RESEARCH CENTER
Nancy - Grand Est

FIELD

Activity Report 2012

Section Scientific Foundations

Edition: 2013-04-24

CAMUS Team

3. Scientific Foundations

3.1. Research directions

The various objectives we are expecting to reach are directly related to the search of adequacy between the sofware and the new multicore processors evolution. They also correspond to the main research directions suggested by Hall, Padua and Pingali in [43]. Performance, correction and productivity must be the users' perceived effects. They will be the consequences of research works dealing with the following issues:

- Issue 1: Static parallelization and optimization
- Issue 2: Profiling and execution behavior modeling
- Issue 3: Dynamic program parallelization and optimization, virtual machine
- Issue 4: Object-oriented programming and compiling for multicores
- Issue 5: Proof of program transformations for multicores

Efficient and correct applications development for multicore processors needs stepping in every application development phase, from the initial conception to the final run.

Upstream, all potential parallelism of the application has to be exhibited. Here static analysis and transformation approaches (issue 1) must be processed, resulting in a *multi-parallel* intermediate code advising the running virtual machine about all the parallelism that can be taken advantage of. However the compiler does not have much knowledge about the execution environment. It obviously knows the instruction set, it can be aware of the number of available cores, but it does not know the effective available resources at any time during the execution (memory, number of free cores, etc.).

That is the reason why a "virtual machine" mechanism will have to adapt the application to the resources (issue 3). Moreover the compiler will be able to take advantage only of a part of the parallelism induced by the application. Indeed some program information (variables values, accessed memory adresses, etc.) being available only at runtime, another part of the available parallelism will have to be generated on-the-fly during the execution, here also, thanks to a dynamic mechanism.

This on-the-fly parallelism extraction will be performed using speculative behavior models (issue 2), such models allowing to generate speculative parallel code (issue 3). Between our behavior modeling objectives, we can add the behavior monitoring, or profiling, of a program version. Indeed current and future architectures complexity avoids assuming an optimal behavior regarding a given program version. A monitoring process will allow to select on-the-fly the best parallelization.

These different parallelizing steps are schematized on figure 1 .

The more and more widespread usage of object-oriented approaches and languages emphasizes the need for specific multicore programming tools. The object and method formalism implies specific execution schemes that translate in the final binary by quite distant elementary schemes. Hence the execution behavior control is far more difficult. Analysis and optimization, either static or dynamic, must take into account from the outset this distortion between object-oriented specification and final binary code: how can object or method parallelization be translated (issue 4).

Our project lies on the conception of a production chain for efficient execution of an application on a multicore architecture. Each link of this chain has to be formally verified in order to ensure correction as well as efficiency. More precisely, it has to be ensured that the compiler produces a correct intermediate code, and that the virtual machine actually performs the parallel execution semantically equivalent to the source code: every transformation applied to the application, either statically by the compiler or dynamically by the virtual machine, must preserve the initial semantics. They must be proved formally (issue 5).

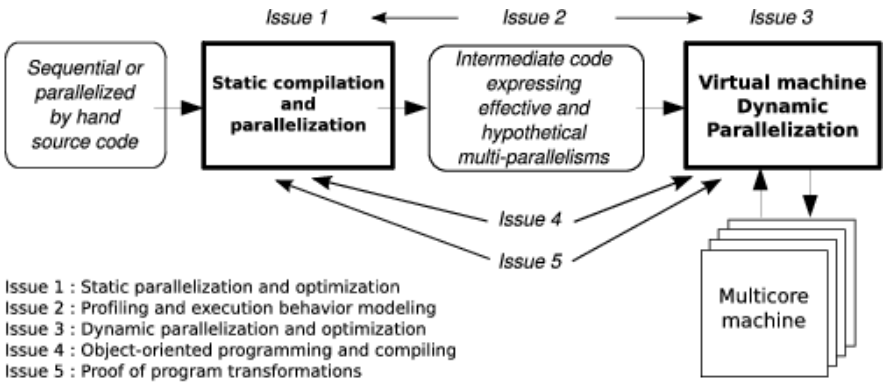

Figure 1. Automatic parallelizing steps for multicore architectures

In the following, those different issues are detailed while forming our global and long term vision of what has to be done.

3.2. Static parallelization and optimization

Participants: Vincent Loechner, Philippe Clauss, Éric Violard, Alexandra Jimborean.

Static optimizations, from source code at compile time, benefit from two decades of research in automatic parallelization: many works address the parallelization of loop nests accessing multi-dimensional arrays, and these works are now mature enough to generate efficient parallel code [26]. Low-level optimizations, in the assembly code generated by the compiler, have also been extensively dealt for single-core and require few adaptations to support multicore architectures. Concerning multicore specific parallelization, we propose to explore two research directions to take full advantage of these architectures. They are described below.

3.2.1. State of the art

Upstream, an easy interprocedural dependence analysis allows to handle complete programs (but recursivity: recursive functions must be transformed into iterative functions). Concerning iterative control we will use the polyhedral model, a formalism developped these last two decades, which allows to represent the execution of a loop nest by scanning a polytope.

When compiling an application, if it contains loop nests with affine bounds accessing scalars or arrays accessed using affine functions, the polyhedral model allows to:

- compute the dependence graph, which describes the order in which the dependent instructions must be executed [34];
- generate a schedule, which extracts some parallelism from the dependence graph [35], [36];
- generate an allocation, which assigns a processor (or a core) to a set of iterations of the loop nest to be scanned.

This last allocation step needs a thorough knowledge of the target architecture, as many crucial choices will result in performance hazards: for example, the volume and flow of inter-processor communications and synchronization; the data locality and the effects of the TLB (Translation Lookaside Buffer) and the various cache levels and distributions; or the register allocation optimizations. There are many techniques to control these parameters, and each architecture needs specific choices, of a valid schedule, of a parallel loop iterations distribution (bloc-, cyclic-, or tiled), of a loop-unrolling factor, as well as a memory data layout and a prefetch strategy (when available). They require powerful mathematical tools, such as counting the number of integer points contained in a parametric polytope.

Our own contributions in this area are significant. Concerning schedule and data placement, we proposed new advances in minimizing the number of communications for parallel architectures [54] and in cache access optimizations [53] [8]. We also proposed essential advances in parametric polytope manipulation [9], [5], developped the first algorithm to count integer points in a parametric polytope as an Ehrhart polynomial [3], and proposed successive improvements of this algorithm [10] [65]. We implemented these results in the free software *PolyLib*, utilized by many researchers around the world.

3.2.2. Adapting parallelization to multicore architecture

The first research direction to be explored is multicore specific efficient optimizations. Indeed, multicore architectures need specific optimizations, or we will get underlinear accelerations, or even decelerations. Multicore architectures may have the following properties: specific memory hierarchy, with distributed low-level cache and (possibly semi-) shared high level caches; software-controlled memory hierarchies (memory hints, local stores or scratchpads for example); optimized access to contiguous memory addresses or to separate memory banks; SIMD or vectorial execution in groups of cores, and synchronous execution; higher register allocation pressure when several threads use the same hardware (as in GPGPUs for example); etc.

A schedule and an allocation must be chosen wisely in order to obtain good performances. On NVIDIA GPGPUs, using the CUDA language, Baskaran et al. [25] obtained interesting results that have been implemented in their PLuTo compiler framework. However, they are based on many empirical and imprecise techniques, and require simulations to fine-tune the optimizations: they can be improved. Memory hierarchy efficient control is a cornerstone of tomorrow's multicore architectures performance. Compiler-optimizers have to evolve to meet this requirement.

Simulation and (partial-) profiling may however remain necessary in some cases, when static analysis reaches its intrinsic limits: when the execution of a program depends on dynamic parameters, when it uses complex pointer arithmetic, or when it performs indirect array accesses for example (as is often the case in *while* loops, out of the scope of the classical polyhedral model). In these cases, the compiler should rely on the profiler, and generate a code that interacts with the dynamic optimizer. This is the link with issues 2 and 3 of this research project.

3.2.3. Expressing many potential parallelisms

The dynamic optimizer (issue 3) must be able to exploit various parallel codes to compare them and the best one to choose, possibly swapping from a code to another during execution. The compiler must therefore generate different potentially efficient versions of a code, depending on fixed parameters such as the schedule or the data layout, and dynamic parameters such as the tile size or the unrolling factor.

The compiler then generates many variants of *effective* parallelism, formally proved by the static analyzer. It may also generate variants of code that have not been formally validated, due to the analyzer limits, and that have to be checked during execution by the dynamic optimizer: *hypothetical* parallelism. Hypothetical parallelism could be expressed as a piece of code, valid under certain conditions. Effective and hypothetical parallelisms are called *potential parallelism*. The variants of potential parallelism will be expressed in an intermediate language that has to be discovered.

