL and PPS labs, and the Jacques Monod Institut. Similarly, the Contraintes research group is also involved in the Inria/INSERM large-scale initiative action COLAGE coordinated by Huges Berry, EPI COMBINING, with François Taddei, Ariel Lindner, INSERM Paris Necker, Hidde de Jong, Delphine Ropers, EPI IBIS, Jean-Luc Gouzé, and Madalena Chaves, EPI COMORE. In this project, we investigate the possibilities to control and reprogram growth and aging in bacteria *E. coli* using synthetic biology approaches.
- **Artificial tissue homeostasis in mammalian cells.** Artificial tissue design is a particularly challenging problem in synthetic biology since the system behavior results from the interplay between intra- and intercellular dynamics. In the framework of the Syne2arti ANR project, coordinated by Grégory Batt, and involving Dirk Draso, EPI BANG, Oded Maler, CNRS Verimag, and Ron Weiss, MIT, USA, we design and genetically-engineer mammalian cells to obtain a tissue having a desired cell density. The long-term correct functioning of the system relies several key aspects, including individual cell decisions, collective, spatial aspects, and cell-to-cell variability.
- **TGFβ signaling** In the framework of the BioTempo ANR project, we recently started to apply the different algorithms available in the BIOCHAM platform to the modeling of the TGFβ signaling network in collaboration with the SeRAIC lab (Rennes, France). The main challenge is to compare and understand crosstalks between the SMAD-dependent fast pathway and the MAPK-dependent slower pathway that is often related to cancer. Both the static network analyzers and the parameter learning methods of BIOCHAM are put to good use in this work.

DREAMPAL Team

4. Application Domains

4.1. Reflective Camera Networks

HiPEAC vision 2011-2012:

> *"reconfiguration, customization and runtime adaptation techniques will facilitate switching between tasks during the deployment of smart camera networks"*

A Smart Camera (SC) is a vision system which, in addition to image-capturing capabilities, is able to extract application-specific information from the captured images and to automatically make intelligent decisions based on them. Dynamicity is inherent in SCs: processing may change depending on the specific observations they make and on the context. For example, an SC may use a low-quality face recognition IP while observing an office during the day, but switch to a high-quality one if it detects an intrusion during the night. Moreover, image processing requires high-performance computing, which is achieved by using parallelism. Thus, the integration of dynamic reconfiguration and parallelism, which is addressed by our project, is naturally present in SCs. Previous work in the DaRT team has already explored efficient uses of FPGAs in an SC network deployed in a retail store. A new proposal concerns embedded reflective camera in the Smart Cities multidisciplinary project developed on the University Lille 1 campus.

4.2. Set-top Boxes

Television sets and set-top-boxes are forming a symbiotic connection, which relies on common standards and protocols such as DLNA, Web standards, Web 2.0, H264, HEVC... As a result, the hardware platform on which applications run is becoming less important: commonly used ISAs like x86 are not mandatory any more. Dedicated pieces of hardware could efficiently provide specific services according to user requests. End-users expect platforms supporting many services with maximum performance, but do not require all of them at the same time. Dynamic reconfiguration is here too a good compromise, and it is efficient enough to support high performance algorithms like H264 or HEVC. It could also provide a ground for supporting on-the-fly codec switching. This may occur because the broadcaster decides to change the encoding of its video signal for safety reasons. Nowadays this operation is performed by a software because changing a hard codec still means flashing the set-top boxes to update it. Dreampal has started a collaboration with Kalray (http://www.kalray.eu) to develop a massively parallel language (without dynamic reconfiguration facilities, for now) on their MPSoC. H264 will be tested on this chip and on special FPGA boards with dedicated extensions for multimedia applications like the Xilinx Zynq.

4.3. Safe and Intelligent Transportation

Safety issues are today a key differentiator in the transportation industrial sector. The supervision and the detection of dangerous situations is a key technological challenge for future transportation systems at the infrastructure and vehicle levels. As an example, various obstacles can be detected on the road or in a Level Crossing (LC) using embedded systems. The proposed system will be based on stereo-vision technology (high definition cameras) and embedded reconfigurable computing and can be integrated either in vehicles or in the rail network. Also, Dreampal has started a collaboration with INDUCT (http://induct-technology.com/) to develop reconfigurable parallel architecture for the detection and the identification of obstacles in the frame of the NAVIA (autonomous electrical vehicle) project. This application will be implemented on Xilinx Zynq-based boards equipped with video processing features.

