



RESEARCH CENTER
Rennes - Bretagne-Atlantique

FIELD

Activity Report 2012

Section Scientific Foundations

Edition: 2013-04-24

ALGORITHMICS, PROGRAMMING, SOFTWARE AND ARCHITECTURE

1. ALF Project-Team 4
 2. CAIRN Project-Team 12
 3. CELTIQUE Project-Team 16
 4. ESPRESSO Project-Team 21
 5. S4 Project-Team 26
 6. TASC Project-Team 28
 7. VERTECS Project-Team 31

APPLIED MATHEMATICS, COMPUTATION AND SIMULATION

8. ASPI Project-Team 37
 9. I4S Team 43
 10. IPSO Project-Team 50

COMPUTATIONAL SCIENCES FOR BIOLOGY, MEDICINE AND THE ENVIRONMENT

11. DYLISS Team 56
 12. FLUMINANCE Project-Team 61
 13. GENSCALE Team 68
 14. SAGE Project-Team 69
 15. SERPICO Team 70
 16. VISAGES Project-Team 72

NETWORKS, SYSTEMS AND SERVICES, DISTRIBUTED COMPUTING

17. ACES Project-Team 74
 18. ASAP Project-Team 78
 19. ASCOLA Project-Team 80
 20. ATLANMOD Team 84
 21. CIDRE Project-Team 86
 22. DIONYSOS Project-Team 90
 23. DISTRIBCOM Project-Team 92
 24. KERDATA Project-Team 95
 25. MYRIADS Project-Team 97
 26. TRISKELL Project-Team 100

PERCEPTION, COGNITION, INTERACTION

27. DREAM Project-Team 104
 28. LAGADIC Project-Team 107
 29. METISS Project-Team 110
 30. MIMETIC Team 116
 31. SIROCCO Project-Team 119
 32. TEXMEX Project-Team 122
 33. VR4I Team 125

ALF Project-Team

3. Scientific Foundations

3.1. Motivations

Multicores have become mainstream in general-purpose as well as embedded computing in the last few years. The integration technology trend allows to anticipate that a 1000-core chip will become feasible before 2020. On the other hand, while traditional parallel application domains, e.g. supercomputing and transaction servers, are benefiting from the introduction of multicores, there are very few new parallel applications that have emerged during the last few years.

In order to allow the end-user to benefit from the technological breakthrough, new architectures have to be defined for the 2020's many-cores, new compiler and code generation techniques as well as new performance prediction/guarantee techniques have to be proposed .

3.2. The context

3.2.1. *Technological context: The advent of multi- and many- cores architecture*

For almost 30 years since the introduction of the first microprocessor, the processor industry was driven by the Moore's law till 2002, delivering performance that doubled every 18-24 months on a uniprocessor. However since 2002 , and despite new progress in integration technology, the efforts to design very aggressive and very complex wide issue superscalar processors have essentially been stopped due to poor performance returns, as well as power consumption and temperature walls.

Since 2002-2003, the microprocessor industry has followed a new path for performance: the so-called multicore approach, i.e., integrating several processors on a single chip. This direction has been followed by the whole processor industry. At the same time, most of the computer architecture research community has taken the same path, focusing on issues such as scalability in multicores, power consumption, temperature management and new execution models, e.g. hardware transactional memory.

In terms of integration technology, the current trend will allow to continue to integrate more and more processors on a single die. Doubling the number of cores every two years will soon lead to up to a thousand processor cores on a single chip. The computer architecture community has coined these future processor chips as many-cores.

3.2.2. *The application context: multicores, but few parallel applications*

For the past five years, small scale parallel processor chips (hyperthreading, dual and quad-core) have become mainstream in general-purpose systems. They are also entering the high-end embedded system market. At the same time, very few (scalable) mainstream parallel applications have been developed. Such development of scalable parallel applications is still limited to niche market segments (scientific applications, transaction servers).

3.2.3. *The overall picture*

Till now, the end-user of multicores is experiencing improved usage comfort because he/she is able to run several applications at the same time. Eventually, in the near future with the 8-core or the 16-core generation, the end-user will realize that he/she is not experiencing any functionality improvement or performance improvement on current applications. The end-user will then realize that he/she needs more effective performance rather than more cores. The end-user will then ask either for parallel applications or for more effective performance on sequential applications.

3.3. Technology induced challenges

3.3.1. The power and temperatures walls

The power and the temperature walls largely contributed to the emergence of the small-scale multicores. For the past five years, mainstream general-purpose multicores have been built by assembling identical superscalar cores on a chip (e.g. IBM Power series). No new complex power hungry mechanisms were introduced in the core architectures, while power saving techniques such as power gating, dynamic voltage and frequency scaling were introduced. Therefore, since 2002, the designers have been able to keep the power consumption budget and the temperature of the chip within reasonable envelopes while scaling the number of cores with the technology.

Unfortunately, simple and efficient power saving techniques have already caught most of the low hanging fruits on energy consumption. Complex power and thermal management mechanisms are now becoming mainstream; e.g. the Intel Montecito (IA64) featured an adjunct (simple) core which unique mission is to manage the power and temperature on two cores. Processor industry will require more and more heroic efforts on this power and temperature management policy to maintain its current performance scaling path. Hence the power and temperature walls might slow the race towards 100's and 1000's cores unless the processor industry takes a new paradigm shift from the current "replicating complex cores" (e.g. Intel Nehalem) towards many simple cores (e.g. Intel Larrabee) or heterogeneous manycores (e.g. new GPUs, IBM Cell).

3.3.2. The memory wall

For the past 20 years, the memory access time has been one of the main bottlenecks for performance in computer systems. This was already true for uniprocessors. Complex memory hierarchies have been defined and implemented in order to limit the visible memory access time as well as the memory traffic demands. Up to three cache levels are implemented for uniprocessors. For multi- and many-cores the problems are even worse. The memory hierarchy must be replicated for each core, memory bandwidth must be shared among the distinct cores, data coherency must be maintained. Maintaining cache coherency for up to 8 cores can be handled through relatively simple bus protocols. Unfortunately, these protocols do not scale for large numbers of cores, and there is no consensus on coherency mechanism for manycore systems. Moreover there is no consensus on core organization (flat ring? flat grid? hierarchical ring or grid?).

Therefore, organizing and dimensioning the memory hierarchy will be a major challenge for the computer architects. The successful architecture will also be determined by the ability of the applications (i.e., the programmers or the compilers or the run-time) to efficiently place data in the memory hierarchy and achieve high performance.

Finally new technology opportunities may demand to revisit the memory hierarchy. As an example, 3D memory stacking enables a huge last-level cache (maybe several gigabytes) with huge bandwidth (several Kbits/ processor cycle). This dwarfs the main memory bandwidth and may lead to other architectural tradeoffs.

3.4. Need for efficient execution of parallel applications

Achieving high performance on future multicores will require the development of parallel applications, but also an efficient compiler/runtime tool chain to adapt codes to the execution platform.

3.4.1. The diversity of parallelisms

Many potential execution parallelism patterns may coexist in an application. For instance, one can express some parallelism with different tasks achieving different functionalities. Within a task, one can expose different granularities of parallelism; for instance a first layer message passing parallelism (processes executing the same functionality on different parts of the data set), then a shared memory thread level parallelism and fine grain loop parallelism (a.k.a vector parallelism).

Current multicores already feature hardware mechanisms to address these different parallelisms: physically distributed memory — e.g. the new Intel Nehalem already features 6 different memory channels — to address task parallelism, thread level parallelism — e.g. on conventional multicores, but also on GPUs or on Cell-based machines —, vector/SIMD parallelism — e.g. multimedia instructions. Moreover they also attack finer instruction level parallelism and memory latency issues. Compilers have to efficiently discover and manage all these forms to achieve effective performance.

3.4.2. Portability is the new challenge

Up to now, most parallel applications were developed for specific application domains in high end computing. They were used on a limited set of very expensive hardware platforms by a limited number of expert users. Moreover, they were executed in batch mode.

In contrast, the expectation of most end-users of the future mainstream parallel applications running on multicores will be very different. The mainstream applications will be used by thousands, maybe millions of non-expert users. These users consider functional portability of codes as granted. They will expect their codes to run faster on new platforms featuring more cores. They will not be able to tune the application environment to optimize performance. Finally, multiple parallel applications may have to be executed concurrently.

The variety of possible hardware platforms, the lack of expertise of the end-users and the varying run-time execution environments will represent major difficulties for applications in the multicore era.

First of all, the end user considers functional portability without recompilation as granted, this is a major challenge on parallel machines. Performance portability/scaling is even more challenging. It will become inconceivable to rewrite/retune each application for each new parallel hardware platform generation to exploit them. Therefore, apart from the initial development of parallel applications, the major challenge for the next decade will be to *efficiently* run parallel applications on hardware architectures radically different from their original hardware target.

3.4.3. The need for performance on sequential code sections

3.4.3.1. Most software will exhibit substantial sequential code sections

For the foreseeable future, the majority of applications will feature important sequential code sections.

First, many legacy codes were developed for uniprocessors. Most of these codes will not be completely redeveloped as parallel applications, but will evolve to applications using parallel sections for the most compute-intensive parts. Second, the overwhelming majority of the programmers have been educated to program in a sequential programming style. Parallel programming is much more difficult, time consuming and error prone than sequential programming. Debugging and maintaining a parallel code is a major issue. Investing in the development of a parallel application will not be cost-effective for the vast majority of software developments. Therefore, sequential programming style will continue to be dominant in the foreseeable future. Most developers will rely on the compiler to parallelize their application and/or use some software components from parallel libraries.

3.4.3.2. Future parallel applications will require high performance sequential processing on 1000's cores chip

With the advent of universal parallel hardware in multicores, large diffusion parallel applications will have to run on a broad spectrum of parallel hardware platforms. They will be used by non-expert users who will not be able to tune the application environment to optimize performance. They will be executed concurrently with other processes which may be interactive.

The variety of possible hardware platforms, the lack of expertise of the end-user and the varying run-time execution environments are major difficulties for parallel applications. This tends to constrain the programming style and therefore reinforces the sequential structure of the control of the application.

Therefore, *most future parallel applications will rely on a single main thread or a few main threads in charge of distinct functionalities of the application. Each main thread will have a general sequential control and can initiate and control the parallel execution of parallel tasks.*

In 1967, Amdahl [34] pointed out that, if only a portion of an application is accelerated, the execution time cannot be reduced below the execution time of the residual part of the application. Unfortunately, even highly parallelized applications exhibit some residual sequential part. For parallel applications, this indicates that the effective performance of the future 1000's cores chip will significantly depend on their ability to be efficient on the execution of the control portions of the main thread as well as on the execution of sequential portions of the application.

3.4.3.3. *The success of 1000's cores architecture will depend on single thread performance*

While the current emphasis of computer architecture research is on the definition of scalable multi- many- core architectures for highly parallel applications, we believe that the success of the future 1000-core architecture will depend not only on their performance on parallel applications including sequential sections, but also on their performance on single thread workloads.

3.5. Performance evaluation/guarantee

Predicting/evaluating the performance of an application on a system without explicitly executing the application on the system is required for several usages. Two of these usages are central to the research of the ALF project-team: microarchitecture research (the system to be evaluated does not exist) and Worst Case Execution Time estimation for real-time systems (the numbers of initial states or possible data inputs is too large).

When proposing a micro-architecture mechanism, its impact on the overall processor architecture has to be evaluated in order to assess its potential performance advantages. For microarchitecture research, this evaluation is generally done through the use of cycle-accurate simulation. Developing such simulators is quite complex and microarchitecture research was helped but also biased by some popular public domain research simulators (e.g. Simplescalar [36]). Such simulations are CPU consuming and simulations cannot be run on a complete application. Sampling representative slices of the application was proposed [5] and popularized by the Simpoint [45] framework.

Real-time systems need a different use of performance prediction; on hard real-time systems, timing constraints must be respected independently from the data inputs and from the initial execution conditions. For such a usage, the Worst Case Execution Time (WCET) of an application must be evaluated and then checked against the timing constraints. While safe and tight WCET estimation techniques and tools exist for reasonably simple embedded processors (e.g. techniques based on abstract interpretation such as [38]), accurate evaluation of the WCET of an algorithm on a complex uniprocessor system is a difficult problem. Accurately modelling data cache behavior [4] and complex superscalar pipelines are still research questions as illustrated by the presence of so-called *timing anomalies* in dynamically scheduled processors, resulting from complex interactions between processor elements (among others, interactions between caching and instruction scheduling) [42].

With the advance of multicores, evaluating / guaranteeing a computer system response time is becoming much more difficult. Interactions between processes occurs at different levels. The execution time on each core depends on the behavior of the other cores. Simulations of 1000's cores micro-architecture will be needed in order to evaluate future many-core proposals. While a few multiprocessor simulators are available for the community, these simulators cannot handle realistic 1000's cores micro-architecture. New techniques have to be invented to achieve such simulations. WCET estimations on multicore platforms will also necessitate radically new techniques, in particular, there are predictability issues on a multicore where many resources are shared; those resources include the memory hierarchy, but also the processor execution units and all the hardware resources if SMT is implemented [49].

3.6. General research directions

The overall performance of a 1000's core system will depend on many parameters including architecture, operating system, runtime environment, compiler technology and application development. In the ALF project, we will essentially focus on architecture, compiler/execution environment as well as performance

predictability, and in particular WCET estimation. Moreover, architecture research, and to a smaller extent, compiler and WCET estimation researches rely on processor simulation. A significant part of the effort in ALF will be devoted to define new processor simulation techniques.

3.6.1. Microarchitecture research directions

The overall performance of a multicore system depends on many parameters including architecture, operating system, runtime environment, compiler technology and application development. Even the architecture dimension of a 1000's core system cannot be explored by a single research project. Many research groups are exploring the parallel dimension of the multicores essentially targeting issues such as coherency and scalability.

We have identified that high performance on single threads and sequential codes is one of the key issues for enabling overall high performance on a 1000's core system and we anticipate that the general architecture of such 1000's core chip will feature many simple cores and a few very complex cores.

Therefore our research in the ALF project will focus on refining the microarchitecture to achieve high performance on single process and/or sequential code sections within the general framework of such an heterogeneous architecture. This leads to two main research directions 1) enhancing the microarchitecture of high-end superscalar processors, 2) exploiting/modifying heterogeneous multicore architecture on a single process. The temperature wall is also a major technological/architectural issue for the design of future processor chips.

3.6.1.1. Enhancing complex core microarchitecture

Research on wide issue superscalar processors was merely stopped around 2002 due to limited performance returns and the power consumption wall.

When considering a heterogeneous architecture featuring hundreds of simple cores and a few complex cores, these two obstacles will partially vanish: 1) the complex cores will represent only a fraction of the chip and a fraction of its power consumption. 2) any performance gain on (critical) sequential threads will result in a performance gain of the whole system

On the complex core, the performance of a sequential code is limited by several factors. At first, on current architectures, it is limited by the peak performance of the processor. To push back this first limitation, we will explore new microarchitecture mechanisms to increase the potential peak performance of a complex core enabling larger instruction issue width. The processor performance is also limited by control dependencies. To push back this limitation, we will explore new branch prediction mechanisms as well as new directions for reducing branch misprediction penalties [14], [13]. As data dependencies may strongly limit performance, we will revisit data prediction. Processor performance is also often highly dependent on the presence or absence of data in a particular level of the memory hierarchy. For the ALF multicore, we will focus on sharing the access to the memory hierarchy in order to adapt the performance of the main thread to the performance of the other cores. All these topics should be studied with the new perspective of quasi unlimited silicon budget.

3.6.1.2. Exploiting heterogeneous multicores on single process

When executing a sequential section on the complex core, the simple cores will be free. Two main research directions to exploit thread level parallelism on a sequential thread have been initiated in late 90's within the context of simultaneous multithreading and early chip multiprocessor proposals: helper threads and speculative multithreading.

Helper threads were initially proposed to improve the performance of the main threads on simultaneous multithreaded architectures [37]. The main idea of helper threads is to execute codes that will accelerate the main thread without modifying its semantic.

In many cases, the compiler cannot determine if two code sections are independent due to some unresolved memory dependency. When no dependency occurs at execution time, the code sections can be executed in parallel. Thread-Level Speculation has been proposed to exploit coarse grain speculative parallelism. Several hardware-only proposals were presented [44], but the most promising solutions integrate hardware support for software thread-level speculation [47].

In the context of future manycores, thread-level speculation and helper threads should be revisited. Many simple cores will be available for executing helper threads or speculative thread execution during the execution of sequential programs or sequential code sections. The availability of these many cores is an opportunity as well as a challenge. For example, one can try to use the simple cores to execute many different helper threads that could not be implemented within a simultaneous multithreaded processor. For thread level speculation, the new challenge is the use of less powerful cores for speculative threads. Moreover the availability of many simple cores may lead to the use of helper threads and thread level speculation at the same time.

3.6.1.3. Temperature issues

Temperature is one of the constraints that have prevented the processor clock frequency to be increased in recent years. Besides techniques to decrease the power consumption, the temperature issue can be tackled with *dynamic thermal management* [10] through techniques such as clock gating or throttling and *activity migration* [46][7].

Dynamic thermal management (DTM) is now implemented on existing processors. For high performance, processors are dimensioned according to the average situation rather than to the worst case situation. Temperature sensors are used on the chip to trigger dynamic thermal management actions, for instance thermal throttling whenever necessary. On multicores, it is possible to migrate the activity from one core to another in order to limit temperature.

A possible way to increase sequential performance is to take advantage of the smaller gate delay that comes with miniaturization, which permits in theory to increase the clock frequency. However increasing the clock frequency generally requires to increase the instantaneous power density. This is why DTM and activity migration will be key techniques to deal with Amdahl's law in future many-core processors.

3.6.2. Processor simulation research

Architecture studies, and in particular microarchitecture studies, require extensive validations through detailed simulations. Cycle accurate simulators are needed to validate the microarchitectural mechanisms.

Within the ALF project, we can distinguish two major requirements on the simulation: 1) single process and sequential code simulations 2) parallel code sections simulations.

For simulating parallel code sections, a cycle-accurate microarchitecture simulator of a 1000-core architecture will be unacceptably slow. In [9], we showed that mixing analytical modeling of the global behavior of a processor with detailed simulation of a microarchitecture mechanism allows to evaluate this mechanism. Karkhanis and Smith [39] further developed a detailed analytical simulation model of a superscalar processor. Building on top of these preliminary researches, simulation methodology mixing analytical modeling of the simple cores with a more detailed simulation of the complex cores is appealing. The analytical model of the simple cores will aim at approximately modeling the impact of the simple core execution on the shared resources (e.g. data bandwidth, memory hierarchy) that are also used by the complex cores.

Other techniques such as regression modeling [40] can also be used for decreasing the time required to explore the large space of microarchitecture parameter values. We will explore these techniques in the context of many-core simulation.

In particular, research on temperature issues will require the definition and development of new simulation tools able to simulate several minutes or even hours of processor execution, which is necessary for modeling thermal effects faithfully.

3.6.3. Compiler research directions

3.6.3.1. General directions

Compilers are keystone solutions for any approach that deals with high performance on 100+ processors systems. But general-purpose compilers try to embrace so many domains and try to serve so many constraints that they frequently fail to achieve very high performance. They need to be deeply revisited. We identify four main compiler/software related issues that must be addressed in order to allow efficient use of multi- and many-cores: 1) programming 2) resource management 3) application deployment 4) portable performance. Addressing these challenges will require to revisit parallel programming and code generation extensively.

The past of parallel programming is scattered with hundreds of parallel languages. Most of these languages were designed to program homogeneous architectures and were targeting a small and well-trained community of HPC programmers. With the new diversity of parallel hardware platforms and the new community of non-expert developers, expressing parallelism is not sufficient anymore. Resource management, application deployment and portable performance are intermingled issues that require to be addressed holistically.

As many decisions should be taken according to the available hardware, resource management cannot be separated from parallel programming. Deploying applications on various systems without having to deal with thousands of hardware configurations (different numbers of cores, accelerators, ...) will become a major concern for software distribution. The grail of parallel computing is to be able to provide portable performance on a large set of parallel machines and varying execution contexts.

Recent techniques are showing promises. Iterative compilation techniques, exploiting the huge CPU cycle count now available, can be used to explore the optimization space at compile-time. Second, machine-learning techniques can be used to automatically improve compilers and code generation strategies. Speculation can be used to deal with necessary but missing information at compile-time. Finally, dynamic techniques can select or generate at run-time the most efficient code adapted to the execution context and available hardware resources.

Future compilers will benefit from past research, but they will also need to combine static and dynamic techniques. Moreover, domain specific approaches might be needed to ensure success. The ALF research effort will focus on these static and dynamic techniques to address the multicore application development challenges.

3.6.3.2. *Portability of applications and performance through virtualization*

The life cycle is much longer for applications than for hardware. Unfortunately the multicore era jeopardizes the old binary compatibility recipe. Binaries cannot automatically exploit additional computing cores or new accelerators available on the silicon. Moreover maintaining backward binary compatibility on future parallel architectures will rapidly become a nightmare, applications will not run at all unless some kind of dynamic binary translation is at work.

Processor virtualization addresses the problem of portability of functionalities. Applications are not compiled to the final native code but to a target independent format. This is the purpose of languages such as Java and .NET. Bytecode formats are often *a priori* perceived as inappropriate for performance intensive applications and for embedded systems. However, it was shown that compiling a C or C++ program to a bytecode format produces a code size similar to dense instruction sets [3]. Moreover, this bytecode representation can be compiled to native code with performance similar to static compilation [2]. Therefore processor virtualization for high performance, i.e., for languages like C or C++, provides significant advantages: 1) it simplifies software engineering with fewer tools to maintain and upgrade; 2) it allows better code readability and easier code maintenancesince it avoids code specialization for specific targets using compile time macros such as `#ifdef` ; 3) the *execution code* deployed on the system is the execution code that has been debugged and validated, as opposed to the same *source code* has been recompiled for another platform; 4) new architectures will come with their JIT compiler. The JIT will (should) automatically take advantage of new architecture features such as SIMD/vector instructions or extra processors.

Our objective is to enrich processor virtualization to allow both functional portability and high performance using JIT at runtime, or bytecode-to-native code offline compiler. Split compilation can be used to annotate the bytecode with relevant information that can be helpful to the JIT at runtime or to the bytecode to native code offline compiler. Because the first compilation pass occurs offline, aggressive analyses can be run and their outcomes encoded in the bytecode. For example, such informations include vectorizability, memory references (in)dependencies, suggestions derived from iterative compilation, polyhedral analysis, or integer linear programming. Virtualization allows to postpone some optimizations to run time, either because they increase the code size and would increase the cost of an embedded system or because the actual hardware platform characteristics are unknown.

3.6.4. *Performance predictability for real-time systems*

While compiler and architecture research efforts often focus on maximizing average case performance, applications with real-time constraints do not need only high performance but also performance guarantees in all situations, including the worst-case situation. Worst-Case Execution Time estimates (WCET) need to be upper bounds of any possible execution time. The safety level required depends on the criticality of applications: missing a frame on a video in the airplane for passenger in seat 20B is less critical than a safety critical decision in the control of the airplane.

Within the ALF project, our objective is to study performance guarantees for both (i) sequential codes running on complex cores ; (ii) parallel codes running on the multicores. Considering the ALF base architecture, this results in two quite distinct problems.

For sequential code executing on a single core, one can expect that, in order to provide real-time possibility, the architecture will feature an execution mode where a given processor will be guaranteed to access a fixed portion of the shared resources (caches, memory bandwidth). Moreover, this guaranteed share could be optimized at compile time to enforce the respect of the time constraints. However, estimating the WCET of an application on a complex micro-architecture is still a research challenge. This is due to the complex interaction of micro-architectural elements (superscalar pipelines, caches, branch prediction, out-of-order execution) [42]. We will continue to explore pure analytical and static methods. However when accurate static hardware modeling methods cannot handle the hardware complexity, new probabilistic methods [41] might be needed to explore to obtain as safe as possible WCET estimates.

Providing performance guarantees for parallel applications executed on a multicore is a new and challenging issue. Entirely new WCET estimation methods have to be defined for these architectures to cope with dynamic resource sharing between cores, in particular on-chip memory (either local memory or caches) are shared, but also buses, network-on-chip and the access to the main memory. Current pure analytical methods are too pessimistic at capturing interferences between cores [50], therefore hardware-based or compiler methods such as [48] have to be defined to provide some degree of isolation between cores. Finally, similarly to simulation methods, new techniques to reduce the complexity of WCET estimation will be explored to cope with manycore architectures.

CAIRN Project-Team

3. Scientific Foundations

3.1. Panorama

The development of complex applications is traditionally split in three stages: a theoretical study of the algorithms, an analysis of the target architecture and the implementation. When facing new emerging applications such as high-performance, low-power and low-cost mobile communication systems or smart sensor-based systems, it is mandatory to strengthen the design flow by a joint study of both algorithmic and architectural issues ¹.

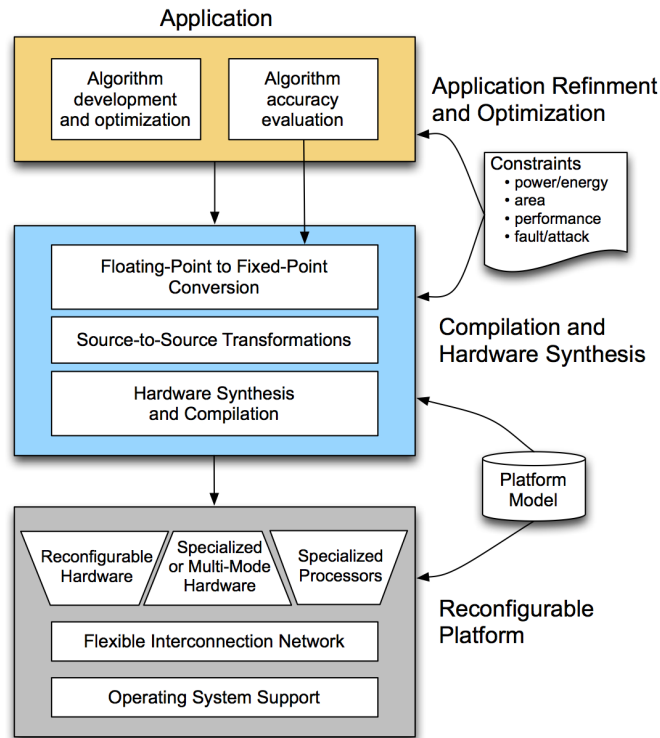


Figure 1. CAIRN's general design flow and related research themes

Figure 1 shows the global design flow that we propose to develop. This flow is organized in levels which refer to our three research themes: application optimization (new algorithms, fixed-point arithmetic and advanced representations of numbers), architecture optimization (reconfigurable and specialized hardware, application-specific processors), and stepwise refinement and code generation (code transformations, hardware synthesis, compilation).

¹ Often referenced as algorithm-architecture mapping or interaction.

In the rest of this part, we briefly describe the challenges concerning **new reconfigurable platforms** in Section 3.2, the issues on **compiler and synthesis tools** related to these platforms in Section 3.3, and the remaining challenges in **algorithm architecture interaction** in Section 3.4.

3.2. Reconfigurable Architecture Design

Over the last two decades, there has been a strong push of the research community to evolve static programmable processors into run-time dynamic and partial reconfigurable (DPR) architectures. Several research groups around the world have hence proposed reconfigurable hardware systems operating at various levels of granularity. For example, functional-level reconfiguration has been proposed to increase the efficiency of programmable processors without having to pay for the FPGAs penalties. These coarse-grained reconfigurable architectures (CGRAs) provide operator-level configurable functional blocks and word-level datapaths. The main goal of this class of architectures is to provide flexibility while minimizing reconfiguration overhead (there exists several recent surveys on this topic [120], [104], [85], [125]). Compared to fine-grained architectures, CGRAs benefit from a massive reduction in configuration memory and configuration delay, as well as a considerable reduction in routing and placement complexity. This, in turns, results in an improvement in the computation volume over energy cost ratio, even if it comes at the price of a loss of flexibility compared to bit-level operations. Such constraints have been taken into account in the design of DART [100][12], CRIP [88], Adres [112] or others [122]. These works have led to commercial products such as the Extreme Processor Platform (XPP) [89] from PACT or Montium² from Recore systems.

Another strong trend is the design of hybrid architectures which combine standard GPP or DSP cores with arrays of *configurable elements* such as the Lx [103], or of *field-configurable elements* such as the Xirisc processor [110] and more recently by commercial platforms such as the Xilinx Zynq-7000. Some of their benefits are the following: functionality on demand (set-top boxes for digital TV equipped with decoding hardware on demand), acceleration on demand (coprocessors that accelerate computationally demanding applications in multimedia or communications applications), and shorter time-to-market (products that target ASIC platforms can be released earlier using reconfigurable hardware).

Dynamic reconfiguration enables an architecture to adapt itself to various incoming tasks. This requires complex resource management and control which can be provided as services by a real-time operating system (RTOS) [111]: communication, memory management, task scheduling [99], [92][1] and task placement [19]. Such an Operating System (OS) based approach has many advantages: it provides a complete design framework, that is independent of the technology and of the underlying hardware architecture, helping to drastically reduce the full platform design time. Due to the unpredictable execution of tasks, the OS must be able to allocate resource to tasks at run-time along with mechanisms to support inter-task communication. An efficient way to support such communications is to resort to a network-on-chip [118]. The role of the communication infrastructure is then to support transactions between different components of the platform, either between macro-components – main processor, dedicated modules, dynamically reconfigurable component – or within the elements of the reconfigurable components themselves.

In CAIRN we mainly target reconfigurable system-on-chip (RSoC) defined as a set of computing and storing resources organized around a flexible interconnection network and integrated within a single silicon chip (or programmable chip such as FPGAs). The architecture is customized for an application domain, and the flexibility is provided by both hardware reconfiguration and software programmability. Computing resources are therefore highly heterogeneous and raise many issues that we discuss in the following:

- **Reconfigurable hardware blocks with a dynamic behavior** where reconfigurability can be achieved at the bit- or operator-level. Our research aims at defining new reconfigurable architectures including computing and memory resources. Since reconfiguration must happen as fast as possible (typically within a few cycles), reducing the configuration time overhead is also a key issue.

²<http://www.recoresystems.com/technology/montium-technology>

- When performance and power consumption are major constraints, it is acknowledged that optimized specialized hardware blocks (often called IPs for Intellectual Properties) are the best (and often the only) solution. Therefore, we also study architecture and tools for **specialized hardware accelerators** and for **multi-mode components**.
- Customized **processors with a specialized instruction-set** also offer a viable solution to trade between energy efficiency and flexibility. They are particularly relevant for modern FPGA platforms where many processor cores can be embedded. For this topic, we focus on the automatic generation of heterogeneous (sequential or parallel) reconfigurable processor extensions that are tightly coupled to processor cores.

3.3. Compilation and Synthesis for Reconfigurable Platforms

In spite of their advantages, reconfigurable architectures lack efficient and standardized compilation and design tools. As of today, this still makes the technology impractical for large scale industrial use. Generating and optimizing the mapping from high-level specifications to reconfigurable hardware platforms is therefore a key research issue, and the problem has received considerable interest over the last years [115], [91], [121], [124]. In the meantime, the complexity (and heterogeneity) of these platforms has also been increasing quite significantly, with complex heterogeneous multi-cores architectures becoming a *de facto* standard. As a consequence, the focus of designers is now geared toward optimizing overall system-level performance and efficiency [106], [115], [114]. Here again, existing tools are not well suited, as they fail at providing a unified programming view of the programmable and/or reconfigurable components implemented on the platform.

In this context we have been pursuing our efforts to propose tools whose design principles are based on a tight coupling between the compiler and the target hardware architectures. We build on the expertise of the team members in High Level Synthesis (HLS) [8], ASIP optimizing compilers [15] and automatic parallelization for massively parallel specialized circuits [6]. We first study how to increase the efficiency of standard programmable processor by extending their instruction set to speed-up compute intensive kernels. Our focus is on efficient and exact algorithms for the identification, selection and scheduling of such instructions [9]. We also propose techniques to synthesize reconfigurable (or multi-mode) architectures. We address these challenges by borrowing techniques from high-level synthesis, optimizing compilers and automatic parallelization, especially when dealing with nested loop kernels. The goal is then either to derive a custom fine-grain parallel architecture and/or to derive the configuration of a Coarse Grain Reconfigurable Architecture (CGRA). In addition, and independently of the scientific challenges mentioned above, proposing such flows also poses significant software engineering issues. As a consequence, we also study how leading edge Object Oriented software engineering techniques (Model Driven Engineering) can help the Computer Aided Design (CAD) and optimizing compiler communities prototyping new research ideas.

Efficient implementation of multimedia and signal processing applications (in software for DSP cores or as special-purpose hardware) often requires, for reasons related to cost, power consumption or silicon area constraints, the use of fixed-point arithmetic, whereas the algorithms are usually specified in floating-point arithmetic. Unfortunately, fixed-point conversion is very challenging and time-consuming, typically demanding up to 50% of the total design or implementation time [93]. Thus, tools are required to automate this conversion. For hardware or software implementation, the aim is to optimize the fixed-point specification. The implementation cost is minimized under a numerical accuracy or an application performance constraint. For DSP-software implementation, methodologies have been proposed [108], [113] to achieve a conversion leading to an ANSI-C code with integer data types. For hardware implementation, the best results are obtained when the word-length optimization process is coupled with the high-level synthesis [107], [96]. Evaluating the effects of finite precision is one of the major and often the most time consuming step while performing fixed-point refinement. Indeed, in the word-length optimization process, the numerical accuracy is evaluated as soon as a new word-length is tested, thus, several times per iteration of the optimization process. Classical approaches are based on fixed-point simulations [97], [119]. They lead to long evaluation times and cannot be used to explore the entire design space. Therefore, our aim is to propose closed-form expressions of errors due to fixed-point approximations that are used by a fast analytical framework for accuracy evaluation.

3.4. Interaction between Algorithms and Architectures

As CAIRN mainly targets domain-specific system-on-chip including reconfigurable capabilities, algorithmic-level optimizations have a great potential on the efficiency of the overall system. Based on the skills and experiences in “signal processing and communications” of some CAIRN’s members, we conduct research on algorithmic optimization techniques under two main constraints: energy consumption and computation accuracy; and for two main application domains: fourth-generation (4G) mobile communications and wireless sensor networks (WSN). These application domains are very conducive to our research activities. The high complexity of the first one and the stringent power constraint of the second one, require the design of specific high-performance and energy-efficient SoCs. We also consider other applications such as video or bioinformatics, but this short state-of-the-art will be limited to wireless applications.

The radio in both transmit and receive modes consumes the bulk of the total power consumption of the system. Therefore, protocol optimization is one of the main sources of significant energy reduction to be able to achieve self-powered autonomous systems. Reducing power due to radio communications can be achieved by two complementary main objectives: (i) minimizing the output transmit power while maintaining sufficient wireless link quality and (ii) minimizing useless wake-up and channel hearing while still being reactive.

As the physical layer affects all higher layers in the protocol stack, it plays an important role in the energy-constrained design of WSNs. The question to answer can be summarized as: *how much signal processing can be added to decrease the transmission energy (i.e. the output power level at the antenna) such that the global energy consumption be decreased?* The temporal and spatial diversity of relay and multiple antenna techniques are very attractive due to their simplicity and their performance for wireless transmission over fading channels. Cooperative MIMO (multiple-input and multiple-output) techniques have been first studied in [101], [109] and have shown their efficiency in terms of energy consumption [98]. Our research aims at finding new energy-efficient cooperative protocols associating distributed MIMO with opportunistic and/or multiple relays and considering wireless channel impairments such as transmitters desynchronisation.

Another way to reduce the energy consumption consists in decreasing the radio activity, controlled by the medium access (MAC) layer protocols. In this regard, low duty-cycle protocols, such as preamble-sampling MAC protocols, are very efficient because they improve the lifetime of the network by reducing the unnecessary energy waste [87]. As the network parameters (data rate, topology, etc.) can vary, we propose new adaptive MAC protocols to avoid overhearing and idle listening.

Finally, MIMO precoding is now recognized as a very interesting technique to enhance the data rate in wireless systems, and is already used in Wi-Max standard (802.16e). This technique can also be used to reduce transmission energy for the same transmission reliability and the same throughput requirement. One of the most efficient precoders is based on the maximization of the minimum Euclidean distance ($\max-d_{min}$) between two received data vectors [94], but it is difficult to define the closed-form of the optimized precoding matrix for large MIMO system with high-order modulations. Our goal is to derive new generic precoders with simple expressions depending only on the channel angle and the modulation order.

CELTIQUE Project-Team

3. Scientific Foundations

3.1. Static program analysis

Static program analysis is concerned with obtaining information about the run-time behaviour of a program without actually running it. This information may concern the values of variables, the relations among them, dependencies between program values, the memory structure being built and manipulated, the flow of control, and, for concurrent programs, synchronisation among processes executing in parallel. Fully automated analyses usually render approximate information about the actual program behaviour. The analysis is correct if the information includes all possible behaviour of a program. Precision of an analysis is improved by reducing the amount of information describing spurious behaviour that will never occur.

Static analysis has traditionally found most of its applications in the area of program optimisation where information about the run-time behaviour can be used to transform a program so that it performs a calculation faster and/or makes better use of the available memory resources. The last decade has witnessed an increasing use of static analysis in software verification for proving invariants about programs. The Celtique project is mainly concerned with this latter use. Examples of static analysis include:

- Data-flow analysis as it is used in optimising compilers for imperative languages. The properties can either be approximations of the values of an expression (“the value of variable x is greater than 0” or x is equal to y at this point in the program”) or more intensional information about program behaviour such as “this variable is not used before being re-defined” in the classical “dead-variable” analysis [71].
- Analyses of the memory structure includes shape analysis that aims at approximating the data structures created by a program. Alias analysis is another data flow analysis that finds out which variables in a program addresses the same memory location. Alias analysis is a fundamental analysis for all kinds of programs (imperative, object-oriented) that manipulate state, because alias information is necessary for the precise modelling of assignments.
- Control flow analysis will find a safe approximation to the order in which the instructions of a program are executed. This is particularly relevant in languages where parameters or functions can be passed as arguments to other functions, making it impossible to determine the flow of control from the program syntax alone. The same phenomenon occurs in object-oriented languages where it is the class of an object (rather than the static type of the variable containing the object) that determines which method a given method invocation will call. Control flow analysis is an example of an analysis whose information in itself does not lead to dramatic optimisations (although it might enable in-lining of code) but is necessary for subsequent analyses to give precise results.

Static analysis possesses strong **semantic foundations**, notably abstract interpretation [51], that allow to prove its correctness. The implementation of static analyses is usually based on well-understood constraint-solving techniques and iterative fixpoint algorithms. In spite of the nice mathematical theory of program analysis and the solid algorithmic techniques available one problematic issue persists, *viz.*, the *gap* between the analysis that is proved correct on paper and the analyser that actually runs on the machine. While this gap might be small for toy languages, it becomes important when it comes to real-life languages for which the implementation and maintenance of program analysis tools become a software engineering task. A *certified static analysis* is an analysis that has been formally proved correct using a proof assistant.

In previous work we studied the benefit of using abstract interpretation for developing **certified static analyses** [49], [74]. The development of certified static analysers is an ongoing activity that will be part of the Celtique project. We use the Coq proof assistant which allows for extracting the computational content of a constructive proof. A Caml implementation can hence be extracted from a proof of existence, for any program, of a correct approximation of the concrete program semantics. We have isolated a theoretical framework based on abstract interpretation allowing for the formal development of a broad range of static analyses. Several case studies for the analysis of Java byte code have been presented, notably a memory usage analysis [50]. This work has recently found application in the context of Proof Carrying Code and have also been successfully applied to particular form of static analysis based on term rewriting and tree automata [3].

3.1.1. *Static analysis of Java*

Precise context-sensitive control-flow analysis is a fundamental prerequisite for precisely analysing Java programs. Bacon and Sweeney's Rapid Type Analysis (RTA) [42] is a scalable algorithm for constructing an initial call-graph of the program. Tip and Palsberg [80] have proposed a variety of more precise but scalable call graph construction algorithms *e.g.*, MTA, FTA, XTA which accuracy is between RTA and O'CFA. All those analyses are not context-sensitive. As early as 1991, Palsberg and Schwartzbach [72], [73] proposed a theoretical parametric framework for typing object-oriented programs in a context-sensitive way. In their setting, context-sensitivity is obtained by explicit code duplication and typing amounts to analysing the expanded code in a context-insensitive manner. The framework accommodates for both call-contexts and allocation-contexts.

To assess the respective merits of different instantiations, scalable implementations are needed. For Cecil and Java programs, Grove *et al.*, [60], [59] have explored the algorithmic design space of contexts for benchmarks of significant size. Latter on, Milanova *et al.*, [66] have evaluated, for Java programs, a notion of context called *object-sensitivity* which abstracts the call-context by the abstraction of the `this` pointer. More recently, Lhotak and Hendren [64] have extended the empiric evaluation of object-sensitivity using a BDD implementation allowing to cope with benchmarks otherwise out-of-scope. Besson and Jensen [46] proposed to use DATALOG in order to specify context-sensitive analyses. Whaley and Lam [81] have implemented a context-sensitive analysis using a BDD-based DATALOG implementation.

Control-flow analyses are a prerequisite for other analyses. For instance, the security analyses of Livshits and Lam [65] and the race analysis of Naik, Aiken [67] and Whaley [68] both heavily rely on the precision of a control-flow analysis.

Control-flow analysis allows to statically prove the absence of certain run-time errors such as "message not understood" or cast exceptions. Yet it does not tackle the problem of "null pointers". Fahrnich and Leino [55] propose a type-system for checking that after object creation fields are non-null. Hubert, Jensen and Pichardie have formalised the type-system and derived a type-inference algorithm computing the most precise typing [63]. The proposed technique has been implemented in a tool called NIT [62]. Null pointer detection is also done by bug-detection tools such as FindBugs [62]. The main difference is that the approach of findbugs is neither sound nor complete but effective in practice.

3.1.2. *Quantitative aspects of static analysis*

Static analyses yield qualitative results, in the sense that they compute a safe over-approximation of the concrete semantics of a program, w.r.t. an order provided by the abstract domain structure. Quantitative aspects of static analysis are two-sided: on one hand, one may want to express and verify (compute) quantitative properties of programs that are not captured by usual semantics, such as time, memory, or energy consumption; on the other hand, there is a deep interest in quantifying the precision of an analysis, in order to tune the balance between complexity of the analysis and accuracy of its result.