Using compiler directives is an interesting way to define this intermediate language. Among the usual directives, we distinguish schedule directives for shared memory architectures (such as the OpenMP [1] *parallel* directive), and placement directives for distributed memory architectures (for example the HPF[2] *ALIGN* directive). These two types of directives are conjointly necessary to take full profit of multicore architectures. However, we have to study their complementarity and solve the interdependence or conflict that may arise between them. Moreover, new directives should allow to control data transfers between different levels of the memory hierarchy.

[1] http://www.openmp.org
[2] http://hpff.rice.edu

We are convinced that the definition of such a language is required in the next advances in compilation for multicore architectures, and there does not exist such an ambitious project to our knowledge. The OpenCL project [3], presented as an general-purpose and efficient multicore programming environment, is too low-level to be exploitable. We propose to define a new high level language based on compilation directives, that could be used by the skilled programmer or automatically generated by a compiler-optimizer (like OpenMP, recently integrated in the *gcc* compiler suite).

3.3. Profiling and execution behavior modeling

Participants: Alain Ketterlin, Philippe Clauss, Aravind Sukumaran-Rajam.

The increasing complexity of programs and hardware architectures makes it ever harder to characterize beforehand a given program's run time behavior. The sophistication of current compilers and the variety of transformations they are able to apply cannot hide their intrinsic limitations. As new abstractions like transactional memories appear, the dynamic behavior of a program strongly conditions its observed performance. All these reasons explain why empirical studies of sequential and parallel program executions have been considered increasingly relevant. Such studies aim at characterizing various facets of one or several program runs, *e.g.*, memory behavior, execution phases, etc. In some cases, such studies characterize more the compiler than the program itself. These works are of tremendous importance to highlight all aspects that escape static analysis, even though their results may have a narrow scope, due to the possible incompleteness of their input data sets.

3.3.1. Selective profiling and interaction with the compiler

In its simplest form, studying a given program's run time behavior consists in collecting and aggregating statistics, *e.g.*, counting how many times routines or basic blocks are executed, or counting the number of cache misses during a certain portion of the execution. In some cases, data can be collected about more abstract events, like the garbage-collector frequency or the number and sizes of sent and received messages. Such measures are relatively easy to obtain, are frequently used to quantify the benefits of some optimization, and may suggest some way to improve performance. These techniques are now well-known, but mostly for sequential programs.

These global studies have often been complemented by local, targeted techniques focused on some program portions, *e.g.*, where static techniques remain inconclusive for some fixed duration. These usages of profiling are usually strongly related to the optimization they complement, and are set up either by the compiler or by the execution environment. Their results may be used immediately at run time, in which case they are considered a form of run time optimization [1]. They can also be used offline to provide hints to a subsequent compilation cycle, in which case they constitute a form of profile-guided compilation, a strategy that is common in general purpose compilers.

For instance, in the context where a set of possible parallelizations have been provided by the compiler (see issue 1), a profiling component can easily be made responsible for testing some relevant condition at run time (*e.g.*, that depends on input data) and for selecting the best between various versions of the code. Beyond such simple tasks, we expect that profiling will, at the beginning of the execution, have enough resources to conduct more elaborate analyzes. We believe that combining an "open" static analysis with an integrated profiling component is a promising approach, first because it may relieve the programmer of a large part of the tedious task of implementing the distribution of computations, and second to free the compiler of the obligation to choose between several optimizations in the absence of enough relevant data. The main open question here is to define precisely the respective roles of the compiler and the profiler, and also the amount and nature of information the former can transmit to the latter.

[3] http://www.khronos.org/opencl

3.3.2. Profiling and dynamic optimization

In the context of dynamic optimization, that is, when the compiler's abilities have been exhausted, a profiler can still do useful work, provided some additional capabilities [1]. If it is able to instrument the code the way, *e.g.*, a PIN-tool does [55], it has access to the whole program, including libraries (or, for example, the code of a low-level library called from a scripting language). This means that it has access to portions of the program that were not under the compiler's control. The profiler can then perform dynamic inter-procedural analyzes, for instance to compute dependencies to detect parallelism that wasn't apparent at compile time because of a function call in the body of a loop. More generally, if the profiler is able to reconstruct at run time some representation of the whole program, as in [74] for example, it is possible to let it search for any construct that can be optimized and/or parallelized in the context of the current execution. Several virtual machines, *e.g.*, for Java or Microsoft CLR, have opened this way of optimizing programs, probably because virtual machines need to maintain an intermediate, structured representation of the running program.

The possibility of running programs on architectures that include a large number of computing cores has given rise to new abstractions [72], [46], [29]. Transactional memories, for instance, aim at simplifying the management of conflicting concurrent accesses to a shared memory, a notoriously difficult problem [48]. However, the performance of a transaction-based application heavily depends on its dynamic behavior, and too many conflicting accesses and rollbacks, severely affect performance. We bet that the need for multicore specific programming tools will lead to other abstractions based on speculative execution. Because of the very nature of speculation, all these abstractions will require run time evaluation, and maybe correction, to avoid pathological cases. The profiler has a central role here, because it can be made responsible for diagnosing inefficient use of speculative execution, and for taking corrective action, which means that it has to be integrated to the execution environment. We also think that the large scope and almost infinite potential uses of a profiling component may well suggest new parallel program abstractions, specially targeted at run time evaluation and adaptation.

3.3.3. Run time program modeling

When profiling goes beyond simple aggregation of counts, it can, for example, sample a program's behavior and split its execution into phases. These phases may help target a subsequent evaluation on a new architecture [66]. When profiling instruments the whole program to obtain a trace, *e.g.*, of memory accesses, it is possible to use this trace for:

- simulation, *e.g.*, by varying the parameters of the memory hierarchy,
- for modeling, *e.g.*, to reconstruct some specific model of the program [74], or to extract dynamic dependencies that help identifying parallel sections [62].

Handling such large execution traces, and especially compressing them, is a research topic by itself [30], [57]. Our contribution to this topic [7] is unusual in that the result of compression is a sequence of loop nests where memory accesses and loop bounds are affine functions of the enclosing loop indices. Modeling a trace this way leads to slightly better average compression rates compared to other, less expressive techniques. But more importantly, it has the advantage to provide a result in symbolic form, and this result can be further analyzed with techniques usually restricted to the static analysis of source code. We plan to apply, in the short term, similar techniques to the modeling of dynamic dependencies, so as to be able to automatically extract parallelism from program traces.

This kind of analysis is representative of a new kind of tools than could be named "parallelization assistants" [52], [62]. Properties that can't be detected by the compiler but that appear to hold in one or several executions of a program can be submitted to the programmer, maybe along a suitable reformulation of its program using some class of abstraction, *e.g.*, compiler directives. The goal is to provide help and guidance in adapting source code, in the same way a classical profiling tool helps pinpoint performance bottlenecks. Control and data dependencies are fundamental to such a tool. An execution trace provides an observed reality; for example a trace of memory addresses. If the observed dynamic dependencies provide a set of constraints, they also suggest a complete family of potential correct executions, be they parallel or sequential, and all these executions are equivalent to the reference execution. Being able to handle large traces, and representing them

in some manageable way, means being able to highlight medium to large grain parallelism, which is especially interesting on multicore architectures and often difficult for compilers to discover, for example because of the use of pointers and the difficulty of eliminating potential aliasing. This can be seen as a machine learning problem, where the goal is to recover a hidden structure from a large sequence of events. This general problem has various incarnations, depending on how much the learner knows about the original program, on the kind of data obtained by profiling, on the class of structures sought, and on the objectives of the analysis. We are convinced that such studies will enrich our understanding of the behavior of programs, and of the programming concepts that are really useful. It will also lead to useful tools, and will open up new directions for dynamic optimization.

3.4. Dynamic parallelization and optimization, virtual machine

Participants: Alexandra Jimborean, Philippe Clauss, Alain Ketterlin, Aravind Sukumaran-Rajam, Vincent Loechner.

This link in the programming chain has become essential with the advent of the new multicore architectures. Still being considered as secondary with mono-core architectures, dynamic analysis and optimization are now one of the keys for controling those new mechanisms complexity. From now on, performed instructions are not only dedicated to the application functionalities, but also to its control and its transformation, and so in its own interest. Behaving like a computer virus, such a process should rather be qualified as a “vitamin”. It perfectly knows the current characteristics of the execution environment and owns some qualitative information thanks to a behavior modeling process (issue 2). It appends a significant part of optimizing ability compared to a static compiler, while observing live resources availability evolution.

3.4.1. State of the art

Dynamic analysis and optimization, that is to say simultaneous to the program execution, have motivated a growing interest during the last decade, mainly because of the hardware architectures and applications growing complexity. Indeed, it has become more and more difficult to anticipate any program run simply from its source code, either because its control structures introduce some unknown objects before run (dynamic memory allocation, pointers, ...), or because the interaction between the target architecture and the program generates unpredictable behaviors. This is notably due to the appearance of more optimizing hardware units (prefetching units, speculative processing, code cache, branch prediction, etc.). With multicore architectures, this interest is growing even more. Works achieved in this area for mono-core processors have permitted to establish some classification of the so-called dynamic approaches, either based on the used methodologies or on the objectives.