INDES Project-Team

4. Application Domains

4.1. Web programming

Along with games, multimedia applications, electronic commerce, and email, the web has popularized computers in everybody's life. The revolution is engaged and we may be at the dawn of a new era of computing where the web is a central element. The web constitutes an infrastructure more versatile, polymorphic, and open, in other words, more powerful, than any dedicated network previously invented. For this very reason, it is likely that most of the computer programs we will write in the future, for professional purposes as well as for our own needs, will extensively rely on the web.

In addition to allowing reactive and graphically pleasing interfaces, web applications are de facto distributed. Implementing an application with a web interface makes it instantly open to the world and accessible from much more than one computer. The web also partially solves the problem of platform compatibility because it physically separates the rendering engine from the computation engine. Therefore, the client does not have to make assumptions on the server hardware configuration, and vice versa. Lastly, HTML is highly durable. While traditional graphical toolkits evolve continuously, making existing interfaces obsolete and breaking backward compatibility, modern web browsers that render on the edge web pages are still able to correctly display the web pages of the early 1990's.

For these reasons, the web is arguably ready to escape the beaten track of n-tier applications, CGI scripting and interaction based on HTML forms. However, we think that it still lacks programming abstractions that minimize the overwhelming amount of technologies that need to be mastered when web programming is involved. Our experience on reactive and functional programming is used for bridging this gap.

4.2. Multimedia

Electronic equipments are less and less expensive and more and more widely spread out. Nowadays, in industrial countries, computers are almost as popular as TV sets. Today, almost everybody owns a mobile phone. Many are equipped with a GPS or a PDA. Modem, routers, NASes and other network appliances are also commonly used, although they are sometimes sealed under proprietary packaging such as the Livebox or the Freebox. Most of us evolve in an electronic environment which is rich but which is also populated with mostly isolated devices.

The first multimedia applications on the web have appeared with the Web 2.0. The most famous ones are Flickr, YouTube, or Deezer. All these applications rely on the same principle: they allow roaming users to access the various multimedia resources available all over the Internet via their web browser. The convergence between our new electronic environment and the multimedia facilities offered by the web will allow engineers to create new applications. However, since these applications are complex to implement this will not happen until appropriate languages and tools are available. In the Indes team, we develop compilers, systems, and libraries that address this problem.

4.3. Home Automation

The web is the de facto standard of communication for heterogeneous devices. The number of devices able to access the web is permanently increasing. Nowadays, even our mobile phones can access the web. Tomorrow it could even be the turn of our wristwatches! The web hence constitutes a compelling architecture for developing applications relying on the "ambient" computing facilities. However, since current programming languages do not allow us to develop easily these applications, ambient computing is currently based on ad-hoc solutions. Programming ambient computing via the web is still to be explored. The tools developed in the Indes team allow us to build prototypes of a web-based home automation platform. For instance, we experiment with controlling heaters, air-conditioners, and electronic shutters with our mobile phones using web GUIs.

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* group 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.

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 nighties. 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 2012 the TASC team was involved in the following application domains:

- *Planning and replanning* in Data Centres SelfXL 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 over a period of several years.

ESPRESSO Project-Team

4. Application Domains

4.1. Embedded systems

The application domains covered by the Polychrony toolbox are engineering areas where a system design-flow requires high-level model transformations and verifications to be applied during the development-cycle. The project-team has focused on developing such integrated design methods in the context of avionics applications, through the European IST projects Sacres, Syrf, Safeair, Speeds, and through the national ANR projects Topcased, OpenEmbeDD, Spacify. In this context, Polychrony is seen as a platform on which the architecture of an embedded system can be specified from the earliest design stages until the late deployment stages through a number of formally verifiable design refinements.

Along the way, the project adopted the policy proposed with project Topcased and continued with OpenEmbeDD to make its developments available to a large community in open-source. The Polychrony environment is now integrated in the OPEES/Polarsys platform and distributed under EPL and GPL v2.0 license for the benefits of a growing community of users and contributors, among which the most active are Virginia Tech's Fermat laboratory and Inria's project-teams Aoste, Dart.