The term of quantitative analysis is often related to probabilistic models for abstract computation devices such as timed automata or process algebras. In the field of programming languages which is more specifically addressed by the Celtique project, several approaches have been proposed for quantifying resource usage: a non-exhaustive list includes memory usage analysis based on specific type systems [61], [41], linear

logic approaches to implicit computational complexity [43], cost model for Java byte code [37] based on size relation inference, and WCET computation by abstract interpretation based loop bound interval analysis techniques [52].

We have proposed an original approach for designing static analyses computing program costs: inspired from a probabilistic approach [75], a quantitative operational semantics for expressing the cost of execution of a program has been defined. Semantics is seen as a linear operator over a dioid structure similar to a vector space. The notion of long-run cost is particularly interesting in the context of embedded software, since it provides an approximation of the asymptotic behaviour of a program in terms of computation cost. As for classical static analysis, an abstraction mechanism allows to effectively compute an over-approximation of the semantics, both in terms of costs and of accessible states [48]. An example of cache miss analysis has been developed within this framework [79].

3.1.3. Semantic analysis for test case generation

The semantic analysis of programs can be combined with efficient constraint solving techniques in order to extract specific information about the program, *e.g.*, concerning the accessibility of program points and feasibility of execution paths [76], [54]. As such, it has an important use in the automatic generation of test data. Automatic test data generation received considerable attention these last years with the development of efficient and dedicated constraint solving procedures and compositional techniques [58].

We have made major contributions to the development of **constraint-based testing**, which is a two-stage process consisting of first generating a constraint-based model of the program's data flow and then, from the selection of a testing objective such as a statement to reach or a property to invalidate, to extract a constraint system to be solved. Using efficient constraint solving techniques allows to generate test data that satisfy the testing objective, although this generation might not always terminate. In a certain way, these constraint techniques can be seen as efficient decision procedures and so, they are competitive with the best software model checkers that are employed to generate test data.

3.2. Software certification

The term "software certification" has a number of meanings ranging from the formal proof of program correctness via industrial certification criteria to the certification of software developers themselves! We are interested in two aspects of software certification:

- industrial, mainly process-oriented certification procedures
- software certificates that convey semantic information about a program

Semantic analysis plays a role in both varieties.

Criteria for software certification such as the Common criteria or the DOA aviation industry norms describe procedures to be followed when developing and validating a piece of software. The higher levels of the Common Criteria require a semi-formal model of the software that can be refined into executable code by traceable refinement steps. The validation of the final product is done through testing, respecting criteria of coverage that must be justified with respect to the model. The use of static analysis and proofs has so far been restricted to the top level 7 of the CC and has not been integrated into the aviation norms.

3.2.1. Process-oriented software certification

The testing requirements present in existing certification procedures pose a challenge in terms of the automation of the test data generation process for satisfying functional and structural testing requirements. For example, the standard document which currently governs the development and verification process of software in airborne system (DO-178B) requires the coverage of all the statements, all the decisions of the program at its higher levels of criticality and it is well-known that DO-178B structural coverage is a primary cost driver on avionics project. Although they are widely used, existing marketed testing tools are currently restricted to test

coverage monitoring and measurements¹ but none of these tools tries to find the test data that can execute a given statement, branch or path in the source code. In most industrial projects, the generation of structural test data is still performed manually and finding automatic methods for this problem remains a challenge for the test community. Building automatic test case generation methods requires the development of precise semantic analysis which have to scale up to software that contains thousands of lines of code.

Static analysis tools are so far not a part of the approved certification procedures. For this to change, the analysers themselves must be accepted by the certification bodies in a process called “Qualification of the tools” in which the tools are shown to be as robust as the software it will certify. We believe that proof assistants have a role to play in building such certified static analysis as we have already shown by extracting provably correct analysers for Java byte code.

3.2.2. Semantic software certificates

The particular branch of information security called “language-based security” is concerned with the study of programming language features for ensuring the security of software. Programming languages such as Java offer a variety of language constructs for securing an application. Verifying that these constructs have been used properly to ensure a given security property is a challenge for program analysis. One such problem is confidentiality of the private data manipulated by a program and a large group of researchers have addressed the problem of tracking information flow in a program in order to ensure that *e.g.*, a credit card number does not end up being accessible to all applications running on a computer [78], [45]. Another kind of problems concern the way that computational resources are being accessed and used, in order to ensure that a given access policy is being implemented correctly and that a given application does not consume more resources than it has been allocated. Members of the Celtique team have proposed a verification technique that can check the proper use of resources of Java applications running on mobile telephones [47]. **Semantic software certificates** have been proposed as a means of dealing with the security problems caused by mobile code that is downloaded from foreign sites of varying trustworthiness and which can cause damage to the receiving host, either deliberately or inadvertently. These certificates should contain enough information about the behaviour of the downloaded code to allow the code consumer to decide whether it adheres to a given security policy.

Proof-Carrying Code (PCC) [69] is a technique to download mobile code on a host machine while ensuring that the code adheres to a specified security policy. The key idea is that the code producer sends the code along with a proof (in a suitably chosen logic) that the code is secure. Upon reception of the code and before executing it, the consumer submits the proof to a proof checker for the logic. Our project focus on two components of the PCC architecture: the proof checker and the proof generator.

In the basic PCC architecture, the only components that have to be trusted are the program logic, the proof checker of the logic, and the formalization of the security property in this logic. Neither the mobile code nor the proposed proof—and even less the tool that generated the proof—need be trusted.

In practice, the *proof checker* is a complex tool which relies on a complex Verification Condition Generator (VCG). VCGs for real programming languages and security policies are large and non-trivial programs. For example, the VCG of the Touchstone verifier represents several thousand lines of C code, and the authors observed that “there were errors in that code that escaped the thorough testing of the infrastructure” [70]. Many solutions have been proposed to reduce the size of the trusted computing base. In the *foundational proof carrying code* of Appel and Felty [40], [39], the code producer gives a direct proof that, in some “foundational” higher-order logic, the code respects a given security policy. Wildmoser and Nipkow [83], [82]. prove the soundness of a *weakest precondition* calculus for a reasonable subset of the Java bytecode. Necula and Schneck [70] extend a small trusted core VCG and describe the protocol that the untrusted verifier must follow in interactions with the trusted infrastructure.

One of the most prominent examples of software certificates and proof-carrying code is given by the Java byte code verifier based on *stack maps*. Originally proposed under the term “lightweight Byte Code Verification”

¹Coverage monitoring answers to the question: what are the statements or branches covered by the test suite ? While coverage measurements answers to: how many statements or branches have been covered ?

by Rose [77], the techniques consists in providing enough typing information (the stack maps) to enable the byte code verifier to check a byte code in one linear scan, as opposed to inferring the type information by an iterative data flow analysis. The Java Specification Request 202 provides a formalization of how such a verification can be carried out.

Inspired by this, Albert *et al.* [38] have proposed to use static analysis (in the form of abstract interpretation) as a general tool in the setting of mobile code security for building a proof-carrying code architecture. In their *abstraction-carrying code* framework, a program comes equipped with a machine-verifiable certificate that proves to the code consumer that the downloaded code is well-behaved.

3.2.3. Certified static analysis

In spite of the nice mathematical theory of program analysis (notably abstract interpretation) and the solid algorithmic techniques available one problematic issue persists, *viz.*, the *gap* between the analysis that is proved correct on paper and the analyser that actually runs on the machine. While this gap might be small for toy languages, it becomes important when it comes to real-life languages for which the implementation and maintenance of program analysis tools become a software engineering task.

A *certified static analysis* is an analysis whose implementation has been formally proved correct using a proof assistant. Such analysis can be developed in a proof assistant like Coq [36] by programming the analyser inside the assistant and formally proving its correctness. The Coq extraction mechanism then allows for extracting a Caml implementation of the analyser. The feasibility of this approach has been demonstrated in [5].

We also develop this technique through certified reachability analysis over term rewriting systems. Term rewriting systems are a very general, simple and convenient formal model for a large variety of computing systems. For instance, it is a very simple way to describe deduction systems, functions, parallel processes or state transition systems where rewriting models respectively deduction, evaluation, progression or transitions. Furthermore rewriting can model every combination of them (for instance two parallel processes running functional programs).

Depending on the computing system modelled using rewriting, reachability (and unreachability) permits to achieve some verifications on the system: respectively prove that a deduction is feasible, prove that a function call evaluates to a particular value, show that a process configuration may occur, or that a state is reachable from the initial state. As a consequence, reachability analysis has several applications in equational proofs used in the theorem provers or in the proof assistants as well as in verification where term rewriting systems can be used to model programs.

For proving unreachability, *i.e.* safety properties, we already have some results based on the over-approximation of the set of reachable terms [56], [57]. We defined a simple and efficient algorithm [53] for computing exactly the set of reachable terms, when it is regular, and construct an over-approximation otherwise. This algorithm consists of a *completion* of a *tree automaton*, taking advantage of the ability of tree automata to finitely represent infinite sets of reachable terms.

To certify the corresponding analysis, we have defined a checker guaranteeing that a tree automaton is a valid fixpoint of the completion algorithm. This consists in showing that for all term recognised by a tree automaton all his rewrites are also recognised by the same tree automaton. This checker has been formally defined in Coq and an efficient Ocaml implementation has been automatically extracted [3]. This checker is now used to certify all analysis results produced by the regular completion tool as well as the optimised version of [44].

ESPRESSO Project-Team

3. Scientific Foundations

3.1. Introduction

Embedded systems are not new, but their pervasive introduction in ordinary-life objects (cars, telephone, home appliances) brought a new focus onto design methods for such systems. New development techniques are needed to meet the challenges of productivity in a competitive environment. Synchronous languages rely on the *synchronous hypothesis*, which lets computations and behaviors be divided into a discrete sequence of *computation steps* which are equivalently called *reactions* or *execution instants*. In itself this assumption is rather common in practical embedded system design.

But the synchronous hypothesis adds to this the fact that, *inside each instant*, the behavioral propagation is well-behaved (causal), so that the status of every signal or variable is established and defined prior to being tested or used. This criterion, which may be seen at first as an isolated technical requirement, is in fact the key point of the approach. It ensures strong semantic soundness by allowing universally recognized mathematical models to be used as supporting foundations. In turn, these models give access to a large corpus of efficient optimization, compilation, and formal verification techniques. The synchronous hypothesis also guarantees full equivalence between various levels of representation, thereby avoiding altogether the pitfalls of non-synthesizability of other similar formalisms. In that sense the synchronous hypothesis is, in our view, a major contribution to the goal of *model-based design* of embedded systems.

We shall describe the synchronous hypothesis and its mathematical background, together with a range of design techniques empowered by the approach. Declarative formalisms implementing the synchronous hypothesis can be cast into a model of computation [8] consisting of a domain of traces or behaviors and of semi-lattice structure that renders the synchronous hypothesis using a timing equivalence relation: clock equivalence. Asynchrony [32] can be superimposed on this model by considering a flow equivalence relation as well as heterogeneous systems [33] by parameterizing composition with arbitrary timing relations.

3.2. Polychronous model of computation

We consider a partially-ordered set of tags t to denote instants seen as symbolic periods in time during which a reaction takes place. The relation $t_1 \leq t_2$ says that t_1 occurs before t_2 . Its minimum is noted 0. A totally ordered set of tags C is called a *chain* and denotes the sampling of a possibly continuous or dense signal over a countable series of causally related tags. Events, signals, behaviors and processes are defined as follows:

- an *evente* is a pair consisting of a value v and a tag t ,
- a *signals* is a function from a *chain* of tags to a set of values,
- a *behaviorb* is a function from a set of names x to signals,
- a *processp* is a set of behaviors that have the same domain.

In the remainder, we write $\text{tags}(s)$ for the tags of a signal s , $\text{vars}(b)$ for the domain of b , $b|_X$ for the projection of a behavior b on a set of names X and b/\bar{X} for its complementary.

Figure 1 depicts a behavior b over three signals named x , y and z . Two frames depict timing domains formalized by chains of tags. Signals x and y belong to the same timing domain: x is a down-sampling of y . Its events are synchronous to odd occurrences of events along y and share the same tags, e.g. t_1 . Even tags of y , e.g. t_2 , are ordered along its chain, e.g. $t_1 < t_2$, but absent from x . Signal z belongs to a different timing domain. Its tags are not ordered with respect to the chain of y .

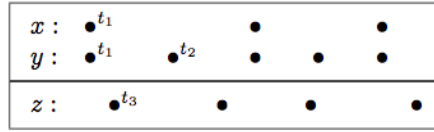


Figure 1. Behavior b over three signals x , y and z in two clock domains

3.2.1. Composition

Synchronous composition is noted $p | q$ and defined by the union $b \cup c$ of all behaviors b (from p) and c (from q) which hold the same values at the same tags $b|_I = c|_I$ for all signal $x \in I = \text{vars}(b) \cap \text{vars}(c)$ they share. Figure 2 depicts the synchronous composition (Figure 2, right) of the behaviors b (Figure 2, left) and the behavior c (Figure 2, middle). The signal y , shared by b and c , carries the same tags and the same values in both b and c . Hence, $b \cup c$ defines the synchronous composition of b and c .

$$\left(\begin{array}{c} x : \bullet^{t_1} \quad \bullet \\ y : \bullet^{t_1} \quad \bullet^{t_2} \quad \bullet \end{array} \right) | \left(\begin{array}{c} y : \bullet^{t_1} \quad \bullet^{t_2} \quad \bullet \\ z : \bullet^{t_3} \quad \bullet \end{array} \right) = \left(\begin{array}{c} x : \bullet^{t_1} \quad \bullet \\ y : \bullet^{t_1} \quad \bullet^{t_2} \quad \bullet \\ z : \bullet^{t_3} \quad \bullet \end{array} \right)$$

Figure 2. Synchronous composition of $b \in p$ and $c \in q$

3.2.2. Scheduling

A scheduling structure is defined to schedule the occurrence of events along signals during an instant t . A scheduling \rightarrow is a pre-order relation between dates x_t where t represents the time and x the location of the event. Figure 3 depicts such a relation superimposed to the signals x and y of Figure 1. The relation $y_{t_1} \rightarrow x_{t_1}$, for instance, requires y to be calculated before x at the instant t_1 . Naturally, scheduling is contained in time: if $t < t'$ then $x_t \rightarrow^b x_{t'}$ for any x and b and if $x_t \rightarrow^b x_{t'}$ then $t' \rightarrow < t$.

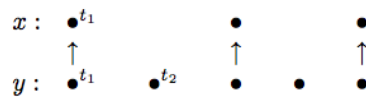


Figure 3. Scheduling relations between simultaneous events

3.2.3. Structure

A synchronous structure is defined by a semi-lattice structure to denote behaviors that have the same timing structure. The intuition behind this relation is depicted in Figure 4. It is to consider a signal as an elastic with

ordered marks on it (tags). If the elastic is stretched, marks remain in the same relative (partial) order but have more space (time) between each other. The same holds for a set of elastics: a behavior. If elastics are equally stretched, the order between marks is unchanged.

In Figure 4, the time scale of x and y changes but the partial timing and scheduling relations are preserved. Stretching is a partial-order relation which defines clock equivalence. Formally, a behavior c is a *stretching* of b of same domain, written $b \leq c$, iff there exists an increasing bijection on tags f that preserves the timing and scheduling relations. If so, c is the image of b by f . Last, the behaviors b and c are said *clock-equivalent*, written $b \sim c$, iff there exists a behavior d s.t. $d \leq b$ and $d \leq c$.

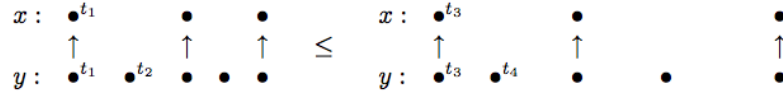


Figure 4. Relating synchronous behaviors by stretching.

3.3. A declarative design language

Signal [34], [48], [49], [41] is a declarative design language expressed within the polychronous model of computation. In Signal, a process P is an infinite loop that consists of the synchronous composition $P | Q$ of simultaneous equations $x := y f z$ over signals named x, y, z . The restriction of a signal name x to a process P is noted P/x .

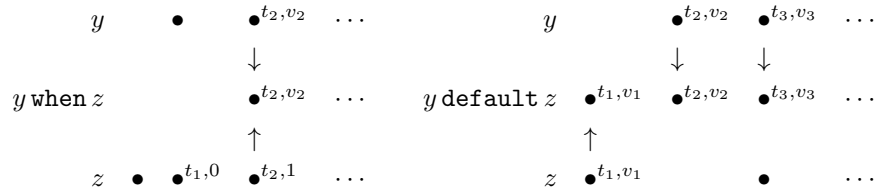
$$P, Q ::= x := y f z \mid P/x \mid P | Q$$

Equations $x := y f z$ in Signal more generally denote processes that define timing relations between input and output signals. There are four primitive combinators in Signal:

- delay $x := y \$ \text{init } v$, initially defines the signal x by the value v and then by the previous value of the signal y . The signal y and its delayed copy $x = y \$ \text{init } v$ are synchronous: they share the same set of tags t_1, t_2, \dots . Initially, at t_1 , the signal x takes the declared value v and then, at tag t_n , the value of y at tag t_{n-1} .

$$\begin{array}{cccc} y & \bullet^{t_1, v_1} & \bullet^{t_2, v_2} & \bullet^{t_3, v_3} \dots \\ y \$ \text{init } v & \bullet^{t_1, v} & \bullet^{t_2, v_1} & \bullet^{t_3, v_2} \dots \end{array}$$

- sampling $x := y \text{ when } z$, defines x by y when z is true (and both y and z are present); x is present with the value v_2 at t_2 only if y is present with v_2 at t_2 and if z is present at t_2 with the value true. When this is the case, one needs to schedule the calculation of y and z before x , as depicted by $y_{t_2} \rightarrow x_{t_2} \leftarrow z_{t_2}$.
- merge $x := y \text{ default } z$, defines x by y when y is present and by z otherwise. If y is absent and z present with v_1 at t_1 then x holds (t_1, v_1) . If y is present (at t_2 or t_3) then x holds its value whether z is present (at t_2) or not (at t_3).



The structuring element of a Signal specification is a process. A process accepts input signals originating from possibly different clock domains to produce output signals when needed. This allows, for instance, to specify a counter where the inputs `tick` and `reset` and the output value have independent clocks. The body of counter consists of one equation that defines the output signal value. Upon the event `reset`, it sets the count to 0. Otherwise, upon a `tick` event, it increments the count by referring to the previous value of `value` and adding 1 to it. Otherwise, if the count is solicited in the context of the counter process (meaning that its clock is active), the counter just returns the previous count without having to obtain a value from the `tick` and `reset` signals.

```
process counter = (? event tick, reset; ! integer value;)
  (| value := (0 when reset)
   default ((value$ init 0 + 1) when tick)
   default (value$ init 0)
  |);
```

A Signal process is a structuring element akin to a hierarchical block diagram. A process may structurally contain sub-processes. A process is a generic structuring element that can be specialized to the timing context of its call. For instance, the definition of a synchronized counter starting from the previous specification consists of its refinement with synchronization. The input `tick` and `reset` clocks expected by the process counter are sampled from the boolean input signals `tick` and `reset` by using the `when tick` and `when reset` expressions. The count is then synchronized to the inputs by the equation `reset ^= tick ^= value`.

```
process synccounter = (? boolean tick, reset; ! integer value;)
  (| value := counter (when tick, when reset)
   | reset ^= tick ^= value
  |);
```

3.4. Compilation of Signal

Sequential code generation starting from a Signal specification starts with an analysis of its implicit synchronization and scheduling relations. This analysis yields the control and data-flow graphs that define the class of sequentially executable specifications and allow to generate code.

3.4.1. Synchronization and scheduling specifications

In Signal, the clock \widehat{x} of a signal x denotes the set of instants at which the signal x is present. It is represented by a signal that is true when x is present and that is absent otherwise. Clock expressions represent control. The clock `when x` (resp. `when not x`) represents the time tags at which a boolean signal x is present and true (resp. false).

The empty clock is written $\widehat{0}$ and clock expressions e combined using conjunction, disjunction and symmetric difference. Clock equations E are Signal processes: the equation $e\widehat{=}e'$ synchronizes the clocks e and e' while $e\widehat{<}e'$ specifies the containment of e in e' . Explicit scheduling relations $x \rightarrow y$ when e allow to schedule the calculation of signals (e.g. x after y at the clock e).

$$\begin{aligned}
e & ::= \hat{x} \mid \text{when } x \mid \text{not } x \mid e^{\wedge} + e' \mid e^{\wedge} - e' \mid e^{\wedge} * e' \mid \hat{0} && \text{(clock expression)} \\
E & ::= () \mid e^{\wedge} = e' \mid e^{\wedge} < e' \mid x \rightarrow y \text{ when } e \mid E \mid E' \mid E/x && \text{(clock relations)}
\end{aligned}$$

3.4.2. Synchronization and scheduling analysis

A Signal process P corresponds to a system of clock and scheduling relations E that denotes its timing structure. It can be defined by induction on the structure of P using the inference system $P : E$ of Figure 5 .

$$\begin{aligned}
x := y \$ \text{init } v & : \hat{x} \hat{=} \hat{v} \\
x := y \text{ when } z & : \hat{x} \hat{=} \hat{y} \text{ when } z \mid y \rightarrow x \text{ when } z \\
x := y \text{ default } z & : \hat{x} \hat{=} \hat{y} \text{ default } \hat{z} \mid y \rightarrow x \text{ when } \hat{y} \mid z \rightarrow x \text{ when } \hat{z} \hat{-} \hat{y}
\end{aligned}$$

Figure 5. Clock inference system

3.4.3. Hierarchization

The clock and scheduling relations E of a process P define the control-flow and data-flow graphs that hold all necessary information to compile a Signal specification upon satisfaction of the property of *endochrony*. A process is said endochronous iff, given a set of input signals and flow-equivalent input behaviors, it has the capability to reconstruct a unique synchronous behavior up to clock-equivalence: the input and output signals are ordered in clock-equivalent ways.

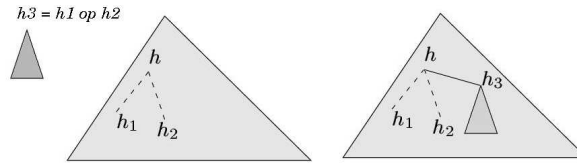


Figure 6. Hierarchization of clocks

To determine the order $x \preceq y$ in which signals are processed during the period of a reaction, clock relations E play an essential role. The process of determining this order is called hierarchization and consists of an insertion algorithm which hooks elementary control flow graphs (in the form of if-then-else structures) one to the others. Figure 6 , right, let $h3$ be a clock computed using $h1$ and $h2$. Let h be the head of a tree from which $h1$ and $h2$ are computed (an if-then-else), $h3$ is computed after $h1$ and $h2$ and placed under h [28].

S4 Project-Team

3. Scientific Foundations

3.1. Scientific Foundations

The research work of the team is built on top of solid foundations, mainly, algebraic, combinatorial or logical theories of transition systems. These theories cover several sorts of systems which have been studied during the last thirty years: sequential, concurrent, synchronous or asynchronous. They aim at modeling the behavior of finite or infinite systems (usually by abstracting computations on data), with a particular focus on the control flow which rules state changes in these systems. Systems can be autonomous or reactive, that is, embedded in an environment with which the system interacts, both receiving an input flow, and emitting an output flow of events and data. System specifications can be explicit (for instance, when the system is specified by an automaton, extensively defined by a set of states and a set of transitions), or implicit (symbolic transition rules, usually parameterized by state or control variables; partially-synchronized products of finite transition systems; Petri nets; systems of equations constraining the transitions of synchronous reactive systems, according to their input flows; etc.). Specifications can be non-ambiguous, meaning that they fully define at most one system (this holds in the previous cases), or they can be ambiguous, in which case more than one system is conforming to the specification (for instance, when the system is described by logical formulas in the modal mu-calculus, or when the system is described by a set of scenario diagrams, such as *Sequence Diagrams* or *Message Sequence Charts*).

Systems can be described in two ways: either the state structure is described, or only the behavior is described. Both descriptions are often possible (this is the case for formal languages, automata, products of automata, or Petri nets), and moving from one representation to the other is achieved by folding/unfolding operations.

Another taxonomy criteria is the concurrency these models can encompass. Automata usually describe sequential systems. Concurrency in synchronous systems is usually not considered. In contrast, Petri nets or partially-synchronized products of automata are concurrent. When these models are transformed, concurrency can be either preserved, reflected or even, infused. An interesting case is whenever the target architecture requires distributing events among several processes. There, communication-efficient implementations require that concurrency is preserved as far as possible and that, at the same time, causality relations are also preserved. These notions of causality and independence are best studied in models such as concurrent automata, Petri nets or Mazurkiewicz trace languages.

Here are our sources of inspiration regarding formal mathematical tools:

1. Jan van Leeuwen (ed.), *Handbook of Theoretical Computer Science - Volume B: Formal Models and Semantics*, Elsevier, 1990.
2. Jörg Desel, Wolfgang Reisig and Grzegorz Rozenberg (eds.), *Lectures on Concurrency and Petri nets*, Lecture Notes in Computer Science, Vol. 3098, Springer, 2004.
3. Volker Diekert and Grzegorz Rozenberg (eds.), *The Book of Traces*, World Scientific, 1995.
4. André Arnold and Damian Niwinski, *Rudiments of Mu-Calculus*, North-Holland, 2001.
5. Gérard Berry, *Synchronous languages for hardware and software reactive systems - Hardware Description Languages and their Applications*, Chapman and Hall, 1997.

Our research exploits decidability or undecidability results on these models (for instance, inclusion of regular languages, bisimilarity on automata, reachability on Petri nets, validity of a formula in the mu-calculus, etc.) and also, representation theorems which provide effective translations from one model to another. For instance, Zielonka's theorem yields an algorithm which maps regular trace languages to partially-synchronized products of finite automata. Another example is the theory of regions, which provides methods for mapping finite or infinite automata, languages, or even *High-Level Message Sequence Charts* to Petri nets. A further example concerns the mu-calculus, in which algorithms computing winning strategies for parity games can be used to synthesize supervisory control of discrete event systems.

Our research aims at providing effective representation theorems, with a particular emphasis on algorithms and tools which, given an instance of one model, synthesize an instance of another model. In particular we have contributed a theory, several algorithms and a tool for synthesizing Petri nets from finite or infinite automata, regular languages, or languages of *High-Level Message Sequence Charts*. This also applies to our work on supervisory control of discrete event systems. In this framework, the problem is to compute a system (the controller) such that its partially-synchronized product with a given system (the plant) satisfies a given behavioral property (control objective, such as a regular language or satisfaction of a mu-calculus formula).

Software engineers often face problems similar to *service adaptation* or *component interfacing*, which in turn, often reduce to particular instances of system synthesis or supervisory control problems.

TASC Project-Team

3. Scientific Foundations

3.1. Overview

Basic research is guided by the challenges raised before: to classify and enrich the models, to automate reformulation and resolution, to dissociate declarative and procedural knowledge, to come up with theories and tools that can handle problems involving both continuous and discrete variables, to develop modelling tools and to come up with solving tools that scale well. On the one hand, **classification aspects** of this research are integrated within a knowledge base about combinatorial problem solving: the global constraint catalog (see <http://www.emn.fr/x-info/sdemasse/gccat/index.html>). On the other hand, **solving aspects** are capitalized within the constraint solving system **CHOCO**. Lastly, within the framework of its activities of valorisation, teaching and of partnership research, the team uses constraint programming for solving various concrete problems. The challenge is, on one side to increase the visibility of the constraints in the others disciplines of computer science, and on the other side to contribute to a broader diffusion of the constraint programming in the industry.

3.2. Fundamental Research Topics

This part presents the research topics investigated by the project:

- Global Constraints Classification, Reformulation and Filtering,
- Convergence between Discrete and Continuous,
- Dynamic, Interactive and over Constrained Problems,
- Solvers.

These research topics are in fact not independent. The work of the team thus frequently relates transverse aspects such as explained global constraints, Benders decomposition and explanations, flexible and dynamic constraints, linear models and relaxations of constraints.

3.2.1. *Constraints Classification, Reformulation and Filtering*

In this context our research is focused (a) first on identifying recurring combinatorial structures that can be used for modelling a large variety of optimization problems, and (b) exploit these combinatorial structures in order to come up with efficient algorithms in the different fields of optimization technology. The key idea for achieving point (b) is that many filtering algorithms both in the context of Constraint Programming, Mathematical Programming and Local Search can be interpreted as the maintenance of invariants on specific domains (e.g., graph, geometry). The systematic classification of global constraints and of their relaxation brings a synthetic view of the field. It establishes links between the properties of the concepts used to describe constraints and the properties of the constraints themselves. Together with **SICS**, the team develops and maintains *a catalog of global constraints*, which describes the semantics of more than 350 constraints, and proposes a unified mathematical model for expressing them. This model is based on graphs, automata and logic formulae and allows to derive filtering methods and automatic reformulation for each constraint in a unified way (see <http://www.emn.fr/x-info/sdemasse/gccat/index.html>). We consider hybrid methods (i.e., methods that involve more than one optimization technology such as constraint programming, mathematical programming or local search), to draw benefit from the respective advantages of the combined approaches. More fundamentally, the study of hybrid methods makes it possible to compare and connect strategies of resolution specific to each approach for then conceiving new strategies. Beside the works on classical, complete resolution techniques, we also investigate local search techniques from a mathematical point of view. These partly random algorithms have been proven very efficient in practice, although we have little theoretical knowledge on their behaviour, which often makes them problem-specific. Our research in that area is focused on a probabilistic model of

local search techniques, from which we want to derive quantified information on their behaviour, in order to use this information directly when designing the algorithms and exploit their performances better. We also consider algorithms that maintain local and global consistencies, for more specific models. Having in mind the trade off between genericity and effectiveness, the effort is put on the efficiency of the algorithms with guarantee on the produced levels of filtering. This effort results in adapting existing techniques of resolution such as graph algorithms. For this purpose we identify necessary conditions of feasibility that can be evaluated by efficient incremental algorithms. Genericity is not neglected in these approaches: on the one hand the constraints we focus on are applicable in many contexts (for example, graph partitioning constraints can be used both in logistics and in phylogeny); on the other hand, this work led to study the portability of such constraints and their independence with specific solvers. This research orientation gathers various work such as strong local consistencies, graph partitioning constraints, geometrical constraints, and optimization and soft constraints. Within the perspective to deal with complex industrial problems, we currently develop meta constraints (e.g. *geost*) handling all together the issues of large-scale problems, dynamic constraints, combination of spatial and temporal dimensions, expression of business rules.

3.2.2. Convergence between Discrete and Continuous

Many industrial problems mix continuous and discrete aspects that respectively correspond to physical (e.g., the position, the speed of an object) and logical (e.g., the identifier, the nature of an object) elements. Typical examples of problems are for instance:

- *Geometrical placement problems* where one has to place in space a set of objects subject to various geometrical constraints (i.e., non-overlapping, distance). In this context, even if the positions of the objects are continuous, the structure of optimal configurations has a discrete nature.
- *Trajectory and mission planning problems* where one has to plan and synchronize the moves of several teams in order to achieve some common goal (i.e., fire fighting, coordination of search in the context of rescue missions, surveillance missions of restricted or large areas).
- *Localization problems in mobile robotic* where a robot has to plan alone (only with its own sensors) its trajectory. This kind of problematic occurs in situations where the GPS cannot be used (e.g., under water or Mars exploration) or when it is not precise enough (e.g., indoor surveillance, observation of contaminated sites).

Beside numerical constraints that mix continuous and integer variables we also have global constraints that involve both type of variables. They typically correspond to graph problems (i.e., graph colouring, domination in a graph) where a graph is dynamically constructed with respect to geometrical and-or temporal constraints. In this context, the key challenge is avoiding decomposing the problem in a discrete and continuous parts as it is traditionally the case. As an illustrative example consider *the wireless network deployment problem*. On the one hand, the continuous part consists of finding out where to place a set of antenna subject to various geometrical constraints. On the other hand, by building an interference graph from the positions of the antenna, the discrete part consists of allocating frequencies to antenna in order to avoid interference. In the context of convergence between discrete and continuous variables, our goals are:

- First to identify and compare typical class of techniques that are used in the context of continuous and discrete solvers.
- To see how one can unify and/or generalize these techniques in order to handle in an integrated way continuous and discrete constraints within the same framework.

3.2.3. Dynamic, Interactive and over Constrained Problems

Some industrial applications are defined by a set of constraints which may change over time, for instance due to an interaction with the user. Many other industrial applications are over-constrained, that is, they are defined by set of constraints which are more or less important and cannot be all satisfied at the same time. Generic, dedicated and explanation-based techniques can be used to deal efficiently with such applications. Especially, these applications rely on the notion of *soft constraints* that are allowed to be (partially) violated. The generic concept that captures a wide variety of soft constraints is the violation measure, which is coupled with specific resolution techniques. Lastly, soft constraints allow to combine the expressive power of global constraints with local search frameworks.

3.2.4. Solvers

Our theoretical work is systematically validated by concrete experimentations. We have in particular for that purpose the **CHOCO** constraint platform. The team develops and maintains **CHOCO** with the assistance of the laboratory e-lab of Bouygues (G. Rochart), the company Amadeus (F. Laburthe), and others researchers such as **H. Cambazard** (4C, INP Grenoble). The functionalities of **CHOCO** are gradually extended with the outcomes of our works: design of constraints, analysis and visualization of explanations, etc. The open source **CHOCO** library is downloaded on average 450 times each month since 2006. **CHOCO** is developed in line with the research direction of the team, in an open-minded scientific spirit. Contrarily to other solvers where the efficiency often relies on problem-specific algorithms, **CHOCO** aims at providing the users both with reusable techniques (based on an up-to-date implementation of the global constraint catalogue) and with a variety of tools to ease the use of these techniques (clear separation between model and resolution, event-based solver, management of the over-constrained problems, explanations, etc.). Since 2009 year, due to the hiring of **G. Chabert**, the team is also involved in the development of the continuous constraint solver **IBEX**. These developments led us to new research topics, suitable for the implementation of discrete and continuous constraint solving systems: portability of the constraints, management of explanations, incrementality and recalculation. They partially use aspect programming (in collaboration with the **InriaASCOLA** team). This work around the design and the development of solvers thus forms the fourth direction of basic research of the project.

VERTECS Project-Team**3. Scientific Foundations****3.1. Underlying models**

The formal models we use are mainly automata-like structures such as labelled transition systems (LTS) and some of their extensions: an LTS is a tuple $M = (Q, \Lambda, \rightarrow, q_o)$ where Q is a non-empty set of states; $q_o \in Q$ is the initial state; A is the alphabet of actions, $\rightarrow \subseteq Q \times \Lambda \times Q$ is the transition relation. These models are adapted for testing and controller synthesis.

To model reactive systems in the testing context, we use Input/Output labeled transition systems (IOLTS for short). In this setting, the interactions between the system and its environment (where the tester lies) must be partitioned into inputs (controlled by the environment), outputs (observed by the environment), and internal (non observable) events modeling the internal behavior of the system. The alphabet Λ is then partitioned into $\Lambda_I \cup \Lambda_O \cup \mathcal{I}$ where Λ_I is the alphabet of outputs, Λ_O the alphabet of inputs, and \mathcal{I} the alphabet of internal actions.

In the controller synthesis theory, we also distinguish between controllable and uncontrollable events ($\Lambda = \Lambda_c \cup \Lambda_{uc}$), observable and unobservable events ($\Lambda = \Lambda_O \cup \mathcal{I}$).

In the context of verification, we also use Timed Automata. A timed automaton is a tuple $A = (L, X, E, \mathcal{J})$ where L is a set of locations, X is a set of clocks whose valuations are positive real numbers, $E \subseteq L \times \mathcal{G}(X) \times 2^X \times L$ is a finite set of edges composed of a source and a target state, a guard given by a finite conjunction of expressions of the form $x \sim c$ where x is a clock, c is a natural number and $\sim \in \{<, \leq, =, \geq, >\}$, a set of resetting clocks, and $\mathcal{J} : L \rightarrow \mathcal{G}(X)$ assigns an invariant to each location [22]. The semantics of a timed automaton is given by a (infinite states) labelled transition system whose states are composed of a location and a valuation of clocks.

Also, for verification purposes, we use graph grammars that are a general tool to define families of graphs. Such grammars are formed by a set of rules whose left-hand sides are hyperedges and right-hand sides are hypergraphs. For graphs with finite degree, these grammars characterise transition graphs of pushdown automata (the correspondence between graphs generated by grammars and transition graphs of pushdown automata is bijective). Graph grammars provide a simple yet powerful setting to define and study infinite state systems.

In order to cope with models closer to practical specification languages, we also need higher level models encompassing both control and data aspects. We defined (input-output) symbolic transition systems ((IO)STS), which are extensions of (IO)LTS that convey or operate on data (i.e., program variables, communication parameters, symbolic constants) through message passing, guards, and assignments. Formally, an IOSTS is a tuple (V, Θ, Σ, T) , where V is a set of variables (including a counter variable encoding the control structure), Θ is the initial condition defined by a predicate on V , Σ is the finite alphabet of actions, where each action has a signature (just like in IOLTS, Σ can be partitioned as e.g. $\Sigma_O \cup \Sigma_I \cup \Sigma_{\tau}$), T is a finite set of symbolic transitions of the form $t = (a, p, G, A)$ where a is an action (possibly with a polarity reflecting its input/output/internal nature), p is a tuple of communication parameters, G is a guard defined by a predicate on p and V , and A is an assignment of variables. The semantics of IOSTS is defined in terms of (IO)LTS where states are vectors of values of variables, and transitions between them are labelled with instantiated actions (action with valued communication parameter). This (IO)LTS semantics allows us to perform syntactical transformations at the (IO)STS level while ensuring semantical properties at the (IO)LTS level. We also consider extensions of these models with added features such as recursion, fifo channels, etc. An alternative to IOSTS for specifying systems with data variables is the model of synchronous dataflow equations.

Our research is based on well established theories: conformance testing, supervisory control, abstract interpretation, and theorem proving. Most of the algorithms that we employ take their origins in these theories:

- graph traversal algorithms (breadth first, depth first, strongly connected components, ...). We use these algorithms for verification as well as test generation and control synthesis.
- BDDs (Binary Decision Diagrams) algorithms, for manipulating Boolean formulae, and their MTB-DDs (Multi-Terminal Decision Diagrams) extension for manipulating more general functions. We use these algorithms for verification, test generation and control.
- abstract interpretation algorithms, specifically in the abstract domain of convex polyhedra (for example, Chernikova's algorithm for the computation of dual forms). Such algorithms are used in verification and test generation.
- logical decision algorithms, such as satisfiability of formulas in Presburger arithmetics. We use these algorithms during generation and execution of symbolic test cases.

3.2. Verification

Verification in its full generality consists in checking that a system, which is specified by a formal model, satisfies a required property. Verification takes place in our research in two ways: on the one hand, a large part of our work, and in particular controller synthesis and conformance testing, relies on the ability to solve some verification problems. Many of these problems reduce to reachability and coreachability questions on a formal model (a state s is *reachable from an initial state* s_i if an execution starting from s_i can lead to s ; s is *coreachable from a final state* s_f if an execution starting from s can lead to s_f). These are important cases of verification problems, as they correspond to the verification of safety properties.

On the other hand we investigate verification on its own in the context of complex systems. For expressivity purposes, it is necessary to be able to describe faithfully and to deal with complex systems. Some particular aspects require the use of infinite state models. For example asynchronous communications with unknown transfer delay (and thus arbitrary large number of messages in transit) are correctly modeled by unbounded FIFO queues, and real time systems require the use of continuous variables which evolve with time. Apart from these aspects requiring infinite state data structure, systems often include uncertain or random behaviours (such as failures, actions from the environment), which it make sense to model through probabilities. To encompass these aspects, we are interested in the verification of systems equipped with infinite data structures and/or probabilistic features.

When the state space of the system is infinite, or when we try to evaluate performances, standard model-checking techniques (essentially graph algorithms) are not sufficient. For large or infinite state spaces, symbolic model-checking or approximation techniques are used. Symbolic verification is based on efficient representations of sets of states and permits exact model-checking of some well-formed infinite-state systems. However, for feasibility reasons, it is often mandatory to use approximate computations, either by computing a finite abstraction and resort to graph algorithms, or preferably by using more sophisticated abstract interpretation techniques. For systems with stochastic aspects, a quantitative analysis has to be performed, in order to evaluate the performances. Here again, either symbolic techniques (e.g. by grouping states with similar behaviour) or approximation techniques should be used.

We detail below verification topics we are interested in: abstract interpretation, quantitative model-checking and analysis of systems defined by graph grammars.

3.2.1. Abstract interpretation and data handling

Most problems in test generation or controller synthesis reduce to state reachability and state coreachability problems which can be solved by fixpoint computations of the form $x = F(x)$, $x \in C$ where C is a lattice. In the case of reachability analysis, if we denote by S the state space of the considered program, C is the lattice $\wp(S)$ of sets of states, ordered by inclusion, and F is roughly the "successor states" function defined by the program.

The big change induced by taking into account the data and not only the (finite) control of the systems under study is that the fixpoints become uncomputable. The undecidability is overcome by resorting to approximations, using the theoretical framework of Abstract Interpretation [24]. The fundamental principles of Abstract Interpretation are:

1. to substitute to the *concrete domain* C a simpler *abstract domain* A (static approximation) and to transpose the fixpoint equation into the abstract domain, so that one has to solve an equation $y = G(y), y \in A$;
2. to use a *widening operator* (dynamic approximation) to make the iterative computation of the least fixpoint of G converge after a finite number of steps to some upper-approximation (more precisely, a post-fixpoint).

Approximations are conservative so that the obtained result is an upper-approximation of the exact result. In simple cases the state space that should be abstracted has a simple structure, but this may be more complicated when variables belong to different data types (Booleans, numerics, arrays) and when it is necessary to establish *relations* between the values of different types.

3.2.2. Model-checking quantitative systems

Model-checking techniques for finite-state systems are now quite developed, and a current challenge is to adapt them as much as possible to infinite-state systems. We detail below two types of models we are interested in: timed automata and infinite-state probabilistic systems.