The first objective for any dynamic approach is to extract some live information at runtime relying on a profiling process. This essential step is the main objective of issue 2 (see sub-section 3.3).

Identifying some “hotspots” thanks to profiling is then used for performance improvement optimizations. Two main approaches can be distinguished:

- the *profile-guided* approach, where analysis and optimization of profile information are performed off-line, that is to say statically. A first run is only performed to extract information for driving a re-compilation. Related to this approach, *iterative compilation* consists in running a code that has been transformed following different optimization possibilities (nature and sequencing of the applied optimizations), and then in re-compiling the transformed code guided by the collected performance information, and so on until obtaining a “best” program version. In order to promote a rapid convergence towards a better solution, some heuristics or some machine learning mechanisms are used [21], [61], [60]. The main drawback of such approaches relates to the quality of the generated code which depends on the reference profiled execution, and more precisely on the used input data set, but also on the used hardware.
- the *on-the-fly* approach consists in performing all steps at each run (profiling, analysis and transformation). The main constraint of this approach is that the time overhead has to be widely compensated

by the benefits it generates. Several works propose such approaches dedicated to specific optimizations. We personally successfully implemented a dynamic data prefetching system for the Itanium processor [1].

Although all these works provided some efficient dynamic mechanisms, their adaptation to multicore architectures yields difficult issues, and even challenges them. It is indeed necessary to control interactions between simultaneous tasks, imposing an additional complexity level which can be fateful for a dynamic system, while becoming too costly in time and space.

Some dynamic parallelizing techniques have been proposed in the last years. They are mainly focusing on parallelizing loop-nests, as programs generally spend most of their execution time in iterative structures.

The LRPD test [64] is certainly one of the foundation strategies. This method consists in speculatively parallelizing loops. Privatization and reduction transformations are applied to promote a successful application of the strategy. During execution, some tests are performed to verify the speculation validity. In case of invalid speculation, the targeted loop is re-executed sequentially. However, the application range is limited to loops accessing arrays; pointers cannot be handled. Moreover the method is not fully dynamic since an initial static analysis is needed.

In [33], Cintra and Llanos present a speculative parallel execution mechanism for loops, where iteration chunks are executed in sliding windows of n threads. The loops are not transformed and the sequential schedule remains as a reference to define a total order on the speculative threads. In order to verify whether some dependencies are violated during the program run, all data structures qualified as speculative, that is to say those being accessed in read-write mode by the threads, are duplicated for each thread and tagged following those states: *not accessed, modified, exposed loaded* or *exposed loaded and later modified*. For example, a *read-after-write* dependency has been violated if a thread owns a data tagged as *exposed loaded* or *exposed loaded and modified*, and if a predecessor thread, following the sequential total order, owns the same data but tagged as *modified* or *exposed loaded and modified*, while this data has not yet been committed in main memory. Such an approach can be memory-costly as each shared data structure is duplicated. It can be tricky to adjust verification frequencies to minimize time overhead. Some other methods based on the same principle of verifying speculation relatively to the sequential schedule have been proposed recently as in [68], where each iteration of a loop is decomposed into a prologue, a speculative body and an epilogue. The speculative bodies are performed in parallel and each body completion induces a verification. This approach seems to be only well suited for loops which bodies represent significant computation time.

Another recent work is the development of SPICE [63] which is a speculative parallelizing system where an entire first run of a loop is initially observed. This observation serves in determining the values reached by some variables during the run. During a next run of the loop, several speculative threads are launched. They consider as initial values of some variables the values that have been observed at the previous run. If a thread reaches the starting value of another thread, it stops. Thus each thread performs a different portion of the loop. But if the loop behavior changes and if another thread starting value is never reached, the run goes on sequentially until completion.

The main limits of these propositions are:

- they do not alter the initial sequential schedule since always contiguous instruction blocks are speculatively parallelized;
- their underlying parallelism is out of control: the characteristics of the generated parallel schedule are completely unknown since they randomly depend on the program instructions, their dependencies and the target machine. If bad performance is encountered, no other parallelization solution can be proposed. Moreover, the effective instruction schedule occurring at program run can significantly vary from one run to another, hence leading to a confusing performance inconsistency.

A strategy that would uniquely be based on a transactional memory mechanism, with rollbacks in the case of data races, yields a totally uncontrolable parallelism where performance can not be ensured and not even strongly expected.

While being based on efficient prediction mechanisms, a better control over parallelization will permit to provide solutions that are well suited to a varying execution context and to parallelize portions of code that can be parallelized only in some particular context. It is indeed crucial to maximize the potential parallelism of the applications to take advantage of the forthcoming processors comprising several tens of cores.

3.4.2. General objective: building a virtual machine

As it has already been mentioned, dynamic parallelization and optimization can take place inside a virtual machine. All the research objectives that are presented in the following are related to its construction.

Notice that the term of "virtual machine" is employed to group a set of dynamic analysis and optimization mechanisms taking as input a binary code, eventually enriched with specific instructions. We refer to a process virtual machine which main role is dynamic binary optimization from one instruction set to the same instruction set. The taxonomy given in [67] includes this kind of virtual machine.

Notice that this virtual machine can run in parallel on the processor cores during the four initial phases (see figure 2), but also simultaneously to the target application, either by sharing some cores with light processes, or by using cores that are useless for the target application. It will also support a transactional memory mechanism, if available. However the foreseen parallelizing strategies do not depend on such a mechanism since our speculative executions are supposed to be as reliable as possible thanks to efficient prediction models, and since they are supported by a specific and higher level rollback mechanism. Anyway if available, a transactional memory mechanism would allow to take advantage of "nearly perfect" prediction models.

The virtual machine takes as input an intermediate code expressing several kinds of parallelism on several code extracts. Those kinds of parallelism are either effective, that is to say that the corresponding parallel execution is obviously semantically correct, or hypothetical, that is to say that there is still some uncertainty on the parallelism correctness. In this case, this uncertainty will have to be resolved at run time. This intermediate "multi-parallel" code is generated by the static parallelization described subsection 3.2 . It also contains generic descriptions of parallelizing or optimizing transformations which parameters will have to be instanciated by the virtual machine, thanks to its knowledge about the target architecture and the program run-time behavior.

3.4.3. Adaptation of the intermediate code to the target architecture

The virtual machine first phase is to adapt this intermediate code to the target multicore architecture. It consists in answering the following questions:

- What is the suitable kind of parallelism?
- What is the suitable parallel task granularity?
- What is the suitable number of parallel tasks?
- Can we take advantage of a specialized instruction set for some operations?
- What are the parameter values for some parallelization or optimization?

The multi-parallel intermediate code exhibits different parameters allowing to adapt some parallelizing and optimizing transformations to the target architecture. For example, a loop unrolling will be parametrized by the number of iterations to be unrolled. This number will depend, for example, on the number of available registers and the size of the instruction cache. A parallelizing transformation will depend on several possible parallel instruction schedules. One or several schedules will be selected, for example, depending on the kind of memory hierarchy and the cache sharing among cores.

Concerning hypothetical parallelism, this first phase will reduce the number of these propositions to solutions that are well suited to the target architecture. This phase also instruments the intermediate code in order to install the dynamic mechanisms related to profiling and speculative parallel execution.

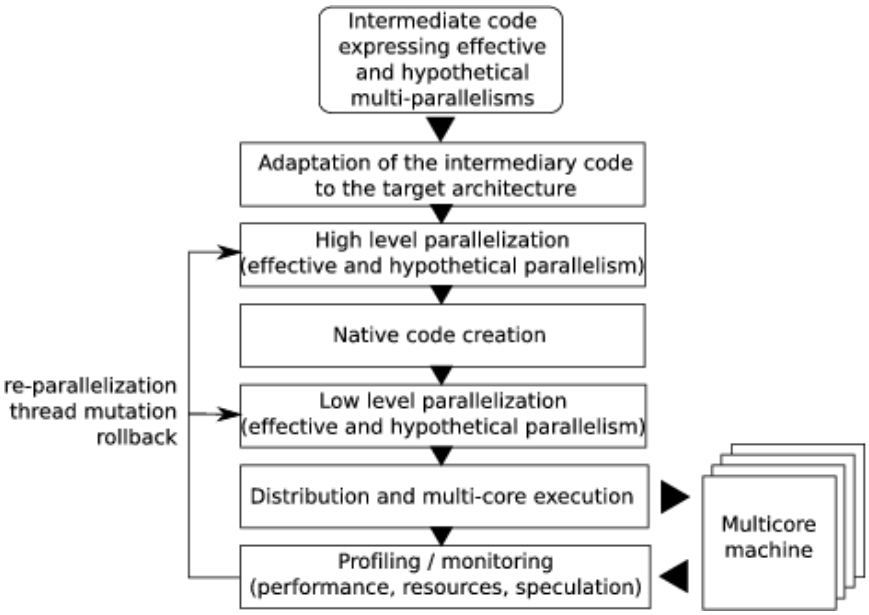

Figure 2. The virtual machine

3.4.4. High level parallelization and native code creation

From these target architecture related adaptations, a parallel intermediate code is generated. It contains instructions that are specific to the dynamic optimizing and parallelizing mechanisms, *i.e.*, instrumentation instructions to feed the profiling process as well as calls to speculative execution management procedures. A translation into native code executable by the target processor follows. This translation also allows to keep trace of the code extracts that have to be modified during the run.