S4 Project-Team

4. Application Domains

4.1. Domain 1

TRIO Team

4. Application Domains

4.1. TRIO application domains

Three main application domains can be underlined.

- In-vehicle embedded systems. The work developed in TRIO is oriented towards transportation systems (cars, airplanes, trains etc.). They mainly cover two points. The first one is the specification of what must be modeled in such a system and how to reach a good accuracy of a model. The second point concerns the verification of dependability properties and temporal properties required by these applications.
- Compilation, memory management and low-power issues for real time embedded systems. It becomes mandatory to design embedded systems that respect performances and reliability constraints while minimizing the energy consumption. Hence, TRIO is involved, on the one hand, in the definition of ad-hoc memory management at compilation time and on the other hand, in joint study of memory management strategies and tasks scheduling for real time critical systems.
- Code analyses and software visualization for embedded systems. Despite important advances, it is still impossible to develop and optimize automatically all the programs with all their variety, especially when deployment constraints are considered. Software design and implementation thus remain highly ad-hoc, poorly automated activities, with a human being in the loop. TRIO is thus involved in the design of better tools for software engineering focusing on helping the human developer understand and develop the system, thanks to powerful automated program analyses and advanced visualizations techniques.

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 in usually tighly coupled with larger aspects of system design, where models from physics are being used already. The formalisms AADL (for avionics) and AutoSar [66] (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 (section vide)

MUTANT Project-Team

4. Application Domains

4.1. Authoring and Performing Interactive Music

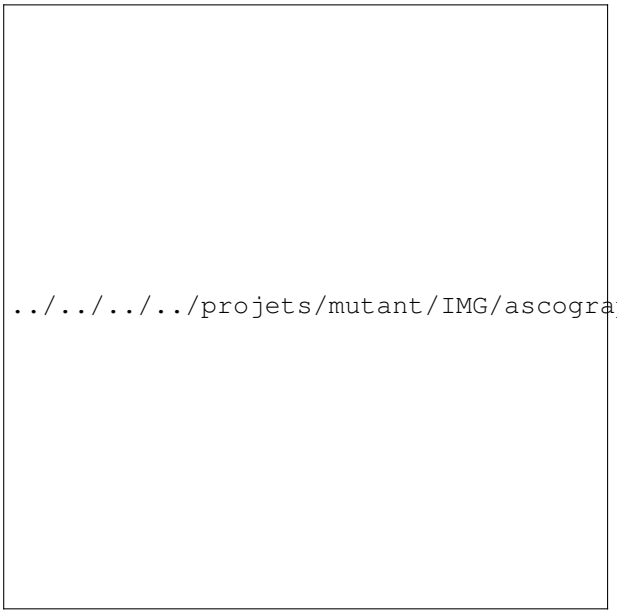

Figure 3. Screenshot of Ascograph, the Antescofo graphical score editor

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 developed in 2013 and shown in Figure 3 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. Music Post-Production.

Outcomes of our recognition and alignment paradigms can improve and ease existing workflows employed by audio engineers for mixing and editing using commercial Digital Audio Workstations (DAW) in post-production. We have initiated collaborations with audio engineers at Ircam and Paris Superior Music Conservatory (CNSMDP) to define the framework [9] and we will continue to develop and integrate our tools into their daily workflow.

4.3. Realtime Music Information Retrieval

We are considering to apply our information geometric approach to well-known and complex MIR problems. A glance of such problems is presented in [6]. Such applications can be used as front-end of many high-level MIR applications such as audio summarisation, audio finger printing, and automatic annotation tools. Besides such low-level enhancements, our information geometric approach can address the well-known (and still to be solved) problem of audio queries over a database.

