



RESEARCH CENTER

FIELD

Algorithmics, Programming, Software and Architecture

Activity Report 2017

Section Application Domains

Edition: 2018-02-19

ALGORITHMICS, COMPUTER ALGEBRA AND CRYPTOLOGY

1. ARIC Project-Team	5
2. AROMATH Project-Team	6
3. CARAMBA Project-Team	8
4. CASCADE Project-Team	10
5. DATASHAPE Project-Team	12
6. GAMBLE Project-Team	13
7. GRACE Project-Team (section vide)	14
8. LFANT Project-Team (section vide)	15
9. POLSYS Project-Team (section vide)	16
10. SECRET Project-Team	17
11. SPECFUN Project-Team (section vide)	18

ARCHITECTURE, LANGUAGES AND COMPILATION

12. CAIRN Project-Team	19
13. CAMUS Team	20
14. CORSE Project-Team	21
15. PACAP Project-Team	22

EMBEDDED AND REAL-TIME SYSTEMS

16. AOSTE2 Team	23
17. HYCOMES Project-Team (section vide)	24
18. KAIROS Team	25
19. PARKAS Project-Team (section vide)	26
20. SPADES Project-Team	27
21. TEA Project-Team	28

PROOFS AND VERIFICATION

22. ANTIQUE Project-Team	30
23. CELTIQUE Project-Team (section vide)	32
24. CONVECS Project-Team	33
25. DEDUCTEAM Project-Team	34
26. GALLIUM Project-Team	35
27. MARELLE Project-Team (section vide)	37
28. MEXICO Project-Team	38
29. PARSIFAL Project-Team	40
30. PIR2 Project-Team (section vide)	42
31. SUMO Project-Team	43
32. TOCCATA Project-Team	45
33. VERIDIS Project-Team	46

SECURITY AND CONFIDENTIALITY

34. CARTE Team	47
35. CIDRE Project-Team	49
36. COMETE Project-Team	50

37. DATASPHERE Team	51
38. PESTO Project-Team	52
39. PRIVATICS Project-Team	53
40. PROSECCO Project-Team	55
41. TAMIS Team	56

ARIC Project-Team

4. Application Domains

4.1. 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.

4.2. Cryptography, Cryptology, Communication Theory

Lattice reduction algorithms have direct applications in

- public-key cryptography;
- diophantine equations;
- communications theory.

AROMATH Project-Team

4. Application Domains

4.1. Geometric modeling for Design and Manufacturing.

The main domain of applications that we consider for the methods we develop is Computer Aided Design and Manufacturing.

Computer-Aided Design (CAD) involves creating digital models defined by mathematical constructions, from geometric, functional or aesthetic considerations. Computer-aided manufacturing (CAM) uses the geometrical design data to control the tools and processes, which lead to the production of real objects from their numerical descriptions.

CAD-CAM systems provide tools for visualizing, understanding, manipulating, and editing virtual shapes. They are extensively used in many applications, including automotive, shipbuilding, aerospace industries, industrial and architectural design, prosthetics, and many more. They are also widely used to produce computer animation for special effects in movies, advertising and technical manuals, or for digital content creation. Their economic importance is enormous. Their importance in education is also growing, as they are more and more used in schools and educational purposes.

CAD-CAM has been a major driving force for research developments in geometric modeling, which leads to very large software, produced and sold by big companies, capable of assisting engineers in all the steps from design to manufacturing.

Nevertheless, many challenges still need to be addressed. Many problems remain open, related to the use of efficient shape representations, of geometric models specific to some application domains, such as in architecture, naval engineering, mechanical constructions, manufacturing Important questions on the robustness and the certification of geometric computation are not yet answered. The complexity of the models which are used nowadays also appeals for the development of new approaches. The manufacturing environment is also increasingly complex, with new type of machine tools including: turning, 5-axis machining and wire EDM (Electrical Discharge Machining), 3D printer. It cannot be properly used without computer assistance, which raises methodological and algorithmic questions. There is an increasing need to combine design and simulation, for analyzing the physical behavior of a model and for optimal design.

The field has deeply changed over the last decades, with the emergence of new geometric modeling tools built on dedicated packages, which are mixing different scientific areas to address specific applications. It is providing new opportunities to apply new geometric modeling methods, output from research activities.

4.2. Geometric modeling for Numerical Simulation and Optimization

A major bottleneck in the CAD-CAM developments is the lack of interoperability of modeling systems and simulation systems. This is strongly influenced by their development history, as they have been following different paths.

The geometric tools have evolved from supporting a limited number of tasks at separate stages in product development and manufacturing, to being essential in all phases from initial design through manufacturing.

Current Finite Element Analysis (FEA) technology was already well established 40 years ago, when CAD-systems just started to appear, and its success stems from using approximations of both the geometry and the analysis model with low order finite elements (most often of degree ≤ 2).

There has been no requirement between CAD and numerical simulation, based on Finite Element Analysis, leading to incompatible mathematical representations in CAD and FEA. This incompatibility makes interoperability of CAD/CAM and FEA very challenging. In the general case today this challenge is addressed by expensive and time-consuming human intervention and software developments.