3.4.5. Low level parallelization

The binary version of the code exhibits new parallelism and optimization sources that are specific to the instruction set and to the target architecture capabilities. Moreover, some dynamic optimizations are dedicated to specific instructions, or instruction blocks, as for example the memory reads which time performances can be dynamically improved by data prefetching [1]. Thus the binary code can be transformed and instrumented as well.

3.4.6. Distribution, execution and profiling

The so built executable code is then distributed among the processor cores to be run. During the run, the instrumentation instructions feed the profiler with information for execution monitoring and for behavior models construction (see subsection 3.3). An accurate knowledge of the binary code, thanks to the control of its generation, also permits at this step to dynamically control the insertion or deletion of some instrumentation instructions. Indeed it is important to manage execution monitoring through sampling based instrumentations in varying frequencies, following the changing behavior frequency (see in [1] and [73] a description of this kind of mechanism), as such instrumentations necessarily induce overheads that have to be minimized.

3.4.7. Re-parallelization, thread mutation or rollback

Depending on the information collected from instrumentation, and depending on the built prediction models, the profiling phase causes a re-transformation of some code parts, thus causing the mutation of the concerned

threads. Such re-transformation is done either on the binary code whether it consists in low level and small modifications, as for example the adjustement of a data prefetching distance, or on the intermediate code if it consists in a complete modification of the parallelizing strategy. For example, such a processing will follow the observation of a bad performance, or of a change in the computing resources availability, or will be caused by the completion of a dependency prediction model allowing the generation of a speculative parallelization. From such a speculative execution, a re-transformation can consist in rolling back to a sequential execution version when the considered hypothetical parallelism, and thus the associated prediction model, has been evaluated wrong.

3.5. Proof of program transformations for multicores

Participants: Éric Violard, Julien Narboux, Nicolas Magaud, Vincent Loechner, Alexandra Jimborean.

3.5.1. State of the art

3.5.1.1. Certification of low-level codes.

Among the languages allowing to exploit the power of multicore architectures, some of them supply the programmer a library of functions that corresponds more or less to the features of the target architecture : for example, CUDA [4] for the architectures of type GPGPU and more recently the standard OpenCL [5] that offers a unifying programming interface allowing the use of most of the existing multicore architectures or a use of heterogeneous aggregate of such architectures. The main advantage of OpenCL is that it allows the programmer to write a code that is portable on a large set of architectures (in the same spirit as the MPI library for multi-processor architectures). However, at this low level, the programming model is very close to the executing model, the control of parallelism is explicit. Proof of program correctness has to take into account low-level mechanisms such as hardware interruptions or thread preemption, which is difficult.

In [38], Feng *et al.* propose a logic inspired from the Hoare logic in order to certify such low-level programs with hardware interrupts and preempted threads. The authors specify this logic by using the meta-logic implemented in the Coq proof assistant [24].

3.5.1.2. Certification of a compiler.

The problem here is to prove that transformations or optimizations preserve the operational behaviour of the compiled programs.

Xavier Leroy in [27], [50] formalizes the analyses and optimizations performed by a C compiler: a big part of this compiler is written in the specification language of Coq and the executable (Caml) code of this compiler is obtained by automatic extraction from the specification.

Optimizing compilers are complex softwares, particularly in the case of multi-threaded programs. They apply some subtle code transformations. Therefore some errors in the compiler may occur and the compiler may produce incorrect executable codes. Work is to be done to remedy this problem. The technique of validation *a posteriori* [69], [70] is an interesting alternative to full verification of a compiler.

3.5.1.3. Semantics of directives.

As it was mentioned in subsection 3.2.3 , the use of directives is an interesting approach to adapt languages to multicore architectures. It is a syntactic means to tackle the increasing need of enriching the operational semantics of programs.

Ideally, these directives are only comments: they do not alter the correction of programs and they are a good means to improve their performance. They allow the separation of concerns: *correction* and *efficiency*.

However, using directives in that sense and in the context of automatic parallelization, raises some questions: for example, assuming that directives are not mandatory, how to ensure that directives are really taken into account? How to know if a directive is better than another? What is the impact of a directive on performance?

[4] http://www.nvidia.com/object/cuda_what_is.html
[5] http://www.khronos.org/opencl

In his thesis [40], that was supervised by Éric Violard, Philippe Gerner addresses similar questionings and states a formal framework in which the semantics of compilation directives can be defined. In this framework, any directive is encoded into one equation which is added to an algebraic specification. The semantics of the directives can be precisely defined via an order relation (called relation of *preference*) on the models of this specification.

3.5.1.4. Definition of a parallel programming model.

Classically, the good definition of a programming model is based on a semantic domain and on the definition of a "toy" language associated with a proof system, which allows to prove the correctness of the programs written in that language. Examples of such "toy" languages are CSP for control parallelism and $\mathcal{L}$ [28] for data parallelism. The proof systems associated with these two languages, are extensions of the Hoare logic.

We have done some significant works on the definition of data parallelism [11]. In particular, a crucial problem for the good definition of this programming model, is the semantics of the various syntactic constructs for data locality. We proposed a semantic domain which unifies two concepts: *alignment* (in a data-parallel language like HPF) and *shape* (in the data-parallel extensions of C).

We defined a "toy" language, called PEI, that is made of a small number of syntactic constructs. One of them, called *change of basis*, allows the programmer to exhibit parallelism in the same way as a placement or a scheduling directive [41].

3.5.1.5. Programming models for multicore architectures.

The multicore emergence questions the existing parallel programming models.

For example, with the programming model supported by OpenMP, it is difficult to master both correctness and efficiency of programs. Indeed, this model does not allow programmers to take optimal advantage of the memory hierarchy and some OpenMP directives may induce unpredictable performances or incorrect results.

Nowadays, some new programming models are experienced to help at designing both efficient and correct programs for multicores. Because memory is shared by the cores and its hierarchy has some distributed parts, some works aim at defining a hybrid model, between task parallelism and data parallelism. For example, languages like UPC (Unified Parallel C) [6] or Chapel [7] combine the advantages of several programming paradigms.

In particular, the model of memory transactions (or transactional memory [47]) retains much attention since it offers the programmer a simple operational semantics including a mutual exclusion mechanism which simplifies program design. However, much work remains to define the precise operational meaning of transactions and the interaction with the other languages features [56]. Moreover, this model leaves the compiler a lot of work to reach a safe and efficient execution on the target architecture. In particular, it is necessary to control the atomicity of transactions [39] and to prove that code transformations preserve the operational semantics.

3.5.1.6. Refinement of programs.

Refinement [22], [42] is a classical approach for gradually building correct programs: it consists in transforming an initial specification by successive steps, by verifying that each transformation preserves the correctness of the previous specification. Its basic principle is to derive simultaneously a program and its own proof. It defines a formal framework in which some rules and strategies can be elaborated to transform specifications written by using the same formalism. Such a set of rules is called a *refinement calculus*.

Unity [32] and Gamma [23] are classical examples of such formalisms, but they are not especially designed for refining programs for multicore architectures. Each of these formalisms is associated with a computing model and thus each specification can be viewed as a program. Starting with an initial specification, a proof logic allows a user to derive a specification which is more suited to the target architecture.

[6] http://upc.gwu.edu
[7] http://chapel.cs.washington.edu

Refinement applies for the programming of a large range of problems and architectures. It allows to pass the limitations of the polyhedral model and of automatic parallelization. We designed a refinement calculus to build data parallel programs [71].

3.5.2. Main objective: formal proof of analyses and transformations

Our main objective consists in certifying the critical modules of our optimization tools (the compiler and the virtual machine). First we will prove the main loop transformation algorithms which constitute the core of our system.

The optimization process can be separated into two stages: the transformations consisting in optimizing the sequential code and in exhibiting parallelism, and those consisting in optimizing the parallel code itself. The first category of optimizations can be proved within a sequential semantics. For the other optimizations, we need to work within a concurrent semantics. We expect the first stage of optimizations to produce data-race free code. For the second stage of optimizations, we will first assume that the input code is data-race free. We will prove those transformations using Appel's concurrent separation logic [44]. Proving transformations involving program which are not data-race free will constitute a longer term research goal.

3.5.3. Proof of transformations in the polyhedral model

The main code transformations used in the compiler and the virtual machine are those carried out in the polyhedral model [49], [37]. We will use the Coq proof assistant to formalize proofs of analyses and transformations based on the polyhedral model. In [31], Cachera and Pichardie formalized nested loops in Coq and showed how to prove *properties* of those loops. Our aim is slightly different as we plan to prove *transformations* of nested loops in the polyhedral model. We will first prove the simplest unimodular transformations, and later we will focus on more complex transformations which are specific to multicore architectures. We will first study scheduling optimizations and then optimizations improving data locality.