Model-checking timed automata The model of timed automata, introduced by Alur and Dill in the 90's [22] is commonly used to represent real-time systems. Timed automata consist of an extension of finite automata with continuous variables, called clocks, that evolve synchronously with time, and can be tested and reset along an execution. Despite their uncountable state space, checking reachability, and more generally ω -regular properties, is decidable *via* the construction of a finite abstraction, the so-called region automaton. The recent developments in model-checking timed automata have aimed at modelling and verifying quantitative aspects encompassing timing constraints, for example costs, probabilities, frequencies. These quantitative questions demand advanced techniques that go far beyond the classical methods.

Model-checking infinite state probabilistic systems Model-checking techniques for finite state probabilistic systems are now quite developed. Given a finite state Markov chain, for example, one can check whether some property holds almost surely (i.e. the set of executions violating the property is negligible), and one can even compute (or at least approximate as close as wanted) the probability that some property holds. In general, these techniques cannot be adapted to infinite state probabilistic systems, just as model-checking algorithms for finite state systems do not carry over to infinite state systems. For systems exhibiting complex data structures (such as unbounded queues, continuous clocks) and uncertainty modeled by probabilities, it can thus be hard to design model-checking algorithms. However, in some cases, especially when considering qualitative verification, symbolic methods can lead to exact results. Qualitative questions aim neither at computing nor at approximating a probability, but are only concerned with almost-sure or non-negligible behaviours (that is events of probability either one or non-zero). In some cases, qualitative model-checking can be derived from a combination of techniques for infinite state systems (such as abstractions) with methods for finite state probabilistic systems. However, when one is interested in computing (or rather approximating) precise probability values (neither 0 nor 1), exact methods are scarce. To deal with these questions, we either try to restrict to classes of systems where exact computations can be made, or look for approximation algorithms.

3.2.3. Analysis of infinite state systems defined by graph grammars

Currently, many techniques (reachability, model checking, ...) from finite state systems have been generalised to pushdown systems, that can be modeled by graph grammars. Several such extensions heavily depend on the actual definition of the pushdown automata, for example, how many top stack symbols may be read, or whether the existence of ε -transitions (silent transitions) is allowed. Many of these restrictions do not affect the actual structure of the graph, and interesting properties like reachability or satisfiability (of a formula) only depend on the structure of a graph.

Deterministic graph grammars enable us to focus on structural properties of systems. The connection with finite graph algorithms is often straightforward: for example reachability is obtained by iterating the finite graph algorithm iterated on the right hand sides of the rules. On the other hand, extending these grammars with time or probabilities is not straightforward: qualitative values associated to different copies (in the graph) of the same vertex (in the grammar) may differ, introducing complex equations. Furthermore, the fact that the left-hand sides of rules are single hyperarcs is a strong restriction. But removing this restriction would lead to non-recursive graphs. Identifying decidable families of graphs defined by contextual graph grammars is also very challenging.

3.3. Automatic test generation

We are mainly interested in conformance testing, which consists in checking whether a black box implementation under test (the real system that is only known by its interface) behaves correctly with respect to its specification (the reference which specifies the intended behavior of the system). In the line of model-based testing, we use formal specifications and their underlying models to unambiguously define the intended behavior of the system, to formally define conformance and to design test case generation algorithms. The difficult problems are to generate test cases that correctly identify faults (the oracle problem) and, as exhaustiveness is impossible to reach in practice, to select an adequate subset of test cases that are likely to detect faults. Hereafter we detail some elements of the models, theories and algorithms we use.

We use IOLTS (or IOSTS) as formal models for specifications, implementations, test purposes, and test cases. We adapt a well established theory of conformance testing [30], which formally defines conformance as a relation between formal models of specifications and implementations. This conformance relation, called **ioco** compares the visible behaviors (called *suspension traces*) of the implementation I (denoted by $S\text{Traces}(I)$) with those of the specification S ($S\text{Traces}(S)$). Suspension traces are sequence of inputs, outputs or quiescence (absence of action denoted by δ), thus abstracting away internal behaviors that cannot be observed by testers. Intuitively, I *ioco* S if after a suspension trace of the specification, the implementation I can only show outputs and quiescences of the specification S . We re-formulated ioco as a partial inclusion of visible behaviors as follows:

$$I \text{ ioco } S \Leftrightarrow S\text{Traces}(I) \cap [S\text{Traces}(S).\Lambda_1^\delta \setminus S\text{Traces}(S)] = \emptyset.$$

In other words, suspension traces of I which are suspension traces of S prolonged by an output or quiescence, should still be suspension traces of S .

Interestingly, this characterization presents conformance with respect to S as a safety property of suspension traces of I . The negation of this property is characterized by a *canonical tester* $Can(S)$ which recognizes exactly $[S\text{Traces}(S).\Lambda_1^\delta \setminus S\text{Traces}(S)]$, the set of non-conformant suspension traces. This *canonical tester* also serves as a basis for test selection.

Test cases are processes executed against implementations in order to detect non-conformance. They are also formalized by IOLTS (or IOSTS) with special states indicating *verdicts*. The execution of test cases against implementations is formalized by a parallel composition with synchronization on common actions. A *Fail* verdict means that the implementation under test (IUT) is rejected and should correspond to non-conformance, a *Pass* verdict means that the IUT exhibited a correct behavior and some specific targeted behaviour has been observed, while an *Inconclusive* verdict is given to a correct behavior that is not targeted.

Test suites (sets of test cases) are required to exhibit some properties relating the verdict they produce to the conformance relation. Soundness means that only non conformant implementations should be rejected by a test suite and exhaustiveness means that every non conformant implementation may be rejected by the test suite. Soundness is not difficult to obtain, but exhaustiveness is not possible in practice and one has to select test cases.

Test selection is often based on the coverage of some criteria (state coverage, transition coverage, etc). But test cases are often associated with *test purposes* describing some abstract behaviors targeted by a test case. In our framework, test purposes are specified as IOLTS (or IOSTS) associated with marked states or dedicated variables, giving them the status of automata or observers accepting runs (or sequences of actions or suspension traces). Selection of test cases amounts to selecting traces of the canonical tester accepted by the test purpose. The resulting test case is then both an observer of the negation of a safety property (non-conformance wrt. S), and an observer of a reachability property (acceptance by the test purpose). Selection can be reduced to a model-checking problem where one wants to identify states (and transitions between them) which are both reachable from the initial state and co-reachable from the accepting states. We have proved that these algorithms ensure soundness. Moreover the (infinite) set of all possibly generated test cases is also exhaustive. Apart from these theoretical results, our algorithms are designed to be as efficient as possible in order to be able to scale up to real applications.

Our first test generation algorithms are based on enumerative techniques, thus adapted to IOLTS models, and optimized to fight the state-space explosion problem. On-the-fly algorithms were designed and implemented in the TGV tool, which consist in computing co-reachable states from a target state during a lazy exploration of the set of reachable states in a product of the specification and the test purpose [25]. However, this enumerative technique suffers from some limitations when specification models contain data.

More recently, we have explored symbolic test generation techniques for IOSTS specifications [29]. The objective is to avoid the state space explosion problem induced by the enumeration of values of variables and communication parameters. The idea consists in computing a test case under the form of an *IOSTS*, i.e., a reactive program in which the operations on data are kept in a symbolic form. Test selection is still based on test purposes (also described as IOSTS) and involves syntactical transformations of IOSTS models that should ensure properties of their IOLTS semantics. However, most of the operations involved in test generation (determinisation, reachability, and coreachability) become undecidable. For determinisation we employ heuristics that allow us to solve the so-called bounded observable non-determinism (i.e., the result of an internal choice can be detected after finitely many observable actions). The product is defined syntactically. Finally test selection is performed as a syntactical transformation of transitions which is based on a semantical reachability and co-reachability analysis. As both problems are undecidable for IOSTS, syntactical transformations are guided by over-approximations using abstract interpretation techniques. Nevertheless, these over-approximations still ensure soundness of test cases [26]. These techniques are implemented in the STG tool (see 5.1), with an interface with NBAC used for abstract interpretation.

3.4. Control synthesis

The supervisory control problem is concerned with ensuring (not only checking) that a computer-operated system works correctly. More precisely, given a system model and a required property, the problem is to control the model's behavior, by coupling it to a supervisor, such that the controlled system satisfies the property [28]. The models used are LTSs and the associated languages, where one makes a distinction between *controllable* and *non-controllable* actions and between *observable* and *non-observable* actions. Typically, the controlled system is constrained by the supervisor, which can block on the system's controllable actions in order to force it to behave as specified by the property. The control synthesis problem can be seen as a constructive verification problem: building a supervisor that prevents the system from violating a property. Several kinds of properties can be enforced such as reachability, invariance (i.e. safety), attractivity, etc. Techniques adapted from model checking are used to compute the supervisor. Optimality must be taken into account as one often wants to obtain a supervisor that constrains the system as few as possible.

Supervisory control theory overview. Supervisory control theory deals with control of Discrete Event Systems. In this theory, the behavior of the system S is assumed not to be fully satisfactory. Hence, it has to be reduced by means of a feedback control (named Supervisor or Controller) in order to achieve a given set of requirements [28]. Namely, if S denotes the model of the system and Φ a safety property to be enforced on S , the problem consists of computing a supervisor \mathcal{C} such that

$$S \parallel \mathcal{C} \models \Phi \quad (1)$$

where \parallel is the classical parallel composition of LTSs. Given S , some events of S are said to be uncontrollable (Σ_{uc}), i.e., the occurrence of these events cannot be prevented by a supervisor, while the others are controllable (Σ_c). It means that all the supervisors satisfying (1) are not good candidates. The behavior of the controlled system must respect an additional condition that happens to be similar to the *ioco* conformance relation previously defined in 3.3. This condition is called the *controllability condition* and it may be stated as

$$\mathcal{L}(S \parallel \mathcal{C})_{\Sigma_{uc}} \cap \mathcal{L}(S) \subseteq \mathcal{L}(S \parallel \mathcal{C}) \quad (2)$$

Namely, when acting on S , a supervisor is not allowed to disable uncontrollable events. Given a safety property Φ , that can be modeled by an LTS A_Φ , there actually exist many different supervisors satisfying both (1) and (2). Among all the valid supervisors, we are interested in computing the supremal one, ie the one that restricts the system as few as possible. It has been shown in [28] that such a supervisor always exists and is unique. It gives access to a behavior of the controlled system that is called the supremal controllable sub-language of A_Φ w.r.t. S and Σ_{uc} . In some situations, it may also be interesting to force the controlled system to be non-blocking (See [28] for details).

The underlying techniques are similar to the ones used for Automatic Test Generation. They consist of computing the product of the system model and A_Φ and to remove the states of the product that may lead to subsequent states violating the property by triggering only uncontrollable events.

ASPI Project-Team

3. Scientific Foundations

3.1. Interacting Monte Carlo methods and particle approximation of Feynman–Kac distributions

Monte Carlo methods are numerical methods that are widely used in situations where (i) a stochastic (usually Markovian) model is given for some underlying process, and (ii) some quantity of interest should be evaluated, that can be expressed in terms of the expected value of a functional of the process trajectory, which includes as an important special case the probability that a given event has occurred. Numerous examples can be found, e.g. in financial engineering (pricing of options and derivative securities) [50], in performance evaluation of communication networks (probability of buffer overflow), in statistics of hidden Markov models (state estimation, evaluation of contrast and score functions), etc. Very often in practice, no analytical expression is available for the quantity of interest, but it is possible to simulate trajectories of the underlying process. The idea behind Monte Carlo methods is to generate independent trajectories of this process or of an alternate instrumental process, and to build an approximation (estimator) of the quantity of interest in terms of the weighted empirical probability distribution associated with the resulting independent sample. By the law of large numbers, the above estimator converges as the size N of the sample goes to infinity, with rate $1/\sqrt{N}$ and the asymptotic variance can be estimated using an appropriate central limit theorem. To reduce the variance of the estimator, many variance reduction techniques have been proposed. Still, running independent Monte Carlo simulations can lead to very poor results, because trajectories are generated *blindly*, and only afterwards are the corresponding weights evaluated. Some of the weights can happen to be negligible, in which case the corresponding trajectories are not going to contribute to the estimator, i.e. computing power has been wasted.

A recent and major breakthrough, has been the introduction of interacting Monte Carlo methods, also known as sequential Monte Carlo (SMC) methods, in which a whole (possibly weighted) sample, called *system of particles*, is propagated in time, where the particles

- *explore* the state space under the effect of a *mutation* mechanism which mimics the evolution of the underlying process,
- and are *replicated* or *terminated*, under the effect of a *selection* mechanism which automatically concentrates the particles, i.e. the available computing power, into regions of interest of the state space.

In full generality, the underlying process is a discrete–time Markov chain, whose state space can be finite, continuous, hybrid (continuous / discrete), graphical, constrained, time varying, pathwise, etc.,

the only condition being that it can easily be *simulated*.

In the special case of particle filtering, originally developed within the tracking community, the algorithms yield a numerical approximation of the optimal Bayesian filter, i.e. of the conditional probability distribution of the hidden state given the past observations, as a (possibly weighted) empirical probability distribution of the system of particles. In its simplest version, introduced in several different scientific communities under the name of *bootstrap filter* [52], *Monte Carlo filter* [57] or *condensation* (conditional density propagation) algorithm [54], and which historically has been the first algorithm to include a redistribution step, the selection mechanism is governed by the likelihood function: at each time step, a particle is more likely to survive and to replicate at the next generation if it is consistent with the current observation. The algorithms also provide as a by–product a numerical approximation of the likelihood function, and of many other contrast functions for parameter estimation in hidden Markov models, such as the prediction error or the conditional least–squares criterion.

Particle methods are currently being used in many scientific and engineering areas

positioning, navigation, and tracking [53], [47], visual tracking [54], mobile robotics [48], [71], ubiquitous computing and ambient intelligence, sensor networks, risk evaluation and simulation of rare events [51], genetics, molecular simulation [49], etc.

Other examples of the many applications of particle filtering can be found in the contributed volume [34] and in the special issue of *IEEE Transactions on Signal Processing* devoted to *Monte Carlo Methods for Statistical Signal Processing* in February 2002, where the tutorial paper [35] can be found, and in the textbook [67] devoted to applications in target tracking. Applications of sequential Monte Carlo methods to other areas, beyond signal and image processing, e.g. to genetics, can be found in [66]. A recent overview can also be found in [37].

Particle methods are very easy to implement, since it is sufficient in principle to simulate independent trajectories of the underlying process. The whole problematic is multidisciplinary, not only because of the already mentioned diversity of the scientific and engineering areas in which particle methods are used, but also because of the diversity of the scientific communities which have contributed to establish the foundations of the field

target tracking, interacting particle systems, empirical processes, genetic algorithms (GA), hidden Markov models and nonlinear filtering, Bayesian statistics, Markov chain Monte Carlo (MCMC) methods.

These algorithms can be interpreted as numerical approximation schemes for Feynman–Kac distributions, a pathwise generalization of Gibbs–Boltzmann distributions, in terms of the weighted empirical probability distribution associated with a system of particles. This abstract point of view [42], [40], has proved to be extremely fruitful in providing a very general framework to the design and analysis of numerical approximation schemes, based on systems of branching and / or interacting particles, for nonlinear dynamical systems with values in the space of probability distributions, associated with Feynman–Kac distributions. Many asymptotic results have been proved as the number N of particles (sample size) goes to infinity, using techniques coming from applied probability (interacting particle systems, empirical processes [74]), see e.g. the survey article [42] or the recent textbook [40], and references therein

convergence in L^p , convergence as empirical processes indexed by classes of functions, uniform convergence in time, see also [63], [64], central limit theorem, see also [59], propagation of chaos, large deviations principle, etc.

The objective here is to systematically study the impact of the many algorithmic variants on the convergence results.

3.2. Statistics of HMM

Hidden Markov models (HMM) form a special case of partially observed stochastic dynamical systems, in which the state of a Markov process (in discrete or continuous time, with finite or continuous state space) should be estimated from noisy observations. The conditional probability distribution of the hidden state given past observations is a well-known example of a normalized (nonlinear) Feynman–Kac distribution, see 3.1. These models are very flexible, because of the introduction of latent variables (non observed) which allows to model complex time dependent structures, to take constraints into account, etc. In addition, the underlying Markovian structure makes it possible to use numerical algorithms (particle filtering, Markov chain Monte Carlo methods (MCMC), etc.) which are computationally intensive but whose complexity is rather small. Hidden Markov models are widely used in various applied areas, such as speech recognition, alignment of biological sequences, tracking in complex environment, modeling and control of networks, digital communications, etc.

Beyond the recursive estimation of a hidden state from noisy observations, the problem arises of statistical inference of HMM with general state space [38], including estimation of model parameters, early monitoring and diagnosis of small changes in model parameters, etc.

Large time asymptotics A fruitful approach is the asymptotic study, when the observation time increases to infinity, of an extended Markov chain, whose state includes (i) the hidden state, (ii) the observation, (iii) the prediction filter (i.e. the conditional probability distribution of the hidden state given observations at all previous time instants), and possibly (iv) the derivative of the prediction filter with respect to the parameter. Indeed, it is easy to express the log-likelihood function, the conditional least-squares criterion, and many other classical contrast processes, as well as their derivatives with respect to the parameter, as additive functionals of the extended Markov chain.

The following general approach has been proposed

- first, prove an exponential stability property (i.e. an exponential forgetting property of the initial condition) of the prediction filter and its derivative, for a misspecified model,
- from this, deduce a geometric ergodicity property and the existence of a unique invariant probability distribution for the extended Markov chain, hence a law of large numbers and a central limit theorem for a large class of contrast processes and their derivatives, and a local asymptotic normality property,
- finally, obtain the consistency (i.e. the convergence to the set of minima of the associated contrast function), and the asymptotic normality of a large class of minimum contrast estimators.

This programme has been completed in the case of a finite state space [7], and has been generalized [43] under an uniform minoration assumption for the Markov transition kernel, which typically does only hold when the state space is compact. Clearly, the whole approach relies on the existence of an exponential stability property of the prediction filter, and the main challenge currently is to get rid of this uniform minoration assumption for the Markov transition kernel [41], [64], so as to be able to consider more interesting situations, where the state space is noncompact.

Small noise asymptotics Another asymptotic approach can also be used, where it is rather easy to obtain interesting explicit results, in terms close to the language of nonlinear deterministic control theory [58]. Taking the simple example where the hidden state is the solution to an ordinary differential equation, or a nonlinear state model, and where the observations are subject to additive Gaussian white noise, this approach consists in assuming that covariances matrices of the state noise and of the observation noise go simultaneously to zero. If it is reasonable in many applications to consider that noise covariances are small, this asymptotic approach is less natural than the large time asymptotics, where it is enough (provided a suitable ergodicity assumption holds) to accumulate observations and to see the expected limit laws (law of large numbers, central limit theorem, etc.). In opposition, the expressions obtained in the limit (Kullback–Leibler divergence, Fisher information matrix, asymptotic covariance matrix, etc.) take here a much more explicit form than in the large time asymptotics.

The following results have been obtained using this approach

- the consistency of the maximum likelihood estimator (i.e. the convergence to the set M of global minima of the Kullback–Leibler divergence), has been obtained using large deviations techniques, with an analytical approach [55],
- if the abovementioned set M does not reduce to the true parameter value, i.e. if the model is not identifiable, it is still possible to describe precisely the asymptotic behavior of the estimators [56]: in the simple case where the state equation is a noise-free ordinary differential equation and using a Bayesian framework, it has been shown that (i) if the rank r of the Fisher information matrix I is constant in a neighborhood of the set M , then this set is a differentiable submanifold of codimension r , (ii) the posterior probability distribution of the parameter converges to a random probability distribution in the limit, supported by the manifold M , absolutely continuous w.r.t. the Lebesgue measure on M , with an explicit expression for the density, and (iii) the posterior probability distribution of the suitably normalized difference between the parameter and its projection on the manifold M , converges to a mixture of Gaussian probability distributions on the normal spaces to the manifold M , which generalized the usual asymptotic normality property,

- it has been shown [65] that (i) the parameter dependent probability distributions of the observations are locally asymptotically normal (LAN) [61], from which the asymptotic normality of the maximum likelihood estimator follows, with an explicit expression for the asymptotic covariance matrix, i.e. for the Fisher information matrix I , in terms of the Kalman filter associated with the linear tangent linear Gaussian model, and (ii) the score function (i.e. the derivative of the log-likelihood function w.r.t. the parameter), evaluated at the true value of the parameter and suitably normalized, converges to a Gaussian r.v. with zero mean and covariance matrix I .

3.3. Multilevel splitting for rare event simulation

See 4.2, and 5.1, 5.6, 5.10 and 5.11.

The estimation of the small probability of a rare but critical event, is a crucial issue in industrial areas such as nuclear power plants, food industry, telecommunication networks, finance and insurance industry, air traffic management, etc.

In such complex systems, analytical methods cannot be used, and naive Monte Carlo methods are clearly inefficient to estimate accurately very small probabilities. Besides importance sampling, an alternate widespread technique consists in multilevel splitting [60], where trajectories going towards the critical set are given offsprings, thus increasing the number of trajectories that eventually reach the critical set. As shown in [5], the Feynman–Kac formalism of 3.1 is well suited for the design and analysis of splitting algorithms for rare event simulation.

Propagation of uncertainty Multilevel splitting can be used in static situations. Here, the objective is to learn the probability distribution of an output random variable $Y = F(X)$, where the function F is only defined pointwise for instance by a computer programme, and where the probability distribution of the input random variable X is known and easy to simulate from. More specifically, the objective could be to compute the probability of the output random variable exceeding a threshold, or more generally to evaluate the cumulative distribution function of the output random variable for different output values. This problem is characterized by the lack of an analytical expression for the function, the computational cost of a single pointwise evaluation of the function, which means that the number of calls to the function should be limited as much as possible, and finally the complexity and / or unavailability of the source code of the computer programme, which makes any modification very difficult or even impossible, for instance to change the model as in importance sampling methods.

The key issue is to learn as fast as possible regions of the input space which contribute most to the computation of the target quantity. The proposed splitting methods consists in (i) introducing a sequence of intermediate regions in the input space, implicitly defined by exceeding an increasing sequence of thresholds or levels, (ii) counting the fraction of samples that reach a level given that the previous level has been reached already, and (iii) improving the diversity of the selected samples, usually using an artificial Markovian dynamics. In this way, the algorithm learns

- the transition probability between successive levels, hence the probability of reaching each intermediate level,
- and the probability distribution of the input random variable, conditioned on the output variable reaching each intermediate level.

A further remark, is that this conditional probability distribution is precisely the optimal (zero variance) importance distribution needed to compute the probability of reaching the considered intermediate level.

Rare event simulation To be specific, consider a complex dynamical system modelled as a Markov process, whose state can possibly contain continuous components and finite components (mode, regime, etc.), and the objective is to compute the probability, hopefully very small, that a critical region of the state space is reached by the Markov process before a final time T , which can be deterministic and fixed, or random (for instance the time of return to a recurrent set, corresponding to a nominal behaviour).

The proposed splitting method consists in (i) introducing a decreasing sequence of intermediate, more and more critical, regions in the state space, (ii) counting the fraction of trajectories that reach an intermediate region before time T , given that the previous intermediate region has been reached before time T , and (iii) regenerating the population at each stage, through redistribution. In addition to the non-intrusive behaviour of the method, the splitting methods make it possible to learn the probability distribution of typical critical trajectories, which reach the critical region before final time T , an important feature that methods based on importance sampling usually miss. Many variants have been proposed, whether

- the branching rate (number of offsprings allocated to a successful trajectory) is fixed, which allows for depth-first exploration of the branching tree, but raises the issue of controlling the population size,
- the population size is fixed, which requires a breadth-first exploration of the branching tree, with random (multinomial) or deterministic allocation of offsprings, etc.

Just as in the static case, the algorithm learns

- the transition probability between successive levels, hence the probability of reaching each intermediate level,
- and the entrance probability distribution of the Markov process in each intermediate region.

Contributions have been given to

- minimizing the asymptotic variance, obtained through a central limit theorem, with respect to the shape of the intermediate regions (selection of the importance function), to the thresholds (levels), to the population size, etc.
- controlling the probability of extinction (when not even one trajectory reaches the next intermediate level),
- designing and studying variants suited for hybrid state space (resampling per mode, marginalization, mode aggregation),

and in the static case, to

- minimizing the asymptotic variance, obtained through a central limit theorem, with respect to intermediate levels, to the Metropolis kernel introduced in the mutation step, etc.

A related issue is global optimization. Indeed, the difficult problem of finding the set M of global minima of a real-valued function V can be replaced by the apparently simpler problem of sampling a population from a probability distribution depending on a small parameter, and asymptotically supported by the set M as the small parameter goes to zero. The usual approach here is to use the cross-entropy method [68], [39], which relies on learning the optimal importance distribution within a prescribed parametric family. On the other hand, multilevel splitting methods could provide an alternate nonparametric approach to this problem.

3.4. Nearest neighbor estimates

This additional topic was not present in the initial list of objectives, and has emerged only recently.

In pattern recognition and statistical learning, also known as machine learning, nearest neighbor (NN) algorithms are amongst the simplest but also very powerful algorithms available. Basically, given a training set of data, i.e. an N -sample of i.i.d. object-feature pairs, with real-valued features, the question is how to generalize, that is how to guess the feature associated with any new object. To achieve this, one chooses some integer k smaller than N , and takes the mean-value of the k features associated with the k objects that are nearest to the new object, for some given metric.

In general, there is no way to guess exactly the value of the feature associated with the new object, and the minimal error that can be done is that of the Bayes estimator, which cannot be computed by lack of knowledge of the distribution of the object–feature pair, but the Bayes estimator can be useful to characterize the strength of the method. So the best that can be expected is that the NN estimator converges, say when the sample size N grows, to the Bayes estimator. This is what has been proved in great generality by Stone [69] for the mean square convergence, provided that the object is a finite–dimensional random variable, the feature is a square–integrable random variable, and the ratio k/N goes to 0. Nearest neighbor estimator is not the only local averaging estimator with this property, but it is arguably the simplest.

The asymptotic behavior when the sample size grows is well understood in finite dimension, but the situation is radically different in general infinite dimensional spaces, when the objects to be classified are functions, images, etc.

Nearest neighbor classification in infinite dimension In finite dimension, the k –nearest neighbor classifier is universally consistent, i.e. its probability of error converges to the Bayes risk as N goes to infinity, whatever the joint probability distribution of the pair, provided that the ratio k/N goes to zero. Unfortunately, this result is no longer valid in general metric spaces, and the objective is to find out reasonable sufficient conditions for the weak consistency to hold. Even in finite dimension, there are exotic distances such that the nearest neighbor does not even get closer (in the sense of the distance) to the point of interest, and the state space needs to be complete for the metric, which is the first condition. Some regularity on the regression function is required next. Clearly, continuity is too strong because it is not required in finite dimension, and a weaker form of regularity is assumed. The following consistency result has been obtained: if the metric space is separable and if some Besicovich condition holds, then the nearest neighbor classifier is weakly consistent. Note that the Besicovich condition is always fulfilled in finite dimensional vector spaces (this result is called the Besicovich theorem), and that a counterexample [3] can be given in an infinite dimensional space with a Gaussian measure (in this case, the nearest neighbor classifier is clearly nonconsistent). Finally, a simple example has been found which verifies the Besicovich condition with a noncontinuous regression function.

Rates of convergence of the functional k –nearest neighbor estimator Motivated by a broad range of potential applications, such as regression on curves, rates of convergence of the k –nearest neighbor estimator of the regression function, based on N independent copies of the object–feature pair, have been investigated when the object is in a suitable ball in some functional space. Using compact embedding theory, explicit and general finite sample bounds can be obtained for the expected squared difference between the k –nearest neighbor estimator and the Bayes regression function, in a very general setting. The results have also been particularized to classical function spaces such as Sobolev spaces, Besov spaces and reproducing kernel Hilbert spaces. The rates obtained are genuine nonparametric convergence rates, and up to our knowledge the first of their kind for k –nearest neighbor regression.

This emerging topic has produced several theoretical advances [1], [2] in collaboration with Gérard Biau (université Pierre et Marie Curie, ENS Paris and EPI CLASSIC, Inria Paris—Rocquencourt), and a possible target application domain has been identified in the statistical analysis of recommendation systems, that would be a source of interesting problems.

I4S Team

3. Scientific Foundations

3.1. Introduction

In this section, the main features for the key monitoring issues, namely identification, detection, and diagnostics, are provided, and a particular instantiation relevant for vibration monitoring is described.

It should be stressed that the foundations for identification, detection, and diagnostics, are fairly general, if not generic. Handling high order linear dynamical systems, in connection with finite elements models, which call for using subspace-based methods, is specific to vibration-based SHM. Actually, one particular feature of model-based sensor information data processing as exercised in I4S, is the combined use of black-box or semi-physical models together with physical ones. Black-box and semi-physical models are, for example, eigenstructure parameterizations of linear MIMO systems, of interest for modal analysis and vibration-based SHM. Such models are intended to be identifiable. However, due to the large model orders that need to be considered, the issue of model order selection is really a challenge. Traditional advanced techniques from statistics such as the various forms of Akaike criteria (AIC, BIC, MDL, ...) do not work at all. This gives rise to new research activities specific to handling high order models.

Our approach to monitoring assumes that a model of the monitored system is available. This is a reasonable assumption, especially within the SHM areas. The main feature of our monitoring method is its intrinsic ability to the early warning of small deviations of a system with respect to a reference (safe) behavior under usual operating conditions, namely without any artificial excitation or other external action. Such a normal behavior is summarized in a reference parameter vector θ_0 , for example a collection of modes and mode-shapes.

3.2. Identification

The behavior of the monitored continuous system is assumed to be described by a parametric model $\{\mathbf{P}_\theta, \theta \in \Theta\}$, where the distribution of the observations (Z_0, \dots, Z_N) is characterized by the parameter vector $\theta \in \Theta$. An *estimating function*, for example of the form :

$$\mathcal{K}_N(\theta) = 1/N \sum_{k=0}^N K(\theta, Z_k)$$

is such that $\mathbf{E}_\theta[\mathcal{K}_N(\theta)] = 0$ for all $\theta \in \Theta$. In many situations, \mathcal{K} is the gradient of a function to be minimized : squared prediction error, log-likelihood (up to a sign), For performing model identification on the basis of observations (Z_0, \dots, Z_N) , an estimate of the unknown parameter is then [34] :

$$\hat{\theta}_N = \arg \{ \theta \in \Theta : \mathcal{K}_N(\theta) = 0 \}$$

Assuming that θ^* is the true parameter value, and that $\mathbf{E}_{\theta^*}[\mathcal{K}_N(\theta)] = 0$ if and only if $\theta = \theta^*$ with θ^* fixed (identifiability condition), then $\hat{\theta}_N$ converges towards θ^* . Thanks to the central limit theorem, the vector $\mathcal{K}_N(\theta^*)$ is asymptotically Gaussian with zero mean, with covariance matrix Σ which can be either computed or estimated. If, additionally, the matrix $\mathcal{J}_N = -\mathbf{E}_{\theta^*}[\mathcal{K}'_N(\theta^*)]$ is invertible, then using a Taylor expansion and the constraint $\mathcal{K}_N(\hat{\theta}_N) = 0$, the asymptotic normality of the estimate is obtained :

$$\sqrt{N}(\hat{\theta}_N - \theta^*) \approx \mathcal{J}_N^{-1} \sqrt{N} \mathcal{K}_N(\theta^*)$$

In many applications, such an approach must be improved in the following directions :

- *Recursive estimation*: the ability to compute $\widehat{\theta}_{N+1}$ simply from $\widehat{\theta}_N$;
- *Adaptive estimation*: the ability to *track* the true parameter θ^* when it is time-varying.

3.3. Detection

Our approach to on-board detection is based on the so-called asymptotic statistical local approach, which we have extended and adapted [5], [4], [2]. It is worth noticing that these investigations of ours have been initially motivated by a vibration monitoring application example. It should also be stressed that, as opposite to many monitoring approaches, our method does not require repeated identification for each newly collected data sample.

For achieving the early detection of small deviations with respect to the normal behavior, our approach generates, on the basis of the reference parameter vector θ_0 and a new data record, indicators which automatically perform :

- The early detection of a slight mismatch between the model and the data;
- A preliminary diagnostics and localization of the deviation(s);
- The tradeoff between the magnitude of the detected changes and the uncertainty resulting from the estimation error in the reference model and the measurement noise level.

These indicators are computationally cheap, and thus can be embedded. This is of particular interest in some applications, such as flutter monitoring, as explained in module 4.4 .

As in most fault detection approaches, the key issue is to design a *residual*, which is ideally close to zero under normal operation, and has low sensitivity to noises and other nuisance perturbations, but high sensitivity to small deviations, before they develop into events to be avoided (damages, faults, ...). The originality of our approach is to :

- *Design* the residual basically as a *parameter estimating function*,
- *Evaluate* the residual thanks to a kind of central limit theorem, stating that the residual is asymptotically Gaussian and reflects the presence of a deviation in the parameter vector through a change in its own mean vector, which switches from zero in the reference situation to a non-zero value.

This is actually a strong result, which transforms any detection problem concerning a parameterized stochastic *process* into the problem of monitoring the mean of a Gaussian *vector*.

The behavior of the monitored system is again assumed to be described by a parametric model $\{\mathbf{P}_\theta, \theta \in \Theta\}$, and the safe behavior of the process is assumed to correspond to the parameter value θ_0 . This parameter often results from a preliminary identification based on reference data, as in module 3.2 .

Given a new N -size sample of sensors data, the following question is addressed : *Does the new sample still correspond to the nominal model \mathbf{P}_{θ_0} ?* One manner to address this generally difficult question is the following. The asymptotic local approach consists in deciding between the nominal hypothesis and a *close* alternative hypothesis, namely :

$$\text{(Safe) } \mathbf{H}_0 : \theta = \theta_0 \quad \text{and} \quad \text{(Damaged) } \mathbf{H}_1 : \theta = \theta_0 + \eta/\sqrt{N} \quad (3)$$

where η is an unknown but fixed change vector. A residual is generated under the form :

$$\zeta_N = 1/\sqrt{N} \sum_{k=0}^N K(\theta_0, Z_k) = \sqrt{N} \mathcal{K}_N(\theta_0) . \quad (4)$$

If the matrix $\mathcal{J}_N = -\mathbf{E}_{\theta_0}[\mathcal{K}'_N(\theta_0)]$ converges towards a limit \mathcal{J} , then the central limit theorem shows [31] that the residual is asymptotically Gaussian :

$$\zeta_N \xrightarrow{N \rightarrow \infty} \begin{cases} \mathcal{N}(0, \Sigma) & \text{under } \mathbf{P}_{\theta_0} , \\ \mathcal{N}(\mathcal{J}\eta, \Sigma) & \text{under } \mathbf{P}_{\theta_0 + \eta/\sqrt{N}} , \end{cases} \quad (5)$$

where the asymptotic covariance matrix Σ can be estimated, and manifests the deviation in the parameter vector by a change in its own mean value. Then, deciding between $\eta = 0$ and $\eta \neq 0$ amounts to compute the following χ^2 -test, provided that \mathcal{J} is full rank and Σ is invertible :

$$\chi^2 = \bar{\zeta}^T \mathbf{F}^{-1} \bar{\zeta} \geq \lambda . \quad (6)$$

where

$$\bar{\zeta} \triangleq \mathcal{J}^T \Sigma^{-1} \zeta_N \quad \text{and} \quad \mathbf{F} \triangleq \mathcal{J}^T \Sigma^{-1} \mathcal{J} \quad (7)$$

With this approach, it is possible to decide, with a quantifiable error level, if a residual value is significantly different from zero, for assessing whether a fault/damage has occurred. It should be stressed that the residual and the sensitivity and covariance matrices \mathcal{J} and Σ can be evaluated (or estimated) for the nominal model. In particular, it is *not* necessary to re-identify the model, and the sensitivity and covariance matrices can be pre-computed off-line.

3.4. Diagnostics

A further monitoring step, often called *fault isolation*, consists in determining which (subsets of) components of the parameter vector θ have been affected by the change. Solutions for that are now described. How this relates to diagnostics is addressed afterwards.

3.4.1. Isolation.

The question: *which (subsets of) components of θ have changed ?*, can be addressed using either nuisance parameters elimination methods or a multiple hypotheses testing approach [29]. Here we only sketch two intuitively simple statistical nuisance elimination techniques, which proceed by projection and rejection, respectively.

The fault vector η is partitioned into an informative part and a nuisance part, and the sensitivity matrix \mathcal{J} , the Fisher information matrix $\mathbf{F} = \mathcal{J}^T \Sigma^{-1} \mathcal{J}$ and the normalized residual $\bar{\zeta} = \mathcal{J}^T \Sigma^{-1} \zeta_N$ are partitioned accordingly

$$\eta = \begin{pmatrix} \eta_a \\ \eta_b \end{pmatrix} , \quad \mathcal{J} = \begin{pmatrix} \mathcal{J}_a & \mathcal{J}_b \end{pmatrix} , \quad \mathbf{F} = \begin{pmatrix} \mathbf{F}_{aa} & \mathbf{F}_{ab} \\ \mathbf{F}_{ba} & \mathbf{F}_{bb} \end{pmatrix} , \quad \bar{\zeta} = \begin{pmatrix} \bar{\zeta}_a \\ \bar{\zeta}_b \end{pmatrix} .$$

A rather intuitive statistical solution to the isolation problem, which can be called *sensitivity* approach, consists in projecting the deviations in η onto the subspace generated by the components η_a to be isolated, and deciding between $\eta_a = \eta_b = 0$ and $\eta_a \neq 0, \eta_b = 0$. This results in the following test statistics :

$$t_a = \bar{\zeta}_a^T \mathbf{F}_{aa}^{-1} \bar{\zeta}_a , \quad (8)$$

where $\bar{\zeta}_a$ is the partial residual (score). If $t_a \geq t_b$, the component responsible for the fault is considered to be a rather than b .

Another statistical solution to the problem of isolating η_a consists in viewing parameter η_b as a nuisance, and using an existing method for inferring part of the parameters while ignoring and being robust to the complementary part. This method is called *min-max approach*. It consists in replacing the nuisance parameter component η_b by its least favorable value, for deciding between $\eta_a = 0$ and $\eta_a \neq 0$, with η_b unknown. This results in the following test statistics :

$$t_a^* = \bar{\zeta}_a^{*T} \mathbf{F}_a^{*-1} \bar{\zeta}_a^* , \quad (9)$$

where $\bar{\zeta}_a^* \triangleq \bar{\zeta}_a - \mathbf{F}_{ab} \mathbf{F}_{bb}^{-1} \bar{\zeta}_b$ is the effective residual (score) resulting from the regression of the informative partial score $\bar{\zeta}_a$ over the nuisance partial score $\bar{\zeta}_b$, and where the Schur complement $\mathbf{F}_a^* = \mathbf{F}_{aa} - \mathbf{F}_{ab} \mathbf{F}_{bb}^{-1} \mathbf{F}_{ba}$ is the associated Fisher information matrix. If $t_a^* \geq t_b^*$, the component responsible for the fault is considered to be a rather than b .

The properties and relationships of these two types of tests are investigated in [28].

3.4.2. Diagnostics.

In most SHM applications, a complex physical system, characterized by a generally non identifiable parameter vector Φ has to be monitored using a simple (black-box) model characterized by an identifiable parameter vector θ . A typical example is the vibration monitoring problem in module 4.2 , for which complex finite elements models are often available but not identifiable, whereas the small number of existing sensors calls for identifying only simplified input-output (black-box) representations. In such a situation, two different diagnosis problems may arise, namely diagnosis in terms of the black-box parameter θ and diagnosis in terms of the parameter vector Φ of the underlying physical model.

The isolation methods sketched above are possible solutions to the former. Our approach to the latter diagnosis problem is basically a detection approach again, and not a (generally ill-posed) inverse problem estimation approach [3]. The basic idea is to note that the physical sensitivity matrix writes $\mathcal{J} \mathcal{J}_{\Phi\theta}$, where $\mathcal{J}_{\Phi\theta}$ is the Jacobian matrix at Φ_0 of the application $\Phi \mapsto \theta(\Phi)$, and to use the sensitivity test (6) for the components of the parameter vector Φ . Typically this results in the following type of directional test :

$$\chi_{\Phi}^2 = \zeta^T \Sigma^{-1} \mathcal{J} \mathcal{J}_{\Phi\theta} (\mathcal{J}_{\Phi\theta}^T \mathcal{J}^T \Sigma^{-1} \mathcal{J} \mathcal{J}_{\Phi\theta})^{-1} \mathcal{J}_{\Phi\theta}^T \mathcal{J}^T \Sigma^{-1} \zeta \geq \lambda . \quad (10)$$

It should be clear that the selection of a particular parameterization Φ for the physical model may have a non negligible influence on such type of tests, according to the numerical conditioning of the Jacobian matrices $\mathcal{J}_{\Phi\theta}$.

As a summary, the machinery in modules 3.2 , 3.3 and 3.4 provides us with a generic framework for designing monitoring algorithms for continuous structures, machines and processes. This approach assumes that a model of the monitored system is available. This is a reasonable assumption within the field of applications described in module 4.2 , since most mechanical processes rely on physical principles which write in terms of equations, providing us with models. These important *modeling* and *parameterization* issues are among the questions we intend to investigate within our research program.

The key issue to be addressed within each parametric model class is the residual generation, or equivalently the choice of the *parameter estimating function*.

3.5. Subspace-based identification and detection

For reasons closely related to the vibrations monitoring applications described in module 4.2 , we have been investigating subspace-based methods, for both the identification and the monitoring of the eigenstructure (λ, ϕ_λ) of the state transition matrix F of a linear dynamical state-space system :

$$\begin{cases} X_{k+1} = F X_k + V_{k+1} \\ Y_k = H X_k \end{cases}, \quad (11)$$

namely the $(\lambda, \varphi_\lambda)$ defined by :

$$\det (F - \lambda I) = 0, \quad (F - \lambda I) \phi_\lambda = 0, \quad \varphi_\lambda \triangleq H \phi_\lambda \quad (12)$$

The (canonical) parameter vector in that case is :

$$\theta \triangleq \begin{pmatrix} \Lambda \\ \text{vec} \Phi \end{pmatrix} \quad (13)$$

where Λ is the vector whose elements are the eigenvalues λ , Φ is the matrix whose columns are the φ_λ 's, and vec is the column stacking operator.