Improving this interaction by using adequate geometric and functional descriptions should boost the interaction between numerical analysis and geometric modeling, with important implications in shape optimization. In particular, it could provide a better feedback of numerical simulations on the geometric model in a design optimization loop, which incorporates iterative analysis steps.

The situation is evolving. In the past decade, a new paradigm has emerged to replace the traditional Finite Elements by B-Spline basis element of any polynomial degree, thus in principle enabling exact representation of all shapes that can be modeled in CAD. It has been demonstrated that the so-called isogeometric analysis approach can be far more accurate than traditional FEA.

It opens new perspectives for the interoperability between geometric modeling and numerical simulation. The development of numerical methods of high order using a precise description of the shapes raises questions on piecewise polynomial elements, on the description of computational domains and of their interfaces, on the construction of good function spaces to approximate physical solutions. All these problems involve geometric considerations and are closely related to the theory of splines and to the geometric methods we are investigating. We plan to apply our work to the development of new interactions between geometric modeling and numerical solvers.

CARAMBA Project-Team

4. Application Domains

4.1. Better Awareness and Avoidance of Cryptanalytic Threats

Our study of the Number Field Sieve family of algorithms aims at showing how the threats underlying various supposedly hard problems are real. Our record computations, as well as new algorithms, contribute to having a scientifically accurate assessment of the feasibility limit for these problems, given academic computing resources. The data we provide in this way is a primary ingredient for government agencies whose purpose includes guidance for the choice of appropriate cryptographic primitives. For example the French ANSSI⁰, German BSI, or the NIST⁰ in the United States base their recommendations on such computational achievements.

The software we make available to achieve these cryptanalytic computations also allows us to give cost estimates for potential attacks to cryptographic systems that are taking the security/efficiency/legacy compatibility trade-offs too lightly. Attacks such as LogJam [24] are understood as being serious concerns thanks to our convincing proof-of-concepts. In the LogJam context, this impact has led to rapid worldwide security advisories and software updates that eventually defeat some potential intelligence threats and improve confidentiality of communications.

4.2. Promotion of Better Cryptography

We also promote the switch to algebraic curves as cryptographic primitives. Those offer nice speed and excellent security, while primitives based on elementary number theory (integer factorization, discrete logarithm in finite fields), which underpin e.g., RSA, are gradually forced to adopt unwieldy key sizes so as to comply with the desired security guarantees of modern cryptography. Our contributions to the ultimate goal of having algebraic curves eventually take over the cryptographic landscape lie in our fast arithmetic contributions, our contributions to the point counting problem, and more generally our expertise on the diverse surrounding mathematical objects, or on the special cases where the discrete logarithm problem is not hard enough and should be avoided.

We also promote cryptographically sound electronic voting, for which we develop the Belenios prototype software, (licensed under the AGPL). It depends on research made in collaboration with the PESTO team, and provides stronger guarantees than current state of the art.

4.3. Key Software Tools

The vast majority of our work is eventually realized as software. We can roughly categorize it in two groups. Some of our software covers truly fundamental objects, such as the GNU MPFR, GNU MPC, GF2X, or MPFQ packages. To their respective extent, these software packages are meant to be included or used in broader projects. For this reason, it is important that the license chosen for this software allows proper reuse, and we favor licenses such as the LGPL, which is not restrictive. We can measure the impact of this software by the way it is used in e.g., the GNU Compiler Collection (GCC), in Victor Shoup's Number Theory Library (NTL), or in the Sage computer algebra system. The availability of these software packages in most Linux distributions is also a good measure for the impact of our work.

⁰In [25], the minimal recommended RSA key size is 2048 bits for usage up to 2030. See also Annex B, in particular Section B.1 "Records de calculs cryptographiques".

⁰The work [36] is one of only two academic works cited by NIST in the initial version (2011) of the report [40].

We also develop more specialized software. Our flagship software package is Cado-NFS, and we also develop some others with various levels of maturity, such as GMP-ECM, CMH, or Belenios, aiming at quite diverse targets. Within the lifespan of the CARAMBA project, we expect more software packages of this kind to be developed, specialized towards tasks relevant to our research targets: important mathematical structures attached to genus 2 curves, generation of cryptographically secure curves, or tools for attacking cryptographically hard problems. Such software both illustrates our algorithms, and provides a base on which further research work can be established. Because of the very nature of these specialized software packages as research topics in their own right, needing both to borrow material from other projects, and being possible source of inspiring material for others, it is again important that these be developed in a free and open-source development model.

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.

DATASHAPE Project-Team

4. Application Domains

4.1. Main application domains

Our work is mostly of a fundamental mathematical and algorithmic nature but finds applications in a variety of application in data analysis, more precisely in Topological Data Analysis (TDA). Although TDA is a quite recent field, it already finds applications in material science, biology, sensor networks, 3D shapes analysis and processing, to name a few.

More specifically, DATASHAPE has recently started to work on the analysis of trajectories obtained from inertial sensors (starting PhD thesis of Bertrand Beaufils) and is exploring some possible new applications in material science.