3.5.4. Validation under hypothesis

In order to prove the correction of a code transformation T it is possible to:

- prove that T is correct in general, *i.e.*, prove that for all x, $T(x)$ is equivalent to x.
- prove *a posteriori* that the applied transformation has been correct in the particular case of a code c.

The second approach relies on the definition of a program called *validator* which verifies if two pieces of program are equivalent. This program can be modeled as a function V such that, given two programs c_1 and c_2, $V(c_1, c_2) = true$ only if c_1 has the same semantics as c_2. This approach has been used in the field of optimizations certification [59], [58]. If the validator itself contains a bug then the certification process is broken. But if the validator is proved formally (as it was achieved by Tristan and Leroy for the Compcert compiler [69], [70]) then we get a transformed program which can be trusted in the same way as if the transformation is proved formally.

This second approach can be used only for the *effective parallelism*, when the static analysis provides enough information to parallelize the code. For the *hypothetical parallelism*, the necessary hypotheses have to be verified at run time.

For instance, the absence of aliases in a piece of code is difficult to decide statically but can be more easily decided at run time.

In this framework, we plan to build a *validator under hypotheses*: a function V' such that, given two programs c_1 and c_2 and an hypothesis H, if $V'(c_1, c_2, H) = true$, then H implies that c_1 has the same semantics as c_2. The validity of the hypothesis H will be verified dynamically by the virtual machine. This verification process, which is part of the virtual machine, will have to be proved as correct as well.

3.5.5. Rejecting incorrect parallelizations

The goal of the project is to exhibit potential parallelism. The source code can contain many sub-routines which could be parallelized under some hypothesis that the static analysis fails to decide. For those optimizations, the virtual machine will have to verify the hypotheses dynamically. Dynamically dealing with the potential parallelism can be complex and costly (profiling, speculative execution with rollbacks). To reduce the overhead of the virtual machine, we will have to provide efficient methods to rule out quickly incorrect parallelism. In this context, we will provide hypotheses which are easy to check dynamically and which can tell when a transformation cannot be applied, *i.e.*, hypotheses which are sufficient conditions for the non-validity of an optimization.

CARAMEL Project-Team

3. Scientific Foundations

3.1. Cryptography, arithmetic: hardware and software

One of the main topics for our project is public-key cryptography. After 20 years of hegemony, the classical public-key algorithms (whose security is based on integer factorization or discrete logarithm in finite fields) are currently being overtaken by elliptic curves. The fundamental reason for this is that the best-known algorithms for factoring integers or for computing discrete logarithms in finite fields have a subexponential complexity, whereas the best known attack for elliptic-curve discrete logarithms has exponential complexity. As a consequence, for a given security level 2^n, the key sizes must grow linearly with n for elliptic curves, whereas they grow like n^3 for RSA-like systems. As a consequence, several governmental agencies, like the NSA or the BSI, now recommend to use elliptic-curve cryptosystems for new products that are not bound to RSA for backward compatibility.

Besides RSA and elliptic curves, there are several alternatives currently under study. There is a recent trend to promote alternate solutions that do not rely on number theory, with the objective of building systems that would resist a quantum computer (in contrast, integer factorization and discrete logarithms in finite fields and elliptic curves have a polynomial-time quantum solution). Among them, we find systems based on hard problems in lattices (NTRU is the most famous), those based on coding theory (McEliece system and improved versions), and those based on the difficulty to solve multivariate polynomial equations (HFE, for instance). None of them has yet reached the same level of popularity as RSA or elliptic curves for various reasons, including the presence of unsatisfactory features (like a huge public key), or the non-maturity (system still alternating between being fixed one day and broken the next day).

Returning to number theory, an alternative to RSA and elliptic curves is to use other curves and in particular genus-2 curves. These so-called hyperelliptic cryptosystems have been proposed in 1989 [17], soon after the elliptic ones, but their deployment is by far more difficult. The first problem was the group law. For elliptic curves, the elements of the group are just the points of the curve. In a hyperelliptic cryptosystem, the elements of the group are points on a 2-dimensional variety associated to the genus-2 curve, called the Jacobian variety. Although there exist polynomial-time methods to represent and compute with them, it took some time before getting a group law that could compete with the elliptic one in terms of speed. Another question that is still not yet fully answered is the computation of the group order, which is important for assessing the security of the associated cryptosystem. This amounts to counting the points of the curve that are defined over the base field or over an extension, and therefore this general question is called point-counting. In the past ten years there have been major improvements on the topic, but there are still cases for which no practical solution is known.

Another recent discovery in public-key cryptography is the fact that having an efficient bilinear map that is hard to invert (in a sense that can be made precise) can lead to powerful cryptographic primitives. The only examples we know of such bilinear maps are associated with algebraic curves, and in particular elliptic curves: this is the so-called Weil pairing (or its variant, the Tate pairing). Initially considered as a threat for elliptic-curve cryptography, they have proven to be quite useful from a constructive point of view, and since the beginning of the decade, hundreds of articles have been published, proposing efficient protocols based on pairings. A long-lasting open question, namely the construction of a practical identity-based encryption scheme, has been solved this way. The first standardization of pairing-based cryptography has recently occurred (see ISO/IEC 14888-3 or IEEE P1363.3), and a large deployment is to be expected in the next years.

Despite the raise of elliptic curve cryptography and the variety of more or less mature other alternatives, classical systems (based on factoring or discrete logarithm in finite fields) are still going to be widely used in the next decade, at least, due to resilience: it takes a long time to adopt new standards, and then an even longer time to renew all the software and hardware that is widely deployed.

This context of public-key cryptography motivates us to work on integer factorization, for which we have acquired expertise, both in factoring moderate-sized numbers, using the ECM (Elliptic Curve Method) algorithm, and in factoring large RSA-like numbers, using the number field sieve algorithm. The goal is to follow the transition from RSA to other systems and continuously assess its security to adjust key sizes. We also want to work on the discrete-logarithm problem in finite fields. This second task is not only necessary for assessing the security of classical public-key algorithms, but is also crucial for the security of pairing-based cryptography.

We also plan to investigate and promote the use of pairing-based and genus-2 cryptosystems. For pairings, this is mostly a question of how efficient can such a system be in software, in hardware, and using all the tools from fast implementation to the search for adequate curves. For genus 2, as said earlier, constructing an efficient cryptosystem requires some more fundamental questions to be solved, namely the point-counting problem.

We summarize in the following table the aspects of public-key cryptography that we address in the CARAMEL team.

public-key primitive	cryptanalysis	design	implementation
RSA	X	–	–
Finite Field DLog	X	–	–
Elliptic Curve DLog	–	–	Soft
Genus 2 DLog	–	X	Soft
Pairings	X	X	Soft/Hard

Another general application for the project is computer algebra systems (CAS), that rely in many places on efficient arithmetic. Nowadays, the objective of a CAS is not only to have more and more features that the user might wish, but also to compute the results fast enough, since in many cases, the CAS are used interactively, and a human is waiting for the computation to complete. To tackle this question, more and more CAS use external libraries, that have been written with speed and reliability as first concern. For instance, most of today's CAS use the GMP library for their computations with big integers. Many of them will also use some external Basic Linear Algebra Subprograms (BLAS) implementation for their needs in numerical linear algebra.

During a typical CAS session, the libraries are called with objects whose sizes vary a lot; therefore being fast on all sizes is important. This encompasses small-sized data, like elements of the finite fields used in cryptographic applications, and larger structures, for which asymptotically fast algorithms are to be used. For instance, the user might want to study an elliptic curve over the rationals, and as a consequence, check its behaviour when reduced modulo many small primes; and then [s]he can search for large torsion points over an extension field, which will involve computing with high-degree polynomials with large integer coefficients.

Writing efficient software for arithmetic as it is used typically in CAS requires the knowledge of many algorithms with their range of applicability, good programming skills in order to spend time only where it should be spent, and finally good knowledge of the target hardware. Indeed, it makes little sense to disregard the specifics of the possible hardware platforms intended, even more so since in the past years, we have seen a paradigm shift in terms of available hardware: so far, it used to be reasonable to consider that an end-user running a CAS would have access to a single-CPU processor. Nowadays, even a basic laptop computer has a multi-core processor and a powerful graphics card, and a workstation with a reconfigurable coprocessor is no longer science-fiction.

In this context, one of our goals is to investigate and take advantage of these influences and interactions between various available computing resources in order to design better algorithms for basic arithmetic objects. Of course, this is not disconnected from the others goals, since they all rely more or less on integer or polynomial arithmetic.

CARTE Project-Team

3. Scientific Foundations

3.1. Computer Virology

From a historical point of view, the first official virus appeared in 1983 on Vax-PDP 11. At the very same time, a series of papers was published which always remains a reference in computer virology: Thompson [70], Cohen [39] and Adleman [29]. The literature which explains and discusses practical issues is quite extensive [43], [45]. However, there are only a few theoretical/scientific studies, which attempt to give a model of computer viruses.