4.4. Automatic Accompaniment/Creative Tools for Entertainment Industry

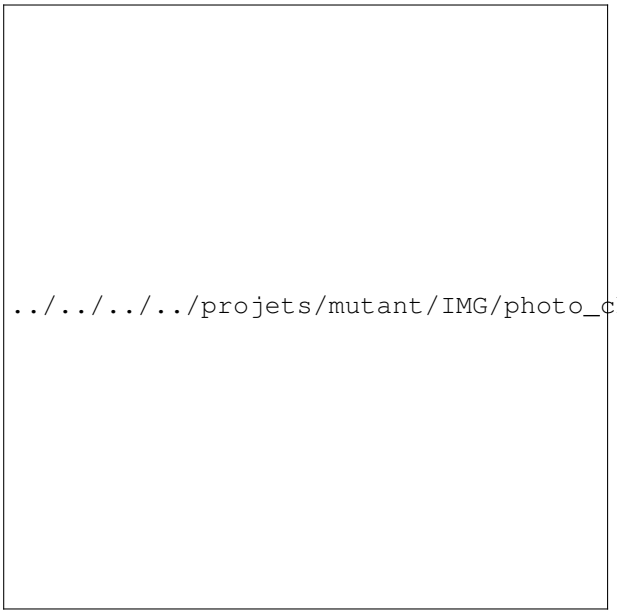

Figure 4. Automatic Accompaniment Session with Antescofo during ACM CHI 2013 Conference

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* and *Tonara*. Technologies developed around *Antescofo* can enhance interaction between user and the computer for such large public applications. We hope to pursue this by licensing our technologies to third-party companies.

PARKAS Project-Team

4. Application Domains

4.1. Domain

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[1]) 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[2] (based on LUSTRE), CONTROLBUILD and RT-BUILDER (based on SIGNAL) from GEENSYS[3] (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.

[1] http://www.dspaceinc.com
[2] http://www.esterel-technologies.com
[3] http://www.geensoft.com

FORMES Team

4. Application Domains

4.1. Proof of Programs

In many life critical application such as nuclear power or transportation, formal proofs of programs are required, and theorem provers provide an essential tool in that area.

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. Certified Compilation for Embedded systems

Many frameworks have been designed in order to make the design and the development of embedded systems more rigourous and secure on the basis of some formal model. All these frameworks implicitly assume the *reliability of the translation* to executable code, in order to guarantee the verified properties in the design level are preserved in the implementation. In other words, they rely on a claim saying that the compilers from high level model description to the implementation will not introduce undesired behaviors or errors in silence. The only safe way to satisfy such a claim is to certify correctness of the compilers, that is, to prove that the code they produce has exactly the semantics of the source code or model.

4.4. Distributed Systems

Many embedded systems run in a distributed environment. Distributed systems raise extremely challenging issues, both for the design and the implementation, because decisions can be made only from a local knowledge, which is imperfect due to communication time and unreliability of transmissions.

4.5. Security

The convergence between embedded technologies and the Internet offers many opportunities to malicious people for breaking the privacy of consumers or of organisations. Using cryptography is not enough for ensuring the protection of data, because of possible flaws in protocols and interfaces, providing opportunities for many well-known attacks. This area is therefore an important target of formal methods.

SECSI Project-Team

4. Application Domains

4.1. Application Domains

Here are a few examples of applications of research done in SECSI:

- Security of electronic voting schemes: the case of the Helios protocol, used in particular at University of Louvain-la-Neuve (2010) and at the International Association for Cryptographic Research (IACR).
- Security of the protocols involved in the TPM (Trusted Platform Module) chip, a chip present in most PC laptops today, and which is meant to act as a trusted base.
- Security of the European electronic passport—and the discovery of an attack on the French implementation of it.
- Intrusion detection with the Orchids tool: several interested partners, among which EADS Cassidian, Thales, Galois Inc. (USA), the French Direction Générale de l'Armement (DGA).

ABSTRACTION Project-Team

4. Application Domains

4.1. Certification of Safety Critical Software

Keywords: Absence of runtime error, Abstract interpretation, Certified compilation, Static analysis, Translation validation, Verifier.

Safety critical software may incur great damage in case of failure, such as human casualties or huge financial losses. These include many kinds of embedded software, such as fly-by-wire programs in aircrafts and other avionic applications, control systems for nuclear power plants, or navigation systems of satellite launchers. For instance, the failure of the first launch of Ariane 5 (flight Ariane 501) was due to overflows in arithmetic computations. This failure caused the loss of several satellites, worth up to $ 500 millions.