Subspace-based methods is the generic name for linear systems identification algorithms based on either time domain measurements or output covariance matrices, in which different subspaces of Gaussian random vectors play a key role [39]. A contribution of ours, minor but extremely fruitful, has been to write the output-only covariance-driven subspace identification method under a form that involves a parameter estimating function, from which we define a *residual adapted to vibration monitoring* [1]. This is explained next.

3.5.1. Covariance-driven subspace identification.

Let $R_i \triangleq \mathbf{E} (Y_k Y_{k-i}^T)$ and:

$$\mathcal{H}_{p+1,q} \triangleq \begin{pmatrix} R_0 & R_1 & \vdots & R_{q-1} \\ R_1 & R_2 & \vdots & R_q \\ \vdots & \vdots & \vdots & \vdots \\ R_p & R_{p+1} & \vdots & R_{p+q-1} \end{pmatrix} \triangleq \text{Hank} (R_i) \quad (14)$$

be the output covariance and Hankel matrices, respectively; and: $G \triangleq \mathbf{E} (X_k Y_k^T)$. Direct computations of the R_i 's from the equations (9) lead to the well known key factorizations :

$$\begin{aligned} R_i &= H F^i G \\ \mathcal{H}_{p+1,q} &= \mathcal{O}_{p+1}(H, F) \mathcal{C}_q(F, G) \end{aligned} \quad (15)$$

where:

$$\mathcal{O}_{p+1}(H, F) \triangleq \begin{pmatrix} H \\ HF \\ \vdots \\ HF^p \end{pmatrix} \quad \text{and} \quad \mathcal{C}_q(F, G) \triangleq (G \quad FG \quad \dots \quad F^{q-1}G) \quad (16)$$

are the observability and controllability matrices, respectively. The observation matrix H is then found in the first block-row of the observability matrix \mathcal{O} . The state-transition matrix F is obtained from the shift invariance property of \mathcal{O} . The eigenstructure (λ, ϕ_λ) then results from (10).

Since the actual model order is generally not known, this procedure is run with increasing model orders.

3.5.2. Model parameter characterization.

Choosing the eigenvectors of matrix F as a basis for the state space of model (9) yields the following representation of the observability matrix:

$$\mathcal{O}_{p+1}(\theta) = \begin{pmatrix} \Phi \\ \Phi \Delta \\ \vdots \\ \Phi \Delta^p \end{pmatrix} \quad (17)$$

where $\Delta \triangleq \text{diag}(\Lambda)$, and Λ and Φ are as in (11). Whether a nominal parameter θ_0 fits a given output covariance sequence $(R_j)_j$ is characterized by [1]:

$$\mathcal{O}_{p+1}(\theta_0) \text{ and } \mathcal{H}_{p+1,q} \text{ have the same left kernel space.} \quad (18)$$

This property can be checked as follows. From the nominal θ_0 , compute $\mathcal{O}_{p+1}(\theta_0)$ using (15), and perform e.g. a singular value decomposition (SVD) of $\mathcal{O}_{p+1}(\theta_0)$ for extracting a matrix U such that:

$$U^T U = I_s \text{ and } U^T \mathcal{O}_{p+1}(\theta_0) = 0 \quad (19)$$

Matrix U is not unique (two such matrices relate through a post-multiplication with an orthonormal matrix), but can be regarded as a function of θ_0 . Then the characterization writes:

$$U(\theta_0)^T \mathcal{H}_{p+1,q} = 0 \quad (20)$$

3.5.3. Residual associated with subspace identification.

Assume now that a reference θ_0 and a new sample Y_1, \dots, Y_N are available. For checking whether the data agree with θ_0 , the idea is to compute the empirical Hankel matrix $\hat{\mathcal{H}}_{p+1,q}$:

$$\hat{\mathcal{H}}_{p+1,q} \triangleq \text{Hank}(\hat{R}_i), \quad \hat{R}_i \triangleq 1/(N-i) \sum_{k=i+1}^N Y_k Y_{k-i}^T \quad (21)$$

and to define the residual vector:

$$\zeta_N(\theta_0) \triangleq \sqrt{N} \text{vec} \left(U(\theta_0)^T \hat{\mathcal{H}}_{p+1,q} \right) \quad (22)$$

Let θ be the actual parameter value for the system which generated the new data sample, and \mathbf{E}_θ be the expectation when the actual system parameter is θ . From (18), we know that $\zeta_N(\theta_0)$ has zero mean when no change occurs in θ , and nonzero mean if a change occurs. Thus $\zeta_N(\theta_0)$ plays the role of a residual.

It is our experience that this residual has highly interesting properties, both for damage detection [1] and localization [3], and for flutter monitoring [8].

3.5.4. Other uses of the key factorizations.

Factorization (3.5.1) is the key for a characterization of the canonical parameter vector θ in (11), and for deriving the residual. Factorization (13) is also the key for :

- Proving consistency and robustness results [6];
- Designing an extension of covariance-driven subspace identification algorithm adapted to the presence and fusion of non-simultaneously recorded multiple sensors setups [7];
- Proving the consistency and robustness of this extension [9];
- Designing various forms of *input-output* covariance-driven subspace identification algorithms adapted to the presence of both known inputs and unknown excitations [10].

IPSO Project-Team

3. Scientific Foundations

3.1. Structure-preserving numerical schemes for solving ordinary differential equations

Participants: François Castella, Philippe Chartier, Erwan Faou, Vilmart Gilles.

ordinary differential equation, numerical integrator, invariant, Hamiltonian system, reversible system, Lie-group system

In many physical situations, the time-evolution of certain quantities may be written as a Cauchy problem for a differential equation of the form

$$\begin{aligned} y'(t) &= f(y(t)), \\ y(0) &= y_0. \end{aligned} \tag{23}$$

For a given y_0 , the solution $y(t)$ at time t is denoted $\varphi_t(y_0)$. For fixed t , φ_t becomes a function of y_0 called the *flow* of (1). From this point of view, a numerical scheme with step size h for solving (1) may be regarded as an approximation Φ_h of φ_h . One of the main questions of *geometric integration* is whether *intrinsic* properties of φ_t may be passed on to Φ_h .

This question can be more specifically addressed in the following situations:

3.1.1. Reversible ODEs

The system (1) is said to be ρ -reversible if there exists an involutive linear map ρ such that

$$\rho \circ \varphi_t = \varphi_t^{-1} \circ \rho = \varphi_{-t} \circ \rho. \tag{24}$$

It is then natural to require that Φ_h satisfies the same relation. If this is so, Φ_h is said to be *symmetric*. Symmetric methods for reversible systems of ODEs are just as much important as *symplectic* methods for Hamiltonian systems and offer an interesting alternative to symplectic methods.

3.1.2. ODEs with an invariant manifold

The system (1) is said to have an invariant manifold g whenever

$$\mathcal{M} = \{y \in \mathbb{R}^n; g(y) = 0\} \tag{25}$$

is kept *globally* invariant by φ_t . In terms of derivatives and for sufficiently differentiable functions f and g , this means that

$$\forall y \in \mathcal{M}, g'(y)f(y) = 0.$$

As an example, we mention Lie-group equations, for which the manifold has an additional group structure. This could possibly be exploited for the space-discretisation. Numerical methods amenable to this sort of problems have been reviewed in a recent paper [56] and divided into two classes, according to whether they use g explicitly or through a projection step. In both cases, the numerical solution is forced to live on the manifold at the expense of some Newton's iterations.

3.1.3. Hamiltonian systems

Hamiltonian problems are ordinary differential equations of the form:

$$\begin{aligned}\dot{p}(t) &= -\nabla_q H(p(t), q(t)) \in \mathbb{R}^d \\ \dot{q}(t) &= \nabla_p H(p(t), q(t)) \in \mathbb{R}^d\end{aligned}\tag{26}$$

with some prescribed initial values $(p(0), q(0)) = (p_0, q_0)$ and for some scalar function H , called the Hamiltonian. In this situation, H is an invariant of the problem. The evolution equation (4) can thus be regarded as a differential equation on the manifold

$$\mathcal{M} = \{(p, q) \in \mathbb{R}^d \times \mathbb{R}^d; H(p, q) = H(p_0, q_0)\}.$$

Besides the Hamiltonian function, there might exist other invariants for such systems: when there exist d invariants in involution, the system (4) is said to be *integrable*. Consider now the parallelogram P originating from the point $(p, q) \in \mathbb{R}^{2d}$ and spanned by the two vectors $\xi \in \mathbb{R}^{2d}$ and $\eta \in \mathbb{R}^{2d}$, and let $\omega(\xi, \eta)$ be the sum of the *oriented* areas of the projections over the planes (p_i, q_i) of P ,

$$\omega(\xi, \eta) = \xi^T J \eta,$$

where J is the *canonical symplectic* matrix

$$J = \begin{bmatrix} 0 & I_d \\ -I_d & 0 \end{bmatrix}.$$

A continuously differentiable map g from \mathbb{R}^{2d} to itself is called *symplectic* if it preserves ω , i.e. if

$$\omega(g'(p, q)\xi, g'(p, q)\eta) = \omega(\xi, \eta).$$

A fundamental property of Hamiltonian systems is that their exact flow is symplectic. Integrable Hamiltonian systems behave in a very remarkable way: as a matter of fact, their invariants persist under small perturbations, as shown in the celebrated theory of Kolmogorov, Arnold and Moser. This behavior motivates the introduction of *symplectic* numerical flows that share most of the properties of the exact flow. For practical simulations of Hamiltonian systems, symplectic methods possess an important advantage: the error-growth as a function of time is indeed linear, whereas it would typically be quadratic for non-symplectic methods.

3.1.4. Differential-algebraic equations

Whenever the number of differential equations is insufficient to determine the solution of the system, it may become necessary to solve the differential part and the constraint part altogether. Systems of this sort are called *differential-algebraic* systems. They can be classified according to their index, yet for the purpose of this expository section, it is enough to present the so-called index-2 systems

$$\begin{aligned}\dot{y}(t) &= f(y(t), z(t)), \\ 0 &= g(y(t)),\end{aligned}\tag{27}$$

where initial values $(y(0), z(0)) = (y_0, z_0)$ are given and assumed to be consistent with the constraint manifold. By constraint manifold, we imply the intersection of the manifold

$$\mathcal{M}_1 = \{y \in \mathbb{R}^n, g(y) = 0\}$$

and of the so-called hidden manifold

$$\mathcal{M}_2 = \{(y, z) \in \mathbb{R}^n \times \mathbb{R}^m, \frac{\partial g}{\partial y}(y)f(y, z) = 0\}.$$

This manifold $\mathcal{M} = \mathcal{M}_1 \cap \mathcal{M}_2$ is the manifold on which the exact solution $(y(t), z(t))$ of (5) lives.

There exists a whole set of schemes which provide a numerical approximation lying on \mathcal{M}_1 . Furthermore, this solution can be projected on the manifold \mathcal{M} by standard projection techniques. However, it is worth mentioning that a projection destroys the symmetry of the underlying scheme, so that the construction of a symmetric numerical scheme preserving \mathcal{M} requires a more sophisticated approach.

3.2. Highly-oscillatory systems

Participants: François Castella, Philippe Chartier, Nicolas Crouseilles, Erwan Faou, Florian Méhats, Mohammed Lemou, Gilles Vilmart.

second-order ODEs, oscillatory solutions, Schrödinger and wave equations, step size restrictions.

In applications to molecular dynamics or quantum dynamics for instance, the right-hand side of (1) involves *fast* forces (short-range interactions) and *slow* forces (long-range interactions). Since *fast* forces are much cheaper to evaluate than *slow* forces, it seems highly desirable to design numerical methods for which the number of evaluations of slow forces is not (at least not too much) affected by the presence of fast forces.

A typical model of highly-oscillatory systems is the second-order differential equations

$$\ddot{q} = -\nabla V(q) \tag{28}$$

where the potential $V(q)$ is a sum of potentials $V = W + U$ acting on different time-scales, with $\nabla^2 W$ positive definite and $\|\nabla^2 W\| \gg \|\nabla^2 U\|$. In order to get a bounded error propagation in the linearized equations for an explicit numerical method, the step size must be restricted according to

$$h\omega < C,$$

where C is a constant depending on the numerical method and where ω is the highest frequency of the problem, i.e. in this situation the square root of the largest eigenvalue of $\nabla^2 W$. In applications to molecular dynamics for instance, *fast* forces deriving from W (short-range interactions) are much cheaper to evaluate than *slow* forces deriving from U (long-range interactions). In this case, it thus seems highly desirable to design numerical methods for which the number of evaluations of slow forces is not (at least not too much) affected by the presence of fast forces.

Another prominent example of highly-oscillatory systems is encountered in quantum dynamics where the Schrödinger equation is the model to be used. Assuming that the Laplacian has been discretized in space, one indeed gets the *time*-dependent Schrödinger equation:

$$i\dot{\psi}(t) = \frac{1}{\varepsilon} H(t)\psi(t), \tag{29}$$

where $H(t)$ is finite-dimensional matrix and where ε typically is the square-root of a mass-ratio (say electron/ion for instance) and is small ($\varepsilon \approx 10^{-2}$ or smaller). Through the coupling with classical mechanics ($H(t)$ is obtained by solving some equations from classical mechanics), we are faced once again with two different time-scales, 1 and ε . In this situation also, it is thus desirable to devise a numerical method able to advance the solution by a time-step $h > \varepsilon$.

3.3. Geometric schemes for the Schrödinger equation

Participants: François Castella, Philippe Chartier, Erwan Faou, Florian Méhats, Gilles Vilmart.

Schrödinger equation, variational splitting, energy conservation.

Given the Hamiltonian structure of the Schrödinger equation, we are led to consider the question of energy preservation for time-discretization schemes.

At a higher level, the Schrödinger equation is a partial differential equation which may exhibit Hamiltonian structures. This is the case of the time-dependent Schrödinger equation, which we may write as

$$i\varepsilon \frac{\partial \psi}{\partial t} = H\psi, \quad (30)$$

where $\psi = \psi(x, t)$ is the wave function depending on the spatial variables $x = (x_1, \dots, x_N)$ with $x_k \in \mathbb{R}^d$ (e.g., with $d = 1$ or 3 in the partition) and the time $t \in \mathbb{R}$. Here, ε is a (small) positive number representing the scaled Planck constant and i is the complex imaginary unit. The Hamiltonian operator H is written

$$H = T + V$$

with the kinetic and potential energy operators

$$T = - \sum_{k=1}^N \frac{\varepsilon^2}{2m_k} \Delta_{x_k} \quad \text{and} \quad V = V(x),$$

where $m_k > 0$ is a particle mass and Δ_{x_k} the Laplacian in the variable $x_k \in \mathbb{R}^d$, and where the real-valued potential V acts as a multiplication operator on ψ .

The multiplication by i in (8) plays the role of the multiplication by J in classical mechanics, and the energy $\langle \psi | H | \psi \rangle$ is conserved along the solution of (8), using the physicists' notations $\langle u | A | u \rangle = \langle u, Au \rangle$ where $\langle \cdot, \cdot \rangle$ denotes the Hermitian L^2 -product over the phase space. In quantum mechanics, the number N of particles is very large making the direct approximation of (8) very difficult.

The numerical approximation of (8) can be obtained using projections onto submanifolds of the phase space, leading to various PDEs or ODEs: see [60], [59] for reviews. However the long-time behavior of these approximated solutions is well understood only in this latter case, where the dynamics turns out to be finite dimensional. In the general case, it is very difficult to prove the preservation of qualitative properties of (8) such as energy conservation or growth in time of Sobolev norms. The reason for this is that backward error analysis is not directly applicable for PDEs. Overwhelming these difficulties is thus a very interesting challenge.

A particularly interesting case of study is given by symmetric splitting methods, such as the Strang splitting:

$$\psi_1 = \exp(-i(\delta t)V/2) \exp(i(\delta t)\Delta) \exp(-i(\delta t)V/2)\psi_0 \quad (31)$$

where δt is the time increment (we have set all the parameters to 1 in the equation). As the Laplace operator is unbounded, we cannot apply the standard methods used in ODEs to derive long-time properties of these schemes. However, its projection onto finite dimensional submanifolds (such as Gaussian wave packets space or FEM finite dimensional space of functions in x) may exhibit Hamiltonian or Poisson structure, whose long-time properties turn out to be more tractable.

3.4. High-frequency limit of the Helmholtz equation

Participant: François Castella.

waves, Helmholtz equation, high oscillations.

The Helmholtz equation models the propagation of waves in a medium with variable refraction index. It is a simplified version of the Maxwell system for electro-magnetic waves.

The high-frequency regime is characterized by the fact that the typical wavelength of the signals under consideration is much smaller than the typical distance of observation of those signals. Hence, in the high-frequency regime, the Helmholtz equation at once involves highly oscillatory phenomena that are to be described in some asymptotic way. Quantitatively, the Helmholtz equation reads

$$i\alpha_\varepsilon u_\varepsilon(x) + \varepsilon^2 \Delta_x u_\varepsilon + n^2(x)u_\varepsilon = f_\varepsilon(x). \quad (32)$$

Here, ε is the small adimensional parameter that measures the typical wavelength of the signal, $n(x)$ is the space-dependent refraction index, and $f_\varepsilon(x)$ is a given (possibly dependent on ε) source term. The unknown is $u_\varepsilon(x)$. One may think of an antenna emitting waves in the whole space (this is the $f_\varepsilon(x)$), thus creating at any point x the signal $u_\varepsilon(x)$ along the propagation. The small $\alpha_\varepsilon > 0$ term takes into account damping of the waves as they propagate.

One important scientific objective typically is to describe the high-frequency regime in terms of *rays* propagating in the medium, that are possibly refracted at interfaces, or bounce on boundaries, etc. Ultimately, one would like to replace the true numerical resolution of the Helmholtz equation by that of a simpler, asymptotic model, formulated in terms of rays.

In some sense, and in comparison with, say, the wave equation, the specificity of the Helmholtz equation is the following. While the wave equation typically describes the evolution of waves between some initial time and some given observation time, the Helmholtz equation takes into account at once the propagation of waves over *infinitely long* time intervals. Qualitatively, in order to have a good understanding of the signal observed in some bounded region of space, one readily needs to be able to describe the propagative phenomena in the whole space, up to infinity. In other words, the “rays” we refer to above need to be understood from the initial time up to infinity. This is a central difficulty in the analysis of the high-frequency behaviour of the Helmholtz equation.

3.5. From the Schrödinger equation to Boltzmann-like equations

Participant: François Castella.

Schrödinger equation, asymptotic model, Boltzmann equation.

The Schrödinger equation is the appropriate way to describe transport phenomena at the scale of electrons. However, for real devices, it is important to derive models valid at a larger scale.

In semi-conductors, the Schrödinger equation is the ultimate model that allows to obtain quantitative information about electronic transport in crystals. It reads, in convenient adimensional units,

$$i\partial_t \psi(t, x) = -\frac{1}{2} \Delta_x \psi + V(x)\psi, \quad (33)$$

where $V(x)$ is the potential and $\psi(t, x)$ is the time- and space-dependent wave function. However, the size of real devices makes it important to derive simplified models that are valid at a larger scale. Typically, one wishes to have kinetic transport equations. As is well-known, this requirement needs one to be able to describe “collisions” between electrons in these devices, a concept that makes sense at the macroscopic level, while it does not at the microscopic (electronic) level. Quantitatively, the question is the following: can one obtain the Boltzmann equation (an equation that describes collisional phenomena) as an asymptotic model for the Schrödinger equation, along the physically relevant micro-macro asymptotics? From the point of view of modelling, one wishes here to understand what are the “good objects”, or, in more technical words, what are the relevant “cross-sections”, that describe the elementary collisional phenomena. Quantitatively, the Boltzmann equation reads, in a simplified, linearized, form :

$$\partial_t f(t, x, v) = \int_{\mathbf{R}^3} \sigma(v, v') [f(t, x, v') - f(t, x, v)] dv'. \quad (34)$$

Here, the unknown is $f(x, v, t)$, the probability that a particle sits at position x , with a velocity v , at time t . Also, $\sigma(v, v')$ is called the cross-section, and it describes the probability that a particle “jumps” from velocity v to velocity v' (or the converse) after a collision process.

DYLISS Team

3. Scientific Foundations

3.1. Knowledge representation with constraint programming

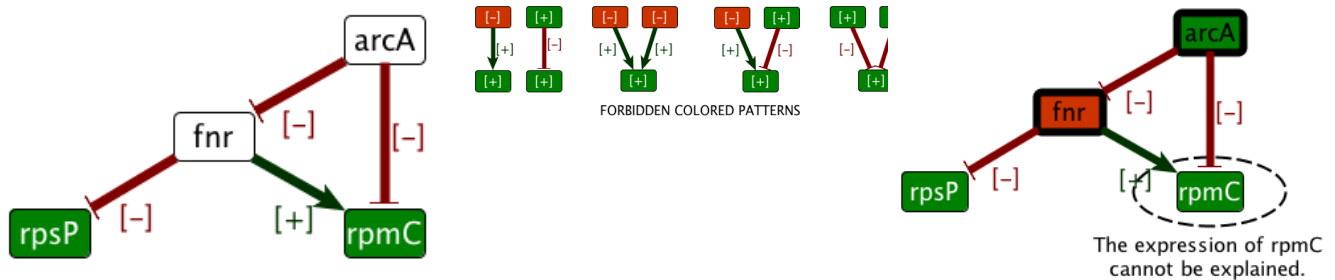
Biological networks are built with data-driven approaches aiming at translating genomic information into a functional map. Most methods are based on a probabilistic framework which defines a probability distribution over the set of models. The reconstructed network is then defined as the most likely model given the data. In the last few years, our team has investigated an alternative perspective where each observation induces a set of constraints - related to the steady state response of the system dynamics - on the set of possible values in a network of fixed topology. The methods that we have developed complete the network with product states at the level of nodes and influence types at the level of edges, able to globally explain experimental data. In other words, the selection of relevant information in the model is no more performed by selecting the network with the highest score, but rather by exploring the complete space of models satisfying constraints on the possible dynamics supported by prior knowledge and observations. Common properties to all solutions are considered as a robust information about the system, as they are independent from the choice of a single solution to the optimization problem[6].

Solving these computational issues requires addressing NP-hard qualitative (non-temporal) issues, based on a notion of causality. We have developed a long-term collaboration with Potsdam University in order to use a logical paradigm named **Answer Set Programming** [27], [30] to solve these optimization issues. Applied on transcriptomic or cancer networks, our methods identified which regions of a large-scale network shall be corrected [1], and proposed robust corrections [5]. The results obtained so far suggest that this approach is compatible with efficiency; scale and expressivity needed by biological systems. Our goal is now to provide **formal models of queries on biological networks** with the focus of integrating dynamical information as explicit logical constraints in the modeling process. This would definitely introduce such logical paradigms as a powerful approach to build and query reconstructed biological systems, in complement to discriminative approaches. Notice that our main issue is in the field of knowledge representation. More precisely, we do not wish to develop new solvers or grounders, a self-contained computational issue which is addressed by specialized teams such as our collaborator team in Potsdam. Our goal is rather to investigate whether progresses in the field of constraint logical programming, shown by the performance of ASP-solvers in several recent competitions, are now sufficient to address the complexity of optimization issues explored in systems biology.

Using these technologies requires to revisit and reformulate optimization problems at hand in order both to decrease the search space size in the grounding part of the process and to optimize the exploration of this search space in the solving part of the process. Concretely, getting logical encoding for the optimization problems forces to clarify the roles and dependencies between parameters involved in the problem. This opens the way to a refinement approach based on a fine investigation of the space of hypotheses in order to make it smaller and gain in the understanding of the system.

3.2. Probabilistic and symbolic dynamics

We work on new techniques to emphasize biological strategies that must occur to reproduce quantitative measurements in order to predict the quantitative response of a system at a larger-scale. Our framework mixes mechanistic and probabilistic modeling [2]. The system is modeled by an Event Transition Graph, that is, a **Markovian qualitative description of its dynamics** together with quantitative laws which describe the effect of the dynamic transitions over higher scale quantitative measurements. Then, a few time-series quantitative measurements are provided. Following an ergodic assumption and average case analysis properties, we know that a multiplicative accumulation law on a Markov chain asymptotically follows a log-normal law with explicit parameters [29]. This property can be derived into constraints to describe the set of admissible



Step 1 . Regulation knowledge is represented as a signed oriented graph. Edge colors stand for regulatory effects (red/green→ inhibition or activation). Vertex colors stand for gene expression data (red/green→ under or over-expression).

Step 2 . Integrity constraints on the whole colored graph come from the necessity to find a consistent explanation of the link between regulation and expression.

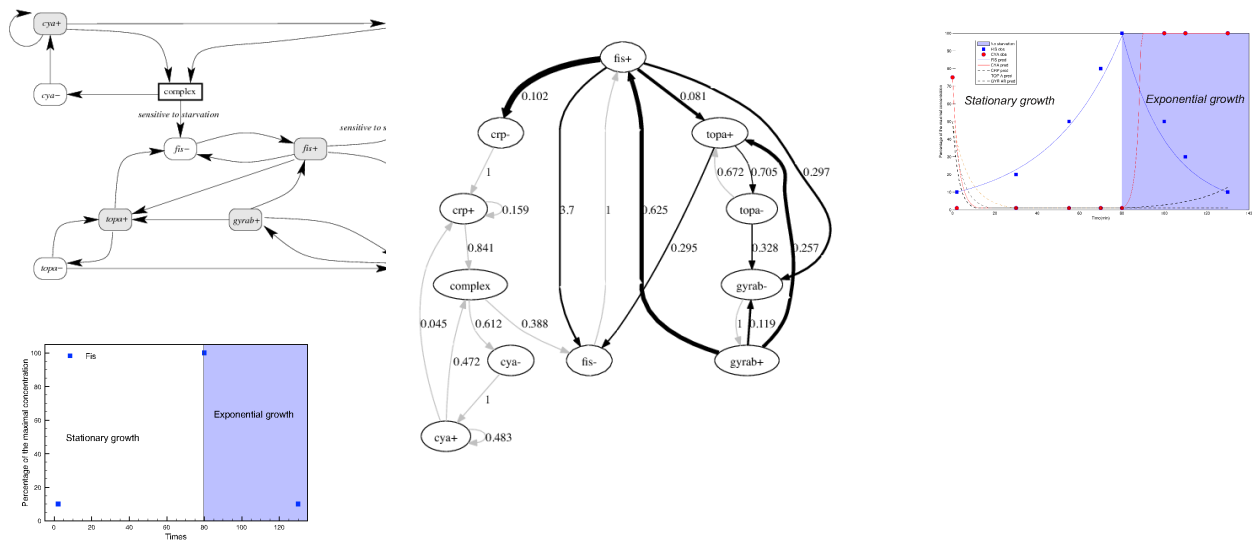
Step 3 . The model allows both the prediction of values (e.g. for fnr in the figure) and the detection of contradictions (e.g. the expression level of rpmC is inconsistent with the regulation in the graph).

Step 4 . Excerpt from the ASP program.

vertex(fnrc).	1 {labelV(I ,+;-)}1 :- vertex(I).	receive(I,+) :- labelE(J,I,S), labelV(J,S).
edge(fnrc,rpsP).	labelV(I ,S) :- observedV(I,S).	receive(I,-) :- labelE(J,I,S), labelV(J,T), S≠T.
observedE(fnrc,rpsP,-).	1 {labelE(J,I,+;-)}1 :- edge(J,I).	:- labelV (I,S), not receive(I,S).
observedV(rpsP,-).	labelE(J,I,S) :- observedE(J,I,S).	

Figure 1. Excerpt from the ASP program identifying which expression of non-observed nodes (white nodes) are fixed by partial observations and rules derived from the system dynamics. The logical approach is flexible enough to model in a single framework network characteristics (products, interactions, partial information on signs of regulations and observations) and static rules about the effects of the dynamics of the system. Extensions of this framework include the exhaustive search for system repair or more constrained dynamical rules [6], [5].

weighted Markov chains whose asymptotic behavior agrees with the quantitative measures at hand. A precise study of this constrained space via local search optimization emphasizes the most important discrete events that must occur to reproduce the information at hand. These methods have been validated on the *E. coli* regulatory network benchmark. We now plan to apply these techniques to reduced networks representing the main pathways and actors automatically generated from the integrative methods developed in Axis 1. This requires to improve the range of dynamics that can be modeled by these techniques, as well as the efficiency and scalability of the local search algorithms.



Input data . Qualitative description of the system dynamics at the transcription level (interaction graph) and 3 concentration measurements of the *fis* protein (population scale).

Event-Transition Graph . Interaction frequencies required to predict the population scale behavior as the asymptotic behavior of an accumulation multiplicative law over a Markov chain. Estimation by local searches in the space of Markov chains consistent with the observed dynamics and whose asymptotic behavior is consistent with quantitative observations at the population scale. Edge thickness reflects their sensitivity in the search space.

Prediction of the *Cya* protein concentration (red curve) fits with observations. Additionally, literature evidences that high sensitivity ETG transitions correspond to key interaction in *E. Coli* response to nutritional stress.

Figure 2. Prediction of the quantitative behavior of a system using average-case analysis of dynamical systems. Identification of key interactions [2].

3.3. Grammatical inference and highly expressive structures

Our main field of expertise in **machine learning** concerns grammatical models with a long-term know-how in finite state automata learning. By introducing a similar fragment merging heuristic approach, we have proposed an algorithm that learns successfully automata modeling families of (non homologous) functional families of proteins [4], leading to a tool named Protomata-learner. As an example, this tool allows to properly model the multi-domain function of the protein family TNF, which is impossible with other existing probabilistic-based approach (see Fig. 3). Our future goal is to demonstrate the relevance of formal language theory by

addressing the question of enzyme prediction, from their genomic or protein sequences, aiming at better sensitivity and specificity. As enzyme-substrate interactions are very specific central relations for integrated genome/metabolome studies and are characterized by faint signatures, we shall rely on models for active sites involved in cellular regulation or catalysis mechanisms. This requires to build models gathering both structural and sequence information in order to describe (potentially nested or crossing) long-term dependencies such as contacts of amino-acids that are far in the sequence but close in the 3D protein folding. We wish to extend our expertise towards inferring Context-Free Grammars including the topological information coming from the structural characterization of active sites.

Moving forward to context-free grammars instead of regular patterns increases **parsing** complexity. Indeed, efficient parsing tools have been developed to identify patterns within genomes but most of them are restricted to simple regular patterns. Definite Clause Grammars (DCG), a particular form of logical context-free grammars have been used in various works to model DNA sequence features [31]. An extended formalism, String Variable Grammars (SVGs), introduces variables that can be associated to a string during a pattern search (see Fig. 4) [34], [33]. This increases the expressivity of the formalism towards midly context sensitive grammars. Thus, those grammars model not only DNA/RNA sequence features but also structural features such as repeats, palindromes, stem/loop or pseudo-knots. We have designed a tool, STAN (suffix-tree analyser) which makes it possible to search for a subset of SVG patterns in full chromosome sequences [7]. This tool was used for the recognition of transposable elements in Arabidopsis thaliana [9]. Our goal is to extend the framework of STAN. Generally, a suitable language for the search of particular components in languages has to meet several needs : expressing existing structures in a compact way, using existing databases of motifs, helping the description of interacting components. In other words, the difficulty is to find a good tradeoff between expressivity and complexity to allow the specification of realistic models at genome scale. In this direction, we are working on Logol, a language and framework based on a systematic introduction of constraints on string variables.

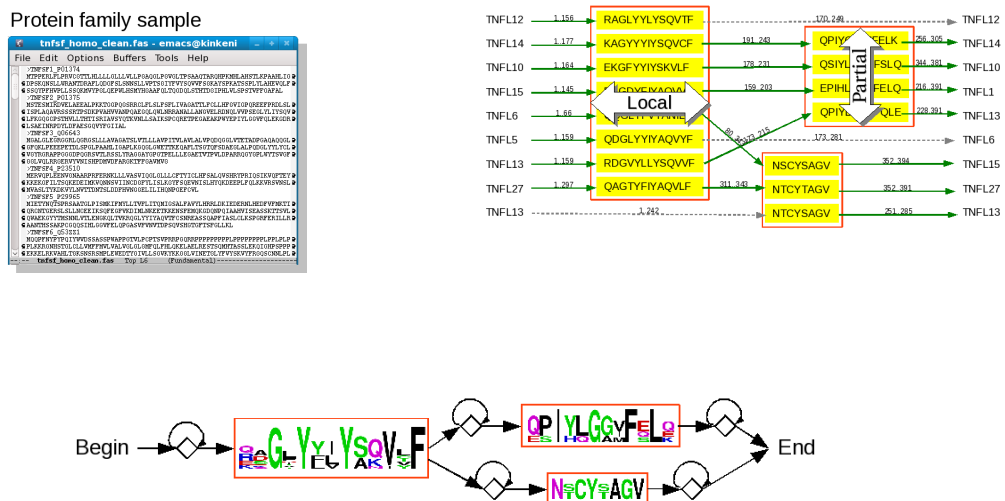


Figure 3. **Protomata Learner workflow.** Starting from a set of protein sequences (up left), a partial local alignment is computed (up right) and an automaton is inferred, which models the family and allows to search for its unknown members (down).

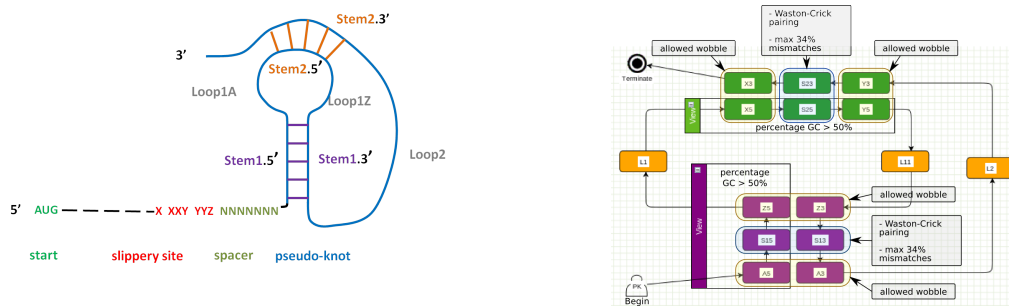


Figure 4. **Left:** A typical RNA structure: the pseudo-knot. **Right:** graphical modeling of a pseudo-knot with String Variable Grammars used in the Logol framework.

FLUMINANCE Project-Team

3. Scientific Foundations

3.1. Fluid flow analysis and modeling

Turbulent fluid flows involved in environmental or industrial applications are complex. In fluid mechanics laboratories, canonical turbulent shear flows have been studied for many years and a relatively clear picture of their underlying structure exists. However, the direct applicability of these efforts to real relevant flows, which often occur in complex geometries and in the presence of multiple non canonical influences, like cross-shear, span wise non-uniform and thermal stratification, is still unknown. In addition, the turbulence can be characterized by Reynolds number ranging between 10^3 and 10^4 , corresponding to transitional regime for which the use of classical turbulence models is limited.

In this context, we have performed research studies on turbulent shear flows of low velocities by tackling crucial topics of measurements, analysis and modeling of environmental and industrial flows in presence of non-canonical influences. This concerns more precisely the study of the interaction between a mixing layer and circular cylinder wake flow, the study of wake flow with span wise non uniformity, the study of mixing layer under the influence of thermal stratification and the study of mixing layer forced between non-uniform flows. The analysis of these flows has required the design of adequate dynamical models, using proper orthogonal decomposition and Galerkin projection. Understanding issues such as the mechanisms of heat and mass transfer involved in these shear flows provides meaningful information for the control of relevant engineering flows and the design of new technologies. To investigate more thoroughly these complex flows numerical and experimental tools have been designed. An immersed boundary method was proposed to mimic complex geometries into Direct Numerical simulation (DNS) and Large Eddy Simulations (LES) codes. A novel anemometer has been designed and implemented for the simultaneous measurement of velocity and temperature in air flows with a single hot-wire probe.

Mixing layer wake interaction

We have investigated the vortex shedding of a circular cylinder immersed in a plane turbulent mixing layer. For a centre span Reynolds number of 7500, the wake flow splits into three regions: a high-velocity wake, a low-velocity wake and a region of interaction in the middle span of the body. A strong unsteady secondary flow is observed, and explained with span wise base pressure gradients. Unexpected features are found for formation length and the base pressure along the span of the cylinder. In the high-velocity side, where the local Reynolds number is the highest, the formation length is longest. Based on the formation length measurements it was shown that as a function of the centre span Reynolds number, the wake flows behaves as circular cylinder in uniform flow. Three cells with a constant frequency with adjacent dislocations are observed. For each cell, a shedding mode was suggested. The relation of the secondary flow to the frequencies was examined. All the observations were analyzed by analogical reasoning with other flows. This pointed out the action of the secondary flow in the high-velocity side regarded as a wake interference mechanism.

Low order complex flow modeling

We have proposed improvements to the construction of low order dynamical systems (LODS) for incompressible turbulent external flows. The reduced model is obtained by means of a Proper Orthogonal Decomposition (POD) basis extracted through a truncated singular value decomposition of the flow auto-correlation matrix built from noisy PIV experimental velocity measurements. The POD modes are then used to formulate a reduced dynamical system that contains the main features of the flow. This low order dynamical system (LODS) is obtained through a Galerkin projection of the Navier-Stokes Equations on the POD basis. Usually, the resulting system of ordinary differential equations presents stability problems due to modes truncation and numerical uncertainties, especially when working on experimental data. The technique we proposed relies on an optimal control approach to estimate the dynamical system coefficients and its initial condition. This allows us to recover a reliable and stable spatio-temporal reconstruction of the large scales of the flow. The technique

has been assessed on the near wake behind a cylinder observed through very noisy PIV measurement. It has been also evaluated for configurations involving a rotating cylinder.

Studies on complex 3D dynamical behavior resulting from the interaction between a plane mixing layer and the wake of a cylinder have been also investigated using POD representation, applied to data from two synchronized 2D PIV systems (Dual-plane PIV). This approach allowed us to construct a 3D-POD representation. An analysis of the correlations shows different length scales in the regions dominated by wake like structures and shear layer type structures [2]. In order to characterize the particular organization in the plane of symmetry, a Galerkin projection from a slice POD has been performed. This led to a low-dimensional dynamical system that allowed the analysis of the relationship between the dominant frequencies. This study led to a reconstruction of the dominant periodic motion suspected from previous studies [41]. This work allowed us to make a link between the three-dimensional organization and the secondary unsteady motion from the low velocity side to the high velocity side of the mixing layer, appearing in this highly 3D flow configuration.

Direct and Large Eddy simulations of complex flows

We have proposed a direct forcing method better suited to the use of compact finite difference schemes in Direct Numerical Simulation. The new forcing creates inside the body an artificial flow preserving the no-slip condition at the surface but reducing the step-like change of the velocity derivatives across the immersed boundary. This modification led to improve results both qualitatively and quantitatively for conventional and complex flow geometries [50].

Three-dimensional direct numerical simulations have been performed for vortex shedding behind cylinders. We focused in particular on cases for which the body diameter and the incoming flow involved span wise linear non-uniformity. Four configurations were considered: the shear flow, the tapered cylinder and their combinations, which gave rise namely to the adverse and aiding cases. In contrast with the observations of other investigators, these computations highlighted distinct vortical features between the shear case and the tapered case. In addition, it was observed that the shear case and the adverse case (respectively the tapered and aiding case), yielded similarities in flow topology. This phenomenon was explained by the span wise variations of the ratio of mean velocity and the cylinder diameter which seemed to govern these flows. Indeed, it was observed that large span wise variations of U/D seemed to enhance three-dimensionality, through the appearance of vortex-adhesions and dislocations. Span wise cellular pattern of vortex shedding were identified. Their modifications in cell size, junction position and number were correlated with the variation of U/D . In the Lee side of the obstacle a wavy secondary motion was identified. Induced secondary flow due to the bending of Karman vortices in the vicinity of vortex-adhesion and dislocations was suggested to explain this result [49].

LES and experimental wake flow database

We contributed to the study of flow over a circular cylinder at Reynolds number $Re = 3900$. Although this classical flow is widely documented in the literature, especially for this precise Reynolds number, which leads to a sub critical flow regime, there is no consensus about the turbulence statistics immediately just behind the obstacle. This flow has been studied both numerically with Large Eddy Simulation and experimentally with Hot-Wire Anemometry and Particle Image Velocimetry. The numerical simulation has been performed using high-order schemes and the specific Immersed Boundary Method previously mentioned. We focused on turbulence statistics and power spectra in the near wake up to 10 diameters. Statistical estimation is shown to need large integration times increasing the computational cost and leading to an uncertainty of about 10% for most flow characteristics considered in this study. The present numerical and experimental results are found to be in good agreement with previous Large Eddy Simulation data. Our study has exhibited significant differences compared with the experimental data found in the literature. The obtained results attenuate previous numerical-experimental controversy for this type of flows [11].

Simultaneous velocity temperature measurements in turbulent flows

We have worked on the design of a novel anemometer for the simultaneous measurement of velocity and temperature in airflows with a single hot wire probe. The principle of periodically varying the overheat ratio of the wire has been selected and applied through a tunable electronic chain. Specific methods were developed for the calibration procedure and the signal processing. The accuracy of the measurements was assessed by means of Monte-Carlo simulations. Accurate results were provided for two types of turbulent non-isothermal flows, a coaxial heated jet and a low speed thermal mixing. The particular interest of the synchronization of the two measurements has been emphasized during the PhD thesis of T. Ndoye.

A new dynamic calibration technique has been developed for hot-wire probes. The technique permits, in a short time range, the combined calibration of velocity, temperature and direction calibration of single and multiple hot-wire probes. The calibration and measurements uncertainties were modeled, simulated and controlled, in order to reduce their estimated values.

3.2. Fluid motion analysis

Flow visualization has been a powerful tool to depict or to understand flow feature properties. Efforts to develop high-quality flow visualization techniques date back over a century. The analysis of the recorded images consisted firstly to a qualitative interpretation of the streak lines leading to an overall global insight into the flow properties but lacking quantitative details on important parameters such as velocity fields or turbulence intensities. Point measurement tools such as hot wire probes or Laser Doppler Velocimetry have typically provided these details. As these probes give information only at the point where they are placed, simultaneous evaluations at different points require to dispose a very large number of probes and the evaluation of unsteady field (most of the flows are unsteady) is almost unachievable with them.