A virus is essentially a self-replicating program inside an adversary environment. Self-replication has a solid background based on works on fixed point in λ-calculus and on studies of von Neumann [75]. More precisely we establish in [35] that Kleene's second recursion theorem [58] is the cornerstone from which viruses and infection scenarios can be defined and classified. The bottom line of a virus behavior is

1. a virus infects programs by modifying them,
2. a virus copies itself and can mutate,
3. it spreads throughout a system.

The above scientific foundation justifies our position to use the word virus as a generic word for self-replicating malwares. There is yet a difference. A malware has a payload, and virus may not have one. For example, worms are an autonous self-replicating malware and so fall into our definition. In fact, the current malware taxonomy (virus, worms, trojans, ...) is unclear and subject to debate.

3.2. Computation over continuous structures

Classical recursion theory deals with computability over discrete structures (natural numbers, finite symbolic words). There is a growing community of researchers working on the extension of this theory to continuous structures arising in mathematics. One goal is to give foundations of numerical analysis, by studying the limitations of machines in terms of computability or complexity, when computing with real numbers. Classical questions are : if a function $f : \mathbb{R} \to \mathbb{R}$ is computable in some sense, are its roots computable? in which time? Another goal is to investigate the possibility of designing new computation paradigms, transcending the usual discrete-time, discrete-space computer model initiated by the Turing machine that is at the base of modern computers.

While the notion of a computable function over discrete data is captured by the model of Turing machines, the situation is more delicate when the data are continuous, and several non-equivalent models exist. In this case, let us mention computable analysis, which relates computability to topology [42], [74]; the Blum-Shub-Smale model (BSS), where the real numbers are treated as elementary entities [34]; the General Purpose Analog Computer (GPAC) introduced by Shannon [68]with continuous time.

3.3. Rewriting

The rewriting paradigm is now widely used for specifying, modelizing, programming and proving. It allows to easily express deduction systems in a declarative way, and to express complex relations on infinite sets of states in a finite way, provided they are countable. Programming languages and environments with a rewriting based semantics have been developed ; see ASF+SDF [36], MAUDE [38], and TOM [64].

For basic rewriting, many techniques have been developed to prove properties of rewrite systems like confluence, completeness, consistency or various notions of termination. Proof methods have also been proposed for extensions of rewriting such as equational extensions, consisting of rewriting modulo a set of axioms, conditional extensions where rules are applied under certain conditions only, typed extensions, where rules are applied only if there is a type correspondence between the rule and the term to be rewritten, and constrained extensions, where rules are enriched by formulas to be satisfied [31], [41], [69].

An interesting aspect of the rewriting paradigm is that it allows automatable or semi-automatable correctness proofs for systems or programs: the properties of rewriting systems as those cited above are translatable to the deduction systems or programs they formalize and the proof techniques may directly apply to them.

Another interesting aspect is that it allows characteristics or properties of the modelled systems to be expressed as equational theorems, often automatically provable using the rewriting mechanism itself or induction techniques based on completion [40]. Note that the rewriting and the completion mechanisms also enable transformation and simplification of formal systems or programs.

Applications of rewriting-based proofs to computer security are various. Approaches using rule-based specifications have recently been proposed for detection of computer viruses [72], [73]. For several years, in our team, we have also been working in this direction. We already have proposed an approach using rewriting techniques to abstract program behaviors for detecting suspicious or malicious programs [32].

CASSIS Project-Team

3. Scientific Foundations

3.1. Introduction

Our main goal is to design techniques and to develop tools for the verification of (safety-critical) systems, such as programs or protocols. To this end, we develop a combination of techniques based on automated deduction for program verification, constraint resolution for test generation, and reachability analysis for the verification of infinite-state systems.

3.2. Automated Deduction

The main goal is to prove the validity of assertions obtained from program analysis. To this end, we develop techniques and automated deduction systems based on rewriting and constraint solving. The verification of recursive data structures relies on inductive reasoning or the manipulation of equations and it also exploits some form of reasoning modulo properties of selected operators (such as associativity and/or commutativity).

Rewriting, which allows us to simplify expressions and formulae, is a key ingredient for the effectiveness of many state-of-the-art automated reasoning systems. Furthermore, a well-founded rewriting relation can be also exploited to implement reasoning by induction. This observation forms the basis of our approach to inductive reasoning, with high degree of automation and the possibility to refute false conjectures.

The constraints are the key ingredient to postpone the activity of solving complex symbolic problems until it is really necessary. They also allow us to increase the expressivity of the specification language and to refine theorem-proving strategies. As an example of this, the handling of constraints for unification problems or for the orientation of equalities in the presence of interpreted operators (e.g., commutativity and/or associativity function symbols) will possibly yield shorter automated proofs.

Finally, decision procedures are being considered as a key ingredient for the successful application of automated reasoning systems to verification problems. A decision procedure is an algorithm capable of efficiently deciding whether formulae from certain theories (such as Presburger arithmetic, lists, arrays, and their combination) are valid or not. We develop techniques to build and to combine decision procedures for the domains which are relevant to verification problems. We also perform experimental evaluation of the proposed techniques by combining propositional reasoning (implemented by means of Boolean solvers, e.g. SAT solvers) and decision procedures to get solvers for the problem of Satisfiability Modulo Theories (SMT).

3.3. Synthesizing and Solving Constraints

Applying constraint logic programming technology in the validation and verification area is currently an active way of research. It usually requires the design of specific solvers to deal with the description language's vocabulary. For instance, we are interested in applying a solver for set constraints [6] to evaluate set-oriented formal specifications. By evaluation, we mean the encoding of the formal model into a constraint system, and the ability for the solver to verify the invariant on the current constraint graph, to propagate preconditions or guards, and to apply a substitution calculus on this graph. The constraint solver is used for animating specifications and automatically generating abstract test cases.

3.4. Rewriting-based Safety Checking

Invariant checking and strenghtening is the dual of reachability analysis, and can thus be used for verifying safety properties of infinite-state systems. In fact, many infinite-state systems are just parameterized systems which become finite state systems when parameters are instantiated. Then, the challenge is to automatically discharge the maximal number of proof obligations coming from the decomposition of the invariance conditions. For parameterized systems, we are interested in a deductive approach where states are defined by first order formulae with equality, and proof obligations are checked by SMT solvers.

PAREO Project-Team

3. Scientific Foundations

3.1. Introduction

It is a common claim that rewriting is ubiquitous in computer science and mathematical logic. And indeed the rewriting concept appears from very theoretical settings to very practical implementations. Some extreme examples are the mail system under Unix that uses rules in order to rewrite mail addresses in canonical forms (see the `/etc/sendmail.cf` file in the configuration of the mail system) and the transition rules describing the behaviors of tree automata. Rewriting is used in semantics in order to describe the meaning of programming languages [28] as well as in program transformations like, for example, re-engineering of Cobol programs [36]. It is used in order to compute, implicitly or explicitly as in Mathematica or MuPAD, but also to perform deduction when describing by inference rules a logic [24], a theorem prover [26] or a constraint solver [27]. It is of course central in systems making the notion of rule an explicit and first class object, like expert systems, programming languages based on equational logic, algebraic specifications, functional programming and transition systems.

In this context, the study of the theoretical foundations of rewriting have to be continued and effective rewrite based tools should be developed. The extensions of first-order rewriting with higher-order and higher-dimension features are hot topics and these research directions naturally encompass the study of the rewriting calculus, of polygraphs and of their interaction. The usefulness of these concepts becomes more clear when they are implemented and a considerable effort is thus put nowadays in the development of expressive and efficient rewrite based programming languages.

3.2. Rule-based programming languages

Programming languages are formalisms used to describe programs, applications, or software which aim to be executed on a given hardware. In principle, any Turing complete language is sufficient to describe the computations we want to perform. However, in practice the choice of the programming language is important because it helps to be effective and to improve the quality of the software. For instance, a web application is rarely developed using a Turing machine or assembly language. By choosing an adequate formalism, it becomes easier to reason about the program, to analyze, certify, transform, optimize, or compile it. The choice of the programming language also has an impact on the quality of the software. By providing high-level constructs as well as static verifications, like typing, we can have an impact on the software design, allowing more expressiveness, more modularity, and a better reuse of code. This also improves the productivity of the programmer, and contributes to reducing the presence of errors.

The quality of a programming language depends on two main factors. First, the *intrinsic design*, which describes the programming model, the data model, the features provided by the language, as well as the semantics of the constructs. The second factor is the programmer and the application which is targeted. A language is not necessarily good for a given application if the concepts of the application domain cannot be easily manipulated. Similarly, it may not be good for a given person if the constructs provided by the language are not correctly understood by the programmer.