This development of safe and secure critical software requires formal methods so as to ensure that they do not go wrong, and will behave as specified. In particular, testing, bug finding methods, checking of models but not programs do not provide any guarantee that no failure will occur, even of a given type such as runtime errors; therefore, their scope is limited for certification purposes. For instance, testing can usually not be performed for *all* possible inputs due to feasibility and cost reasons, so that it does not prove anything about a large number of possible executions.

By contrast, program analysis methods such as abstract-interpretation-based static analysis are not subject to unsoundness, since they can *formally prove* the absence of bugs directly on the program, not on a model that might be erroneous. Yet, these techniques are generally incomplete since the absence of runtime errors is undecidable. Therefore, in practice, they are prone to false alarms (*i.e.*, they may fail to prove the absence of runtime errors for a program which is safe). The objective of certification is to ultimately eliminate all false alarms.

It should be noted that, due to the size of the critical codes (typically from 100 to 1000 kLOCs), only scalable methods can succeed (in particular, software model checking techniques are subject to state explosion issues). As a consequence, this domain requires efficient static analyses, where costly abstractions should be used only parsimoniously.

Furthermore, many families of critical software have similar features, such as the reliance on floating-point intensive computations for the implementation of control laws, including linear and non-linear control with feedback, interpolations, and other DSP algorithms. Since we stated that a proof of absence of runtime errors is required, very precise analyses are required, which should be able to yield no false alarm on wide families of critical applications. To achieve that goal, significant advantages can be found in the design of domain specific analyzers, such as ASTRÉE [30], [46], which has been initially designed specifically for synchronous embedded software.

Last, some specific critical software qualification procedures may require additional properties being proved. As an example, the DO-178 regulations (which apply to avionics software) require a tight, documented, and certified relation to be established between each development stage. In particular, compilation of high level programs into executable binaries should also be certified correct.

The ABSTRACTION project-team has been working on both proof of absence of runtime errors and certified compilation over the decade, using abstract interpretation techniques. Successful results have been achieved on industrial applications using the ASTRÉE analyzer. Following this success, ASTRÉE has been licensed to AbsInt Angewandte Informatik GmbH to be industrialized, and the ABSTRACTION project-team has strong plans to continue research on this topic.

4.2. Abstraction of Biological Cell Signaling Networks

Keywords: Biology, Health, Static analysis.

Protein-protein interactions consist in complexations and post translational modifications such as phosphorilation. These interactions enable biological organisms to receive, propagate, and integrate signals that are expressed as proteins concentrations in order to make decisions (on the choice between cell division and cell death for instance). Models of such interaction networks suffer from a combinatorial blow up in the number of species (number of non-isomorphic ways in which some proteins can be connected to each others). This large number of species makes the design and the analysis of these models a highly difficult task. Moreover the properties of interest are usually quantitative observations on stochastic or differential trajectories, which are difficult to compute or abstract.

Contextual graph-rewriting systems allow a concise description of these networks, which leads to a scalable method for modeling them. Then abstract interpretation allows the abstraction of these systems properties. First qualitative abstractions (such as over approximation of complexes that can be built) provide both debugging information in the design phases (of models) and static information that are necessary in order to make other computations (such as stochastic simulations) scale up. Then qualitative invariants also drive efficient quantitative abstractions (such as the reduction of ordinary differential semantics).

The work of the ABSTRACTION project-team on biological cell signaling networks ranges from qualitative abstractions to quantitative abstractions.

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. Tools for proofs in B

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* [10]. From this work presented in the Doctoral dissertation of Mélanie Jacquel, the *Super Zenon* tool [5] 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* [22], [23] (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.

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 [45] and enforcement of data confidentiality through type-based inference of information flows and noninterference properties [49].

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. Languages such as CDuce and OCamlDuce extend these benefits to the handling of semi-structured XML data [39]. 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 librairies 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, Serge Haddad, 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.

PARSIFAL Project-Team

4. Application Domains

4.1. Automated Reasoning

Automated reasoning has traditionally focused on classical first-order logic but it is increasingly important for automation to other logics. We are applying our research to the following extensions to this traditional focus.