GAMBLE Project-Team

4. Application Domains

4.1. Applications of computational geometry

Many domains of science can benefit from the results developed by GAMBLE. Curves and surfaces are ubiquitous in all sciences to understand and interpret raw data as well as experimental results. Still, the non-linear problems we address are rather basic and fundamental, and it is often difficult to predict the impact of solutions in that area. The short-term industrial impact is likely to be small because, on basic problems, industries have used ad hoc solutions for decades and have thus got used to it. The example of our work on quadric intersection is typical: even though we were fully convinced that intersecting 3D quadrics is such an elementary/fundamental problem that it ought to be useful, we were the first to be astonished by the scope of the applications of our software ⁰ (which was the first and still is the only one –to our knowledge– to compute robustly and efficiently the intersection of 3D quadrics) which has been used by researchers in, for instance, photochemistry, computer vision, statistics, and mathematics. Our work on certified drawing of plane (algebraic) curves falls in the same category. It seems obvious that it is widely useful to be able to draw curves correctly (recall also that part of the problem is to determine where to look in the plane) but it is quite hard to come up with specific examples of fields where this is relevant. A contrario, we know that certified meshing is critical in mechanical-design applications in robotics, which is a non-obvious application field. There, the singularities of a manipulator often have degrees higher than 10 and meshing the singular locus in a certified way is currently out of reach. As a result, researchers in robotics can only build physical prototypes for validating, or not, the approximate solutions given by non-certified numerical algorithms.

The fact that several of our pieces of software for computing non-Euclidean triangulations have already been requested by users long before they become public is a good sign for their wide future impact once in CGAL. This will not come as a surprise, since most of the questions that we have been studying followed from discussions with researchers outside computer science and pure mathematics. Such researchers are either users of our algorithms and software, or we meet them in workshops. Let us only mention a few names here. We have already referred above to our collaboration with Rien van de Weijgaert [37], [52] (astrophysicist, Groningen, NL). Michael Schindler [48] (theoretical physicist, ENSPCI, CNRS, France) is using our prototype software for 3D periodic weighted triangulations. Stephen Hyde and Vanessa Robins (applied mathematics and physics at Australian National University) have recently signed a software license agreement with INRIA that allows their group to use our prototype for 3D periodic meshing. Olivier Faugeras (neuromathematics, Inria Sophia Antipolis) had come to us and mentioned his needs for good meshes of the Bolza surface [27] before we started to study them. Such contacts are very important both to get feedback about our research and to help us choose problems that are relevant for applications. These problems are at the same time challenging from the mathematical and algorithmic points of view. Note that our research and our software are generic, i.e., we are studying fundamental geometric questions, which do not depend on any specific application. This recipe has made the success of the CGAL library.

Probabilistic models for geometric data are widely used to model various situations ranging from cell phone distribution to quantum mechanics. The impact of our work on probabilistic distributions is twofold. On the one hand, our studies of properties of geometric objects built on such distributions will yield a better understanding of the above phenomena and has potential impact in many scientific domains. On the other hand, our work on simulations of probabilistic distributions will be used by other teams, more maths oriented, to study these distributions.

⁰QI: <http://vegas.loria.fr/qi/>.

GRACE Project-Team (section vide)

LFANT Project-Team (section vide)

POLSYS Project-Team (section vide)

SECRET Project-Team

4. Application Domains

4.1. Cryptographic primitives

Our major application domain is the design of cryptographic primitives, especially for platforms with restricting implementation requirements. For instance, we aim at recommending (or designing) low-cost (or extremely fast) encryption schemes, or primitives which remain secure against quantum computers.

4.2. Code Reconstruction

To evaluate the quality of a cryptographic algorithm, it is usually assumed that its specifications are public, as, in accordance with Kerckhoffs principle, it would be dangerous to rely, even partially, on the fact that the adversary does not know those specifications. However, this fundamental rule does not mean that the specifications are known to the attacker. In practice, before mounting a cryptanalysis, it is necessary to strip off the data. This reverse-engineering process is often subtle, even when the data formatting is not concealed on purpose. A typical case is interception: some raw data, not necessarily encrypted, is observed out of a noisy channel. To access the information, the whole communication system has first to be disassembled and every constituent reconstructed. A transmission system actually corresponds to a succession of elements (symbol mapping, scrambler, channel encoder, interleaver...), and there exist many possibilities for each of them. In addition to the “preliminary to cryptanalysis” aspect, there are other links between those problems and cryptology. They share some scientific tools (algorithmics, discrete mathematics, probability...), but beyond that, there are some very strong similarities in the techniques.

SPECFUN Project-Team (section vide)

CAIRN Project-Team

4. Application Domains

4.1. Panorama

keywords: Wireless (Body) Sensor Networks, High-Rate Optical Communications, Wireless Communications, Applied Cryptography.

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.

Wireless Communication is our privileged application domain. Our research includes the prototyping of (subsets of) such applications on reconfigurable and programmable platforms. For this application domain, the high computational complexity of the 5G Wireless Communication Systems calls for the design of high-performance and energy-efficient architectures. In **Wireless Sensor Networks** (WSN), where each wireless node is expected to operate without battery replacement for significant periods of time, energy consumption is the most important constraint. 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.