In the *Pareo* group we target a population of programmers interested in improving the long-term maintainability and the quality of their software, as well as their efficiency in implementing complex algorithms. Our privileged domain of application is large since it concerns the development of *transformations*. This ranges from the transformation of textual or structured documents such as XML, to the analysis and the transformation of programs and models. This also includes the development of tools such as theorem provers, proof assistants, or model checkers, where the transformations of proofs and the transitions between states play a crucial role. In that context, the *expressiveness* of the programming language is important. Indeed, complex encodings into low level data structures should be avoided, in contrast to high level notions such as abstract types and transformation rules that should be provided.

It is now well established that the notions of *term* and *rewrite rule* are two universal abstractions well suited to model tree based data types and the transformations that can be done upon them. Over the last ten years we have developed a strong experience in designing and programming with rule based languages [29], [20], [18]. We have introduced and studied the notion of *strategy* [19], which is a way to control how the rules should be applied. This provides the separation which is essential to isolate the logic and to make the rules reusable in different contexts.

To improve the quality of programs, it is also essential to have a clear description of their intended behaviors. For that, the *semantics* of the programming language should be formally specified.

There is still a lot of progress to be done in these directions. In particular, rule based programming can be made even more expressive by extending the existing matching algorithms to context-matching or to new data structures such as graphs or polygraphs. New algorithms and implementation techniques have to be found to improve the efficiency and make the rule based programming approach effective on large problems. Separating the rules from the control is very important. This is done by introducing a language for describing strategies. We still have to invent new formalisms and new strategy primitives which are both expressive enough and theoretically well grounded. A challenge is to find a good strategy language we can reason about, to prove termination properties for instance.

On the static analysis side, new formalized typing algorithms are needed to properly integrate rule based programming into already existing host languages such as Java. The notion of traversal strategy merits to be better studied in order to become more flexible and still provide a guarantee that the result of a transformation is correctly typed.

3.3. Rewriting calculus

The huge diversity of the rewriting concept is obvious and when one wants to focus on the underlying notions, it becomes quickly clear that several technical points should be settled. For example, what kind of objects are rewritten? Terms, graphs, strings, sets, multisets, others? Once we have established this, what is a rewrite rule? What is a left-hand side, a right-hand side, a condition, a context? And then, what is the effect of a rule application? This leads immediately to defining more technical concepts like variables in bound or free situations, substitutions and substitution application, matching, replacement; all notions being specific to the kind of objects that have to be rewritten. Once this is solved one has to understand the meaning of the application of a set of rules on (classes of) objects. And last but not least, depending on the intended use of rewriting, one would like to define an induced relation, or a logic, or a calculus.

In this very general picture, we have introduced a calculus whose main design concept is to make all the basic ingredients of rewriting explicit objects, in particular the notions of rule *application* and *result*. We concentrate on *term* rewriting, we introduce a very general notion of rewrite rule and we make the rule application and result explicit concepts. These are the basic ingredients of the *rewriting*- or ρ-calculus whose originality comes from the fact that terms, rules, rule application and application strategies are all treated at the object level (a rule can be applied on a rule for instance).

The λ-calculus is usually put forward as the abstract computational model underlying functional programming. However, modern functional programming languages have pattern-matching features which cannot be directly expressed in the λ-calculus. To palliate this problem, pattern-calculi [34], [31], [25] have been introduced. The rewriting calculus is also a pattern calculus that combines the expressiveness of pure functional calculi and algebraic term rewriting. This calculus is designed and used for logical and semantical purposes. It could be equipped with powerful type systems and used for expressing the semantics of rule based as well as object oriented languages. It allows one to naturally express exception handling mechanisms and elaborated rewriting strategies. It can be also extended with imperative features and cyclic data structures.

The study of the rewriting calculus turns out to be extremely successful in terms of fundamental results and of applications [22]. Different instances of this calculus together with their corresponding type systems have been proposed and studied. The expressive power of this calculus was illustrated by comparing it with similar

formalisms and in particular by giving a typed encoding of standard strategies used in first-order rewriting and classical rewrite based languages like *ELAN* and *Tom*.

TRIO Project-Team

3. Scientific Foundations

3.1. Fondation

In order to check for the timing behavior and the reliability of distributed systems, the TRIO team developed several techniques based on deterministic approaches ; in particular, we apply and extend analytical evaluation of worst case response times and when necessary, e.g. for large-scale communication systems as Internet based applications, we use techniques based on network calculus.

When the environment might lead to hazards (e.g. electromagnetic interferences causing transmission errors and bit-flips in memory), or when some characteristics of the system are not perfectly known or foreseeable beforehand, we model and analyze the uncertainties using stochastic models, for instance, models of the frame transmission patterns or models of the transmission errors. In the context of real time computing, we are in general much more interested by worst-case results over a given time window than by average and asymptotic results, and dedicated analyses in that area have been developed in our team over the last 10 years.

In the design of discrete event systems with hard real time constraints, the scheduling of the system's activities is of crucial importance. This means that we have to devise scheduling policies that ensure the respect of time constraints on line and / or optimize the behavior of the system according to some other application-dependent performance criteria.

In order to foster the best quality for programs, their understanding has to be automated, or at made significantly easier. Thus, we focus on analyzing and modeling program code, program structure and program behavior, and presenting these pieces of information to the user (in our case, program designers and program developers). Modeling user interaction is to come as well.

In the design of embedded, autonomous systems, power and energy usage is of paramount importance. We thus strive to model energy usage, based on actual hardware, and derive context-aware optimizations to decrease peak power and overall energy usage.

VEGAS Project-Team (section vide)

VERIDIS Project-Team

3. Scientific Foundations

3.1. Automated and interactive theorem proving

The VeriDis team unites experts in techniques and tools for interactive and automated verification, and specialists in methods and formalisms for the proved development of concurrent and distributed systems and algorithms. Our common objective is to advance the state of the art of combining interactive with automated methods resulting in powerful tools for the (semi-)automatic verification of distributed systems and protocols. Our techniques and tools will support methods for the formal development of trustworthy distributed systems that are grounded in mathematically precise semantics and that scale to algorithms relevant for practical applications.

The VeriDis members from Saarbrücken are developing Spass [7], one of the leading automated theorem provers for first-order logic based on the superposition calculus [33]. Recent extensions to the system include the integration of dedicated reasoning procedures for specific theories, such as linear arithmetic [44], [31], that are ubiquitous in the verification of systems and algorithms. The group also studies general frameworks for the combination of theories such as the locality principle [45] and automated reasoning mechanisms these induce.

The VeriDis members from Nancy develop veriT [1], an SMT (Satisfiability Modulo Theories [35]) solver that combines decision procedures for different fragments of first-order logic and that integrates an automatic theorem prover for full first-order logic. The veriT solver is designed to produce detailed proofs; this makes it particularly suitable as a component of a robust cooperation of deduction tools.

We rely on interactive theorem provers for reasoning about specifications at a high level of abstraction. Members of VeriDis have ample experience in the specification and subsequent machine-assisted, interactive verification of algorithms. In particular, we participate in a project at the joint MSR-Inria Centre in Saclay on the development of methods and tools for the formal proof of TLA$^+$ [41] specifications. Our prover relies on a declarative proof language and includes several automatic backends [3].

3.2. Methodology of proved system development

Powerful theorem provers are not a panacea for system verification: they support sound methodologies for modeling and verifying systems. In this respect, members of VeriDis have gained expertise and recognition in making contributions to formal methods for concurrent and distributed algorithms and systems [2], [6], and in applying them to concrete use cases. In particular, the concept of *refinement* [30], [34], [43] in state-based modeling formalisms is central to our approach. Its basic idea is to derive an algorithm or implementation by providing a series of models, starting from a high-level description that precisely states the problem, and gradually adding details in intermediate models. An important goal in designing such methods is to reduce the number of generated proof obligations and/or support their proof by automatic tools. This requires taking into account specific characteristics of certain classes of systems and tailoring the model to concrete computational models. Our research in this area is supported by carrying out case studies for academic and industrial developments. This activity benefits from and influences the development of our proof tools.

Our vision for the integration of our expertise can be resumed as follows. Based on our experience and related work on specification languages, logical frameworks, and automatic theorem proving tools, we develop an approach that is suited for specification, interactive theorem proving, and for eventual automated analysis and verification, possibly through appropriate translation methods. While specifications are developed by users inside our framework, they are analyzed for errors by our SMT based verification tools. Eventually, properties are proved by a combination of interactive and automatic theorem proving tools, potentially again with support of SMT procedures for specific sub-problems, or with the help of interactive proof guidance.

Today, the formal verification of a new algorithm is typically the subject of a PhD thesis, if it is addressed at all. This situation is not sustainable given the move towards more and more parallelism in mainstream systems: algorithm developers and system designers must be able to productively use verification tools for validating their algorithms and implementations. On a high level, the goal of VeriDis is to make formal verification standard practice for the development of distributed algorithms and systems, just as symbolic model checking has become commonplace in the development of embedded systems and as security analysis for cryptographic protocols is becoming standard practice today. Although the fundamental problems in distributed programming, such as mutual exclusion, leader election, group membership or consensus, are well-known, they pose new challenges in the context of current system paradigms, including ad-hoc and overlay networks or peer-to-peer systems.