- Non-classical logics are increasingly becoming important in the specification and analysis of software. Most type systems are based on (possibly second-order) propositional intuitionistic logic, for example, while resource-sensitive and concurrent systems are most naturally expressed in linear logic. The members of the Parsifal team have a strong expertise in the design and implementation of performant automated reasoning systems for such non-classical logics. In particular, the Linprover suite of provers [38] continue to be the fastest automated theorem provers for propositional and first-order linear logic.
- Automated reasoning uses a broad range of techniques whose soundness and completeness relate to the existence of proofs. The research programme of the ANR PSI project at Parsifal is to build a finer-grained connection by specifying automated reasoning techniques as the step-by-step construction of proofs, as we know it from proof theory and logic programming. The goal is to do this in a unifying framework, namely proof-search in a polarized and focused logic. One of the advantages of this approach is that it allows combining and extending such techniques. For example, the PSI project has applied this approach to proof to the problem of SAT-modulo-Theory. In that domain, logical reasoning is combined with domain-specific decision procedures. The PSI project has shown how to incorporate the call to decision procedures in the proof-theoretical framework of focused sequent calculi and the proof-search mechanisms that are related to it.

4.2. Mechanized Metatheory

There has been increasing interest in the use of formal methods to provide proofs of properties of programs and programming languages. Tony Hoare's Grand Challenge titled "Verified Software: Theories, Tools, Experiments" has as a goal the construction of "verifying compilers" for a world where programs would only be produced with machine-verified guarantees of adherence to specified behavior. There is also the POPLMark challenge [37] which envisions "a world in which mechanically verified software is commonplace: a world in which theorem proving technology is used routinely by both software developers and programming language researchers alike." The proposers of this challenge go on to say that a "crucial step towards achieving these goals is mechanized reasoning about language metatheory."

The Parsifal team has been applying their research results to design and building systems to directly aid in both of these challenges. One important requirements for reasoning about programming languages is the ability to reason about data structures with binding constructs up to α-equivalence. The use of higher-order syntax and nominal techniques for such data structures was pioneered by Miller, Nadathur and Tiu. The Abella system (see Section 3.2) implements a refinement of a number of these ideas and has been used to give full solutions to sections of the POPLMark challenge in addition to fully formal proofs of a number of other theorems in the meta-theory of the λ-calculus. Also, our colleague Alwen Tiu from the Australian National University has also been building on our Bedwyr model checking tool so that we can build on top of it his SPEC system for doing model checking of spi-calculus expressions. We have adopted his enhancements to Bedwyr and are developing further improvements within the context of the BATT project (see Section 5.2).

4.3. Proof Certificates

Within the context of the ProofCert project, various members of the team have been building a flexible framework for the definition of the semantics of proof evidence. The emphasis is to attempt to capture as many forms of proof evidence as is possible. Using this framework, we have defined the semantics of all the following forms of proof evidence: natural deduction, expansion trees, matings, proof nets, resolution refutations, and Frege proofs. Given our framework, there is one kernel that can check all of these different forms of proof. Thus, one only needs to trust this one kernel in order to trust the output of a very wide range of theorem provers working in either intuitionistic or classical logics (see [20], [19], and [32].

PI.R2 Project-Team (section vide)

SUMO Team

4. Application Domains

4.1. Telecommunication network management

The domain of autonomic network management, under its new hype names, will remain an important playground for SUMO. It covers a wide variety of problems, ranging from distributed (optimal) control to distributed diagnosis, optimization, reconfiguration, 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 accross 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 coexists, 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 controler 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 (see above), 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 then 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, initiated in [21], 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).

TOCCATA 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

- Transportation: aeronautics, railroad, space flight, automotive
- Communications: mobile phones, smart phones, Web applications
- Financial applications, banking
- Medicine: diagnostic devices, computer-assisted surgery
- Databases with confidentiality requirements (e.g. health records, electronic voting)

Currently our industrial collaborations mainly belong the first of these domains: transportation. These include, in the context of the ANR U3CAT project (Airbus France, Toulouse; Dassault Aviation, Saint-Cloud; Sagem Défense et Sécurité):

- proof of C programs via *Frama-C*/*Jessie*/*Why*;
- proof of floating-point programs;
- use of the *Alt-Ergo* prover via CAVEAT tool (CEA) or *Frama-C*/WP.

In the context of the FUI project Hi-Lite, the Adacore (Paris) uses *Why3* and *Alt-Ergo* as back-end to GnatProve, an environment for verification of Ada programs. This is applied in the domain of aerospace (Thales, EADS Astrium).