Other important fields are also considered: hardware cryptographic and security modules, high-rate optical communications, machine learning, and multimedia processing.

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 primary 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.

CORSE Project-Team

4. Application Domains

4.1. Transfer

The main industrial sector related to the research activities of CORSE is the one of semi-conductor (programmable architectures spanning from embedded systems to servers). Obviously any computing application which has the objective of exploiting as much as possible the resources (in terms of high-performance but also low energy consumption) of the host architecture is intended to take advantage of advances in compiler and run-time technology. These applications are based over numerical kernels (linear algebra, FFT, convolution...) that can be adapted on a large spectrum of architectures. Members of CORSE already maintain fruitful and strong collaborations with several companies such as STMicroelectronics, Bull, UpMem, Kalray, or Aselta.

PACAP Project-Team

4. Application Domains

4.1. Any computer usage

The PACAP team is working on the fundamental technologies for computer science: processor architecture, performance-oriented compilation and guaranteed response time for real-time. The research results may have impacts on any application domain that requires high performance execution (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.

AOSTE2 Team

4. Application Domains

4.1. Avionics

Participants: Liliana Cucu, Keryan Didier, Adriana Gogonel, Cristian Maxim, Dumitru Potop-Butucaru, Yves Sorel.

A large number of our activities, in analysis, modelling, design and implementation of real-time embedded systems addresses specific applications mainly in the avionics field (with partners such as Airbus, Thales, Safran, etc.) (in the CAPACITES and ASSUME projects [9.1.2.1](#) , [9.2.1.1](#)).

4.2. Many-Core Embedded Architectures

Participants: Liliana Cucu, Keryan Didier, Dumitru Potop-Butucaru, Anselme Revuz, Yves Sorel.

The AAA approach (fitting embedded applications onto embedded architectures) requires a sufficiently precise description of (a model of) the architecture (description platform). Such platforms become increasingly heterogeneous, and we had to consider a number of emerging ones with that goal in mind, such as Kalray MPPA (in the CAPACITES and ASSUME projects [9.1.2.1](#) , [9.2.1.1](#)).

4.3. Railways

Participants: Liliana Cucu, Adriana Gogonel, Walid Talaboulma.

The statistical estimation of bounds on the execution time of a program on a processor is applied in the context of railroad crossing in the context of the collaborative project DEPARTS [9.1.2.2](#) .

HYCOMES Project-Team (section vide)

KAIROS Team

4. Application Domains

4.1. Model-Based Embedded System Design

Kairos general area of activity is embedded systems, as found in transportation (automotive, avionics), connected objects (smartphones,...) satellites, smart AE... More specifically we are focused on the design methods and tools to model and deploy such systems, where software is tightly coupled to its physical environment and its heterogenous running platform. Inversely we are not focused in designing new nature of cyber-physical models, but rather to study tools that help collaborative design between engineers of these distinct fields.

PARKAS Project-Team (section vide)

SPADES Project-Team

4. Application Domains

4.1. Industrial Applications

Our applications are in the embedded system area, typically: transportation, energy production, robotics, telecommunications, the Internet of things (IoT), 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/ dSPACE⁰) 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: SCADE⁰ (based on LUSTRE), CONTROLBUILD and RT-BUILDER (based on SIGNAL) from GEENSYs⁰ (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 Thales on schedulability analysis for evolving or underspecified real-time embedded systems, with Orange Labs on software architecture for cloud services and with Daimler on reduction of nondeterminism and analysis of deadline miss models for the design of automotive systems.

⁰<http://www.dspaceinc.com>

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

⁰<http://www.geensoft.com>

TEA Project-Team

4. Application Domains

4.1. Automotive and Avionics

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

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

Together with the importance of open-source software, one of these projects, the FUI Project P (section 8), demonstrated that a centralized model for system design could not just be a domain-specific programming language, such as discrete Simulink data-flows or a synchronous language. Synchronous languages implement a fixed model of time using logical clocks that are abstraction of time as sensed by software. They correspond to a fixed viewpoint in system design, and in a fixed hardware location in the system, which is not adequate to our purpose and must be extended.

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

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

4.2. Factory Automation

In collaboration with Mitsubishi R&D, we explore another application domain where time and domain heterogeneity are prime concerns: factory automation. In factory automation alone, a system is conventionally built from generic computing modules: PLCs (Programmable Logic Controllers), connected to the environment with actuators and detectors, and linked to a distributed network. Each individual, physically distributed, PLC module must be timely programmed to perform individually coherent actions and fulfill the global physical, chemical, safety, power efficiency, performance and latency requirements of the whole production chain. Factory chains are subject to global and heterogeneous (physical, electronic, functional) requirements whose enforcement must be orchestrated for all individual components.

Model-based analysis in factory automation emerges from different scientific domains and focus on different CPS abstractions that interact in subtle ways: logic of PLC programs, real-time electromechanical processing, physical and chemical environments. This yields domain communication problems that render individual domain analysis useless. For instance, if one domain analysis (e.g. software) modifies a system model in a way that violates assumptions made by another domain (e.g. chemistry) then the detection of its violation may well be impossible to explain to either of the software and chemistry experts. As a consequence, cross-domain analysis issues are discovered very late during system integration and lead to costly fixes. This is particularly prevalent in multi-tier industries, such as avionic, automotive, factories, where systems are prominently integrated from independently-developed parts.