In an effort to avoid the limitations of these probes, the Particle Image Velocimetry (PIV), a non-intrusive diagnostic technique, has been developed in the last two decades [40]. The PIV technique enables obtaining velocity fields by seeding the flow with particles (e.g. dye, smoke, particles) and observing the motion of these tracers. In computer vision, the estimation of the projection of the apparent motion of a 3D scene onto the image plane, refereed in the literature as optical-flow, is an intensive subject of researches since the 80's and the seminal work of B. Horn and B. Schunk [45]. Unlike to dense optical flow estimators, the former approach provides techniques that supply only sparse velocity fields. These methods have demonstrated to be robust and to provide accurate measurements for flows seeded with particles. These restrictions and their inherent discrete local nature limit too much their use and prevent any evolutions of these techniques towards the devising of methods supplying physically consistent results and small scale velocity measurements. It does not authorize also the use of scalar images exploited in numerous situations to visualize flows (image showing the diffusion of a scalar such as dye, pollutant, light index refraction, flurocein,...). At the opposite, variational techniques enable in a well-established mathematical framework to estimate spatially continuous velocity fields, which should allow more properly to go towards the measurement of smaller motion scales. As these methods are defined through PDE's systems they allow quite naturally including as constraints the kinematical and dynamical laws governing the observed fluid flows. Besides, within this framework it is also much easier to define characteristic features estimation on the basis of physically grounded data model that describes the relation linking the observed luminance function and some state variables of the observed flow. This route has demonstrated to be much more robust to scalar image. Several studies in this vein have strengthened our skills in this domain. All the following approaches have been either formulated within a statistical Markov Random Fields modeling or either within a variational framework. For a thorough description of these approaches see [7].

ICE data model and div-curl regularization This fluid motion estimator is constructed on a data model derived from the Integration of the Continuity Equation (ICE data model) [5] and includes a second order regularization scheme enabling to preserve blobs of divergence and curl. Intensive evaluations of this estimator on flow prototypes mastered in laboratory have shown that this estimator led to the same order of accuracy as the best PIV techniques but for an increase information density. This ability to get dense flow fields allowed us estimating proper vorticity or divergence maps without resorting to additional post-processing interpolation schemes.

Schlieren Image velocimetry We have addressed the problem of estimating the motion of fluid flows visualized with the Schlieren technique. Such an experimental visualization system is well known in fluid mechanics and it enables the visualization of unseeded flows. This technique authorizes the capture of phenomena that are impossible to visualize with particle seeding such as natural convection, phonation flow, breath flow and allows the setting of large scale experiments. Since the resulting images exhibit very low intensity contrasts, classical motion estimation methods based on the brightness constancy assumption (correlation-based approaches, optical flow methods) are completely inefficient. The global energy function we have defined for Schlieren images is composed of i) a specific data model accounting for the fact that the observed luminance is related to the gradient of the fluid density, and ii) a specific constrained div-curl regularization term. To date there exists no motion estimator allowing estimating accurately dense velocity fields on Schlieren images.

Low order fluid motion estimator This low-dimensional fluid motion estimator [6] is based on the Helmholtz decomposition, which consists in representing the velocity field as the sum of a divergence-free component and a curl-free one. In order to provide a low-dimensional solution, both components have been approximated using a discretization of the vorticity (curl of the velocity vector) and divergence maps through regularized Dirac measures [44]. The resulting so-called irrotational (resp. solenoidal) field is then represented by a linear combination of basis functions obtained by a convolution product of the Green kernel gradient and the vorticity map (resp. the divergence map). The coefficient values and the basis function parameters are obtained by minimizing a function formed by an integrated version of the mass conservation principle of fluid mechanics.

Potential functions estimation and finite mimetic differences We have studied a direct estimation approach of the flow potential functions (respectively the *stream* function and the *velocity* potential) from two consecutive images. The estimation has been defined on the basis of a high order regularization scheme and has been implemented through mimetic difference methods[12]. With these approaches the discretization preserves basic relationships of continuous vector analysis. Compared to previous discretization scheme based on auxiliary div-curl variables, the considered technique appeared to be numerically much more stable and led to an improve accuracy.

2D and 3D atmospheric motion layer estimation In this study, we have explored the problem of estimating mesoscales dynamics of atmospheric layers from satellite image sequences. Due to the intrinsic sparse 3-dimensional nature of clouds and to large occluded zones caused by the successive overlapping of cloud layers, the estimation of accurate layered dense motion fields is an intricate issue. Relying on a physically sound vertical decomposition of the atmosphere into layers, we have proposed two dense motion estimators for the extraction of multi-layer horizontal (2D) and 3D wind fields. These estimators are expressed as the minimization of a global function that includes a data-driven term and a spatio-temporal smoothness term. A robust data term relying on shallow-water mass conservation model has been proposed to fit sparse observations related to each layer. In the 3D case, the layers are interconnected through a term modeling mass exchanges at the layers surfaces frontiers [9].

A novel spatio-temporal regularizer derived from the shallow-water momentum conservation model has been considered to enforce temporal consistency of the solution along time. These constraints are combined with a robust second-order regularizer preserving divergent and vorticity structures of the flow. Besides, a two-level motion estimation scheme has been settled to overcome the limitations of the multiresolution incremental estimation scheme when capturing the dynamics of fine mesoscale structures. This alternative approach relies on the combination of correlation and optical-flow observations. An exhaustive evaluation of the novel method has been first performed on a scalar image sequence generated by Direct Numerical Simulation of a turbulent bi-dimensional flow. Based on qualitative experimental comparisons, the method has also been assessed on a Meteosat infrared image sequence.

3.3. Data assimilation and Tracking of characteristic fluid features

Classical motion estimation techniques usually proceed on pairs of two successive images, and do not enforce temporal consistency. This often induces an estimation drift which is essentially due to the fact that motion estimation is formulated as a local process in time. No adequate physical dynamics law, or conservation law,

related to the observed flow, is taken into account over long time intervals by the usual motion estimators. The estimation of an unknown state variable trajectory on the basis of specified dynamical laws and some incomplete and noisy measurements of the variable of interest can be either conducted through optimal control techniques or through stochastic filtering approach. These two frameworks have their own advantages and deficiencies. We rely indifferently on both approaches.

Stochastic filtering for fluid motion tracking We have proposed a recursive Bayesian filter for tracking velocity fields of fluid flows. The filter combines an Itô diffusion process associated to 2D vorticity-velocity formulation of Navier-Stokes equation and discrete image error reconstruction measurements. In contrast to usual filters, designed for visual tracking problem, our filter combines a continuous law for the description of the vorticity evolution with discrete image measurements. We resort to a Monte-Carlo approximation based on particle filtering. The designed tracker provides a robust and consistent estimation of instantaneous motion fields along the whole image sequence. In order to handle a state space of reasonable dimension for the stochastic filtering problem, the motion field is represented as a combination of adapted basis functions. The basis functions are derived from a mollification of Biot-Savart integral and a discretization of the vorticity and divergence maps of the fluid vector field. The output of such a tracking is a set of motion fields along the whole time range of the image sequence. As the time discretization is much finer than the frame rate, the method provides consistent motion interpolation between consecutive frames. In order to reduce further the dimensionality of the associated state space when we are facing a large number of motion basis functions, we have explored a new dimensional reduction approach based on dynamical systems theory. The study of the stable and unstable directions of the continuous dynamics enables to construct an adaptive dimension reduction procedure. It consists in sampling only in the unstable directions, while the stable ones are treated deterministically [6].

When the likelihood of the measurement can be modeled as Gaussian law, we have also investigated the use of so-called ensemble Kalman filtering for fluid tracking problems. This kind of filters introduced for the analysis of geophysical fluids is based on the Kalman filter update equation. Nevertheless, unlike traditional Kalman filtering setting, the covariances of the estimation errors, required to compute the Kalman gain, rely on an ensemble of forecasts. Such a process gives rise to a Monte Carlo approximation for a family of stochastic non linear filters enabling to handle state spaces of large dimension. We have recently proposed an extension of this technique that combines sequential importance sampling and the propagation law of ensemble Kalman filter. This technique leads to ensemble Kalman filter with an improved efficiency. This appears to be a generalization of the optimal importance sampling strategy we proposed in the context of partial conditional Gaussian trackers [1].

Variational assimilation technique

We investigated the use of variational framework for the tracking from image sequence of features belonging to high dimensional spaces. This framework relies on optimal control principles as developed in environmental sciences to analyze geophysical flows [46], [47]. Within the PhD of Nicolas Papadakis [10], we have first devised a data assimilation technique for the tracking of closed curves and their associated motion fields. The proposed approach enables a continuous tracking along an image sequence of both a deformable curve and its associated velocity field. Such an approach has been formalized through the minimization of a global spatio-temporal continuous cost functional, with respect to a set of variables representing the curve and its related motion field. The resulting minimization sequence consists in a forward integration of an evolution law followed by a backward integration of an adjoint evolution model. The latter pde includes a term related to the discrepancy between the state variables evolution law and discrete noisy measurements of the system. The closed curves are represented through implicit surface modeling [48], whereas the motion is described either by a vector field or through vorticity and divergence maps according to the type of targeted application. The efficiency of the approach has been demonstrated on two types of image sequences showing deformable objects and fluid motions.

More recently assimilation technique for the direct estimation of atmospheric wind field from pressure images have been proposed [4]. These techniques rely on a brightness variation model of the intensity function. They do not include anymore motion measurements provided by external motion estimators. The resulting estimator

allows us to recover accurate fluid motion fields and enables tracking dense vorticity maps along an image sequence.

3.4. Visual servoing

Nowadays, visual servoing is a widely used technique in robot control. It consists in using data provided by a vision sensor for controlling the motions of a robot [43]. Various sensors can be considered such as perspective cameras, omnidirectional cameras, 2D ultrasound probes or even virtual cameras. In fact, this technique is historically embedded in the larger domain of sensor-based control [51] so that other sensors than vision sensors can be properly used. On the other hand, this approach was first dedicated to robot arms control. Today, much more complex systems can be considered like humanoid robots, cars, submarines, airships, helicopters, aircrafts. Therefore, visual servoing is now seen as a powerful approach to control the state of dynamic systems.

Classically, to achieve a visual servoing task, a set of visual features \mathbf{s} has to be selected from visual measurements \mathbf{m} extracted from the image. A control law is then designed so that these visual features reach a desired value \mathbf{s}^* related to the desired state of the system. The control principle is thus to regulate to zero the error vector $\mathbf{e} = \mathbf{s} - \mathbf{s}^*$. To build the control law, the knowledge of the so-called *interaction matrix* \mathbf{L}_s is usually required. This matrix links the time variation of \mathbf{s} to the camera instantaneous velocity \mathbf{v}

$$\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{v} + \frac{\partial \mathbf{s}}{\partial t} \quad (35)$$

where the term $\frac{\partial \mathbf{s}}{\partial t}$ describes the non-stationary behavior of \mathbf{s} . Typically, if we try to ensure an exponential decoupled decrease of the error signal and if we consider the camera velocity as the input of the robot controller, the control law writes as follow

$$\mathbf{v} = -\lambda \widehat{\mathbf{L}}_s^+ \mathbf{e} - \widehat{\mathbf{L}}_s^+ \frac{\partial \mathbf{e}}{\partial t} \quad (36)$$

with λ a proportional gain that has to be tuned to minimize the time-to-convergence, $\widehat{\mathbf{L}}_s^+$ the pseudo-inverse of a model or an approximation of \mathbf{L}_s and $\widehat{\frac{\partial \mathbf{e}}{\partial t}}$ an estimation of $\frac{\partial \mathbf{e}}{\partial t}$.

The behavior of the closed-loop system is then obtained, from (2), by expressing the time variation of the error \mathbf{e}

$$\dot{\mathbf{e}} = -\lambda \mathbf{L}_s \widehat{\mathbf{L}}_s^+ \mathbf{e} - \mathbf{L}_s \widehat{\mathbf{L}}_s^+ \frac{\partial \mathbf{e}}{\partial t} + \frac{\partial \mathbf{e}}{\partial t}. \quad (37)$$

As can be seen, visual servoing explicitly relies on the choice of the visual features \mathbf{s} and then on the related interaction matrix; that is the key point of this approach. Indeed, this choice must be performed very carefully. Especially, an isomorphism between the camera pose and the visual features is required to ensure that the convergence of the control law will lead to the desired state of the system. An optimal choice would result in finding visual features leading to a diagonal and constant interaction matrix and, consequently, to a linear decoupled system for which the control problem is well known. Thereafter, the isomorphism as well as the global stability would be guaranteed. In addition, since the interaction matrix would present no more nonlinearities, a suitable robot trajectory would be ensured.

However, finding such visual features is a very complex problem and it is still an open issue. Basically, this problem consists in building the visual features \mathbf{s} from the nonlinear visual measurements \mathbf{m} so that the interaction matrix related to \mathbf{s} becomes diagonal and constant or, at least, as simple as possible.

On the other hand, a robust extraction, matching (between the initial and desired measurements) and real-time spatio-temporal tracking (between successive measurements) have to be ensured but have proved to be a complex task, as testified by the abundant literature on the subject. Nevertheless, this image process is, to date, a necessary step and often considered as one of the bottlenecks of the expansion of visual servoing. That is why more and more non-geometric visual measurements are proposed [3].

3.5. Sparse Representations and Bayesian model selection

Sparse representation methods aim at finding representations of a signal with a small number of components taken from an over-complete dictionary of elementary functions or vectors. Sparse representations are of interest in a number of applications in Physics and signal processing. In particular, they provide a simple characterization of certain families of signals encountered in practice. For example, smooth signals can be shown to have a sparse representation in over-complete Fourier or wavelet dictionaries. More recently, it has been emphasized in [42] that the solutions of certain differential equations (*e.g.*, diffusion or transport equation) have a sparse representation in dictionaries made up of curvelets.

Finding the sparse representation of a signal typically requires to solve an under-determined system of equations under the constraint that the solution is composed of the minimum number of non-zero elements. Unfortunately, this problem is known to be NP-hard and sub-optimal procedures have to be devised to find practical solutions. Among the various algorithms that find approximate solutions, let us mention for example the matching pursuit, orthogonal matching pursuit or basis pursuit algorithms.

Choosing appropriate models and fixing hyper-parameters is a tricky and often hidden process in optic-flow estimation. Most of the motion estimators proposed so far have generally to rely on successive trials and an empirical strategy for fixing the hyper-parameters values and choosing the adequate model. Besides of its computational inefficiency, this strategy may produce catastrophic estimates without any relevant feedback for the end-user, especially when motions are difficult to apprehend as for instance for complex deformations or non-conventional imagery. Imposing hard values to these parameters may also yield poor results when the lighting conditions or the underlying motions differ from those the system has been calibrated with. At the extreme, the estimate may be either too smooth or at the opposite non-existent strong motion discontinuities.

Bayesian model selection offers an attractive solution to this problem. The Bayesian paradigm implicitly requires the definition of several competing observation and prior *probabilistic* model(s). The observation model relates the motion of the physical system to the spatial and temporal variations of the image intensity. The prior models define the spatio-temporal constraints that the motion has to satisfy. Considering these competing models, the Bayesian theory provides methodologies to select the best models under objective performance criterion (minimum probability of error, minimum mean square error, etc). Moreover, due to the generality of this problem, numerous algorithms and approximations exist in the literature to implement efficient and effective practical solutions: Monte-Carlo integration, mean-field and Laplace approximations, EM algorithm, graphical models, etc.

GENSCALE Team

3. Scientific Foundations

3.1. Introduction

To tackle challenges brought by the processing of huge amount of genomic data, the main strategy of GenScale is to merge the following computer science expertise:

- Data structure;
- Combinatorial optimization;
- Parallelism.

3.2. Data structure

To face the genomic data tsunami, the design of efficient algorithms involves the optimization of memory fingerprints. A key point is the design of innovative data structures to represent large genomic datasets into computer memories. Today's limitations come from their size, their construction time, or their centralized (sequential) access. Random accesses to large data structures poorly exploit the sophisticated processor cache memory system. New data structures including compression techniques, probabilistic filters, approximate string matching, or techniques to improve spatial/temporal memory access are developed [3].

3.3. Combinatorial optimization

For wide genome analysis, Next Generation Sequencing (NGS) data processing or protein structure applications, the main issue concerns the exploration of sets of data by time-consuming algorithms, with the aim of identifying solutions that are optimal in a predefined sense. In this context, speeding up such algorithms requires acting on many directions: (1) optimizing the search with efficient heuristics and advanced combinatorial optimization techniques [2], [5] or (2) targeting biological sub-problems to reduce the search space [7], [9]. Designing algorithms with adapted heuristics, and able to scale from protein (a few hundreds of amino acids) to full genome (millions to billions of nucleotides) is one of the competitive challenges addressed in the GenScale project.

3.4. Parallelism

The traditional parallelization approach, which consists in moving from a sequential to a parallel code, must be transformed into a direct design and implementation of high performance parallel software. All levels of parallelism (vector instructions, multi-cores, many-cores, clusters, grid, clouds) need to be exploited in order to extract the maximum computing power from current hardware resources [6], [8], [1]. An important specificity of GenScale is to systematically adopt a design approach where all levels of parallelism are potentially considered.

SAGE Project-Team

3. Scientific Foundations

3.1. Numerical algorithms and high performance computing

Linear algebra is at the kernel of most scientific applications, in particular in physical or chemical engineering. For example, steady-state flow simulations in porous media are discretized in space and lead to a large sparse linear system. The target size is 10^7 in 2D and 10^{10} in 3D. For transient models such as diffusion, the objective is to solve about 10^4 linear systems for each simulation. Memory requirements are of the order of Giga-bytes in 2D and Tera-bytes in 3D. CPU times are of the order of several hours to several days. Several methods and solvers exist for large sparse linear systems. They can be divided into three classes: direct, iterative or semi-iterative. Direct methods are highly efficient but require a large memory space and a rapidly increasing computational time. Iterative methods of Krylov type require less memory but need a scalable preconditioner to remain competitive. Iterative methods of multigrid type are efficient and scalable, used by themselves or as preconditioners, with a linear complexity for elliptic or parabolic problems but they are not so efficient for hyperbolic problems. Semi-iterative methods such as subdomain methods are hybrid direct/iterative methods which can be good tradeoffs. The convergence of iterative and semi-iterative methods and the accuracy of the results depend on the condition number which can blow up at large scale. The objectives are to analyze the complexity of these different methods, to accelerate convergence of iterative methods, to measure and improve the efficiency on parallel architectures, to define criteria of choice.

In geophysics, a main concern is to solve inverse problems in order to fit the measured data with the model. Generally, this amounts to solve a linear or nonlinear least-squares problem. Complex models are in general coupled multi-physics models. For example, reactive transport couples advection-diffusion with chemistry. Here, the mathematical model is a set of nonlinear Partial Differential Algebraic Equations. At each timestep of an implicit scheme, a large nonlinear system of equations arise. The challenge is to solve efficiently and accurately these large nonlinear systems.

Approximation in Krylov subspace is in the core of the team activity since it provides efficient iterative solvers for linear systems and eigenvalue problems as well. The later are encountered in many fields and they include the singular value problem which is especially useful when solving ill posed inverse problems.

3.2. Numerical models applied to hydrogeology and physics

The team Sage is strongly involved in numerical models for hydrogeology and physics. There are many scientific challenges in the area of groundwater simulations. This interdisciplinary research is very fruitful with cross-fertilizing subjects. For example, high performance simulations were very helpful for finding out the asymptotic behaviour of the plume of solute transported by advection-dispersion. Numerical models are necessary to understand flow transfer in fractured media.

The team develops stochastic models for groundware simulations. Numerical models must then include Uncertainty Quantification methods, spatial and time discretization. Then, the discrete problems must be solved with efficient algorithms. The team develops parallel algorithms for complex numerical simulations and conducts performance analysis. Another challenge is to run multiparametric simulations. They can be multiple samples of a non intrusive Uncertainty Quantification method, or multiple samples of a stochastic method for inverse problems, or multiple samples for studying the sensitivity to a given model parameter. Thus these simulations are more or less independent and are well-suited to grid computing but each simulation requires powerful CPU and memory resources.

A strong commitment of the team is to develop the scientific software platform H2OLab for numerical simulations in heterogeneous hydrogeology.

SERPICO Team

3. Scientific Foundations

3.1. Glossary

- WF** Optical Wide-Field microscopy.
- SDC** (Spinning-Disk Confocal microscopy): illumination of the sample with a rotating pattern of several hundred of pinholes for complete simultaneous confocal illumination.
- FLIM** (Fluorescence Lifetime Microscopy Imaging): imaging of fluorescent molecule lifetimes.
- PALM** (Photo-Activated Localization Microscopy): high-resolution microscopy using stochastic photo-activation of fluorophores and adjustment of point spread functions [20].
- SIM** (Structured Illumination Microscopy): high-resolution light microscopy using structured patterns and interference analysis [30].
- TIRF** (Total Internal Reflectance): 2D optical microscopy using evanescent waves and total reflectance [19].
- Cryo-EM** (Cryo-Electron Tomography): 3D representation of sub-cellular and molecular objects of 5-20 nanometres, frozen at very low temperatures, from 2D projections using a transmission electron microscope.

3.2. Image restoration for high-resolution microscopy

In order to produce images compatible with the dynamic processes in living cells as seen in video-microscopy, we study the potential of non-local neighborhood filters and image denoising algorithms (e.g. ND-SAFIR software) [6], [2], [7], [4]. The major advantage of these approaches is to acquire images at very low SNR while recovering denoised 2D+T(ime) and 3D+T(ime) images [1]. Such post-acquisition processing can improve the rate of image acquisition by a factor of 100 to 1000 times [5], reducing the sensitivity threshold and allowing imaging for long time regime without cytotoxic effect and photodamages. This approach has been successfully applied to WF, SDC [1], TIRF [19], fast live imaging and 3D-PALM using the OMX system in collaboration with J. Sedat and M. Gustafsson at UCSF [5]. The ND-SAFIR software (see Section 5.1) has been licensed to a large set of laboratories over the world (see Figure 2). New information restoration and image denoising methods are currently investigated to make SIM imaging compatible with the imaging of molecular dynamics in live cells. Unlike other optical sub-diffraction limited techniques (e.g. STED [32], PALM [20]) SIM has the strong advantage of versatility when considering the photo-physical properties of the fluorescent probes [30]. Such developments are also required to be compatible with “high-throughput microscopy” since several hundreds of cells are observed at the same time and the exposure times are typically reduced.

3.3. Dynamic analysis and trajectory computation

3.3.1. Motion analysis and tracking

In time-lapse microscopy, the challenge is to detect and track moving objects. Classical tracking methods have limitations as the number of objects and clutter increase. It is necessary to correctly associate measurements with tracked objects, i.e. to solve the difficult data association problem [37]. Data association even combined with sophisticated particle filtering techniques [40] or matching techniques [38] is problematic when tracking several hundreds of similar objects with variable velocities. Developing new optical flow and tracking methods and models in this area is then very stimulating since the problems we have to solve are really challenging and new for applied mathematics. The goal is to formulate the problem of optical flow estimations in ways that take physical causes of brightness violations into account [26], [31]. In addition, the interpretation of computed flow fields enables to provide spatio-temporal signatures of particular dynamic processes and could help to complete the traffic modelling.

3.3.2. Event detection

Several approaches can be considered for the automatic detection of appearing and vanishing particles (or spots) in WF and TIRF microscopy images. The difficulty is to distinguish motions due to trafficking from the appearing and vanishing spots. Ideally this could be performed by tracking all the vesicles contained in the cell [40], [29]. Among the methods proposed to detect particles in microscopy images [43], [39], none is dedicated to the detection of a small number of particles appearing or disappearing suddenly between two time steps. Our way of handling small blob appearances/disappearances originates from the observation that two successive images are redundant and that occlusions correspond to blobs in one image which cannot be reconstructed from the other image [1] (see also [24]).

3.4. Computational simulation and modelling of membrane transport

Mathematical biology is a field in expansion, which has evolved into various branches and paradigms to address problems at various scales ranging from ecology to molecular structures. Nowadays, system biology [33], [45] aims at modelling systems as a whole in an integrative perspective instead of focusing on independent biophysical processes. One of the goals of these approaches is the cell in silico as investigated at Harvard Medical School (<http://vcp.med.harvard.edu/>) or the VCell of the University of Connecticut Health Center (<http://www.nrcam.uchc.edu/>). Previous simulation-based methods have been investigated to explain the spatial organization of microtubules [35] but the method is not integrative and a single scale is used to describe the visual patterns. In this line of work, we propose several contributions to combine imaging, traffic and membrane transport modelling in cell biology.

In this area, we focus on the analysis of transport intermediates (vesicles) that deliver cellular components to appropriate places within cells. We have already investigated the concept of Network Tomography (NT) [44] mainly developed for internet traffic estimation. The idea is to determine mean traffic intensities based on statistics accumulated over a period of time. The measurements are usually the number of vesicles detected at each destination region receiver. The NT concept has been investigated also for simulation [3] since it can be used to statistically mimic the contents of real traffic image sequences. In the future, we plan to incorporate more prior knowledge on dynamics to improve representation. An important challenge will be to correlate stochastic and dynamical 1D and in silico models studied at the nano-scale in biophysics, to 3D images acquired in vivo at the scale of few hundred nanometres. A difficulty is related to the scale change and statistical aggregation problems (in time and space).

VISAGES Project-Team

3. Scientific Foundations

3.1. Scientific Foundations

The scientific foundations of our team concern the development of new processing algorithms in the field of medical image computing : image fusion (registration and visualization), image segmentation and analysis, management of image related information. Since this is a very large domain, which can endorse numerous types of application; for seek of efficiency, the purpose of our methodological work primarily focuses on clinical aspects and for the most part on head and neck related diseases. In addition, we emphasize our research efforts on the neuroimaging domain. Concerning the scientific foundations, we have pushed our research efforts:

- In the field of image fusion and image registration (rigid and deformable transformations) with a special emphasis on new challenging registration issues, especially when statistical approaches based on joint histogram cannot be used or when the registration stage has to cope with loss or appearance of material (like in surgery or in tumour imaging for instance).
- In the field of image analysis and statistical modelling with a new focus on image feature and group analysis problems. A special attention was also to develop advanced frameworks for the construction of atlases and for automatic and supervised labelling of brain structures.
- In the field of image segmentation and structure recognition, with a special emphasis on the difficult problems of *i*) image restoration for new imaging sequences (new Magnetic Resonance Imaging protocols, 3D ultrasound sequences...), and *ii*) structure segmentation and labelling based on shape, multimodal and statistical information.
- Following the Neurobase national project where we had a leading role, we wanted to enhance the development of distributed and heterogeneous medical image processing systems.

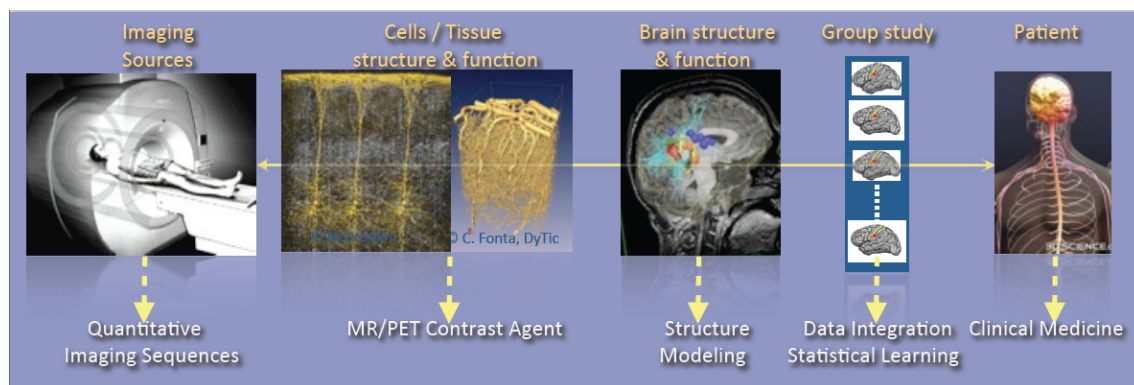


Figure 1. The major overall scientific foundation of the team concerns the integration of data from the Imaging source to the patient at different scales : from the cellular or molecular level describing the structure and function, to the functional and structural level of brain structures and regions, to the population level for the modelling of group patterns and the learning of group or individual imaging markers

As shown in figure 1, research activities of the VISAGES U746 team are tightly coupling observations and models through integration of clinical and multi-scale data, phenotypes (cellular, molecular or structural patterns). We work on personalized models of central nervous system organs and pathologies, and intend to confront these models to clinical investigation studies for quantitative diagnosis, prevention of diseases, therapy planning and validation. This approaches developed in a translational framework where the data integration process to build the models inherits from specific clinical studies, and where the models are assessed on prospective clinical trials for diagnosis and therapy planning. All of this research activity is conducted in tight links with the **Neurinfo** imaging platform environments and the engineering staff of the platform. In this context, some of our major challenges in this domain concern:

- The elaboration of new descriptors to study the brain structure and function (e.g. variation of brain perfusion with and without contrast agent, evolution in shape and size of an anatomical structure in relation with normal, pathological or functional patterns, computation of asymmetries from shapes and volumes).
- The integration of additional spatio-temporal imaging sequences covering a larger range of observation, from the molecular level to the organ through the cell (Arterial Spin Labeling, diffusion MRI, MR relaxometry, MR cell labeling imaging, PET molecular imaging, ...). This includes the elaboration of new image descriptors coming from spatio-temporal quantitative or contrast-enhanced MRI.
- The creation of computational models through data fusion of molecular, cellular, structural and functional image descriptors from group studies of normal and/or pathological subjects.
- The evaluation of these models on acute pathologies especially for the study of degenerative, psychiatric or developmental brain diseases (e.g. Multiple Sclerosis, Epilepsy, Parkinson, Dementia, Strokes, Depression, Schizophrenia, ...) in a translational framework.

In terms of methodological developments, we are particularly working on statistical methods for multidimensional image analysis, and feature selection and discovery, which includes:

- The development of specific shape and appearance models, construction of atlases better adapted to a patient or a group of patients in order to better characterize the pathology;
- The development of advanced segmentation and modeling methods dealing with longitudinal and multidimensional data (vector or tensor fields), especially with the integration of new prior models to control the integration of multiscale data and aggregation of models;
- The development of new models and probabilistic methods to create water diffusion maps from MRI;
- The integration of machine learning procedures for classification and labeling of multidimensional features (from scalar to tensor fields and/or geometric features): pattern and rule inference and knowledge extraction are key techniques to help in the elaboration of knowledge in the complex domains we address;
- The development of new dimensionality reduction techniques for problems with massive data, which includes dictionary learning for sparse model discovery. Efficient techniques have still to be developed to properly extract from a raw mass of images derived data that are easier to analyze.

ACES Project-Team

3. Scientific Foundations

3.1. Programming Context

The goal of ambient computing is to seamlessly merge virtual and real environments. A real environment is composed of objects from the physical world, e.g., people, places, machines. A virtual environment is any information system, e.g., the Web. The integration of these environments must permit people and their information systems to implicitly interact with their surrounding environment.

Ambient computing applications are able to evaluate the state of the real world through sensing technologies. This information can include the position of a person (caught with a localization system like GPS), the weather (captured using specialized sensors), etc. Sensing technologies enable applications to automatically update digital information about events or entities in the physical world. Further, interfaces can be used to act on the physical world based on information processed in the digital environment. For example, the windows of a car can be automatically closed when it is raining.

This real-world and virtual-world integration must permit people to implicitly interact with their surrounding environment. This means that manual device manipulation must be minimal since this constrains person mobility. In any case, the relative small size of personal devices can make them awkward to manipulate. In the near future, interaction must be possible without people being aware of the presence of neighbouring processors.

Information systems require tools to *capture* data in its physical environment, and then to *interpret*, or process, this data. A context denotes all information that is pertinent to a person-centric application. There are three classes of context information:

- The *digital context* defines all parameters related to the hardware and software configuration of the device. Examples include the presence (or absence) of a network, the available bandwidth, the connected peripherals (printer, screen), storage capacity, CPU power, available executables, etc.
- The *personal context* defines all parameters related to the identity, preferences and location of the person who owns the device. This context is important for deciding the type of information that a personal device needs to acquire at any given moment.
- The *physical context* relates to the person's environment; this includes climatic condition, noise level, luminosity, as well as date and time.

All three forms of context are fundamental to person-centric computing. Consider for instance a virtual museum guide service that is offered via a PDA. Each visitor has his own PDA that permits him to receive and visualise information about surrounding artworks. In this application, the *pertinent* context of the person is made up of the artworks situated near the person, the artworks that interest him as well as the degree of specialisation of the information, i.e., if the person is an art expert, he will desire more detail than the occasional museum visitor.

There are two approaches to organising data in a real to virtual world mapping: a so-called *logical* approach and a *physical* approach. The logical approach is the traditional way, and involves storing all data relevant to the physical world on a service platform such as a centralised database. Context information is sent to a person in response to a request containing the person's location co-ordinates and preferences. In the example of the virtual museum guide, a person's device transmits its location to the server, which replies with descriptions of neighbouring artworks.

The main drawbacks of this approach are scalability and complexity. Scalability is a problem since we are evolving towards a world with billions of embedded devices; complexity is a problem since the majority of physical objects are unrelated, and no management body can cater for the integration of their data into a service platform. Further, the model of the physical world must be up to date, so the more dynamic a system, the more updates are needed. The services platform quickly becomes a potential bottleneck if it must deliver services to all people.

The physical approach does not rely on a digital model of the physical world. The service is computed wherever the person is located. This is done by spreading data onto the devices in the physical environment; there are a sufficient number of embedded systems with wireless transceivers around to support this approach. Each device manages and stores the data of its associated object. In this way, data are physically linked to objects, and there is no need to update a positional database when physical objects move since the data *physically* moves with them.

With the physical approach, computations are done on the personal and available embedded devices. Devices interact when they are within communication range. The interactions constitute delivery of service to the person. Returning to the museum example, data is directly embedded in a painting's frame. When the visitor's guide meets (connects) to a painting's devices, it receives the information about the painting and displays it.

3.2. Spatial Information Systems

One of the major research efforts in ACES over the last few years has been the definition of the Spread programming model to cater for spacial context. The model is derived from the Linda [10] tuple-space model. Each information item is a *tuple*, which is a sequence of typed data items. For example, $\langle 10, 'Peter', -3.14 \rangle$ is a tuple where the first element is the integer 10, the second is the string "Peter" and the third is the real value -3.14. Information is addressed using patterns that match one or a set of tuples present in the tuple-space. An example pattern that matches the previous tuple is $\langle \text{int}, 'Peter', \text{float} \rangle$. The tuple-space model has the advantage of allowing devices that meet for the first time to exchange data since there is no notion of names or addresses.

Data items are not only addressed by their type, but also by the physical space in which they reside. The size of the space is determined by the strength of the radio signal of the device. The important difference between Spread and other tuple-space systems (e.g., Sun's JavaSpaces [9], IBM's T-Space [13]) is that when a program issues a matching request, only the tuples filling the *physical space* of the requesting program are tested for matching. Thus, though SIS (Spatial Information Systems) applications are highly distributed by nature, they only rely on localised communications; they do not require access to a global communication infrastructure. Figure 1 shows an example of a physical tuple space, made of tuples arranged in the space and occupying different spaces.

As an example of the power of this model, consider two of the applications that we have developed using it.

- *Ubi-bus* is a spatial information application whose role is to help blind and partially blind people use public transport. When taking a bus, a blind person uses his PDA to signal his intention to a device embedded in the bus stop; this device then contacts the bus on the person's behalf. This application illustrates how data is distributed over the objects of the physical world, and generally, how devices complement human means of communication.
- *Ubi-board* is a spatial information application designed for public electronic billboards. Travel hotspots like airports and major train stations have an international customer base, so bill-board announcements need to be made in several languages. In Ubi-bus, a billboard has an embedded device. When a person comes within communication range of the billboard, his device sends a request to the billboard asking it to print the message in the language of the person. In the case where several travellers are in proximity of the billboard, the board sends a translation of its information message to each person. The Ubi-board application illustrates personal context in use, i.e., the choice of natural language, and also how actions can be provoked in the physical world without explicit intervention by the person.

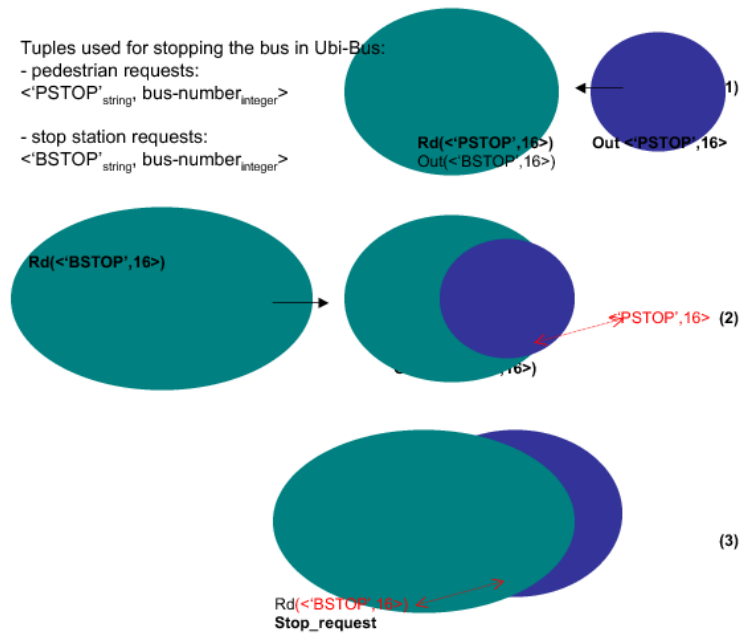


Figure 1. Physical Tuple Space

3.3. Coupled objects

Integrity checking is an important concern in many activities, both in the real world and in the information society. The basic purpose is to verify that a set of objects, parts, components, people remains the same along some activity or process, or remains consistent against a given property (such as a part count).

In the real world, it is a common step in logistic: objects to be transported are usually checked by the sender (for their conformance to the recipient expectation), and at arrival by the recipient. When a school get a group of children to a museum, people responsible for the children will regularly check that no one is missing. Yet another common example is to check for our personal belongings when leaving a place, to avoid lost. While important, these verification are tedious, vulnerable to human errors, and often forgotten.

Because of these vulnerabilities, problems arise: E-commerce clients sometimes receive incomplete packages, valuable and important objects (notebook computers, passports etc.) get lost in airports, planes, trains, hotels, etc. with sometimes dramatic consequences.

While there are very few automatic solutions to improve the situation in the real world, integrity checking in the computing world is a basic and widely used mechanism: magnetic and optical storage devices, network communications are all using checksums and error checking code to detect information corruption, to name a few.

The emergence of ubiquitous computing and the rapid penetration of RFID devices enable similar integrity checking solutions to work for physical objects. We introduced the concept of *coupled object*, which offers simple yet powerful mechanisms to check and ensure integrity properties for set of physical objects.

Essentially, coupled objects are a set of physical objects which defines a logical group. An important feature is that the group information is self contained on the objects which allow to verify group properties, such as completeness, only with the objects. Said it another way, the physical objects can be seen as fragments of

a composite object. A trivial example could be a group made of a person, his jacket, his mobile phone, his passport and his cardholder.

The important feature of the concept are its distributed, autonomous and anonymous nature: it allows the design and implementation of pervasive security applications without any database tracking or centralized information system support. This is a significant advantage of this approach given the strong privacy issues that affect pervasive computing.

ASAP Project-Team

3. Scientific Foundations

3.1. Distributed Computing

Distributed computing was born in the late seventies when people started taking into account the intrinsic characteristics of physically distributed systems. The field then emerged as a specialized research area distinct from networks, operating systems and parallelism. Its birth certificate is usually considered as the publication in 1978 of Lamport's most celebrated paper "*Time, clocks and the ordering of events in a distributed system*" [56] (that paper was awarded the Dijkstra Prize in 2000). Since then, several high-level journals and (mainly ACM and IEEE) conferences have been devoted to distributed computing. The distributed systems area has continuously been evolving, following the progresses of all the above-mentioned areas such as networks, computing architecture, operating systems.

The last decade has witnessed significant changes in the area of distributed computing. This has been acknowledged by the creation of several conferences such as NSDI and IEEE P2P. The NSDI conference is an attempt to reassemble the networking and system communities while the IEEE P2P conference was created to be a forum specialized in peer-to-peer systems. At the same time, the EuroSys conference originated as an initiative of the European Chapter of the ACM SIGOPS to gather the system community in Europe.

3.2. Theory of distributed systems

Finding models for distributed computations prone to asynchrony and failures has received a lot of attention. A lot of research in this domain focuses on what can be computed in such models, and, when a problem can be solved, what are its best solutions in terms of relevant cost criteria. An important part of that research is focused on distributed computability: what can be computed when failure detectors are combined with conditions on process input values for example. Another part is devoted to model equivalence. What can be computed with a given class of failure detectors? Which synchronization primitives is a given failure class equivalent to? These are among the main topics addressed in the leading distributed computing community. A second fundamental issue related to distributed models, is the definition of appropriate models suited to dynamic systems. Up to now, the researchers in that area consider that nodes can enter and leave the system, but do not provide a simple characterization, based on properties of computation instead of description of possible behaviors [57], [50], [51]. This shows that finding dynamic distributed computing models is today a "Holy Grail", whose discovery would allow a better understanding of the essential nature of dynamic systems.

3.3. Peer-to-peer overlay networks

A standard distributed system today is related to thousand or even millions of computing entities scattered all over the world and dealing with a huge amount of data. This major shift in scalability requirements has led to the emergence of novel computing paradigms. In particular, the peer-to-peer communication paradigm imposed itself as the prevalent model to cope with the requirements of large scale distributed systems. Peer-to-peer systems rely on a symmetric communication model where peers are potentially both clients and servers. They are fully decentralized, thus avoiding the bottleneck imposed by the presence of servers in traditional systems. They are highly resilient to peers arrivals and departures. Finally, individual peer behavior is based on a local knowledge of the system and yet the system converges toward global properties.