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

Apart from the domain of transportation, the Cubicle model checker modulo theories based on the *Alt-Ergo* SMT prover (collaboration with Intel Strategic Cad Labs, Hillsboro, OR, USA) can be applied to verification of concurrent programs and protocols (http://cubicle.lri.fr/).

VERIDIS Project-Team

4. Application Domains

4.1. Application Domains

Our work focuses on the formal modeling and verification of distributed algorithms and protocols. These are or will be 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. Distributed protocols underly computing infrastructure that must be highly available and mostly invisible to the end user, therefore correctness is important. 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, they are becoming more and more common due to the cost associated with malfunctioning system components and software. 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 is 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 [50] in order to understand the limits of this method. The second one consists in analysing the behavior of a program by monitoring it. Following [52], 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 [70].

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. This project of "high security lab" in one of the main project of the CPER 2007-2013.

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. [35]), control theory (see e.g. [43]), neural networks (see e.g. [71]), and so on. We are interested in the formal decidability of properties of dynamical systems, such as reachability [62], the Skolem-Pisot problem [39], the computability of the ω-limit set [61]. 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 [72], on recursive analysis [77], on the algebraic approach [68] and on computability in a probabilistic context [64]. 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 [53], [54], [55], to weak termination [56], sufficient completeness [57] and probabilistic termination [59]. 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 [58], [60]. 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 [65], [66], [67]. 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 platform. As an extension of the AVISPA specification language, we are working on a new environment called *CASRUL* for handling more general protocols like e-business protocols for example.

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 [69] and Java Card Virtual Machine Transaction mechanism [71]), information system and for embedded software [80].

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. [76]): 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 extend the coverage of method for 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 adressed. 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 consitute 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 availablity, 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 quantitive 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 π-calculus, and the application of model-checking to compute the matrices associated to the corresponding channels.

DICE Team

4. Application Domains

4.1. Standard software stack for Intermediation

As linux has emerged as the reference stack for operating systems, LAMP (Linux, Apache, Mysql, Php) has emerged for the pre-Web 2.0 stack. With the emergence of intermediation system, such as those used by Facebook and other intermediation platforms, new reference stacks are used, with open architecture, which must enable the development of new intermediation businesses in a matter of days of development. Most of the tools have been developed. We are confident that the low-level toolbox is mostly designed and is based on JavaScript both at the client and the server parts, based on nosql databases such as redis or mongodb at the data layer, based on web development framework at the client and the server side, and finally on social network plugins at the intermediation layer. MEAN (MongoDB, Express, AngularJS, NodeJS) is a first proposal towards the kind of software stack we focus on, that is not exclusively devoted to intermediation purposes. We propose our own stack, based on these toolboxes to handle the future intermediation systems we envision.

4.2. Intermediation systems

Intermediation systems are going to govern most of our activities. Intermediation systems link all people and provide them with the best services, the most appropriate to them. They exist currently in the realm of the Web 2.0 systems such as search engine, social networks, blogging, etc. related to accessing knowledge and communicating or exchanging with people. In the near future they will make their ways in most of our systems, energy, transport, education, employment, etc. We believe that political systems will evolve as well, with a new interaction between governing bodies and citizens. The surveillance programs that are currently widely debated give an increased information on their citizens to government as well as to corporations. The trend towards open data will provide information to citizens on government actions, to an extend that we probably fail to understand today. Intermediation platforms will play a crucial role to carry on the right information or service to the right people. Our research is devoted to better understand these challenging evolutions, and propose solutions to specific aspects, in particular in the realm of elections. (cf http://www.inriality.fr/vie-citoyenne/open-data/geopolitique/va-t-delocaliser-aussi-nos/)

PRIVATICS Team

3. Application Domains

3.1. Domain 1: Privacy in smart environments.

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 n 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 implementations

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, 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-terms 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 authenticate authorize users using a single sign-on protocol such as OAuth, a cloud storage service may encrypt user files on the server-side using XML encryption, and a password manager may encrypt passwords in the browser using a JavaScript cryptographic library. We build verification tools that can analyze such usages in commercial web applicaitons and evaluate their security against sophisticated web-based attacks.