ANTIQUÉ Project-Team

4. Application Domains

4.1. Verification of safety critical embedded software

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

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

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

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

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

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

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

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

4.2. Static analysis of software components and libraries

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

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

4.3. Biological systems

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

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

CELTIQUE Project-Team (section vide)

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.

DEDUCTEAM Project-Team

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. Termination certificate verification

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

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 OCaml 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 dereferences, 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 OCaml 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 [46] and enforcement of data confidentiality through type-based inference of information flow and noninterference properties [49].

4.3. Processing of complex structured data

Like most functional languages, OCaml is very well suited to expressing processing and transformations of complex, structured data. It provides concise, high-level declarations for data structures; a very expressive pattern-matching mechanism to destructure data; and compile-time exhaustiveness tests. Therefore, OCaml 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 OCaml 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 OCaml 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 family 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 (section vide)

MEXICO Project-Team

4. Application Domains

4.1. Telecommunications

Participants: Stefan Haar, Serge Haddad.

Telecommunications

MEXICO's research is motivated by problems of system management in several domains, such as:

- 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.

Currently, we have no active cooperation on these subjects.

4.2. Biological Systems

Participants: Thomas Chatain, Stefan Haar, Serge Haddad, Stefan Schwoon.

We have begun in 2014 to examine concurrency issues in systems biology, and are currently enlarging the scope of our research's applications in this direction. To see the context, note that in recent years, a considerable shift of biologists' interest can be observed, from the mapping of static genotypes to gene expression, i.e. the processes in which genetic information is used in producing functional products. These processes are far from being uniquely determined by the gene itself, or even jointly with static properties of the environment; rather, regulation occurs throughout the expression processes, with specific mechanisms increasing or decreasing the production of various products, and thus modulating the outcome. These regulations are central in understanding cell fate (how does the cell differentiate ? Do mutations occur ? etc), and progress there hinges on our capacity to analyse, predict, monitor and control complex and variegated processes. We have applied Petri net unfolding techniques for the efficient computation of attractors in a regulatory network; that is, to identify strongly connected reachability components that correspond to stable evolutions, e.g. of a cell that differentiates into a specific functionality (or mutation). This constitutes the starting point of a broader research with Petri net unfolding techniques in regulation. In fact, the use of ordinary Petri nets for capturing regulatory network (RN) dynamics overcomes the limitations of traditional RN models : those impose e.g. Monotonicity properties in the influence that one factor had upon another, i.e. always increasing or always decreasing, and were thus unable to cover all actual behaviours (see [75]). Rather, we follow the more refined model of boolean networks of automata, where the local states of the different factors jointly determine which state transitions are possible. For these connectors, ordinary PNs constitute a first approximation, improving greatly over the literature but leaving room for improvement in terms of introducing more refined logical connectors. Future work thus involves transcending this class of PN models. Via unfoldings, one has access – provided efficient techniques are available – to all behaviours of the model, rather than over- or under-approximations as previously. This opens the way to efficiently searching in particular for determinants of the cell fate : which attractors are reachable from a given stage, and what are the factors that decide in favor of one or the other attractor, etc. Our current research focusses on *cellular reprogramming*.

4.3. Autonomous Vehicles

The validation of safety properties is a crucial concern for the design of computer guided systems, in particular for automated transport systems. Our approach consists in analyzing the interactions of a randomized environment (roads, cross-sections, etc.) with a vehicle controller. This requires to :

- define the relevant case studies;
- extend our tool COSMOS to handle general hybrid systems;
- conduct experimentations and analyze their results.

In [SIA2017], we have shown that this approach scales pretty well but with a controller written in C. The next step will be to combine Simulink models with Petri nets since Simulink is widely used for specifying hybrid systems in industry. In order to do so, we need to define an operational semantic for Simulink and to design an elegant way for specifying the interface between nets and Simulink models. Then we will implement the solution in Cosmos.

PARSIFAL Project-Team

4. Application Domains

4.1. Integrating a model checker and a theorem prover

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

Our first step to providing an integration of model checking and (inductive) theorem proving was the development of a strong logic, that we call \mathcal{G} , which extends intuitionistic logic with notions of least and greatest fixed points. We had developed the proof theory of this logic in earlier papers [4] [57]. We have now recently converted the Bedwyr system so that it formally accepts almost all definitions and theorem statements that are accepted by the inductive theorem prover Abella. Thus, these two systems are proving theorems in the same logic and their results can now be shared.

Bedwyr's tabling mechanism has been extended so that it can make use of previously proved lemmas. For instance, when trying to prove that some board position has a winning strategy, an available stored lemma can now be used to obtain the result if some symmetric board position is already in the table.

Heath and Miller have shown how model checking can be seen as constructing proof in (linear) logic [64]. For more about recent progress on providing checkable proof certificates for model checking, see the web site for Bedwyr <http://slimmer.gforge.inria.fr/bedwyr/>.