A peer-to-peer overlay network logically connects peers on top of IP. Two main classes of such overlays dominate, structured and unstructured. The differences relate to the choice of the neighbors in the overlay, and the presence of an underlying naming structure. Overlay networks represent the main approach to build large-scale distributed systems that we retained. An overlay network forms a logical structure connecting participating entities on top of the physical network, be it IP or a wireless network. Such an overlay might form a structured overlay network [58], [59], [60] following a specific topology or an unstructured network [55], [61] where participating entities are connected in a random or pseudo-random fashion. In between, lie weakly structured peer-to-peer overlays where nodes are linked depending on a proximity measure providing more flexibility than structured overlays and better performance than fully unstructured ones. Proximity-aware overlays connect participating entities so that they are connected to close neighbors according to a given proximity metric reflecting some degree of affinity (computation, interest, etc.) between peers. We extensively use this approach to provide algorithmic foundations of large-scale dynamic systems.

3.4. Epidemic protocols

Epidemic algorithms, also called gossip-based algorithms [53], [52], constitute a fundamental topic in our research. In the context of distributed systems, epidemic protocols are mainly used to create overlay networks and to ensure a reliable information dissemination in a large-scale distributed system. The principle underlying the technique, in analogy with the spread of a rumor among humans via gossiping, is that participating entities continuously exchange information about the system in order to spread it gradually and reliably. Epidemic algorithms have proved efficient to build and maintain large-scale distributed systems in the context of many applications such as broadcasting [52], monitoring, resource management, search, and more generally in building unstructured peer-to-peer networks.

3.5. Malicious process behaviors

When assuming that processes fail by simply crashing, bounds on resiliency (maximum number of processes that may crash), number of exchanged messages, number of communication steps, etc. either in synchronous and augmented asynchronous systems (recall that in purely asynchronous systems some problems are impossible to solve) are known. If processes can exhibit malicious behaviors, these bounds are seldom the same. Sometimes, it is even necessary to change the specification of the problem. For example, the consensus problem for correct processes does not make sense if some processes can exhibit a Byzantine behavior and thus propose arbitrary value. In this case, the validity property of consensus, which is normally "a decided value is a proposed value", must be changed to "if all correct processes propose the same value then only this value can be decided". Moreover, the resilience bound of less than half of faulty processes is at least lowered to "less than a third of Byzantine processes". These are some of the aspects that underlie our studies in the context of the classical model of distributed systems, in peer-to-peer systems and in sensor networks.

3.6. Online Social Networks

Social Networks have rapidly become a fundamental component of today's distributed applications. Web 2.0 applications have dramatically changed the way users interact with the Internet and with each other. The number of users of websites like Flickr, Delicious, Facebook, or MySpace is constantly growing, leading to significant technical challenges. On the one hand, these websites are called to handle enormous amounts of data. On the other hand, news continue to report the emergence of privacy threats to the personal data of social-network users. Our research aims to exploit our expertise in distributed systems to lead to a new generation of scalable, privacy-preserving, social applications.

ASCOLA Project-Team

3. Scientific Foundations

3.1. Overview

Since we mainly work on new software structuring concepts and programming language design, we first briefly introduce some basic notions and problems of software components (understood in a broad sense, i.e., including modules, objects, architecture description languages and services), aspects, and domain-specific languages. We conclude by presenting the main issues related to distribution and concurrency that are relevant to our work.

3.2. Software Components

Modules and services. The idea that building *software components*, i.e., composable prefabricated and parametrized software parts, was key to create an effective software industry was realized very early [67]. At that time, the scope of a component was limited to a single procedure. In the seventies, the growing complexity of software made it necessary to consider a new level of structuring and programming and led to the notions of information hiding, *modules*, and module interconnection languages [76], [48]. Information hiding promotes a black-box model of program development whereby a module implementation, basically a collection of procedures, is strongly encapsulated behind an interface. This makes it possible to guarantee logical invariant *properties* of the data managed by the procedures and, more generally, makes *modular reasoning* possible. In a first step, it is possible to reason locally, about the consistency between the module implementation and the module interface. In a second step, it is possible to reason about composing modules by only considering their interfaces. Modern module systems also consider types as module elements and consider, typically static, modules as a unit of separate compilation, with the most sophisticated ones also supporting modules parametrized by modules [65].

In the context of today's Internet-based information society, components and modules have given rise to *software services* whose compositions are governed by explicit *orchestration or choreography* specifications that support notions of global properties of a service-oriented architecture. These horizontal compositions have, however, to be frequently adapted dynamically. Dynamic adaptations, in particular in the context of software evolution processes, often conflict with a black-box composition model either because of the need for invasive modifications, for instance, in order to optimize resource utilization or modifications to the vertical compositions implementing the high-level services.

Object-Oriented Programming. *Classes* and *objects* provide another kind of software component, which makes it necessary to distinguish between *component types* (classes) and *component instances* (objects). Indeed, unlike modules, objects can be created dynamically. Although it is also possible to talk about classes in terms of interfaces and implementations, the encapsulation provided by classes is not as strong as the one provided by modules. This is because, through the use of inheritance, object-oriented languages put the emphasis on *incremental programming* to the detriment of modular programming. This introduces a white-box model of software development and more flexibility is traded for safety as demonstrated by the *fragile base class* issue [71].

Architecture Description Languages. The advent of distributed applications made it necessary to consider more sophisticated connections between the various building blocks of a system. The *software architecture* [79] of a software system describes the system as a composition of *components* and *connectors*, where the connectors capture the *interaction protocols* between the components [39]. It also describes the rationale behind such a given architecture, linking the properties required from the system to its implementation. *Architecture Description Languages* (ADLs) are languages that support architecture-based development [68]. A number of these languages make it possible to generate executable systems from architectural descriptions, provided implementations for the primitive components are available. However, guaranteeing that the implementation conforms to the architecture is an issue.

3.3. Aspect-Oriented Programming

The main driving force for the structuring means, such as components and modules, is the quest for clean *separation of concerns* [50] on the architectural and programming levels. It has, however, early been noted that concern separation in the presence of crosscutting functionalities requires specific language and implementation level support. Techniques of so-called *computational reflection*, for instance, Smith's 3-Lisp or Kiczales's CLOS meta-object protocol [80], [62] as well as metaprogramming techniques have been developed to cope with this problem but proven unwieldy to use and not amenable to formalization and property analysis due to their generality.

Aspect-Oriented Software Development [61], [37] has emerged over the previous decade as the domain of systematic exploration of crosscutting concerns and corresponding support throughout the software development process. The corresponding research efforts have resulted, in particular, in the recognition of *crosscutting* as a fundamental problem of virtually any large-scale application, and the definition and implementation of a large number of aspect-oriented models and languages.

However, most current aspect-oriented models, notably AspectJ [60], rely on pointcuts and advice defined in terms of individual execution events. These models are subject to serious limitations concerning the modularization of crosscutting functionalities in distributed applications, the integration of aspects with other modularization mechanisms such as components, and the provision of correctness guarantees of the resulting AO applications. They do, in particular, only permit the manipulation of distributed applications on a per-host basis, that is, without direct expression of coordination properties relating different distributed entities [81]. Similarly, current approaches for the integration of aspects and (distributed) components do not directly express interaction properties between sets of components but rather seemingly unrelated modifications to individual components [47]. Finally, current formalizations of such aspect models are formulated in terms of low-level semantic abstractions (see, e.g., Wand's et al semantics for AspectJ [83]) and provide only limited support for the analysis of fundamental aspect properties.

Recently, first approaches have been put forward to tackle these problems, in particular, in the context of so-called *stateful* or *history-based aspect languages* [51], [52], which provide pointcut and advice languages that directly express rich relationships between execution events. Such languages have been proposed to directly express coordination and synchronization issues of distributed and concurrent applications [75], [41], [54], provide more concise formal semantics for aspects and enable analysis of their properties [40], [53], [51], [38]. Due to the novelty of these approaches, they represent, however, only first results and many important questions concerning these fundamental issues remain open.

3.4. Protocols

Today, protocols constitute a frequently used means to precisely define, implement, and analyze contracts between two or more hardware or software entities. They have been used to define interactions between communication layers, security properties of distributed communications, interactions between objects and components, and business processes.

Object interactions [74], component interactions [84], [77] and service orchestrations [49] are most frequently expressed in terms of *regular interaction protocols* that enable basic properties, such as compatibility, substitutability, and deadlocks between components to be defined in terms of basic operations and closure properties of finite-state automata. Furthermore, such properties may be analyzed automatically using, e.g., model checking techniques [44], [56].

However, the limited expressive power of regular languages has led to a number of approaches using more expressive *non-regular* interaction protocols that often provide distribution-specific abstractions, e.g., session types [59], or context-free or turing-complete expressiveness [78], [43]. While these protocol types allow conformance between components to be defined (e.g., using unbounded counters), property verification can only be performed manually or semi-automatically.

Furthermore, first approaches for the definition of *aspects over protocols* have been proposed, as well as over regular structures [51] and non-regular ones [82], [73]. The modification of interaction protocols by aspects seems highly promising for the *integration of aspects and components*.

3.5. Patterns

Patterns provide a kind of abstraction that is complementary to the modularization mechanisms discussed above. They have been used, in particular, to define general *architectural styles* either by defining entire computation and communication topologies [72], connectors between (complex) software artifacts [69], or (based on, possibly concretizations of, *design patterns* [58]) as building blocks for object-oriented software architectures. The resulting pattern-based architectures are similar to common component-based architectures and are frequently used to implement the latter, see, for instance, Sun's J2EE patterns.

Patterns have also been used to implement architectural abstractions. This is the case, for instance, for the numerous variants of the *publish/subscribe pattern* [55] as well as the large set of so-called *skeletons* [46], that is, patterns for the implementation of distributed and concurrent systems. While these patterns are essentially similar to architecture-level patterns, their fine-grained application to multiple code entities often results in crosscutting code structures. An important open issue consists in the lack of pattern-based representations for the implementation of general distributed applications — in sharp contrast to their use for the derivation of massively parallel programs.

3.6. Domain-Specific Languages

Domain-specific languages (DSLs) represent domain knowledge in terms of suitable basic language constructs and their compositions at the language level. By trading generality for abstraction, they enable complex relationships among domain concepts to be expressed concisely and their properties to be expressed and formally analyzed. DSLs have been applied to a large number of domains; they have been particularly popular in the domain of software generation and maintenance [70], [85].

Many modularization techniques and tasks can be naturally expressed by DSLs that are either specialized with respect to the type of modularization constructs, such as a specific brand of software component, or to the compositions that are admissible in the context of an application domain that is targeted by a modular implementation. Moreover, software development and evolution processes can frequently be expressed by transformations between applications implemented using different DSLs that represent an implementation at different abstraction levels or different parts of one application.

Functionalities that crosscut a component-based application, however, complicate such a DSL-based transformational software development process. Since such functionalities belong to another domain than that captured by the components, different DSLs should be composed. Such compositions (including their syntactic expression, semantics and property analysis) have only very partially been explored until now. Furthermore, restricted composition languages and many aspect languages that only match execution events of a specific domain (e.g., specific file accesses in the case of security functionality) and trigger only domain-specific actions clearly are quite similar to DSLs but remain to be explored.

3.7. Distribution and Concurrency

While ASCOLA does not investigate distribution and concurrency as research domains per se (but rather from a software engineering and modularization viewpoint), there are several specific problems and corresponding approaches in these domains that are directly related to its core interests that include the structuring and modularization of large-scale distributed infrastructures and applications. These problems include crosscutting functionalities of distributed and concurrent systems, support for the evolution of distributed software systems, and correctness guarantees for the resulting software systems.

Underlying our interest in these domains is the well-known observation that large-scale distributed applications are subject to *numerous crosscutting functionalities* (such as the transactional behavior in enterprise information systems, the implementation of security policies, and fault recovery strategies). These functionalities are typically partially encapsulated in distributed infrastructures and partially handled in an ad hoc manner by using infrastructure services at the application level. Support for a more principled approach to the development and evolution of distributed software systems in the presence of crosscutting functionalities has been investigated in the field of *open adaptable middleware* [42], [64]. Open middleware design exploits the concept of reflection to provide the desired level of configurability and openness. However, these approaches are subject to several fundamental problems. One important problem is their insufficient, framework-based support that only allows partial modularization of crosscutting functionalities.

There has been some *criticism* on the use of *AspectJ-like aspect models* (which middleware aspect models like that of JBoss AOP are an instance of) for the modularization of distribution and concurrency related concerns, in particular, for transaction concerns [63] and the modularization of the distribution concern itself [81]. Both criticisms are essentially grounded in AspectJ's inability to explicitly represent sophisticated relationships between execution events in a distributed system: such aspects therefore cannot capture the semantic relationships that are essential for the corresponding concerns. History-based aspects, as those proposed by the ASCOLA project-team provide a starting point that is not subject to this problem.

From a point of view of language design and implementation, aspect languages, as well as domain specific languages for distributed and concurrent environments share many characteristics with existing distributed languages: for instance, event monitoring is fundamental for pointcut matching, different synchronization strategies and strategies for code mobility [57] may be used in actions triggered by pointcuts. However, these relationships have only been explored to a small degree. Similarly, the formal semantics and formal properties of aspect languages have not been studied yet for the distributed case and only rudimentarily for the concurrent one [40], [54].

ATLANMOD Team

3. Scientific Foundations

3.1. MDE Foundations

MDE can be seen as a generalization and abstraction of object technology allowing to map more abstract organizations on class-based implementations. In MDE, (software) models are considered as the unifying concept [44].

Traditionally, models were often used as initial design sketches mainly aimed for communicating ideas among developers. On the contrary, MDE promotes models as the primary artifacts that drive all software engineering activities. Therefore, rigorous techniques for model definition and manipulation are the basis of any MDE framework.

The MDE community distinguishes three levels of models: (terminal) model, metamodel, and metametamodel. A terminal model is a (partial) representation of a system/domain that captures some of its characteristics (different models can provide different knowledge views on the domain and be combined later on to provide a global view). In MDE we are interested in terminal models expressed in precise modeling languages. The abstract syntax of a language, when expressed itself as a model, is called a metamodel. A complete language definition is given by an abstract syntax (a metamodel), one or more concrete syntaxes (the graphical or textual syntaxes that designers use to express models in that language) plus one or more definition of its semantics. The relation between a model expressed in a language and the metamodel of that language is called *conformsTo*. Metamodels are in turn expressed in a modeling language called metamodeling language. Similar to the model/metamodel relationship, the abstract syntax of a metamodeling language is called a metametamodel and metamodels defined using a given metamodeling language must conform to its metametamodel. Terminal models, metamodels, and metametamodel form a three-level architecture with levels respectively named M1, M2, and M3. A formal definition of these concepts is provided in [52] and [45]. MDE promotes *unification by models*, like object technology proposed in the eighties *unification by objects* [43]. These MDE principles may be implemented in several standards. For example, OMG proposes a standard metametamodel called Meta Object Facility (MOF) while the most popular example of metamodel in the context of OMG standards is the UML metamodel.

In our view the main way to automate MDE is by providing model manipulation facilities in the form of model transformation operations that taking one or more models as input generate one or more models as output (where input and output models are not necessarily conforming to the same metamodel). More specifically, a model transformation Mt defines the production of a model Mb from a model Ma . When the source and target metamodels are identical ($MMa = MMb$), we say that the transformation is endogenous. When this is not the case ($MMa \neq MMb$) we say the transformation is exogenous. An example of an endogenous transformation is a UML refactoring that transforms public class attributes into private attributes while adding accessor methods for each transformed attribute. Many other operations may be considered as transformations as well. For example verifications or measurements on a model can be expressed as transformations [47]. One can see then why large libraries of reusable modeling artifacts (mainly metamodels and transformations) will be needed.

Another important idea is the fact that a model transformation is itself a model [4]. This means that the transformation program Mt can be expressed as a model and as such conforms to a metamodel MMt . This allows an homogeneous treatment of all kinds of terminal models, including transformations. Mt can be manipulated using the same existing MDE techniques already developed for other kinds of models. For instance, it is possible to apply a model transformation Mt' to manipulate Mt models. In that case, we say that Mt' is a higher order transformation (HOT), i.e. a transformation taking other transformations (expressed as transformation models) as input or/and producing other transformations as output.

As MDE developed, it became apparent that this was a branch of language engineering [46]. In particular, MDE offers an improved way to develop DSLs (Domain-Specific Languages). DSLs are programming or modeling languages that are tailored to solve specific kinds of problems in contrast with General Purpose Languages (GPLs) that aim to handle any kind of problem. Java is an example of a programming GPL and UML an example of a modeling GPL. DSLs are already widely used for certain kinds of programming; probably the best-known example is SQL, a language specifically designed for the manipulation of relational data in databases. The main benefit of DSLs is that they allow everybody to write programs/models using the concepts that actually make sense to their domain or to the problem they are trying to solve (for instance Matlab has matrices and lets the user express operations on them, Excel has cells, relations between cells, and formulas and allows the expression of simple computations in a visual declarative style, etc.). As well as making domain code programmers more productive, DSLs also tend to offer greater optimization opportunities. Programs written with these DSLs may be independent of the specific hardware they will eventually run on. Similar benefits are obtained when using modeling DSLs. In MDE, new DSLs can be easily specified by using the metamodel concept to define their abstract syntax. Models specified with those DSLs can then be manipulated by means of model transformations (with ATL for example [8]).

When following the previously described principles, one may take advantage of the uniformity of the MDE organization. Considering similarly models of the static architecture and models of the dynamic behavior of a system allows at the same time economy of concepts and economy of implementation. Considering models of products (e.g., software artifacts like UML) and models of processes (e.g., software processes like SPEM) may lead to a dual process/product organization. Considering transformation models, weaving models, and traceability models as special cases of correspondence models may also lead to simplicity and efficiency of implementations. These are some of the use cases that are being explored in the team.

CIDRE Project-Team

3. Scientific Foundations

3.1. Introduction

For many aspects of our everyday life, we rely heavily on information systems, many of which are based on massively networked devices that support a population of interacting and cooperating entities. While these information systems become increasingly open and complex, accidental and intentional failures get considerably more frequent and severe.

Two research communities traditionally address the concern of accidental and intentional failures: the distributed computing community and the security community. While both these communities are interested in the construction of systems that are correct and secure, an ideological gap and a lack of communication exist between them that is often explained by the incompatibility of the assumptions each of them traditionally makes. Furthermore, in terms of objectives, the distributed computing community has favored systems availability while the security community has focused on integrity and confidentiality, and more recently on privacy.

By contrast with this traditional conception, we are convinced that by looking at information systems as a combination of possibly revisited basic protocols, each one specified by a set of properties such as synchronization and agreement, security properties should emerge. This vision is shared by others and in particular by Myers *et al.* [56], whose objectives are to explore new methods for constructing distributed systems that are trustworthy in the aggregate even when some nodes in the system have been compromised by malicious attackers. In accordance with this vision, the first main characteristic of the CIDRE group is to gather researchers from the two aforementioned communities in order to address in a complementary manner both the concerns of accidental and intentional failures.

The second main characteristic of the CIDRE group lies in the scope of the systems it considers. Indeed, during our research, we will consider three complementary levels of study: the Node Level, the Group Level, and the Open Network Level:

- **Node Level:** The term node either refers to a device that hosts a network client or service or to the process that runs this client or service. Node security management must be the focus of a particular attention, since from the user point of view, security of his own devices is crucial. Sensitive information and services must therefore be locally protected against various forms of attacks. This protection may take a dual form, namely prevention and detection.
- **Group Level:** Distributed applications often rely on the identification of sets of interacting entities. These subsets are either called groups, clusters, collections, neighborhoods, spheres, or communities according to the criteria that define the membership. Among others, the adopted criteria may reflect the fact that its members are administrated by a unique person, or that they share the same security policy. It can also be related to the localization of the physical entities, or the fact that they need to be strongly synchronized, or even that they share mutual interests. Due to the vast number of possible contexts and terminologies, we refer to a single type of set of entities, that we call set of nodes. We assume that a node can locally and independently identify a set of nodes and modify the composition of this set at any time. The node that manages one set has to know the identity of each of its members and should be able to communicate directly with them without relying on a third party. Despite these two restrictions, this definition remains general enough to include as particular cases most of the examples mentioned above. Of course, more restrictive behaviors can be specified by adding other constraints. We are convinced that security can benefit from the existence and the identification of sets of nodes of limited size as they can help in improving the efficiency of the detection and prevention mechanisms.

- **Open Network Level:** In the context of large-scale distributed and dynamic systems, interaction with unknown entities becomes an unavoidable habit despite the induced risk. For instance, consider a mobile user that connects his laptop to a public Wifi access point to interact with his company. At this point, data (regardless it is valuable or not) is updated and managed through non trusted undedicated entities (i.e., communication infrastructure and nodes) that provide multiple services to multiple parties during that user connection. In the same way, the same device (e.g., laptop, PDA, USB key) is often used for both professional and private activities, each activity accessing and manipulating decisive data.

The third characteristic of the CIDRE group is to focus on three different aspects of security, i.e., trust, intrusion detection, and privacy, and on the different bridges that exist between these aspects. Indeed, we believe that to study new security solutions for nodes, set of nodes and open network levels, one must take into account that it is now a necessity to interact with devices whose owners are unknown. To reduce the risk to rely on dishonest entities, a trust mechanism is an essential prevention tool that aims at measuring the capacity of a remote node to provide a service compliant with its specification. Such a mechanism should allow to overcome ill-founded suspicions and to be aware of established misbehaviors. To identify such misbehaviors, intrusion detection systems are necessary. Such systems aim at detecting, by analyzing data flows, whether violations of the security policies have occurred. Finally, Privacy Protection which is now recognized as a basic user right, should be respected despite the presence of tools that continuously observe or even control users actions or behaviors.

3.2. Intrusion Detection

By exploiting vulnerabilities in operating systems, applications, or network services, an attacker can defeat the preventive security mechanisms and violate the security policy of the whole system. The goal of intrusion detection systems (IDS) is to be able to detect, by analyzing some data generated on a monitored system, violations of the security policy. From our point of view, while useful in practice, misuse detection is intrinsically limited. Indeed, it requires to update the signatures database in real-time similarly to what has to be done for antivirus tools. Given that there are thousands of machines that are every day victims of malware, such an approach may appear as insufficient especially due to the incredible expansion of malware, drastically limiting the capabilities of human intervention and response. The CIDRE group takes the alternative approach, i.e. the anomaly approach, which consists in detecting a deviation from a referenced behavior. Specifically, we propose to study two complementary methods:

- **Illegal Flow Detection:** This first method intends to detect information flows that violate the security policy [59], [55]. Our goal is here to detect information flows in the monitored system that are allowed by the access control mechanism, but are illegal from the security policy point of view.
- **Data Corruption Detection:** This second method aims at detecting intrusions that target specific applications, and make them execute illegal actions by using these applications incorrectly [54], [58]. This approach complements the previous one in the sense that the incorrect use of the application can possibly be legal from the point of view of the information flows and access control mechanisms, but is incorrect considering the security policy.

In both approaches, the access control mechanisms or the monitored applications can be either configured and executed on a single node, or distributed on a set of nodes. Thus, our approach must be studied at least at these first two levels. Moreover, we plan to work on intrusion detection system evaluation methods. For that research, we set a priori aside no particular IDS approach or technique. Here are some concrete examples of our research goals (both short term and long term objectives) in the intrusion detection field:

- at node level, we are going to apply the defensive programming approach (coming from the dependency field) to data corruption detection. The challenge is to determine which invariant/properties must be and can be verified either at runtime or statically. Regarding illegal flow detection, we plan to extend this method to build anti-viruses and DBMS tools by determining viruses signatures.

- at the set of nodes level, we are going to revisit the distributed problems such as clock synchronization, logical clocks, consensus, properties detection, to extend the solutions proposed at node levels to cope with distributed flow control checking mechanisms. Regarding illegal flow detection, one of the challenges is to enforce the collaboration and consistency at nodes and set of nodes levels to obtain a global intrusion detection mechanism. Regarding the data corruption detection approach, the challenge is to identify local predicates/properties/invariants so that global predicates/properties/invariants would emerge at the system level.

3.3. Privacy

In our world of ubiquitous technologies, each individual constantly leaves digital traces related to his activities and interests which can be linked to his identity. In forthcoming years, the protection of privacy is one of the greatest challenge that lies ahead and also an important condition for the development of the Information Society. Moreover, due to legality and confidentiality issues, problematics linked to privacy emerge naturally for applications working on sensitive data, such as medical records of patients or proprietary datasets of enterprises. Privacy Enhancing Technologies (PETs) are generally designed to respect both the principles of data minimization and data sovereignty. The data minimization principle states that only the information necessary to complete a particular application should be disclosed (and no more). This principle is a direct application of the legitimacy criteria defined by the European data protection directive (Article 7). The data sovereignty principle states that data related to an individual belong to him and that he should stay in control of how this data is used and for which purpose. This principle can be seen as an extension of many national legislations on medical data that consider that a patient record belongs to the patient, and not to the doctors that create or update it, nor to the hospital that stores it. In the CIDRE project, we will investigate PETs that operate at the three different levels (node, set of nodes or open distributed system) and are generally based on a mix of different foundations such as cryptographic techniques, security policies and access control mechanisms just to name a few. Examples of domains where privacy and utility aspects collide and that will be studied within the context of CIDRE include: identity and privacy, geo-privacy, distributed computing and privacy, privacy-preserving data mining and privacy issues in social networks. Here are some concrete examples of our research goals in the privacy field:

- at the node level, we aim at designing privacy preserving identification scheme, automated reasoning on privacy policies [57], and policy-based adaptive PETs.
- at the set of nodes level, we plan to augment distributed algorithms (i.e., consensus) with privacy properties such as anonymity, unlinkability, and unobservability.
- at the open distributed system level, we plan to target both geo-privacy concerns (that typically occur in geolocalized systems) and privacy issues in social networks. In the former case, we will adopt a sanitization approach while in the latter one we plan to define privacy policies at user level, and their enforcement by all the intervening actors (e.g, at the social network sites providers).

3.4. Trust Management

While the distributed computing community relies on the trustworthiness of its algorithms to ensure systems availability, the security community historically makes the hypothesis of a Trusted Computing Base (TCB) that contains the security mechanisms (such as access controls, and cryptography) that implement the security policy. Unfortunately, as information systems get increasingly complex and open, the TCB management may itself get very complex, dynamic and error-prone. From our point of view, an appealing approach is to distribute and manage the TCB on each node and to leverage the trustworthiness of the distributed algorithms in order to strengthen each node's TCB. Accordingly, the CIDRE group proposes to study automated trust management systems at all the three identified levels:

- at the node level, such a system should allow each node to evaluate by itself the trustworthiness of its neighborhood and to self-configure the security mechanisms it implements;
- at the group level, such a system might rely on existing trust relations with other nodes of the group to enhance the significance and the reliability of the gathered information;

- at the open network level, such a system should rely on reputation mechanisms to estimate the trustworthiness of the peers the node interacts with. The system might also benefit from the information provided by a priori trusted peers that, for instance, would belong to the same group (see previous item).

For the last two items, the automated trust management system will de facto follow the distributed computing approach. As such, emphasis will be put on the trustworthiness of the designed distributed algorithms. Thus, the proposed approach will provide both the adequate security mechanisms and a trustworthy distributed way of managing them. By way of examples of our research goals regarding the trust management field, we briefly list some of our short and long term objectives at node, group and open networks levels:

1. at node level, we are going to investigate how implicit trust relationships, identified and deduced by a node during its interactions with its neighborhood, could be explicitly used by the node (for instance by means of a series of rules) to locally evaluate the trustworthiness of its neighborhood. The impact of trust on the local security policy, and on its enforcement will be studied accordingly.
2. at the set of nodes level, we plan to take advantage of the pre-existing trust relationship among the set of nodes to design composition mechanisms that would guarantee that automatically configured security policies are consistent with each group member security policy.
3. at the open distributed system level, we are going to design reputation mechanisms to both defend the system against specific attacks (whitewashing, bad mouthing, ballot stuffing, isolation) by relying on the properties guaranteed at nodes and set of nodes levels, and guaranteeing persistent and safe feedback, and for specific cases in guaranteeing the right to oblivion (i.e., the right to data erasure).

DIONYSOS Project-Team

3. Scientific Foundations

3.1. Introduction

The scientific foundations of our work are those of network design and network analysis. Specifically, this concerns the principles of packet switching and in particular of IP networks (protocol design, protocol testing, routing, scheduling techniques), and the mathematical and algorithmic aspects of the associated problems, on which our methods and tools are based.

These foundations are described in the following paragraphs. We begin by a subsection dedicated to Quality of Service (QoS) and Quality of Experience (QoE), since they can be seen as unifying concepts in our activities. Then we briefly describe the specific sub-area of models' evaluation and about the particular multidisciplinary domain of network economics.

3.2. Quality of Service and Quality of Experience

Since it is difficult to develop as many communication solutions as possible applications, the scientific and technological communities aim towards providing general *services* allowing to give to each application or user a set of properties nowadays called "Quality of Service" (QoS), a terminology lacking a precise definition. This QoS concept takes different forms according to the type of communication service and the aspects which matter for a given application: for performance it comes through specific metrics (delays, jitter, throughput, ...), for dependability it also comes through appropriate metrics: reliability, availability, or vulnerability, in the case for instance of WAN (Wide Area Network) topologies, etc.

QoS is at the heart of our research activities: we look for methods to obtain specific "levels" of QoS and for techniques to evaluate the associated metrics. Our ultimate goal is to provide tools (mathematical tools and/or algorithms, under appropriate software "containers" or not) allowing users and/or applications to attain specific levels of QoS, or to improve the provided QoS, if we think of a particular system, with an optimal use of the resources available. Obtaining a good QoS level is a very general objective. It leads to many different areas, depending on the systems, applications and specific goals being considered. Our team works on several of these areas. We also investigate the impact of network QoS on multimedia payloads to reduce the impact of congestion.

Some important aspects of the behavior of modern communication systems have subjective components: the quality of a video stream or an audio signal, *as perceived by the user*, is related to some of the previous mentioned parameters (packet loss, delays, ...) but in an extremely complex way. We are interested in analyzing these types of flows from this user-oriented point of view. We focus on the *user perceived quality*, the main component of what is nowadays called Quality of Experience (in short, QoE), to underline the fact that, in this case, we want to center the analysis on the user. In this context, we have a global project called PSQA, which stands for Pseudo-Subjective Quality Assessment, and which refers to a methodology allowing to automatically measuring the QoE (see 3.2).

Another special case to which we devote research efforts in the team is the analysis of qualitative properties related to interoperability assessment. This refers to the act of determining if end-to-end functionality between at least two communicating systems is as required by the base standards for those systems. Conformance is the act of determining to what extent a single component conforms to the individual requirements of the standard it is based on. Our purpose is to provide such a formal framework (methods, algorithms and tools) for interoperability assessment, in order to help in obtaining efficient interoperability test suites for new generation networks, mainly around IPv6-related protocols. The interoperability test suites generation is based on specifications (standards and/or RFCs) of network components and protocols to be tested.

3.3. Stochastic modeling

The scientific foundations of our modeling activities are composed of stochastic processes theory and, in particular, Markov processes, queuing theory, stochastic graphs theory, etc. The objectives are either to develop numerical solutions, or analytical ones, or possibly discrete event simulation or Monte Carlo (and Quasi-Monte Carlo) techniques. We are always interested in models' evaluation techniques for dependability and performability analysis, both in static (network reliability) and dynamic contexts (depending on the fact that time plays an explicit role in the analysis or not). We look at systems from the classical so-called *call level*, leading to standard models (for instance, queues or networks of queues) and also at the *burst level*, leading to *fluid models*.

In recent years, our work on the design of the topologies of WANs led us to optimization techniques, in particular in the case of very large optimization problems, usually formulated in terms of graphs. The associated methods we are interested in are composed of simulated annealing, genetic algorithms, TABU search, etc. For the time being, we have obtained our best results with GRASP techniques.

Network pricing is a good example of a multi-disciplinary research activity half-way between applied mathematics, economy and networking, centered on stochastic modeling issues. Indeed, the Internet is facing a tremendous increase of its traffic volume. As a consequence, real users complain that large data transfers take too long, without any possibility to improve this by themselves (by paying more, for instance). A possible solution to cope with congestion is to increase the link capacities; however, many authors consider that this is not a viable solution as the network must respond to an increasing demand (and experience has shown that demand of bandwidth has always been ahead of supply), especially now that the Internet is becoming a commercial network. Furthermore, incentives for a fair utilization between customers are not included in the current Internet. For these reasons, it has been suggested that the current flat-rate fees, where customers pay a subscription and obtain an unlimited usage, should be replaced by usage-based fees. Besides, the future Internet will carry heterogeneous flows such as video, voice, email, web, file transfers and remote login among others. Each of these applications requires a different level of QoS: for example, video needs very small delays and packet losses, voice requires small delays but can afford some packet losses, email can afford delay (within a given bound) while file transfer needs a good average throughput and remote login requires small round-trip times. Some pricing incentives should exist so that each user does not always choose the best QoS for her application and so that the final result is a fair utilization of the bandwidth. On the other hand, we need to be aware of the trade-off between engineering efficiency and economic efficiency; for example, traffic measurements can help in improving the management of the network but is a costly option. These are some of the various aspects often present in the pricing problems we address in our work. More recently, we have switched to the more general field of network economics, dealing with the economic behavior of users, service providers and content providers, as well as their relations.

DISTRIBCOM Project-Team

3. Scientific Foundations

3.1. Overview of the needed paradigms

Management of telecommunications networks and services, and Web services, involves the following algorithmic tasks:

Observing, monitoring, and testing large distributed systems: Alarm or message correlation is one of the five basic tasks in network and service management. It consists in causally relating the various alarms collected throughout the considered infrastructure—be it a network or a service sitting on top of a transport infrastructure. Fault management requires in particular reconstructing the set of all state histories that can explain a given log of observations. Testing amounts to understanding and analyzing the responses of a network or service to a given set of stimuli; stimuli are generally selected according to given test purposes. All these are variants of the general problem of *observing* a network or service. Networks and services are large distributed systems, and we aim at observing them in a distributed way as well, namely: logs are collected in a distributed way and observation is performed by a distributed set of supervising peers.

Quality of Service (QoS) evaluation, negotiation, and monitoring: QoS issues are a well established topic for single domain networks or services, for various protocols — e.g., Diffserv for IP. Performance evaluation techniques are used that follow a “closed world” point of view: the modeling involves the overall traffic, and resource characteristics are assumed known. These approaches extend to some telecommunication services as well, e.g., when considering (G)MPLS over an IP network layer.

However, for higher level applications, including composite Web services (also called *orchestrations*), this approach to QoS is no longer valid. For instance, an orchestration using other Web services has no knowledge of how many users are calling the same Web services. In addition, it has no knowledge of the transport resources it is using. Therefore, the well developed “closed world” approach can no longer be used. *Contract-based* approaches are considered instead, in which a given orchestration offers promises to its users on the basis of promises it has from its subcontracting services. In this context, contract composition becomes a central issue. Monitoring is needed to check for possible breaching of the contract. Countermeasures would consist in reconfiguring the orchestration by replacing the failed subcontracted services by alternative ones.

The DistribCom team focuses on the algorithms supporting the above tasks. Therefore models providing an adequate framework are fundamental. We focus on models of discrete systems, not models of streams or fluid types of models. And we address the distributed and asynchronous nature of the underlying systems by using models involving only local, not global, states, and local, not global, time. These models are reviewed in section 3.2. We use these mathematical models to support our algorithms and we use them also to study and develop formalisms of Web services orchestrations and workflow management in a more general setting.

3.2. Models of concurrency: nets, scenarios, event structures, and their variants

For Finite State Machines (FSM), a large body of theory has been developed to address problems such as: observation (the inference of hidden state trajectories from incomplete observations), control, diagnosis, and learning. These are difficult problems, even for simple models such as FSM's. One of the research tracks of DistribCom consists in extending such theories to distributed systems involving concurrency, i.e., systems in which both time and states are local, not global. For such systems, even very basic concepts such as “trajectories” or “executions” need to be deeply revisited. Computer scientists have for a long time recognized this topic of concurrent and distributed systems as a central one. In this section, we briefly introduce the reader to the models of scenarios, event structures, nets, languages of scenarios, graph grammars, and their variants.

3.2.1. Scenarios.

The simplest concept related to concurrency is that of a finite execution of a distributed machine. To this end, scenarios have been informally used by telecom engineers for a long time. In scenarios, so-called “instances” exchange asynchronous messages, thus creating events that are totally ordered on a given instance, and only partially ordered by causality on different instances (emission and reception of a message are causally related). The formalization of scenarios was introduced by the work done in the framework of ITU and OMG on High-level Message Sequence Charts and on UML Sequence Diagrams in the last ten years, see [52], [57]. This allowed in particular to formally define infinite scenarios, and to enhance them with variables, guards, etc [61], [59], [60]. Today, scenarios are routinely offered by UML and related software modeling tools.

3.2.2. Event structures.

Event structures were invented by Glynn Winskel and co-authors in 1980 [56], [62]. Executions are sets of events that are partially ordered by a *causality* relation. Event structures collect all the executions by superimposing shared prefixes. Events not belonging to a same execution are said in *conflict*. Events that are neither causally related nor in conflict are called *concurrent*. Concurrent processes model the “parallel progress” of components.

Categories of event structures have been defined, with associated morphisms, products, and co-products, see [63]. Products and co-products formalize the concepts of parallel composition and “union” of event structures, respectively. This provides the needed apparatus for composing and projecting (or abstracting) systems. Event structures have been mostly used to give the semantics of various formalisms or languages, such as Petri nets, CCS, CSP, etc [56], [62]. We in DistribCom make a nonstandard use of these, e.g., we use them as a structure to compute and express the solutions of observation or diagnosis problems, for concurrent systems.

3.2.3. Nets and languages of scenarios.

The next step is to have finite representations of systems having possibly infinite executions. In DistribCom, we use two such formalisms: *Petri nets* [58], [45] and *languages of scenarios* such as High-level Message Sequence Charts (HMSC) [52], [60]. Petri nets are well known, at least in their basic form, we do not introduce them here. We use so-called *safe* Petri Nets, in which markings are boolean (tokens can be either 0 or 1); and we use also variants, see below.

3.2.4. Extensions and variants.

Two extensions of the basic concepts of nets or scenario languages are useful for us. Nets or scenario languages enriched with variables, actions, and guards, are useful to model general concurrent and distributed dynamical systems in which a certain discrete abstraction of the control is represented by means of a net or a scenario language. Manipulating such *symbolic nets* requires using abstraction techniques. Time Petri nets and network of timed automata are particular cases of symbolic nets. Probabilistic Nets or event structures: Whereas a huge literature exists on stochastic Petri nets or stochastic process algebras (in computer science), randomizing *concurrent models*, i.e., with ω 's being concurrent trajectories, not sequential ones, has been addressed only since the 21st century. We have contributed to this new area of research.

3.2.5. Handling dynamic changes in the systems.

The last and perhaps most important issue, for our applications, is the handling of dynamic changes in the systems model. This is motivated by the constant use of dynamic reconfigurations in management systems. Extensions of net models have been proposed to capture this, for example the *dynamic nets* of Vladimiro Sassone [44] and *net systems* [46]. For the moment, such models lack a suitable theory of unfoldings.

3.3. Modal logics for distributed systems

Modal logics are a family of logics that were developed originally to reason about different modalities occurring in natural language, such as for example the modality of knowledge (epistemic logic), the modalities of obligation and permission (deontic logic) and the modality of time (temporal logic). Temporal logics (CTL, LTL, μ -calculus...) are the most prominent (modal) logics used in computer science nowadays, especially in the field of verification.

3.3.1. Epistemic logic and distributed systems.

In the 1980's, epistemic logic was propounded by computer scientists such as Fagin, Halpern, Moses and Vardi to address problems in distributed systems, resulting in the TARK conference series (Theoretical Aspects of Rationality and Knowledge) and the books [48], [54]. This interest in epistemic logic was due to their observation that the notion of knowledge plays a central role in the informal reasoning used in the design of distributed protocols. This led these authors to "hope that a theory of knowledge, communication and action will prove rich enough to provide general foundations for a unified theoretical treatment of distributed systems" [50]. The research pursued in DistribCom follows this line of thought, although we also strive to feed and confront our theoretical developments with actual problems stemming from diverse areas of application of distributed systems.

In [48], the behavior of a distributed system is represented by a set of *runs*, each run being a possible execution of the distributed system, determined by a given protocol. Processors are called *agents* and their partial observation of the system is represented at any point in the run by indistinguishability relations between local states of different runs (the local state of a processor represents the state of this processor at a moment of time). This model was used to show for example that the specific notion of common knowledge of epistemic logic is necessary to reach agreement and to coordinate actions [50]. Dynamic Epistemic Logic (DEL) is another logical framework that can be used to represent and reason about distributed systems (connections between these two logical frameworks were made in [64]). DEL deals with the representation of global states of synchronous distributed systems. The global state of the system at a moment in time is represented directly by means of an *epistemic model*. Events occurring in this distributed system are represented by means of *event models* and their effects on the local states of agents (processors) are represented by means of a *product update*.

The contributions in this sub-module are described in Section 6.4 .

3.3.2. Deontic logic and privacy in distributed systems.

We also use deontic logic in combination with epistemic logic for the formalization of privacy regulations. We intend to use this formalization to reason about privacy in the composition of web-services. The combination of these two modal logics can be used to express statements such as "it is *forbidden* for agent 1 to *know* that agent 2 sent message *m*" or "if agent 1 is an administrator of the system, then it is *permitted* for him to *know* information *i*". This provides a formal language very close to the natural language used in actual privacy regulations by law legislators. In the long run, we expect this formal language to be used at the level of interfaces of the web-service in order to:

1. check that the privacy policy declared by the web-service on its interface is indeed compliant (coherent) with respect to the privacy regulations expressed by law makers;
2. check that the web-service does enforce and apply the privacy policy it has declared on its interface.

The contributions in this sub-module are described in Section 6.8 .

3.4. Statistical Model Checking

Complex systems pose two particular challenges to formal verification: (i) the non-determinism caused by concurrency and unpredictable environmental conditions and (ii) the size of the state space. Our interest is probabilistic model checking, that can verify intricate details of a system's dynamical behavior and where non-determinism is handled by assigning probabilistic distributions to unknowns and quantifying results with a probability. Exact probabilistic model checking quantifies these probabilities to the limit of numerical precision by an exhaustive exploration of the state space, but is restricted by what can be conveniently stored in memory. Our focus is therefore statistical model checking (SMC), that avoids an explicit representation of the state space by building a statistical model of the executions of a system and giving results within confidence bounds. The key challenges of this approach are to reduce the length (simulation steps and cpu time) and number of simulation traces necessary to achieve a result with given confidence. Rare properties pose a particular problem in this respect, since they are not only difficult to observe but their probability is difficult to bound. A further goal is to make a tool where the choice of modeling language and logic are flexible.