4.2. Implementing trusted proof checkers

Traditionally, theorem provers—whether interactive or automatic—are usually monolithic: if any part of a formal development was to be done in a particular theorem prover, then the whole of it would need to be done in that prover. Increasingly, however, formal systems are being developed to integrate the results returned from several, independent and high-performance, specialized provers: see, for example, the integration of Isabelle with an SMT solver [56] as well as the Why3 and ESC/Java systems.

Within the Parsifal team, we have been working on foundational aspects of this multi-prover integration problem. As we have described above, we have been developing a formal framework for defining the semantics of proof evidence. We have also been working on prototype checkers of proof evidence which are capable of executing such formal definitions. The proof definition language described in the papers [54], [53] is currently given an implementation in the λ Prolog programming language [74]. This initial implementation will be able to serve as a “reference” proof checker: others who are developing proof evidence definitions will be able to use this reference checker to make sure that they are getting their definitions to do what they expect.

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

4.3. Trustworthy implementations of theorem proving techniques

Instead of integrating different provers by exchanging proof evidence and relying on a backend proof-checker, another approach to integration consists in re-implementing the theorem proving techniques as proof-search strategies, on an architecture that guarantees correctness.

Inference systems in general, and focused sequent calculi in particular, can serve as the basis of such an architecture, providing primitives for the exploration of the search space. These form a trusted *Application Programming Interface* that can be used to program and experiment various proof-search heuristics without worrying about correctness. No proof-checking is needed if one trusts the implementation of the API.

This approach has led to the development of the Psyche engine, and to its latest branch CDSAT.

Three major research directions are currently being explored, based on the above:

- The first one is about formulating automated reasoning techniques in terms of inference systems, so that they fit the approach described above. While this is rather standard for technique used in first-order Automated Theorem Provers (ATP), such as resolution, superposition, etc, this is much less standard in SMT-solving, the branch of automated reasoning that can natively handle reasoning in a combination of mathematical theories: the traditional techniques developed there usually organise the collaborations between different reasoning black boxes, whose opaque mechanisms less clearly connect to proof-theoretical inference systems. We are therefore investigating new foundations for reasoning in combinations of theories, expressed as fine-grained inference systems, and developed the *Conflict-Driven Satisfiability framework* for these foundations [19].
- The second one is about understanding how to deal with quantifiers in presence of one or more theories: On the one hand, traditional techniques for quantified problems, such as *unification* [41] or *quantifier elimination* are usually designed for either the empty theory or very specific theories. On the other hand, the industrial techniques for combining theories (Nelson-Oppen, Shostak, MCSAT [78], [82], [86], [66]) are designed for quantifier-free problems, and quantifiers there are dealt with incomplete *clause instantiation* methods or *trigger*-based techniques [55]. We are working on making the two approaches compatible.
- The above architecture’s modular approach raises the question of how its different modules can safely cooperate (in terms of guaranteed correctness), while some of them are trusted and others are not. The issue is particularly acute if some of the techniques are run concurrently and exchange data at unpredictable times. For this we explore new solutions based on Milner’s *LCF* [77]. In [60], we argued that our solutions in particular provide a way to fulfil the “Strategy Challenge for SMT-solving” set by De Moura and Passmore [87].

PI.R2 Project-Team (section vide)

SUMO Project-Team

4. Application Domains

4.1. Smart transportation systems

The smart-city trend aims at optimizing all functions of future cities with the help of digital technologies. We focus on the segment of urban trains, which will evolve from static and scheduled offers to reactive and eventually on-demand transportation offers. We address two challenges in this field. The first one concerns the optimal design of robust subway lines. The idea is to be able to evaluate, at design time, the performance of time tables and of different regulations policies. In particular, we focus on robustness issues: how can small perturbations and incidents be accommodated by the system, how fast will return to normality occur, when does the system become unstable? The second challenge concerns the design of new robust regulation strategies to optimize delays, recovery times, and energy consumption at the scale of a full subway line. These problems involve large-scale discrete-event systems, with temporal and stochastic features, and translate into robustness assessment, stability analysis and joint numerical/combinatorial optimization problems on the trajectories of these systems.

4.2. Management of telecommunication networks and of data centers

Telecommunication-network management is a rich provider of research topics for the team, and some members of SUMO have a long background of contacts and transfer with industry in this domain. Networks are typical examples of large distributed dynamic systems, and their management raises numerous problems ranging from diagnosis (or root-cause analysis), to optimization, reconfiguration, provisioning, planning, verification, etc. They also bring new challenges to the community, for example on the modeling side: building or learning a network model is a complex task, specifically because these models should reflect features like the layering, the multi-resolution view of components, the description of both functions, protocols and configuration, and they should also reflect dynamically-changing architectures. Besides modeling, management algorithms are also challenged by features like the size of systems, the need to work on abstractions, on partial models, on open systems, etc. The networking technology is now evolving toward software-defined networks, virtualized-network functions, multi-tenant systems, etc., which reinforces the need for more automation in the management of such systems.