KERDATA Project-Team

3. Scientific Foundations

3.1. Our goals and methodology

Data-intensive applications demonstrate common requirements with respect to the need for data storage and I/O processing. These requirements lead to several core challenges discussed below.

Challenges related to cloud storage. In the area of cloud data management, a significant milestone is the emergence of the Map-Reduce [32] parallel programming paradigm, currently used on most cloud platforms, following the trend set up by Amazon [28]. At the core of the Map-Reduce frameworks stays a key component, which must meet a series of specific requirements that have not fully been met yet by existing solutions: the ability to provide efficient *fine-grain access* to the files, while sustaining a *high throughput* in spite of *heavy access concurrency*. Additionally, as thousands of clients simultaneously access shared data, it is critical to preserve *fault-tolerance* and *security* requirements.

Challenges related to data-intensive HPC applications. The requirements exhibited by climate simulations specifically highlights a major, more general research topic. It has been clearly identified by international panels of experts like IESP [30] and EESI [29], in the context of HPC simulations running on post-Petascale supercomputers. A jump of one order of magnitude in the size of numerical simulations is required to address some of the fundamental questions in several communities such as climate modeling, solid earth sciences or astrophysics. In this context, the lack of data-intensive infrastructure and methodology to analyze huge simulations is a growing limiting factor. The challenge is to find new ways to store and analyze massive outputs of data during and after the simulation without impacting the overall performance.

The overall goal of the KerData project-team is to bring a substantial contribution to the effort of the research community to address the above challenges. KerData aims to design and implement distributed algorithms for scalable data storage and input/output management for efficient large-scale data processing. We target two main execution infrastructures: cloud platforms and post-Petascale HPC supercomputers. We are also looking at other kinds of infrastructures (that we are considering as secondary), e.g. hybrid platforms combining enterprise desktop grids extended to cloud platforms. Our collaboration portfolio includes international teams that are active in this area both in Academia (e.g., Argonne National Lab, University of Illinois at Urbana-Champaign, University of Tsukuba) and Industry (Microsoft, IBM).

The highly experimental nature of our research validation methodology should be stressed. Our approach relies on building prototypes and on their large-scale experimental validation on real testbeds and experimental platforms. We strongly rely on the ALADDIN-Grid'5000 platform. Moreover, thanks to our projects and partnerships, we have access to reference software and physical infrastructures in the cloud area (Microsoft Azure, Amazon clouds, Nimbus clouds); in the post-Petascale HPC area we have access to the Jaguar and Kraken supercomputers (ranked 3rd and 11th respectively in the Top 500 supercomputer list) and, hopefully soon, to the Blue Waters supercomputer). This provides us with excellent opportunities to validate our results on realistic platforms.

Moreover, the consortiums of our current projects include application partners in the areas of Bio-Chemistry, Neurology and Genetics, and Climate Simulations. This is an additional asset, it enables us to take into account application requirements in the early design phase of our solutions, and to validate those solutions with real applications. We intend to continue increasing our collaborations with application communities, as we believe that this a key to perform effective research with a high potential impact.

3.2. Our research agenda

Three typical application scenarios are described in Section 4.1 :

- Joint genetic and neuroimaging data analysis on Azure clouds
- Structural protein analysis on Nimbus clouds
- I/O intensive climate simulations for the Blue Waters post-Petascale machine

They illustrate the above challenges in some specific ways. They all exhibit a common scheme: massively concurrent processes which access massive data at a fine granularity, where data is shared and distributed at a large scale. To efficiently address the aforementioned challenges we have started to work out an approach called BlobSeer, which stands today at the center of our research efforts. This approach relies on the design and implementation of *scalable* distributed algorithms for data storage and access. They combine advanced techniques for decentralized metadata and data management, with versioning-based concurrency control to optimize the performance of applications under heavy access concurrency.

Preliminary experiments with our BlobSeer BLOB management system within today's cloud software infrastructures proved very promising. Recently, we used the BlobSeer approach as a starting point to address more in depth two usage scenarios, which led to two more specific approaches: 1) Pyramid (which borrows many concepts from BlobSeer), with a specific focus on array-oriented storage; and 2) Damaris (totally independent of BlobSeer), which exploits multicore parallelism in post-Petascale supercomputers. All these directions are described below.

Our short- and medium-term research plan is devoted to storage challenges in two main contexts: clouds and post-Petascale HPC architectures. Consequently, our research plan is split in two main themes, which correspond to their respective challenges. For each of those themes, we have initiated several actions through collaborative projects coordinated by KerData, which define our agenda for the next 4 years.

Based on very promising results demonstrated by this approach in preliminary experiments [37], we have initiated several collaborative projects led by KerData in the area of cloud data management, e.g., the MapReduce ANR project, the A-Brain Microsoft-Inria project. Such frameworks are for us concrete and efficient means to work in close connection with strong partners already well positioned in the area of cloud computing research. Thanks to those projects, we have already started to enjoy a visible scientific positioning at the international level.

The particularly active DataCloud@work Associate Team creates the framework for an enlarged research activity involving a large number of young researchers and students. It serves as a basis for extended research activities based on our approaches, carried out beyond the frontiers of our team. In the HPC area, our presence in the research activities of the Joint UIUC-Inria Lab for Petascale Computing at Urbana-Champaign is a very exciting opportunity that we have started to leverage. It facilitates high-quality collaborations and access to some of the most powerful supercomputers, an important asset which already helped us produce and transfer some results, as described in Section 6.5 .

MYRIADS Project-Team

3. Scientific Foundations

3.1. Introduction

Research activity within the MYRIADS team encompasses several areas: distributed systems, middleware and programming models. We have chosen to provide a brief presentation of some of the scientific foundations associated with them: autonomic computing, future internet and SOA, distributed operating systems, and unconventional/nature-inspired programming.

3.2. Autonomic Computing

During the past years the development of raw computing power coupled with the proliferation of computer devices has grown at exponential rates. This phenomenal growth along with the advent of the Internet have led to a new age of accessibility - to other people, other applications and others systems. It is not just a matter of numbers. This boom has also led to unprecedented levels of complexity for the design and the implementation of these applications and systems, and of the way they work together. The increasing system scale is reaching a level beyond human ability to master its complexity.

This points towards an inevitable need to automate many of the functions associated with computing today. Indeed we want to interact with applications and systems intuitively, and we want to be far less involved in running them. Ideally, we would like computing systems to entirely manage themselves.

IBM [68] has named its vision for the future of computing "autonomic computing." According to IBM this new computer paradigm means the design and implementation of computer systems, software, storage and support that must exhibit the following basic fundamentals:

Flexibility. An autonomic computing system must configure and reconfigure itself under varying, even unpredictable, conditions.

Accessibility. The nature of the autonomic system is that it is always on.

Transparency. The system will perform its tasks and adapt to a user's needs without dragging the user into the intricacies of its workings.

In the Myriads team we will act to satisfy these fundamentals.

3.3. Future Internet and SOA

Traditional information systems were built by integrating applications into a communication framework, such as CORBA or with an Enterprise Application Integration system (EAI). Today, companies need to be able to reconfigure themselves; they need to be able to include other companies' business, split or externalize some of their works very quickly. In order to do this, the information systems should react and adapt very efficiently. EAI's approaches did not provide the necessary agility because they were too tightly coupled and a large part of business processes were "hard wired" into company applications.

Web services and Service Oriented Architectures (SOA) partly provide agility because in SOA business processes are completely separated from applications which can only be viewed as providing services through an interface. With SOA technologies it is easily possible to modify business processes, change, add or remove services.

However, SOA and Web services technologies are mainly market-driven and sometimes far from the state-of-the-art of distributed systems. Achieving dependability or being able to guarantee Service Level Agreement (SLA) needs much more agility of software elements. Dynamic adaptability features are necessary at many different levels (business processes, service composition, service discovery and execution) and should be coordinated. When addressing very large scale systems, autonomic behaviour of services and other parts of service oriented architectures is necessary.

SOAs will be part of the "Future Internet". The "Future Internet" will encompass traditional Web servers and browsers to support companies and people interactions (Internet of services), media interactions, search systems, etc. It will include many appliances (Internet of things). The key research domains in this area are network research, cloud computing, Internet of services and advanced software engineering.

The Myriads team will address adaptability and autonomy of SOAs in the context of Grids, Clouds and at large scale.

3.4. Distributed Operating Systems

An operating system provides abstractions such as files, processes, sockets to applications so that programmers can design their applications independently of the computer hardware. At execution time, the operating system is in charge of finding and managing the hardware resources necessary to implement these abstractions in a secure way. It also manages hardware and abstract resource sharing between different users and programs.

A distributed operating system makes a network of computer appear as a single machine. The structure of the network and the heterogeneity of the computation nodes are hidden to users. Members of the Myriads team members have a long experience in the design and implementation of distributed operating systems, for instance in Kerrighed, Vigne and XtremOS projects.

Clouds can be defined as platforms for on-demand resource provisioning over the Internet. These platforms rely on networked computers. Three flavours of cloud platforms have emerged corresponding to different kinds of service delivery:

- IaaS (Infrastructure as a Service) refers to clouds for on-demand provisioning of elastic and customizable execution platforms (from physical to virtualized hardware).
- PaaS (Platform as a Service) refers to clouds providing an integrated environment to develop, build, deploy, host and maintain scalable and adaptable applications.
- SaaS (Software as a Service) refers to clouds providing customers access to ready-to-use applications.

The cloud computing model [65], [62] introduces new challenges in the organization of the information infrastructure: security, identity management, adaptation to the environment (costs). The organization of large organization IT infrastructures is also impacted as their internal data-centers, sometimes called private clouds, need to cooperate with resources and services provisioned from the cloud in order to cope with workload variations. The advent of cloud and green computing introduces new challenges in the domain of distributed operating systems: resources can be provisioned and released dynamically, the distribution of the computations on the resources must be reevaluated periodically in order to reduce power consumption and resource usage costs. Distributed cloud operating system must adapt to these new challenges in order to reduce cost and energy, for instance, through the redistribution of the applications and services on a smaller set of resources.

The Myriads team will work on the design and implementation of system services to autonomously manage cloud and cloud federations resources and support collaboration between cloud users.

3.5. Unconventional/Nature-inspired Programming

Facing the complexity of the emerging ICT landscape in which highly heterogeneous digital services evolve and interact in numerous different ways in an autonomous fashion, there is a strong need for rethinking programming models. The question is *“what programming paradigm can efficiently and naturally express this great number of interactions arising concurrently on the platform?”*.

It has been suggested [63] that observing nature could be of great interest to tackle the problem of modeling and programming complex computing platforms, and overcome the limits of traditional programming models. Innovating unconventional programming paradigms are requested to provide a high-level view of these interactions, then allowing to clearly separate what is a matter of expression from what is a question of implementation. Towards this, nature is of high inspiration, providing examples of self-organising, fully decentralized coordination of complex and large scale systems.

As an example, chemical computing [66] has been proposed more than twenty years ago for a natural way to program parallelism. Even after significant spread of this approach, it appears today that chemical computing exposes a lot of good properties (implicit autonomy, decentralization, and parallelism) to be leveraged for programming service infrastructures.

The Myriads team will investigate nature-inspired programming such as chemical computing for autonomous service computing.

TRISKELL Project-Team

3. Scientific Foundations

3.1. Model Driven Engineering for Distributed Software

Objects, design patterns, software components, contracts, aspects, models, UML, product lines

3.1.1. Software Product Lines

It is seldom the case nowadays that we can any longer deliver software systems with the assumption that one-size-fits-all. We have to handle many variants accounting not only for differences in product functionalities (range of products to be marketed at different prices), but also for differences in hardware (e.g.; graphic cards, display capacities, input devices), operating systems, localization, user preferences for GUI (“skins”). Obviously, we do not want to develop from scratch and independently all of the variants the marketing department wants. Furthermore, all of these variant may have many successive versions, leading to a two-dimensional vision of product-lines.

3.1.2. Object-Oriented Software Engineering

The object-oriented approach is now widespread for the analysis, the design, and the implementation of software systems. Rooted in the idea of modeling (through its origin in Simula), object-oriented analysis, design and implementation takes into account the incremental, iterative and evolutive nature of software development [76], [71]: large software system are seldom developed from scratch, and maintenance activities represent a large share of the overall development effort.

In the object-oriented standard approach, objects are instances of classes. A class encapsulates a single abstraction in a modular way. A class is both *closed*, in the sense that it can be readily instantiated and used by clients objects, and *open*, that is subject to extensions through inheritance [79].

3.1.3. Design Pattern

Since by definition objects are simple to design and understand, complexity in an object-oriented system is well known to be in the *collaboration* between objects, and large systems cannot be understood at the level of classes and objects. Still these complex collaborations are made of recurring patterns, called design patterns. The idea of systematically identifying and documenting design patterns as autonomous entities was born in the late 80's. It was brought into the mainstream by such people as Beck, Ward, Coplien, Booch, Kerth, Johnson, etc. (known as the Hillside Group). However the main event in this emerging field was the publication, in 1995, of the book *Design Patterns: Elements of Reusable Object Oriented Software* by the so-called Gang of Four (GoF), that is E. Gamma, R. Helm, R. Johnson and J. Vlissides [75]. Today, design patterns are widely accepted as useful tools for guiding and documenting the design of object-oriented software systems. Design patterns play many roles in the development process. They provide a common vocabulary for design, they reduce system complexity by naming and defining abstractions, they constitute a base of experience for building reusable software, and they act as building blocks from which more complex designs can be built. Design patterns can be considered reusable micro-architectures that contribute to an overall system architecture. Ideally, they capture the intent behind a design by identifying the component objects, their collaborations, and the distribution of responsibilities. One of the challenges addressed in the Triskell project is to develop concepts and tools to allow their formal description and their automatic application.

3.1.4. Component

The object concept also provides the basis for *software components*, for which Szyperski's definition [86] is now generally accepted, at least in the industry:

A software component is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to composition by third party.

Component based software relies on assemblies of components. Such assemblies rely in turn on fundamental mechanisms such as precise definitions of the mutual responsibility of partner components, interaction means between components and their non-component environment and runtime support (e.g. .Net, EJB, Corba Component Model CCM, OSGI or Fractal).

Components help reducing costs by allowing reuse of application frameworks and components instead of redeveloping applications from scratch (product line approach). But more important, components offer the possibility to radically change the behaviors and services offered by an application by substitution or addition of new components, even a long time after deployment. This has a major impact of software lifecycle, which should now handle activities such as the design of component frameworks, the design of reusable components as deployment units, the validation of component compositions coming from various origins and the component life-cycle management.

Empirical methods without real component composition models have appeared during the emergence of a real component industry (at least in the Windows world). These methods are now clearly the cause of untractable validation and of integration problems that can not be transposed to more critical systems (see for example the accidental destruction of Ariane 501 [78]).

Providing solutions for formal component composition models and for verifiable quality (notion of *trusted components*) are especially relevant challenges. Also the methodological impact of component-based development (for example within the maturity model defined by the SEI) is also worth attention.

3.1.5. Contracts

Central to this trusted component notion is the idea of *contract*. A software contract captures mutual requirements and benefits among stake-holder components, for example between the client of a service and its suppliers (including subcomponents). Contracts strengthen and deepen interface specifications. Along the lines of abstract data type theory, a common way of specifying software contracts is to use boolean assertions called pre- and post-conditions for each service offered, as well as class invariants for defining general consistency properties. Then the contract reads as follows: The client should only ask a supplier for a service in a state where the class invariant and the precondition of the service are respected. In return, the supplier promises that the work specified in the post-condition will be done, and the class invariant is still respected. In this way rights and obligations of both client and supplier are clearly delineated, along with their responsibilities. This idea was first implemented in the Eiffel language [80] under the name *Design by Contract*, and is now available with a range of expressive power into several other programming languages (such as Java) and even in the Unified Modeling Language (UML) with the Object Constraint Language (OCL) [87]. However, the classical predicate based contracts are not enough to describe the requirements of modern applications. Those applications are distributed, interactive and they rely on resources with random quality of service. We have shown that classical contracts can be extended to take care of synchronization and extrafunctional properties of services (such as throughput, delays, etc) [69].

3.1.6. Models and Aspects

As in other sciences, we are increasingly resorting to modelling to master the complexity of modern software development. According to Jeff Rothenberg,

Modeling, in the broadest sense, is the cost-effective use of something in place of something else for some cognitive purpose. It allows us to use something that is simpler, safer or cheaper than reality instead of reality for some purpose. A model represents reality for the given purpose; the model is an abstraction of reality in the sense that it cannot represent all aspects of reality. This allows us to deal with the world in a simplified manner, avoiding the complexity, danger and irreversibility of reality.

So modeling is not just about expressing a solution at a higher abstraction level than code. This has been useful in the past (assembly languages abstracting away from machine code, 3GL abstracting over assembly languages, etc.) and it is still useful today to get a holistic view on a large C++ program. But modeling goes well beyond that.

Modeling is indeed one of the touchstone of any scientific activity (along with validating models with respect to experiments carried out in the real world). Note by the way that the specificity of engineering is that engineers build models of artefacts that usually do not exist yet (with the ultimate goal of building them).

In engineering, one wants to break down a complex system into as many models as needed in order to address all the relevant concerns in such a way that they become understandable enough. These models may be expressed with a general purpose modeling language such as the Unified Modeling Language (UML), or with Domain Specific Languages when it is more appropriate.

Each of these models can be seen as the abstraction of an aspect of reality for handling a given concern. The provision of effective means for handling such concerns makes it possible to establish critical trade-offs early on in the software life cycle, and to effectively manage variation points in the case of product-lines.

Note that in the Aspect Oriented Programming community, the notion of aspect is defined in a slightly more restricted way as the modularization of a cross-cutting concern. If we indeed have an already existing “main” decomposition paradigm (such as object orientation), there are many classes of concerns for which clear allocation into modules is not possible (hence the name “cross-cutting”). Examples include both allocating responsibility for providing certain kinds of functionality (such as logging) in a cohesive, loosely coupled fashion, as well as handling many non-functional requirements that are inherently cross-cutting e.g.; security, mobility, availability, distribution, resource management and real-time constraints.

However now that aspects become also popular outside of the mere programming world [84], there is a growing acceptance for a wider definition where an aspect is a concern that can be modularized. The motivation of these efforts is the systematic identification, modularization, representation, and composition of these concerns, with the ultimate goal of improving our ability to reason about the problem domain and the corresponding solution, reducing the size of software model and application code, development costs and maintenance time.

3.1.7. Design and Aspect Weaving

So really modeling is the activity of separating concerns in the problem domain, an activity also called *analysis*. If solutions to these concerns can be described as aspects, the design process can then be characterized as a weaving of these aspects into a detailed design model (also called the solution space). This is not new: this is actually what designers have been effectively doing forever. Most often however, the various aspects are not *explicit*, or when there are, it is in the form of informal descriptions. So the task of the designer is to do the weaving in her head more or less at once, and then produce the resulting detailed design as a big tangled program (even if one decomposition paradigm, such as functional or object-oriented, is used). While it works pretty well for small problems, it can become a major headache for bigger ones.

Note that the real challenge here is not on how to design the system to take a particular aspect into account: there is a huge design know-how in industry for that, often captured in the form of Design Patterns (see above). Taking into account more than one aspect at the same time is a little bit more tricky, but many large scale successful projects in industry are there to show us that engineers do ultimately manage to sort it out.

The real challenge in a product-line context is that the engineer wants to be able to change her mind on which version of which variant of any particular aspect she wants in the system. And she wants to do it cheaply, quickly and safely. For that, redoing by hand the tedious weaving of every aspect is not an option.

3.1.8. Model Driven Engineering

Usually in science, a model has a different nature than the thing it models (“do not take the map for the reality” as Sun Tse put it many centuries ago). Only in software and in linguistics a model has the same nature as the thing it models. In software at least, this opens the possibility to automatically derive software from its

model. This property is well known from any compiler writer (and others), but it was recently made quite popular with an OMG initiative called the Model Driven Architecture (MDA). This requires that models are no longer informal, and that the weaving process is itself described as a program (which is as a matter of facts an executable meta-model) manipulating these models to produce a detailed design that can ultimately be transformed to code or at least test suites.

The OMG has built a meta-data management framework to support the MDA. It is mainly based on a unique M3 “meta-meta-model” called the Meta-Object Facility (MOF) and a library of M2 meta-models, such as the UML (or SPEM for software process engineering), in which the user can base his M1 model.

The MDA core idea is that it should be possible to capitalize on platform-independent models (PIM), and more or less automatically derive platform-specific models (PSM) –and ultimately code– from PIM through model transformations. But in some business areas involving fault-tolerant, distributed real-time computations, there is a growing concern that the added value of a company not only lies in its know-how of the business domain (the PIM) but also in the design know-how needed to make these systems work in the field (the transformation to go from PIM to PSM). Reasons making it complex to go from a simple and stable business model to a complex implementation include:

- Various modeling languages used beyond UML,
- As many points of views as stakeholders,
- Deliver software for (many) variants of a platform,
- Heterogeneity is the rule,
- Reuse technical solutions across large product lines (e.g. fault tolerance, security, etc.),
- Customize generic transformations,
- Compose reusable transformations,
- Evolve and maintain transformations for 15+ years.

This wider context is now known as Model Driven Engineering.

DREAM Project-Team

3. Scientific Foundations

3.1. Computer assisted monitoring and diagnosis of physical systems

keywords: monitoring, diagnosis, deep model, fault model, simulation, chronicle acquisition

Our work on monitoring and diagnosis relies on model-based approaches developed by the Artificial Intelligence community since the seminal studies by R. Reiter and J. de Kleer [63], [74]. Two main approaches have been proposed then: (i) the consistency-based approach, relying on a model of the expected correct behavior ; (ii) the abductive approach which relies on a model of the failures that might affect the system, and which identifies the failures or the faulty behavior explaining the anomalous observations. See the references [21], [23] for a detailed exposition of these investigations.

Since 1990, the researchers in the field have studied dynamic system monitoring and diagnosis, in a similar way as researchers in control theory do. What characterizes the AI approach is the use of qualitative models instead of quantitative ones and the importance given to the search for the actual source/causes of the faulty behavior. Model-based diagnosis approaches rely on qualitative simulation or on causal graphs in order to look for the causes of the observed deviations. The links between the two communities have been enforced, in particular for what concerns the work about discrete events systems and hybrid systems. Used formalisms are often similar (automata, Petri nets ,...) [28], [27].

Our team focuses on monitoring and on-line diagnosis of discrete events systems and in particular on monitoring by alarm management.

Two different methods have been studied by our team in the last years:

- In the first method, the automaton used as a model is transformed off-line into an automaton adapted to diagnosis. This automaton is called a *diagnoser*. This method has first been proposed by M. Sampath and colleagues [65]. The main drawback of this approach is its centralized nature that requires to explicitly build the global model of the system, which is most of the time unrealistic. It is why we proposed a decentralized approach in [60].
- In the second method, the idea is to associate each failure that we want to detect with a *chronicle* (or a scenario), i.e. a set of observable events interlinked by time constraints. The chronicle recognition approach consists in monitoring and diagnosing dynamic systems by recognizing those chronicles on-line [43], [62], [41].

One of our research focus is to extend the chronicle recognition methods to a distributed context. Local chronicle bases and local recognizers are used to detect and diagnose each component. However, it is important to take into account the interaction model (messages exchanged by the components). Computing a global diagnosis requires then to check the synchronisation constraints between local diagnoses.

Another issue is the chronicle base acquisition. An expert is often needed - create the chronicle base, and that makes the creation and the maintenance of the base very expensive. That is why we are working on an automatic method to acquire the base.

Developing diagnosis methodologies is not enough, especially when on-line monitoring is required. Two related concerns must be tackled, and are the topics of current research in the team:

- The ultimate goal is usually not merely to diagnose, but to put the system back in some acceptable state after the occurrence of a fault. One of our aim is to develop self-healable systems able to self-diagnose and -repair.

- When designing a system and equipping it with diagnosis capabilities, it may be crucial to be able to check off-line that the system will behave correctly, i.e., that the system is actually 'diagnosable'. A lot of techniques have been developed in the past (see Lafortune and colleagues [64]), essentially in automata models. We extended them to cope with temporal patterns. A recent focus has been to study the self-healability of systems (ability to self-diagnose and -repair).

3.2. Machine learning and data mining

keywords: machine learning, Inductive Logic Programming (ILP), temporal data mining, temporal abstraction, data-streams

The machine learning and data mining techniques investigated in the group aim at acquiring and improving models automatically. They belong to the field of machine or artificial learning [38]. In this domain, the goal is the induction or the discovery of hidden objects characterizations from their descriptions by a set of features or attributes. For several years we investigated Inductive Logic Programming (ILP) but now we are also working on data-mining techniques.

We are especially interested in structural learning which aims at making explicit dependencies among data where such links are not known. The relational (temporal or spatial) dimension is of particular importance in applications we are dealing with, such as process monitoring in health-care, environment or telecommunications. Being strongly related to the dynamics of the observed processes, attributes related to temporal or spatial information must be treated in a special way. Additionally, we consider that the legibility of the learned results is of crucial importance as domain experts must be able to evaluate and assess these results.

The discovery of spatial patterns or temporal relations in sequences of events involve two main steps: the choice of a data representation and the choice of a learning technique.

We are mainly interested in symbolic supervised and unsupervised learning methods. Furthermore, we are investigating methods that can cope with temporal or spatial relationships in data. In the sequel, we will give some details about relational learning, relational data-mining and data streams mining.

3.2.1. Relational learning

) Relational learning, also called inductive logic programming (ILP), lies at the intersection of machine learning, logic programming and automated deduction. Relational learning aims at inducing classification or prediction rules from examples and from domain knowledge. As relational learning relies on first order logic, it provides a very expressive and powerful language for representing learning hypotheses especially those learnt from temporal data. Furthermore, domain knowledge represented in the same language can also be used. This is a very interesting feature which enables taking into account already available knowledge and avoids starting learning from scratch.

Concerning temporal data, our work is more concerned with applying relational learning rather than developing or improving the techniques. Nevertheless, as noticed by Page and Srinivasan [59], the target application domains (such as signal processing in health-care) can benefit from adapting relational learning scheme to the particular features of the application data. Therefore, relational learning makes use of constraint programming to infer numerical values efficiently [66]. Extensions, such as QSIM [49], have also been used for learning a model of the behavior of a dynamic system [44]. Precisely, we investigate how to associate temporal abstraction methods to learning and to chronicle recognition. We are also interested in constraint clause induction, particularly for managing temporal aspects. In this setting, the representation of temporal phenomena uses specific variables managed by a constraint system [61] in order to deal efficiently with the associated computations (such as the covering tests).

For environmental data, we have investigated tree structures where a set of attributes describe nodes. Our goal is to find patterns expressed as sub-trees [37] with attribute selectors associated to nodes.

3.2.2. Data mining

Data mining is an unsupervised learning method which aims at discovering interesting knowledge from data. Association rule extraction is one of the most popular approach and has deserved a lot of interest in the last 10 years. For instance, many enhancements have been proposed to the well-known Apriori algorithm [25]. It is based on a level-wise generation of candidate patterns and on efficient candidate pruning having a sufficient relevance, usually related to the frequency of the candidate pattern in the data-set (i.e., the support): the most frequent patterns should be the most interesting. Later, Agrawal and Srikant proposed a framework for "mining sequential patterns" [26], which extends Apriori by coping with the order of elements in patterns.

In [54], Mannila and Toivonen extended the work of Aggrawal et al. by introducing an algorithm for mining patterns involving temporal episodes with a distinction between parallel and sequential event patterns. Later, in [42], Dousson and Vu Duong introduced an algorithm for mining chronicles. Chronicles are sets of events associated with temporal constraints on their occurrences. They generalize the temporal patterns of Mannila and Toivonen. The candidate generation is an Apriori-like algorithm. The chronicle recognizer CRS [40] is used to compute the support of patterns. Then, the temporal constraints are computed as an interval whose bounds are the minimal and the maximal temporal extent of the delay separating the occurrences of two given events in the data-set. Chronicles are very interesting because they can model a system behavior with sufficient precision to compute fine diagnoses. Their extraction from a data-set is reasonably efficient. They can be efficiently recognized on an input data stream.

Relational data-mining [22] can be seen as generalizing these works to first order patterns. In this field, the work of Dehaspe for extracting first-order association rules have strong links with chronicles. Another interesting research concerns inductive databases which aim at giving a theoretical and logical framework to data-mining [50], [39]. In this view, the mining process means to query a database containing raw data as well as patterns that are implicitly coded in the data. The answer to a query is, either the solution patterns that are already present in the database, or computed by a mining algorithm, e.g., Apriori. The original work concerns sequential patterns only [53]. We have investigated an extension of inductive database where patterns are very close to chronicles [69].

3.2.3. Mining data streams

) During the last years, a new challenge has appeared in the data mining community: mining from data streams [24]. Data coming for example from monitoring systems observing patients or from telecommunication systems arrive in such huge volumes that they cannot be stored in totality for further processing: the key feature is that "you get only one look at the data" [46]. Many investigations have been made to adapt existing mining algorithms to this particular context or to propose new solutions: for example, methods for building synopses of past data in the form of or summaries have been proposed, as well as representation models taking advantage of the most recent data. Sequential pattern stream mining is still an issue [55]. At present, research topics such as, sampling, summarizing, clustering and mining data streams are actively investigated.

A major issue in data streams is to take into account the dynamics of process generating data, i.e., the underlying model is evolving and, so, the extracted patterns have to be adapted constantly. This feature, known as *concept drift* [71], [51], occurs within an evolving system when the state of some hidden system variables changes. This is the source of important challenges for data stream mining [45] because it is impossible to store all the data for off-line processing or learning. Thus, changes must be detected on-line and the current mined models must be updated on line as well.

LAGADIC Project-Team

3. Scientific Foundations

3.1. Visual servoing

Basically, visual servoing techniques consist in using the data provided by one or several cameras in order to control the motions of a dynamic system [1]. Such systems are usually robot arms, or mobile robots, but can also be virtual robots, or even a virtual camera. A large variety of positioning tasks, or mobile target tracking, can be implemented by controlling from one to all the degrees of freedom of the system. Whatever the sensor configuration, which can vary from one on-board camera on the robot end-effector to several free-standing cameras, a set of visual features has to be selected at best from the image measurements available, allowing to control the desired degrees of freedom. A control law has also to be designed so that these visual features $\mathbf{s}(t)$ reach a desired value \mathbf{s}^* , defining a correct realization of the task. A desired planned trajectory $\mathbf{s}^*(t)$ can also be tracked. The control principle is thus to regulate to zero the error vector $\mathbf{s}(t) - \mathbf{s}^*(t)$. With a vision sensor providing 2D measurements, potential visual features are numerous, since 2D data (coordinates of feature points in the image, moments, ...) as well as 3D data provided by a localization algorithm exploiting the extracted 2D features can be considered. It is also possible to combine 2D and 3D visual features to take the advantages of each approach while avoiding their respective drawbacks.

More precisely, a set \mathbf{s} of k visual features can be taken into account in a visual servoing scheme if it can be written:

$$\mathbf{s} = \mathbf{s}(\mathbf{x}(\mathbf{p}(t)), \mathbf{a}) \quad (38)$$

where $\mathbf{p}(t)$ describes the pose at the instant t between the camera frame and the target frame, \mathbf{x} the image measurements, and \mathbf{a} a set of parameters encoding a potential additional knowledge, if available (such as for instance a coarse approximation of the camera calibration parameters, or the 3D model of the target in some cases).

The time variation of \mathbf{s} can be linked to the relative instantaneous velocity \mathbf{v} between the camera and the scene:

$$\dot{\mathbf{s}} = \frac{\partial \mathbf{s}}{\partial \mathbf{p}} \dot{\mathbf{p}} = \mathbf{L}_s \mathbf{v} \quad (39)$$

where \mathbf{L}_s is the interaction matrix related to \mathbf{s} . This interaction matrix plays an essential role. Indeed, if we consider for instance an eye-in-hand system and the camera velocity as input of the robot controller, we obtain when the control law is designed to try to obtain an exponential decoupled decrease of the error:

$$\mathbf{v}_c = -\lambda \widehat{\mathbf{L}}_s^+ (\mathbf{s} - \mathbf{s}^*) - \widehat{\mathbf{L}}_s^+ \frac{\partial \mathbf{s}}{\partial t} \quad (40)$$

where λ is a proportional gain that has to be tuned to minimize the time-to-convergence, $\widehat{\mathbf{L}}_s^+$ is the pseudo-inverse of a model or an approximation of the interaction matrix, and $\widehat{\frac{\partial \mathbf{s}}{\partial t}}$ an estimation of the features velocity due to a possible own object motion.

From the selected visual features and the corresponding interaction matrix, the behavior of the system will have particular properties as for stability, robustness with respect to noise or to calibration errors, robot 3D trajectory, etc. Usually, the interaction matrix is composed of highly non linear terms and does not present any decoupling properties. This is generally the case when s is directly chosen as x . In some cases, it may lead to inadequate robot trajectories or even motions impossible to realize, local minimum, tasks singularities, etc. It is thus extremely important to design adequate visual features for each robot task or application, the ideal case (very difficult to obtain) being when the corresponding interaction matrix is constant, leading to a simple linear control system. To conclude in few words, **visual servoing is basically a non linear control problem. Our Holy Grail quest is to transform it into a linear control problem.**

Furthermore, embedding visual servoing in the task function approach allows solving efficiently the redundancy problems that appear when the visual task does not constrain all the degrees of freedom of the system. It is then possible to realize simultaneously the visual task and secondary tasks such as visual inspection, or joint limits or singularities avoidance. This formalism can also be used for tasks sequencing purposes in order to deal with high level complex applications.

3.2. Visual tracking

Elaboration of object tracking algorithms in image sequences is an important issue for researches and applications related to visual servoing and more generally for robot vision. A robust extraction and real time spatio-temporal tracking process of visual cues is indeed one of the keys to success of a visual servoing task. If fiducial markers may still be useful to validate theoretical aspects in modeling and control, natural scenes with non cooperative objects and subject to various illumination conditions have to be considered for addressing large scale realistic applications.

Most of the available tracking methods can be divided into two main classes: feature-based and model-based. The former approach focuses on tracking 2D features such as geometrical primitives (points, segments, circles,...), object contours, regions of interest...The latter explicitly uses a model of the tracked objects. This can be either a 3D model or a 2D template of the object. This second class of methods usually provides a more robust solution. Indeed, the main advantage of the model-based methods is that the knowledge about the scene allows improving tracking robustness and performance, by being able to predict hidden movements of the object, detect partial occlusions and acts to reduce the effects of outliers. The challenge is to build algorithms that are fast and robust enough to meet our applications requirements. Therefore, even if we still consider 2D features tracking in some cases, our researches mainly focus on real-time 3D model-based tracking, since these approaches are very accurate, robust, and well adapted to any class of visual servoing schemes. Furthermore, they also meet the requirements of other classes of application, such as augmented reality.

3.3. Slam

Most of the applications involving mobile robotic systems (ground vehicles, aerial robots, automated submarines,...) require a reliable localization of the robot in its environment. A challenging problem is when neither the robot localization nor the map is known. Localization and mapping must then be considered concurrently. This problem is known as Simultaneous Localization And Mapping (Slam). In this case, the robot moves from an unknown location in an unknown environment and proceeds to incrementally build up a navigation map of the environment, while simultaneously using this map to update its estimated position.

Nevertheless, solving the Slam problem is not sufficient for guaranteeing an autonomous and safe navigation. The choice of the representation of the map is, of course, essential. The representation has to support the different levels of the navigation process: motion planning, motion execution and collision avoidance and, at the global level, the definition of an optimal strategy of displacement. The original formulation of the Slam problem is purely metric (since it basically consists in estimating the Cartesian situations of the robot and a set of landmarks), and it does not involve complex representations of the environment. However, it is now well recognized that **several complementary representations are needed to perform exploration, navigation, mapping, and control tasks successfully. We propose to use composite models of the environment that**

mix topological, metric, and grid-based representations. Each type of representation is well adapted to a particular aspect of autonomous navigation: the metric model allows one to locate the robot precisely and plan Cartesian paths, the topological model captures the accessibility of different sites in the environment and allows a coarse localization, and finally the grid representation is useful to characterize the free space and design potential functions used for reactive obstacle avoidance. However, ensuring the consistency of these various representations during the robot exploration, and merging observations acquired from different viewpoints by several cooperative robots, are difficult problems. This is particularly true when different sensing modalities are involved. New studies to derive efficient algorithms for manipulating the hybrid representations (merging, updating, filtering...) while preserving their consistency are needed.

METISS Project-Team

3. Scientific Foundations

3.1. Introduction

probabilistic modeling, statistical estimation, bayesian decision theory gaussian mixture modeling, Hidden Markov Model, adaptive representation, redundant system, sparse decomposition, sparsity criterion, source separation

Probabilistic approaches offer a general theoretical framework [92] which has yielded considerable progress in various fields of pattern recognition. In speech processing in particular [89], the probabilistic framework indeed provides a solid formalism which makes it possible to formulate various problems of segmentation, detection and classification. Coupled to statistical approaches, the probabilistic paradigm makes it possible to easily adapt relatively generic tools to various applicative contexts, thanks to estimation techniques for training from examples.

A particularly productive family of probabilistic models is the Hidden Markov Model, either in its general form or under some degenerated variants. The stochastic framework makes it possible to rely on well-known algorithms for the estimation of the model parameters (EM algorithms, ML criteria, MAP techniques, ...) and for the search of the best model in the sense of the exact or approximate maximum likelihood (Viterbi decoding or beam search, for example).

More recently, Bayesian networks [94] have emerged as offering a powerful framework for the modeling of musical signals (for instance, [90], [95]).

In practice, however, the use of probabilistic models must be accompanied by a number of adjustments to take into account problems occurring in real contexts of use, such as model inaccuracy, the insufficiency (or even the absence) of training data, their poor statistical coverage, etc...

Another focus of the activities of the METISS research group is dedicated to sparse representations of signals in redundant systems [93]. The use of criteria of sparsity or entropy (in place of the criterion of least squares) to force the unicity of the solution of a underdetermined system of equations makes it possible to seek an economical representation (exact or approximate) of a signal in a redundant system, which is better able to account for the diversity of structures within an audio signal.

The topic of sparse representations opens a vast field of scientific investigation : sparse decomposition, sparsity criteria, pursuit algorithms, construction of efficient redundant dictionaries, links with the non-linear approximation theory, probabilistic extensions, etc... and more recently, compressive sensing [88]. The potential applicative outcomes are numerous.

This section briefly exposes these various theoretical elements, which constitute the fundamentals of our activities.

3.2. Probabilistic approach

probability density function, gaussian model, gaussian mixture model, Hidden Markov Model, Bayesian network, maximum likelihood, maximum a posteriori, EM algorithm, inference, Viterbi algorithm, beam search, classification, hypotheses testing, acoustic parameterisation

For several decades, the probabilistic approaches have been used successfully for various tasks in pattern recognition, and more particularly in speech recognition, whether it is for the recognition of isolated words, for the retranscription of continuous speech, for speaker recognition tasks or for language identification. Probabilistic models indeed make it possible to effectively account for various factors of variability occurring in the signal, while easily lending themselves to the definition of metrics between an observation and the model of a sound class (phoneme, word, speaker, etc...).

3.2.1. Probabilistic formalism and modeling

The probabilistic approach for the representation of an (audio) class X relies on the assumption that this class can be described by a probability density function (PDF) $P(.|X)$ which associates a probability $P(Y|X)$ to any observation Y .

In the field of speech processing, the class X can represent a phoneme, a sequence of phonemes, a word from a vocabulary, or a particular speaker, a type of speaker, a language, Class X can also correspond to other types of sound objects, for example a family of sounds (word, music, applause), a sound event (a particular noise, a jingle), a sound segment with stationary statistics (on both sides of a rupture), etc.

In the case of audio signals, the observations Y are of an acoustical nature, for example vectors resulting from the analysis of the short-term spectrum of the signal (filter-bank coefficients, cepstrum coefficients, time-frequency principal components, etc.) or any other representation accounting for the information that is required for an efficient separation of the various audio classes considered.

In practice, the PDF P is not accessible to measurement. It is therefore necessary to resort to an approximation \hat{P} of this function, which is usually referred to as the likelihood function. This function can be expressed in the form of a parametric model.

The models most used in the field of speech and audio processing are the Gaussian Model (GM), the Gaussian Mixture Model (GMM) and the Hidden Markov Model (HMM). But recently, more general models have been considered and formalised as graphical models.

Choosing a particular family of models is based on a set of considerations ranging from the general structure of the data, some knowledge on the audio class making it possible to size the model, the speed of calculation of the likelihood function, the number of degrees of freedom of the model compared to the volume of training data available, etc.

3.2.2. Statistical estimation

The determination of the model parameters for a given class is generally based on a step of statistical estimation consisting in determining the optimal value for model parameters.

The Maximum Likelihood (ML) criterion is generally satisfactory when the number of parameters to be estimated is small w.r.t. the number of training observations. However, in many applicative contexts, other estimation criteria are necessary to guarantee more robustness of the learning process with small quantities of training data. Let us mention in particular the Maximum a Posteriori (MAP) criterion which relies on a prior probability of the model parameters expressing possible knowledge on the estimated parameter distribution for the class considered. Discriminative training is another alternative to these two criteria, definitely more complex to implement than the ML and MAP criteria.

In addition to the fact that the ML criterion is only one particular case of the MAP criterion, the MAP criterion happens to be experimentally better adapted to small volumes of training data and offers better generalization capabilities of the estimated models (this is measured for example by the improvement of the classification performance and recognition on new data). Moreover, the same scheme can be used in the framework of incremental adaptation, i.e. for the refinement of the parameters of a model using new data observed for instance, in the course of use of the recognition system.

3.2.3. Likelihood computation and state sequence decoding

During the recognition phase, it is necessary to evaluate the likelihood function of the observations for one or several models. When the complexity of the model is high, it is generally necessary to implement fast calculation algorithms to approximate the likelihood function.

In the case of HMM models, the evaluation of the likelihood requires a decoding step to find the most probable sequence of hidden states. This is done by implementing the Viterbi algorithm, a traditional tool in the field of speech recognition. However, when the acoustic models are combined with a syntagmatic model, it is necessary to call for sub-optimal strategies, such as beam search.