Data centers are another example of large-scale modular dynamic and reconfigurable systems: they are composed of thousands of servers, on which virtual machines are activated, migrated, resized, etc. Their management covers issues like troubleshooting, reconfiguration, optimal control, in a setting where failures are frequent and mitigated by the performance of the management plane. We have a solid background in the coordination of the various autonomic managers that supervise the different functions/layers of such systems (hardware, middleware, web services, ...) Virtualization technologies now reach the domain of networking, and telecommunication operators/vendors evolve towards providers of distributed open clouds. This convergence of IT and networking strongly calls for new management paradigms, which is an opportunity for the team.

This application domain will be revived in the team by a collaboration with Orange Labs (1 Cifre PhD in the common lab Orange/Inria) and a collaboration with Nokia Bell Labs (1 Cifre PhD, and participation to the joint research team “Softwarization of Everything” of the common lab Nokia Bell Labs/Inria).

4.3. Collaborative workflows

A current trend is to involve end-users in collection and analysis of data. Examples of this trend are contributive science, crisis-management systems, and crowds. All these applications are data-centric and user-driven. They often are distributed and involve complex, and sometimes dynamic workflows. In many cases, there are

strong interactions between data and control flows: indeed, decisions taken regarding the next tasks to be launched highly depend on collected data. For instance, in an epidemic-surveillance system, the aggregation of various reported disease cases may trigger alerts. Another example is crowds where user skills are used to complete tasks that are better performed by humans than computers. In return, this requires addressing imprecise and sometimes unreliable answers. We address several issues related to complex workflows and data. We study declarative and dynamic models that can handle workflows, data, uncertainty, and competence management.

Once these models are mature enough, we plan to experiment them on real use cases from contributive science, health-management systems, and crowd platforms using prototypes. We also plan to define abstraction schemes allowing formal reasoning on these systems.

4.4. Systems Biology

Systems Biology is a recent topic in SUMO. In systems biology, many continuous variables interact together. Biological systems are thus good representatives for large complex quantitative systems, for which we are developing analysis and management methods. For instance, the biological pathway of apoptosis explains how many molecules interact inside a cell, triggered by some outside signal (drug, etc.), eventually leading to the death of the cell by apoptosis. While intrinsically quantitative in nature and in problems, data are usually noisy and problems need not be answered with ultimate precision. It thus seems reasonable to resort to approximations in order to handle the state-space explosion resulting from the high dimensionality of biological systems.

We are developing models and abstraction tools for systems biology. Studying these models suggests new reduction methods, such as considering populations instead of explicitly representing every single element into play (be it cells, molecules, etc): we thus develop algorithm handling population symbolically, either in a continuous (distributions) or a discrete (parametric) way. An intermediate goal is to speed-up analysis of such systems using abstractions, and a long term goal is to develop top-down model-checking methods that can be run on these abstractions.

TOCCATA Project-Team

4. Application Domains

4.1. Domain

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. Currently our industrial collaborations mainly belong to the domain of transportation, including aeronautics, railroad, space flight, automotive.

Verification of C programs, Alt-Ergo at Airbus Transportation is the domain considered in the context of the ANR U3CAT project, led by CEA, in partnership with Airbus France, Dassault Aviation, Sagem Défense et Sécurité. It included proof of C programs via Frama-C/Jessie/Why, proof of floating-point programs [114], the use of the Alt-Ergo prover via CAVEAT tool (CEA) or Frama-C/WP. Within this context, we contributed to a qualification process of Alt-Ergo with Airbus industry: the technical documents (functional specifications and benchmark suite) have been accepted by Airbus, and these documents were submitted by Airbus to the certification authorities (DO-178B standard) in 2012. This action is continued in the new project Soprano.

Certified compilation, certified static analyzers Aeronautics is the main target of the Verasco project, led by Verimag, on the development of certified static analyzers, in partnership with Airbus. This is a follow-up of the transfer of the CompCert certified compiler (Inria team Gallium) to which we contributed to the support of floating-point computations [64].

Transfer to the community of Ada development The former FUI project Hi-Lite, led by Adacore company, introduced the use of Why3 and Alt-Ergo as back-end to SPARK2014, an environment for verification of Ada programs. This is applied to the domain of aerospace (Thales, EADS Astrium). At the very beginning of that project, Alt-Ergo was added in the Spark Pro toolset (predecessor of SPARK2014), developed by Altran-Praxis: Alt-Ergo can be used by customers as an alternate prover for automatically proving verification conditions. Its usage is described in the new edition of the Spark book ⁰ (Chapter “Advanced proof tools”). This action is continued in the new joint laboratory ProofInUse. A recent paper [72] provides an extensive list of applications of SPARK, a major one being the British air control management *iFacts*.

Transfer to the community of Atelier B In the current ANR project BWare, we investigate the use of Why3 and Alt-Ergo as an alternative back-end for checking proof obligations generated by *Atelier B*, whose main applications are railroad-related software, a collaboration with Mitsubishi Electric R&D Centre Europe (Rennes) (joint publication [119]) and ClearSy (Aix-en-Provence).

SMT-based Model-Checking: Cubicle S. Conchon (with A. Mebsout and F. Zaidi from VALS team at LRI) has a long-term collaboration with S. Krstic and A. Goel (Intel Strategic Cad Labs in Hillsboro, OR, USA) that aims in the development of the SMT-based model checker Cubicle (<http://cubicle.lri.fr/>) based on Alt-Ergo [116][5]. It is particularly targeted to the verification of concurrent programs and protocols.