3.2.4. Bayesian decision

When the task to solve is the classification of an observation into one class among several closed-set possibilities, the decision usually relies on the maximum a posteriori rule.

In other contexts (for instance, in speaker verification, word-spotting or sound class detection), the problem of classification can be formulated as a binary hypotheses testing problem, consisting in deciding whether the tested observation is more likely to be pertaining to the class under test or not pertaining to it. In this case, the decision consists in acceptance or rejection, and the problem can be theoretically solved within the framework of Bayesian decision by calculating the ratio of the PDFs for the class and the non-class distributions, and comparing this ratio to a decision threshold.

In theory, the optimal threshold does not depend on the class distribution, but in practice the quantities provided by the probabilistic models are not the true PDFs, but only likelihood functions which approximate the true PDFs more or less accurately, depending on the quality of the model of the class.

The optimal threshold must be adjusted for each class by modeling the behaviour of the test on external (development) data.

3.2.5. Graphical models

In the past years, increasing interest has focused on graphical models for multi-source audio signals, such as polyphonic music signals. These models are particularly interesting, since they enable a formulation of music modelisation in a probabilistic framework.

It makes it possible to account for more or less elaborate relationship and dependencies between variables representing multiple levels of description of a music piece, together with the exploitation of various priors on the model parameters.

Following a well-established metaphor, one can say that the graphical model expresses the notion of modularity of a complex system, while probability theory provides the glue whereby the parts are combined. Such a data structure lends itself naturally to the design of efficient general-purpose algorithms.

The graphical model framework provides a way to view a number of existing models (including HMMs) as instances of a common formalism and all of them can be addressed via common machine learning tools.

A first issue when using graphical models is the one of the model design, i.e. the chosen variables for parameterizing the signal, their priors and their conditional dependency structure.

The second problem, called the inference problem, consists in estimating the activity states of the model for a given signal in the maximum a posteriori sense. A number of techniques are available to achieve this goal (sampling methods, variational methods belief propagation, ...), whose challenge is to achieve a good compromise between tractability and accuracy [94].

3.3. Sparse representations

wavelet, dictionary, adaptive decomposition, optimisation, parcimony, non-linear approximation, pursuit, greedy algorithm, computational complexity, Gabor atom, data-driven learning, principal component analysis, independent component analysis

Over the past decade, there has been an intense and interdisciplinary research activity in the investigation of sparsity and methods for sparse representations, involving researchers in signal processing, applied mathematics and theoretical computer science. This has led to the establishment of sparse representations as a key methodology for addressing engineering problems in all areas of signal and image processing, from the data acquisition to its processing, storage, transmission and interpretation, well beyond its original applications in enhancement and compression. Among the existing sparse approximation algorithms, L1-optimisation principles (Basis Pursuit, LASSO) and greedy algorithms (e.g., Matching Pursuit and its variants) have in particular been extensively studied and proved to have good decomposition performance, provided that the sparse signal model is satisfied with sufficient accuracy.

The large family of audio signals includes a wide variety of temporal and frequential structures, objects of variable durations, ranging from almost stationary regimes (for instance, the note of a violin) to short transients (like in a percussion). The spectral structure can be mainly harmonic (vowels) or noise-like (fricative consonants). More generally, the diversity of timbers results in a large variety of fine structures for the signal and its spectrum, as well as for its temporal and frequential envelope. In addition, a majority of audio signals are composite, i.e. they result from the mixture of several sources (voice and music, mixing of several tracks, useful signal and background noise). Audio signals may have undergone various types of distortion, recording conditions, media degradation, coding and transmission errors, etc.

Sparse representations provide a framework which has shown increasingly fruitful for capturing, analysing, decomposing and separating audio signals

3.3.1. Redundant systems and adaptive representations

Traditional methods for signal decomposition are generally based on the description of the signal in a given basis (i.e. a free, generative and constant representation system for the whole signal). On such a basis, the representation of the signal is unique (for example, a Fourier basis, Dirac basis, orthogonal wavelets, ...). On the contrary, an adaptive representation in a redundant system consists of finding an optimal decomposition of the signal (in the sense of a criterion to be defined) in a generating system (or dictionary) including a number of elements (much) higher than the dimension of the signal.

Let y be a monodimensional signal of length T and D a redundant dictionary composed of $N > T$ vectors g_i of dimension T .

$$y = [y(t)]_{1 \leq t \leq T} \quad D = \{g_i\}_{1 \leq i \leq N} \quad \text{with} \quad g_i = [g_i(t)]_{1 \leq t \leq T}$$

If D is a generating system of R^T , there is an infinity of exact representations of y in the redundant system D , of the type:

$$y(t) = \sum_{1 \leq i \leq N} \alpha_i g_i(t)$$

We will denote as $\alpha = \{\alpha_i\}_{1 \leq i \leq N}$, the N coefficients of the decomposition.

The principles of the adaptive decomposition then consist in selecting, among all possible decompositions, the best one, i.e. the one which satisfies a given criterion (for example a sparsity criterion) for the signal under consideration, hence the concept of adaptive decomposition (or representation). In some cases, a maximum of T coefficients are non-zero in the optimal decomposition, and the subset of vectors of D thus selected are referred to as the basis adapted to y . This approach can be extended to approximate representations of the type:

$$y(t) = \sum_{1 \leq i \leq M} \alpha_{\phi(i)} g_{\phi(i)}(t) + e(t)$$

with $M < T$, where ϕ is an injective function of $[1, M]$ in $[1, N]$ and where $e(t)$ corresponds to the error of approximation to M terms of $y(t)$. In this case, the optimality criterion for the decomposition also integrates the error of approximation.

3.3.2. Sparsity criteria

Obtaining a single solution for the equation above requires the introduction of a constraint on the coefficients α_i . This constraint is generally expressed in the following form :

$$\alpha^* = \arg \min_{\alpha} F(\alpha)$$

Among the most commonly used functions, let us quote the various functions L_γ :

$$L_\gamma(\alpha) = \left[\sum_{1 \leq i \leq N} |\alpha_i|^\gamma \right]^{1/\gamma}$$

Let us recall that for $0 < \gamma < 1$, the function L_γ is a sum of concave functions of the coefficients α_i . Function L_0 corresponds to the number of non-zero coefficients in the decomposition.

The minimization of the quadratic norm L_2 of the coefficients α_i (which can be solved in an exact way by a linear equation) tends to spread the coefficients on the whole collection of vectors in the dictionary. On the other hand, the minimization of L_0 yields a maximally parsimonious adaptive representation, as the obtained solution comprises a minimum of non-zero terms. However the exact minimization of L_0 is an untractable NP-complete problem.

An intermediate approach consists in minimizing norm L_1 , i.e. the sum of the absolute values of the coefficients of the decomposition. This can be achieved by techniques of linear programming and it can be shown that, under some (strong) assumptions the solution converges towards the same result as that corresponding to the minimization of L_0 . In a majority of concrete cases, this solution has good properties of sparsity, without reaching however the level of performance of L_0 .

Other criteria can be taken into account and, as long as the function F is a sum of concave functions of the coefficients α_i , the solution obtained has good properties of sparsity. In this respect, the entropy of the decomposition is a particularly interesting function, taking into account its links with the information theory.

Finally, let us note that the theory of non-linear approximation offers a framework in which links can be established between the sparsity of exact decompositions and the quality of approximate representations with M terms. This is still an open problem for unspecified redundant dictionaries.

3.3.3. Decomposition algorithms

Three families of approaches are conventionally used to obtain an (optimal or sub-optimal) decomposition of a signal in a redundant system.

The “Best Basis” approach consists in constructing the dictionary D as the union of B distinct bases and then to seek (exhaustively or not) among all these bases the one which yields the optimal decomposition (in the sense of the criterion selected). For dictionaries with tree structure (wavelet packets, local cosine), the complexity of the algorithm is quite lower than the number of bases B , but the result obtained is generally not the optimal result that would be obtained if the dictionary D was taken as a whole.

The “Basis Pursuit” approach minimizes the norm L_1 of the decomposition resorting to linear programming techniques. The approach is of larger complexity, but the solution obtained yields generally good properties of sparsity, without reaching however the optimal solution which would have been obtained by minimizing L_0 .

The “Matching Pursuit” approach consists in optimizing incrementally the decomposition of the signal, by searching at each stage the element of the dictionary which has the best correlation with the signal to be decomposed, and then by subtracting from the signal the contribution of this element. This procedure is repeated on the residue thus obtained, until the number of (linearly independent) components is equal to the dimension of the signal. The coefficients α can then be reevaluated on the basis thus obtained. This greedy algorithm is sub-optimal but it has good properties for what concerns the decrease of the error and the flexibility of its implementation.

Intermediate approaches can also be considered, using hybrid algorithms which try to seek a compromise between computational complexity, quality of sparsity and simplicity of implementation.

3.3.4. Dictionary construction

The choice of the dictionary D has naturally a strong influence on the properties of the adaptive decomposition : if the dictionary contains only a few elements adapted to the structure of the signal, the results may not be very satisfactory nor exploitable.

The choice of the dictionary can rely on a priori considerations. For instance, some redundant systems may require less computation than others, to evaluate projections of the signal on the elements of the dictionary. For this reason, the Gabor atoms, wavelet packets and local cosines have interesting properties. Moreover, some general hint on the signal structure can contribute to the design of the dictionary elements : any knowledge on the distribution and the frequential variation of the energy of the signals, on the position and the typical duration of the sound objects, can help guiding the choice of the dictionary (harmonic molecules, chirplets, atoms with predetermined positions, ...).

Conversely, in other contexts, it can be desirable to build the dictionary with data-driven approaches, i.e. training examples of signals belonging to the same class (for example, the same speaker or the same musical instrument, ...). In this respect, Principal Component Analysis (PCA) offers interesting properties, but other approaches can be considered (in particular the direct optimization of the sparsity of the decomposition, or properties on the approximation error with M terms) depending on the targeted application.

In some cases, the training of the dictionary can require stochastic optimization, but one can also be interested in EM-like approaches when it is possible to formulate the redundant representation approach within a probabilistic framework.

Extension of the techniques of adaptive representation can also be envisaged by the generalization of the approach to probabilistic dictionaries, i.e. comprising vectors which are random variables rather than deterministic signals. Within this framework, the signal $y(t)$ is modeled as the linear combination of observations emitted by each element of the dictionary, which makes it possible to gather in the same model several variants of the same sound (for example various waveforms for a noise, if they are equivalent for the ear). Progress in this direction are conditioned to the definition of a realistic generative model for the elements of the dictionary and the development of effective techniques for estimating the model parameters.

3.3.5. Compressive sensing

The theoretical results around sparse representations have laid the foundations for a new research field called compressed sensing, emerging primarily in the USA. Compressed sensing investigates ways in which we can sample signals at roughly the lower information rate rather than the standard Shannon-Nyquist rate for sampled signals.

In a nutshell, the principle of Compressed Sensing is, at the acquisition step, to use as samples a number of random linear projections. Provided that the underlying phenomenon under study is sufficiently sparse, it is possible to recover it with good precision using only a few of the random samples. In a way, Compressed Sensing can be seen as a generalized sampling theory, where one is able to trade bandwidth (i.e. number of samples) with computational power. There are a number of cases where the latter is becoming much more accessible than the former; this may therefore result in a significant overall gain, in terms of cost, reliability, and/or precision.

MIMETIC Team

3. Scientific Foundations

3.1. Biomechanics and Motion Control

Human motion control is a very complex phenomenon that involves several layered systems, as shown in figure 3 . Each layer of this controller is responsible for dealing with perceptual stimuli in order to decide the actions that should be applied to the human body and his environment. Due to the intrinsic complexity of the information (internal representation of the body and mental state, external representation of the environment) used to perform this task, it is almost impossible to model all the possible states of the system. Even for simple problems, there generally exist infinity of solutions. For example, from the biomechanical point of view, there are much more actuators (i.e. muscles) than degrees of freedom leading to infinity of muscle activation patterns for a unique joint rotation. From the reactive point of view there exist infinity of paths to avoid a given obstacle in navigation tasks. At each layer, the key problem is to understand how people select one solution among these infinite state spaces. Several scientific domains have addressed this problem with specific points of view, such as physiology, biomechanics, neurosciences and psychology.

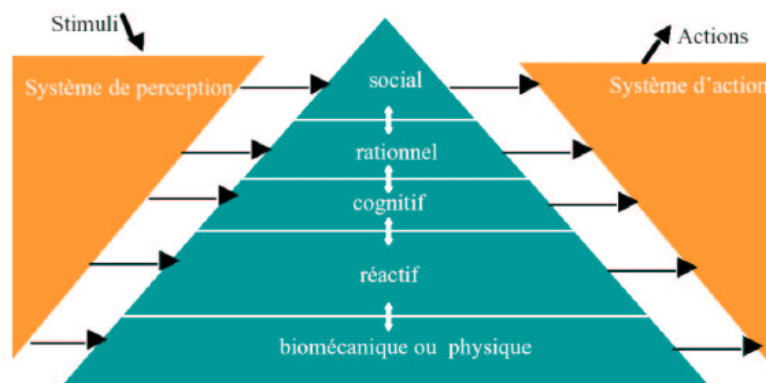


Figure 3. Layers of the motion control natural system in humans.

In biomechanics and physiology, researchers have proposed hypotheses based on accurate joint modeling (to identify the real anatomical rotational axes), energy minimization, force and torques minimization, comfort maximization (i.e. avoiding joint limits), and physiological limitations in muscle force production. All these constraints have been used in optimal controllers to simulate natural motions. The main problem is thus to define how these constraints are composed altogether such as searching the weights used to linearly combine these criteria in order to generate a natural motion. Musculoskeletal models are stereotyped examples for which there exist infinity of muscle activation patterns, especially when dealing with antagonist muscles. An unresolved problem is to define how using the above criteria to retrieve the actual activation patterns while optimization approaches still lead to unrealistic ones. It is still an open problem that will require multidisciplinary skills including computer simulation, constraint solving, biomechanics, optimal control, physiology and neurosciences.

In neuroscience, researchers have proposed other theories, such as coordination patterns between joints driven by simplifications of the variables used to control the motion. The key idea is to assume that instead of controlling all the degrees of freedom, people control higher level variables which correspond to combination of joint angles. In walking, data reduction techniques such as Principal Component Analysis have shown that lower-limb joint angles are generally projected on a unique plan whose angle in the state space is associated with energy expenditure. Although there exists knowledge on specific motion, such as locomotion or grasping, this type of approach is still difficult to generalize. The key problem is that many variables are coupled and it is very difficult to objectively study the behavior of a unique variable in various motor tasks. Computer simulation is a promising method to evaluate such type of assumptions as it enables to accurately control all the variables and to check if it leads to natural movements.

Neurosciences also address the problem of coupling perception and action by providing control laws based on visual cues (or any other senses), such as determining how the optical flow is used to control direction in navigation tasks, while dealing with collision avoidance or interception. Coupling of the control variables is enhanced in this case as the state of the body is enriched by the big amount of external information that the subject can use. Virtual environments inhabited with autonomous characters whose behavior is driven by motion control assumptions is a promising approach to solve this problem. For example, an interesting problem in this field is navigation in an environment inhabited with other people. Typically, avoiding static obstacles together with other people displacing into the environment is a combinatory problem that strongly relies on the coupling between perception and action.

One of the main objectives of MimeTIC is to enhance knowledge on human motion control by developing innovative experiments based on computer simulation and immersive environments. To this end, designing experimental protocols is a key point and some of the researchers in MimeTIC have developed this skill in biomechanics and perception-action coupling. Associating these researchers to experts in virtual human simulation, computational geometry and constraints solving enable us to contribute to enhance fundamental knowledge in human motion control.

3.2. Experiments in Virtual Reality

Understanding interaction between humans is very challenging because it addresses many complex phenomena including perception, decision-making, cognition and social behaviors. Moreover, all these phenomena are difficult to isolate in real situations, it is thus very complex to understand the influence of each of them on the interaction. It is then necessary to find an alternative solution that can standardize the experiments and that allows the modification of only one parameter at a time. Video was first used since the displayed experiment is perfectly repeatable and cut-offs (stop the video at a specific time before its end) allow having temporal information. Nevertheless, the absence of adapted viewpoint and stereoscopic vision does not provide depth information that are very meaningful. Moreover, during video recording session, the real human is acting in front of a camera and not an opponent. The interaction is then not a real interaction between humans.

Virtual Reality (VR) systems allow full standardization of the experimental situations and the complete control of the virtual environment. It is then possible to modify only one parameter at a time and observe its influence on the perception of the immersed subject. VR can then be used to understand what information are picked up to make a decision. Moreover, cut-offs can also be used to obtain temporal information about when these information are picked up. When the subject can moreover react as in real situation, his movement (captured in real time) provides information about his reactions to the modified parameter. Not only is the perception studied, but the complete perception-action loop. Perception and action are indeed coupled and influence each other as suggested by Gibson in 1979.

Finally, VR allows the validation of the virtual human models. Some models are indeed based on the interaction between the virtual character and the other humans, such as a walking model. In that case, there are two ways to validate it. First, they can be compared to real data (e.g. real trajectories of pedestrians). But such data are not always available and are difficult to get. The alternative solution is then to use VR. The validation of the realism of the model is then done by immersing a real subject in a virtual environment in which a virtual

character is controlled by the model. Its evaluation is then deduced from how the immersed subject reacts when interacting with the model and how realistic he feels the virtual character is.

3.3. Computational geometry

Computational geometry is a branch of computer science devoted to the study of algorithms which can be stated in terms of geometry. It aims at studying algorithms for combinatorial, topological and metric problems concerning sets of points in Euclidian spaces. Combinatorial computational geometry focuses on three main problem classes: static problems, geometric query problems and dynamic problems.

In static problems, some input is given and the corresponding output needs to be constructed or found. Such problems include linear programming, Delaunay triangulations, and Euclidian shortest paths for instance. In geometric query problems, commonly known as geometric search problems, the input consists of two parts: the search space part and the query part, which varies over the problem instances. The search space typically needs to be preprocessed, in a way that multiple queries can be answered efficiently. Some typical problems are range searching, point location in a partitioned space, nearest neighbor queries for instance. In dynamic problems, the goal is to find an efficient algorithm for finding a solution repeatedly after each incremental modification of the input data (addition, deletion or motion of input geometric elements). Algorithms for problems of this type typically involve dynamic data structures. Both of previous problem types can be converted into a dynamic problem, for instance, maintaining a Delaunay triangulation between moving points.

The Mimetic team works on problems such as crowd simulation, spatial analysis, path and motion planning in static and dynamic environments, camera planning with visibility constraints for instance. The core of those problems, by nature, relies on problems and techniques belonging to computational geometry. Proposed models pay attention to algorithms complexity to propose models compatible with performance constraints imposed by interactive applications.

SIROCCO Project-Team

3. Scientific Foundations

3.1. Introduction

The research activities on analysis, compression and communication of visual data mostly rely on tools and formalisms from the areas of statistical image modelling, of signal processing, of coding and information theory. However, the objective of better exploiting the Human Visual System (HVS) properties in the above goals also pertains to the areas of perceptual modelling and cognitive science. Some of the proposed research axes are also based on scientific foundations of computer vision (e.g. multi-view modelling and coding). We have limited this section to some tools which are central to the proposed research axes, but the design of complete compression and communication solutions obviously rely on a large number of other results in the areas of motion analysis, transform design, entropy code design, etc which cannot be all described here.

3.2. Parameter estimation and inference

Bayesian estimation, Expectation-Maximization, stochastic modelling

Parameter estimation is at the core of the processing tools studied and developed in the team. Applications range from the prediction of missing data or future data, to extracting some information about the data in order to perform efficient compression. More precisely, the data are assumed to be generated by a given stochastic data model, which is partially known. The set of possible models translates the a priori knowledge we have on the data and the best model has to be selected in this set. When the set of models or equivalently the set of probability laws is indexed by a parameter (scalar or vectorial), the model is said parametric and the model selection resorts to estimating the parameter. Estimation algorithms are therefore widely used at the encoder in order to analyze the data. In order to achieve high compression rates, the parameters are usually not sent and the decoder has to jointly select the model (i.e. estimate the parameters) and extract the information of interest.

3.3. Data Dimensionality Reduction

manifolds, locally linear embedding, non-negative matrix factorization, principal component analysis

A fundamental problem in many data processing tasks (compression, classification, indexing) is to find a suitable representation of the data. It often aims at reducing the dimensionality of the input data so that tractable processing methods can then be applied. Well-known methods for data dimensionality reduction include the principal component analysis (PCA) and independent component analysis (ICA). The methodologies which will be central to several proposed research problems will instead be based on sparse representations, on locally linear embedding (LLE) and on the “non negative matrix factorization” (NMF) framework.

The objective of *sparse representations* is to find a sparse approximation of a given input data. In theory, given $A \in \mathbb{R}^{m \times n}$, $m < n$, and $\mathbf{b} \in \mathbb{R}^m$ with $m \ll n$ and A is of full rank, one seeks the solution of $\min\{\|\mathbf{x}\|_0 : A\mathbf{x} = \mathbf{b}\}$, where $\|\mathbf{x}\|_0$ denotes the L_0 norm of x , i.e. the number of non-zero components in x . There exist many solutions x to $Ax = b$. The problem is to find the sparsest, the one for which x has the fewest non zero components. In practice, one actually seeks an approximate and thus even sparser solution which satisfies $\min\{\|\mathbf{x}\|_0 : \|A\mathbf{x} - \mathbf{b}\|_p \leq \rho\}$, for some $\rho \geq 0$, characterizing an admissible reconstruction error. The norm p is usually 2, but could be 1 or ∞ as well. Except for the exhaustive combinatorial approach, there is no known method to find the exact solution under general conditions on the dictionary A . Searching for this sparsest representation is hence unfeasible and both problems are computationally intractable. Pursuit algorithms have been introduced as heuristic methods which aim at finding approximate solutions to the above problem with tractable complexity.

Non negative matrix factorization (NMF) is a non-negative approximate data representation³. NMF aims at finding an approximate factorization of a non-negative input data matrix V into non-negative matrices W and H , where the columns of W can be seen as *basis vectors* and those of H as coefficients of the linear approximation of the input data. Unlike other linear representations like principal component analysis (PCA) and independent component analysis (ICA), the non-negativity constraint makes the representation purely additive. Classical data representation methods like PCA or Vector Quantization (VQ) can be placed in an NMF framework, the differences arising from different constraints being placed on the W and H matrices. In VQ, each column of H is constrained to be unary with only one non-zero coefficient which is equal to 1. In PCA, the columns of W are constrained to be orthonormal and the rows of H to be orthogonal to each other. These methods of data-dependent dimensionality reduction will be at the core of our visual data analysis and compression activities.

3.4. Perceptual Modelling

Saliency, visual attention, cognition

The human visual system (HVS) is not able to process all visual information of our visual field at once. To cope with this problem, our visual system must filter out the irrelevant information and reduce redundant information. This feature of our visual system is driven by a selective sensing and analysis process. For instance, it is well known that the greatest visual acuity is provided by the fovea (center of the retina). Beyond this area, the acuity drops down with the eccentricity. Another example concerns the light that impinges on our retina. Only the visible light spectrum lying between 380 nm (violet) and 760 nm (red) is processed. To conclude on the selective sensing, it is important to mention that our sensitivity depends on a number of factors such as the spatial frequency, the orientation or the depth. These properties are modeled by a sensitivity function such as the Contrast Sensitivity Function (CSF).

Our capacity of analysis is also related to our visual attention. Visual attention which is closely linked to eye movement (note that this attention is called overt while the covert attention does not involve eye movement) allows us to focus our biological resources on a particular area. It can be controlled by both top-down (i.e. goal-directed, intention) and bottom-up (stimulus-driven, data-dependent) sources of information⁴. This detection is also influenced by prior knowledge about the environment of the scene⁵. Implicit assumptions related to Prior knowledge or beliefs form play an important role in our perception (see the example concerning the assumption that light comes from above-left). Our perception results from the combination of prior beliefs with data we gather from the environment. A Bayesian framework is an elegant solution to model these interactions⁶. We define a vector \vec{v}_l of local measurements (contrast of color, orientation, etc.) and vector \vec{v}_c of global and contextual features (global features, prior locations, type of the scene, etc.). The salient locations S for a spatial position \vec{x} are then given by:

$$S(\vec{x}) = \frac{1}{p(\vec{v}_l | \vec{v}_c)} \times p(s, \vec{x} | \vec{v}_c) \quad (41)$$

The first term represents the bottom-up saliency. It is based on a kind of contrast detection, following the assumption that rare image features are more salient than frequent ones. Most of existing computational models of visual attention rely on this term. However, different approaches exist to extract the local visual features as well as the global ones. The second term is the contextual priors. For instance, given a scene, it indicates which parts of the scene are likely the most salient.

³D.D. Lee and H.S. Seung, "Algorithms for non-negative matrix factorization", *Nature* 401, 6755, (Oct. 1999), pp. 788-791.

⁴L. Itti and C. Koch, "Computational Modelling of Visual Attention", *Nature Reviews Neuroscience*, Vol. 2, No. 3, pp. 194-203, 2001.

⁵J. Henderson, "Regarding scenes", *Directions in Psychological Science*, vol. 16, pp. 219-222, 2007.

⁶L. Zhang, M. Tong, T. Marks, H. Shan, H. and G.W. Cottrell, "SUN: a Bayesian framework for saliency using natural statistics", *Journal of Vision*, vol. 8, pp. 1-20, 2008.

3.5. Coding theory

OPTA limit (Optimum Performance Theoretically Attainable), Rate allocation, Rate-Distortion optimization, lossy coding, joint source-channel coding multiple description coding, channel modelization, oversampled frame expansions, error correcting codes

Source coding and channel coding theory ⁷ is central to our compression and communication activities, in particular to the design of entropy codes and of error correcting codes. Another field in coding theory which has emerged in the context of sensor networks is Distributed source coding (DSC). It refers to the compression of correlated signals captured by different sensors which do not communicate between themselves. All the signals captured are compressed independently and transmitted to a central base station which has the capability to decode them jointly. DSC finds its foundation in the seminal Slepian-Wolf⁸ (SW) and Wyner-Ziv⁹ (WZ) theorems. Let us consider two binary correlated sources X and Y . If the two coders communicate, it is well known from Shannon's theory that the minimum lossless rate for X and Y is given by the joint entropy $H(X, Y)$. Slepian and Wolf have established in 1973 that this lossless compression rate bound can be approached with a vanishing error probability for long sequences, even if the two sources are coded separately, provided that they are decoded jointly and that their correlation is known to both the encoder and the decoder.

In 1976, Wyner and Ziv considered the problem of coding of two correlated sources X and Y , with respect to a fidelity criterion. They have established the rate-distortion function $R_{*X|Y}(D)$ for the case where the side information Y is perfectly known to the decoder only. For a given target distortion D , $R_{*X|Y}(D)$ in general verifies $R_{X|Y}(D) \leq R_{*X|Y}(D) \leq R_X(D)$, where $R_{X|Y}(D)$ is the rate required to encode X if Y is available to both the encoder and the decoder, and R_X is the minimal rate for encoding X without SI. These results give achievable rate bounds, however the design of codes and practical solutions for compression and communication applications remain a widely open issue.

⁷T. M. Cover and J. A. Thomas, Elements of Information Theory, Second Edition, July 2006.

⁸D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources." IEEE Transactions on Information Theory, 19(4), pp. 471-480, July 1973.

⁹A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder." IEEE Transactions on Information Theory, pp. 1-10, January 1976.

TEXMEX Project-Team

3. Scientific Foundations

3.1. Image description

In most contexts where images are to be compared, a direct comparison is impossible. Images are compressed in different formats, most formats are error-prone, images are re-sized, cropped, etc. The solution consists in computing descriptors, which are invariant to these transformations.

The first description methods associate a unique global descriptor with each image, *e.g.*, a color histogram or correlogram, a texture descriptor. Such descriptors are easy to compute and use, but they usually fail to handle cropping and cannot be used for object recognition. The most successful approach to address a large class of transformations relies on the use of local descriptors, extracted on regions of interest detected by a detector, for instance the Harris detector [82] or the Difference of Gaussian method proposed by David Lowe [84].

The detectors select a square, circular or elliptic region that is described in turn by a patch descriptor, usually referred to as a local descriptor. The most established description method, namely the SIFT descriptor [84], was shown robust to geometric and photometric transforms. Each local SIFT descriptor captures the information provided by the gradient directions and intensities in the region of interest in each region of a 4×4 grid, thereby taking into account the spatial organization of the gradient in a region. As a matter of fact, the SIFT descriptor has become a standard for image and video description.

Local descriptors can be used in many applications: image comparison for object recognition, image copy detection, detection of repeats in television streams, etc. While they are very reliable, local descriptors are not without problems. As many descriptors can be computed for a single image, a collection of one million images generates in the order of a billion descriptors. That is why specific indexing techniques are required. The problem of taking full advantage of these strong descriptors on a large scale is still an open and active problem. Most of the recent techniques consists in computing a global descriptor from local ones, such as proposed in the so-called bag-of-visual-word approach [89]. Recently, global description computed from local descriptors has been shown successful in breaking the complexity problem. We are active in designing methods that aggregate local descriptors into a single vector representation without losing too much of the discriminative power of the descriptors.

3.2. Corpus-based text description and machine learning

Our work on textual material (textual documents, transcriptions of speech documents, captions in images or videos, etc.) is characterized by a chiefly corpus-based approach, as opposed to an introspective one. A corpus is for us a huge collection of textual documents, gathered or used for a precise objective. We thus exploit specialized (abstracts of biomedical articles, computer science texts, etc.) or non specialized (newspapers, broadcast news, etc.) collections for our various studies. In TEXMEX, according to our applications, different kinds of knowledge can be extracted from the textual material. For example, we automatically extract terms characteristic of each successive topic in a corpus with no a priori knowledge; we produce representations for documents in an indexing perspective [88]; we acquire lexical resources from the collections (morphological families, semantic relations, translation equivalences, etc.) in order to better grasp relations between segments of texts in which a same idea is expressed with different terms or in different languages...

In the domain of the corpus-based text processing, many researches have been undergone in the last decade. While most of them are essentially based on statistical methods, symbolic approaches also present a growing interest [78]. For our various problems involving language processing, we use both approaches, making the most of existing machine learning techniques or proposing new ones. Relying on advantages of both methods, we aim at developing machine learning solutions that are automatic and generic enough to make it possible to extract, from a corpus, the kind of elements required by a given task.

3.3. Stochastic models for multimodal analysis

Describing multimedia documents, *i.e.*, documents that contain several modalities (*e.g.*, text, images, sound) requires taking into account all modalities, since they contain complementary pieces of information. The problem is that the various modalities are only weakly synchronized, they do not have the same rate and combining the information that can be extracted from them is not obvious. Of course, we would like to find generic ways to combine these pieces of information. Stochastic models appear as a well-dedicated tool for such combinations, especially for image and sound information.

Markov models are composed of a set of states, of transition probabilities between these states and of emission probabilities that provide the probability to emit a given symbol at a given state. Such models allow generating sequences. Starting from an initial state, they iteratively emit a symbol and then switch in a subsequent state according to the respective probability distributions. These models can be used in an indirect way. Given a sequence of symbols (called observations), hidden Markov models (HMMs, [87]) aim at finding the best sequence of states that can explain this sequence. The Viterbi algorithm provides an optimal solution to this problem.

For such HMMs, the structure and probability distributions need to be a priori determined. They can be fixed manually (this is the case for the structure: number of states and their topology), or estimated from example data (this is often the case for the probability distributions). Given a document, such an HMM can be used to retrieve its structure from the features that can be extracted. As a matter of fact, these models allow an audiovisual analysis of the videos, the symbols being composed of a video and an audio component.

Two of the main drawbacks of the HMMs is that they can only emit a unique symbol per state, and that they imply that the duration in a given state follows an exponential distribution. Such drawbacks can be circumvented by segment models [86]. These models are an extension of HMMs where each state can emit several symbols and contains a duration model that governs the number of symbols emitted (or observed) for this state. Such a scheme allows us to process features at different rates.

Bayesian networks are an even more general model family. Static Bayesian networks [80] are composed of a set of random variables linked by edges indicating their conditional dependency. Such models allow us to learn from example data the distributions and links between the variables. A key point is that both the network structure and the distributions of the variables can be learned. As such, these networks are difficult to use in the case of temporal phenomena. Dynamic Bayesian [85] networks are a generalization of the previous models. Such networks are composed of an elementary network that is replicated at each time stamp. Duration variable can be added in order to provide some flexibility on the time processing, like it was the case with segment models. While HMMs and segment models are well suited for dense segmentation of video streams, Bayesian networks offer better capabilities for sparse event detection. Defining a trash state that corresponds to non event segments is a well known problem in speech recognition: computing the observation probabilities in such a state is very difficult.

3.4. Multidimensional indexing techniques

Techniques for indexing multimedia data are needed to preserve the efficiency of search processes as soon as the data to search in becomes large in volume and/or in dimension. These techniques aim at reducing the number of I/Os and CPU cycles needed to perform a search. Multi-dimensional indexing methods either perform exact nearest neighbor (NN) searches or approximate NN-search schemes. Often, approximate techniques are faster as speed is traded off against accuracy.

Traditional multidimensional indexing techniques typically group high dimensional features vectors into cells. At querying time, few such cells are selected for searching, which, in turn, provides performance as each cell contains a limited number of vectors [79]. Cell construction strategies can be classified in two broad categories: *data-partitioning* indexing methods that divide the data space according to the distribution of data, and *space-partitioning* indexing methods that divide the data space along predefined lines and store each descriptor in the appropriate cell.

Unfortunately, the “curse of dimensionality” problem strongly impacts the performance of many techniques. Some approaches address this problem by simply relying on dimensionality reduction techniques. Other approaches abort the search process early, after having accessed an arbitrary and predetermined number of cells. Some other approaches improve their performance by considering approximations of cells (with respect to their true geometry for example).

Recently, several approaches make use of quantization operations. This, somehow, transforms costly nearest neighbor searches in multidimensional space into efficient uni-dimensional accesses. One seminal approach, the LSH technique [81], uses a structured scalar quantizer made of projections on segmented random lines, acting as spatial locality sensitive hash-functions. In this approach, several hash functions are used such that co-located vectors are likely to collide in buckets. Other approaches use unstructured quantization schemes, sometimes together with a vector aggregation mechanism [89] to boost performance.

3.5. Data mining methods

Data Mining (DM) is the core of knowledge discovery in databases whatever the contents of the databases are. Here, we focus on some aspects of DM we use to describe documents and to retrieve information. There are two major goals to DM: description and prediction. The descriptive part includes unsupervised and visualization aspects while prediction is often referred to as supervised mining.

The description step very often includes feature extraction and dimensional reduction. As we deal mainly with contingency tables crossing "documents and words", we intensively use factorial correspondence analysis. "Documents" in this context can be a text as well as an image.

Correspondence analysis is a descriptive/exploratory technique designed to analyze simple two-way and multi-way tables containing some measure of correspondence between the rows and columns. The results provide information, which is similar in nature to those produced by factor analysis techniques, and they allow one to explore the structure of categorical variables included in the table. The most common kind of table of this type is the two-way frequency cross-tabulation table. There are several parallels in interpretation between correspondence analysis and factor analysis: suppose one could find a lower-dimensional space, in which to position the row points in a manner that retains all, or almost all, of the information about the differences between the rows. One could then present all information about the similarities between the rows in a simple 1, 2, or 3-dimensional graph. The presentation and interpretation of very large tables could greatly benefit from the simplification that can be achieved via correspondence analysis (CA).

One of the most important concepts in CA is inertia, *i.e.*, the dispersion of either row points or column points around their gravity center. The inertia is linked to the total Pearson χ^2 for the two-way table. Some rows and/or some columns will be more important due to their quality in a reduced dimensional space and their relative inertia. The quality of a point represents the proportion of the contribution of that point to the overall inertia that can be accounted for by the chosen number of dimensions. However, it does not indicate whether or not, and to what extent, the respective point does in fact contribute to the overall inertia (χ^2 value). The relative inertia represents the proportion of the total inertia accounted for by the respective point, and it is independent of the number of dimensions chosen by the user. We use the relative inertia and quality of points to characterize clusters of documents. The outputs of CA are generally very large. At this step, we use different visualization methods to focus on the most important results of the analysis.

In the supervised classification task, a lot of algorithms can be used; the most popular ones are the decision trees and more recently the Support Vector Machines (SVM). SVMs provide very good results in supervised classification but they are used as "black boxes" (their results are difficult to explain). We use graphical methods to help the user understanding the SVM results, based on the data distribution according to the distance to the separating boundary computed by the SVM and another visualization method (like scatter matrices or parallel coordinates) to try to explain this boundary. Other drawbacks of SVM algorithms are their computational cost and large memory requirement to deal with very large datasets. We have developed a set of incremental and parallel SVM algorithms to classify very large datasets on standard computers.

VR4I Team

3. Scientific Foundations

3.1. Panorama

Our main concern is to allow real users to interact naturally within shared virtual environments as interaction can be the result of an individual interaction of one user with one object or a common interaction of several users on the same object. The long-term purpose of the project is to propose interaction modalities within virtual environments that bring **acting in Virtual Reality as natural as acting in reality**.

Complex physically based models have to be proposed to represent the virtual environment, complex multi-modal interaction models have to be proposed to represent natural activity and complex collaborative environments have to be proposed to ensure effective collaborative interactions.

The long term objectives of VR4i are:

- Improving the accuracy of the virtual environment representation for more interactivity and better perception of the environment;
- Improving the multi-modal interaction for more natural interactions and better perception of the activity;
- Improving the use of virtual environments for real activity and open to human science for evaluation and to engineering science for applications.

Thus, we propose three complementary research axes:

- Physical modeling and simulation of the environment
- Multimodal immersive interaction
- Collaborative work in Collaborative Virtual Environments (CVE)

3.2. Physical modeling and simulation

The first aspect is the modeling and the simulation of the virtual world that represents properly the physical behavior of the virtual world that sustains a natural interaction through the different devices. The main challenge is the search of the trade-off between accuracy and performance to allow effective manipulation, in interactive time, by the user. This trade-off is a key point while the user closes the interaction loop. Namely, the accuracy of the simulation drives the quality of the phenomenon to perceive and the performance drives the sensori-motor feelings of the user. Proposing new controlled algorithms for physical based simulation of the virtual world is certainly a key point for meeting this trade-off. We believe that the mechanical behavior of objects as to be more studied and to be as close as possible to their real behavior. The devices may act as a both way filter on the action and on the perception of the simulated world, but improving the representation of rigid objects submitted to contact, of deformable objects, of changing state object and of environments that include mixed rigid and deformable objects is needed in order to compute forces and positions that have a physical meaning. The interaction between tools and deformable objects is still a challenge in assembly applications and in medical applications. The activity of the user in interaction with the immersive environment will allow to provide method to qualify the quality of the environment and of the interaction by proposing a bio-mechanical user's Alter Ego. We believe that the analysis of the forces involved during an immersive activity will give us keys to design more acceptable environments. As the goal is to achieve more and more accurate simulation that will require more and more computation time, the coupling between physical modeling and related simulation algorithms is of first importance. Looking for genericity will ensure correct deployment on new advanced hardware platforms that we will use to ensure adapted performance. The main aim of this topic is to improve the simulation accuracy satisfying the simulation time constraints for improving the naturalness of interactions.

3.3. Multimodal immersive interaction

The second aspect concerns the design and evaluation of novel approaches for multimodal immersive interaction with virtual environments.

We aim at improving capabilities of selection and manipulation of virtual objects, as well as navigation in the virtual scene and control of the virtual application. We target a wide spectrum of sensory modalities and interfaces such as tangible devices, haptic interfaces (force-feedback, tactile feedback), visual interfaces (e.g., gaze tracking), locomotion and walking interfaces, and brain-computer interfaces. We consider this field as a strong scientific and technological challenge involving advanced user interfaces, but also as strongly related to user's perceptual experience. We promote a perception-based approach for multimodal interaction, based on collaborations with laboratories of the Perception and Neuroscience research community.

The introduction of a third dimension when interacting with a virtual environment makes inappropriate most of the classical techniques used successfully in the field of 2D interaction with desktop computers up to now. Thus, it becomes successfully used to design and evaluate new paradigms specifically oriented towards interaction within 3D virtual environments.

We aim at improving the immersion of VR users by offering them natural ways for navigation, interaction and application control, as these are the three main tasks within 3D virtual environments. Here we consider interactions as multimodal interactions, as described in the previous section. We also want to make the users forget their physical environment in benefit of the virtual environment that surrounds them and contribute to improve the feeling of immersion and of presence. To achieve this goal, we must ensure that users can avoid collisions with their surrounding real environment (the screens of the rendering system, the walls of the room) and can avoid lost of interaction tracking (keeping the user within the range of the physical interaction devices). To do that, we propose to take into account the surrounding real physical environment of the user and to include it in the virtual environment through a virtual representation. This explicit model of the real environment of the users will help users to forget it: throughout this model, the user will be aware (with visual, auditive or haptic feedback) of these virtual objects when he comes near their boundaries. We also have to investigate which physical limitations are the most important ones to perceive, and what are the best ways to make the users aware of their physical limitations.

3.4. Collaborative work in CVE's

The third aspect is to propose Collaborative Virtual Environments for several local or distant users. In these environments, distant experts could share their expertise for project review, for collaborative design or for analysis of data resulting from scientific computations in HPC context. Sharing the virtual environment is certainly a key point that leads to propose new software architectures ensuring the data distribution and the synchronization of the users.

In terms of interaction, new multi-modal interaction metaphors have to be proposed to tackle with the awareness of other users' activity. Here it is important to see a virtual representation of the other users, of their activity, and of the range of their action field, in order to better understand both their potential and their limitation for collaboration: what they can see, what they can reach, what their interaction tools are and which possibilities they offer.

Simultaneous collaborative interactions upon the same data through local representations of these data should be tackled by new generic algorithms dedicated to consistency management. Some solutions have to be proposed for distant collaboration, where it is not possible any more to share tangible devices to synchronize co-manipulation: we should offer some new haptic rendering to enforce users' coordination. Using physics engines for realistic interaction with virtual objects is also a challenge if we want to offer low latency feedback to the users. Indeed, the classical centralized approach for physics engines is not able to offer fast feedback to distant users, so this approach must be improved.