⁰<http://www.altran-praxis.com/book/>

VERIDIS Project-Team

4. Application Domains

4.1. Application Domains

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

Formal verification techniques can contribute to certifying 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 encourage the use of formal methods. While initially the requirements of certified development have mostly been restricted to safety-critical systems, the cost of unavailable services due to malfunctioning system components and software provides wider incentives for verification. For example, we have been working on modeling and verifying medical devices that require closed-loop models of both the system and its environment.

CARTE Team

4. Application Domains

4.1. 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., [39]), control theory (see e.g., [47]), neural networks (see e.g., [72]), and so on. We are interested in the formal decidability of properties of dynamical systems, such as reachability [63], the Skolem-Pisot problem [44], the computability of the ω -limit set [62]. 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 analyzing 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 [75], on recursive analysis [79], on the algebraic approach [70] and on Markov computability [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. 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 modelling 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 [54], [55], [56], to weak termination [57], sufficient completeness [58] and probabilistic termination [60]. 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 [59], [61]. Provided program code can be translated into rule-based specifications, this approach can be applied to correctness proofs 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 [66], [67], [68]. 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.

CIDRE Project-Team

4. Application Domains

4.1. Security is Required Everywhere

With the infiltration of computers and software in almost all aspects of our modern life, security can nowadays be seen as an absolutely general concern. As such, the results of the research targeted by CIDRE apply to a wide range of domains. It is clear that critical systems, in which security (and safety) is a major concern, can benefit from ideas such as dynamic security policy monitoring. On the other hand, systems used by the general public (basically, the internet and services such as web or cloud services, social networks, location-based services, etc.) can also benefit from results obtained by CIDRE, in particular to solve some of the privacy issues raised by these systems that manipulate huge amount of personal data. In addition, systems are getting more and more complex, decentralized, distributed, or spontaneous. Cloud computing, in particular, brings many challenges that could benefit from ideas, approaches and solutions studied by CIDRE in the context of distributed systems.

Industrial Control Systems (ICS) and in particular Supervisory Control and Data Acquisition are also new application domains for intrusion detection. The Stuxnet attack has emphasized the vulnerability of such critical systems which are not totally isolated anymore. Securing ICS is challenging since modifications of the systems, for example to patch them, are often not possible. High availability requirements also often conflict with preventive approaches. In this case, security monitoring is appealing to protect such systems against malicious activities. Intrusion detection in ICS is not fundamentally different from traditional approaches. However, new hypotheses and constraints need to be taken into account, which also bring interesting new research challenges.

COMETE Project-Team

4. Application Domains

4.1. Security and privacy

Participants: Konstantinos Chatzikokolakis, Catuscia Palamidessi, Ali Kassem, Anna Pazii, Tymofii Prokopenko.

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

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

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

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

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

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

DATASPHERE Team

4. Application Domains

4.1. Governance

Application domains include:

- City governance, data and interaction with citizens.
- New voting mechanisms.
- Local governance vs global norms and control.
- Security in an unstable world.
- Adaptation to the conditions of the anthropocene, resources and homeostasis.
- Data strategy for digital economy

PESTO Project-Team

4. Application Domains

4.1. Formal methods for cryptographic protocols

Security protocols, such as TLS, Kerberos or ssh, are the main tool for securing our communications. The aim of our work is to propose models that are expressive enough to formally represent protocol executions in the presence of an adversary, formal definitions of the security properties to be satisfied by these protocols, and design automated tools able to analyse them and possibly exhibit design flaws.

4.2. Automated reasoning

Many techniques for symbolic verification of security are rooted in automated reasoning. A typical example is equational reasoning used to model the algebraic properties of a cryptographic primitive. Our work therefore aims to improve and adapt existing techniques or propose new ones when needed for reasoning about security.

4.3. Electronic voting

Electronic elections have in the last years been used in several countries for politically binding elections. The use in professional elections is even more widespread. The aim of our work is to increase our understanding of the security properties needed for secure elections, propose techniques for analysing e-voting protocols, design of state-of-the-art voting protocols, but also to highlight the limitations of e-voting solutions.

4.4. Privacy in social networks

Treatment of information released by users on social networks can violate a user's privacy. The goal of our work is to allow one a controlled information release while guaranteeing a user's privacy.

PRIVATICS Project-Team

3. Application Domains

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

3.2. Big Data and Privacy

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

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

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

PROSECCO Project-Team

4. Application Domains

4.1. Cryptographic Protocol Libraries

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

4.2. Hardware-based security APIs

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

4.3. Web application security

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

TAMIS Team

4. Application Domains

4.1. System analysis

The work performed in Axes 1 and 2 and the methods developed there are applicable to the domain of system analysis, both wrt. program analysis and hardware analysis.

4.2. Cybersecurity

The work done in the 3 axes above aims at improving cybersecurity, be it via vulnerability analyses, malware analyses and the development of safer networking mechanisms.

4.3. Safe Internet

The work done in Axis 3 above very directly contributes to the goal of a safer Internet.