CALVI Project-Team

3. Scientific Foundations

3.1. Kinetic models for plasma and beam physics

plasma physics, beam physics, kinetic models, reduced models, Vlasov equation, modeling, mathematical analysis, asymptotic analysis, existence, uniqueness

Plasmas and particle beams can be described by a hierarchy of models including N-body interaction, kinetic models and fluid models. Kinetic models in particular are posed in phase-space and involve specific difficulties. We perform a mathematical analysis of such models and try to find and justify approximate models using asymptotic analysis.

3.1.1. Models for plasma and beam physics

The **plasma state** can be considered as the **fourth state of matter**, obtained for example by bringing a gas to a very high temperature ($10^4\ K$ or more). The thermal energy of the molecules and atoms constituting the gas is then sufficient to start ionization when particles collide. A globally neutral gas of neutral and charged particles, called **plasma**, is then obtained. Intense charged particle beams, called nonneutral plasmas by some authors, obey similar physical laws.

The hierarchy of models describing the evolution of charged particles within a plasma or a particle beam includes N-body models where each particle interacts directly with all the others, kinetic models based on a statistical description of the particles and fluid models valid when the particles are at a thermodynamical equilibrium.

In a so-called *kinetic model*, each particle species s in a plasma or a particle beam is described by a distribution function $f_s(\mathbf{x}, \mathbf{v}, t)$ corresponding to the statistical average of the particle distribution in phase-space corresponding to many realisations of the physical system under investigation. The product $f_s\, d\mathbf{x}\, d\mathbf{v}$ is the average number of particles of the considered species, the position and velocity of which are located in a bin of volume $d\mathbf{x}\, d\mathbf{v}$ centered around $(\mathbf{x}, \mathbf{v})$. The distribution function contains a lot more information than what can be obtained from a fluid description, as it also includes information about the velocity distribution of the particles.

A kinetic description is necessary in collective plasmas where the distribution function is very different from the Maxwell-Boltzmann (or Maxwellian) distribution which corresponds to the thermodynamical equilibrium, otherwise a fluid description is generally sufficient. In the limit when collective effects are dominant with respect to binary collisions, the corresponding kinetic equation is the *Vlasov equation*

$$\frac{\partial f_s}{\partial t} + \mathbf{v} \cdot \frac{\partial f_s}{\partial \mathbf{x}} + \frac{q}{m}(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \frac{\partial f_s}{\partial \mathbf{v}} = 0,$$

which expresses that the distribution function f is conserved along the particle trajectories which are determined by their motion in their mean electromagnetic field. The Vlasov equation which involves a self-consistent electromagnetic field needs to be coupled to the Maxwell equations in order to compute this field

$$\begin{aligned} -\frac{1}{c^2}\frac{\partial \mathbf{E}}{\partial t} + \nabla \times \mathbf{B} &= \mu_0\, \mathbf{J}, \\ \frac{\partial \mathbf{B}}{\partial t} + \nabla \times \mathbf{E} &= 0, \\ \div \mathbf{E} &= \frac{\rho}{\varepsilon_0}, \\ \div \mathbf{B} &= 0, \end{aligned}$$

which describes the evolution of the electromagnetic field generated by the charge density

$$\rho(\mathbf{x},t) = \sum_s q_s \int f_s(\mathbf{x},\mathbf{v},t)\, d\mathbf{v},$$

and current density

$$\mathbf{J}(\mathbf{x},t) = \sum_s q_s \int f_s(\mathbf{x},\mathbf{v},t)\mathbf{v}\, d\mathbf{v},$$

associated to the charged particles.

When binary particle-particle interactions are dominant with respect to the mean-field effects then the distribution function f obeys the Boltzmann equation

$$\frac{\partial f}{\partial t} + \mathbf{v}\cdot\frac{\partial f}{\partial \mathbf{x}} = Q(f,f),$$

where Q is the nonlinear Boltzmann collision operator. In some intermediate cases, a collision operator needs to be added to the Vlasov equation.

The numerical solution of the three-dimensional Vlasov-Maxwell system represents a considerable challenge due to the huge size of the problem. Indeed, the Vlasov-Maxwell system is nonlinear and posed in phase space. It thus depends on seven variables: three configuration space variables, three velocity space variables and time, for each species of particles. This feature makes it essential to use every possible option to find a reduced model wherever possible, in particular when there are geometrical symmetries or small terms which can be neglected.

3.1.2. Mathematical and asymptotic analysis of kinetic models

The mathematical analysis of the Vlasov equation is essential for a thorough understanding of the model as well for physical as for numerical purposes. It has attracted many researchers since the end of the 1970s. Among the most important results which have been obtained, we can cite the existence of strong and weak solutions of the Vlasov-Poisson system by Horst and Hunze [74], see also Bardos and Degond [51]. The existence of a weak solution for the Vlasov-Maxwell system has been proved by Di Perna and Lions [59]. An overview of the theory is presented in a book by Glassey [71].

Many questions concerning for example uniqueness or existence of strong solutions for the three-dimensional Vlasov-Maxwell system are still open. Moreover, their is a realm of approached models that need to be investigated. In particular, the Vlasov-Darwin model for which we could recently prove the existence of global solutions for small initial data [52].

On the other hand, the asymptotic study of the Vlasov equation in different physical situations is important in order to find or justify reduced models. One situation of major importance in tokamaks, used for magnetic fusion as well as in atmospheric plasmas, is the case of a large external magnetic field used for confining the particles. The magnetic field tends to incurve the particle trajectories which eventually, when the magnetic field is large, are confined along the magnetic field lines. Moreover, when an electric field is present, the particles drift in a direction perpendicular to the magnetic and to the electric field. The new time scale linked to the cyclotron frequency, which is the frequency of rotation around the magnetic field lines, comes in addition to the other time scales present in the system like the plasma frequencies of the different particle species. Thus, many different time scales as well as length scales linked in particular to the different Debye length are present in the system. Depending on the effects that need to be studied, asymptotic techniques allow to find reduced models. In this spirit, in the case of large magnetic fields, recent results have been obtained by Golse and Saint-Raymond [72], [80] as well as by Brenier [57]. Our group has also contributed to this problem using homogenization techniques to justify the guiding center model and the finite Larmor radius model which are used by physicist in this setting [67], [65], [66].

Another important asymptotic problem yielding reduced models for the Vlasov-Maxwell system is the fluid limit of collisionless plasmas. In some specific physical situations, the infinite system of velocity moments of the Vlasov equations can be closed after a few of those, thus yielding fluid models.

3.2. Development of simulation tools

Numerical methods, Vlasov equation, unstructured grids, adaptivity, numerical analysis, convergence, Semi-Lagrangian method The development of efficient numerical methods is essential for the simulation of plasmas and beams. Indeed, kinetic models are posed in phase space and thus the number of dimensions is doubled. Our main effort lies in developing methods using a phase-space grid as opposed to particle methods. In order to make such methods efficient, it is essential to consider means for optimizing the number of mesh points. This is done through different adaptive strategies. In order to understand the methods, it is also important to perform their mathematical analysis. Since a few years we are interested also with solvers that uses Particle In Cell method. This new issue allows us to enrich some parts of our research activities previously centered on the Semi-Lagrangian approach.

3.2.1. Introduction

The numerical integration of the Vlasov equation is one of the key challenges of computational plasma physics. Since the early days of this discipline, an intensive work on this subject has produced many different numerical schemes. One of those, namely the Particle-In-Cell (PIC) technique, has been by far the most widely used. Indeed it belongs to the class of Monte Carlo particle methods which are independent of dimension and thus become very efficient when dimension increases which is the case of the Vlasov equation posed in phase space. However these methods converge slowly when the number of particles increases, hence if the complexity of grid based methods can be decreased, they can be the better choice in some situations. This is the reason why one of the main challenges we address is the development and analysis of adaptive grid methods.

3.2.2. Convergence analysis of numerical schemes

Exploring grid based methods for the Vlasov equation, it becomes obvious that they have different stability and accuracy properties. In order to fully understand what are the important features of a given scheme and how to derive schemes with the desired properties, it is essential to perform a thorough mathematical analysis of this scheme, investigating in particular its stability and convergence towards the exact solution.

3.2.3. The semi-Lagrangian method

The semi-Lagrangian method consists in computing a numerical approximation of the solution of the Vlasov equation on a phase space grid by using the property of the equation that the distribution function f is conserved along characteristics. More precisely, for any times s and t, we have

$$f(\mathbf{x},\mathbf{v},t) = f(\mathbf{X}(s;\mathbf{x},\mathbf{v},t),\mathbf{V}(s;\mathbf{x},\mathbf{v},t),s),$$

where $(\mathbf{X}(s;\mathbf{x},\mathbf{v},t),\mathbf{V}(s;\mathbf{x},\mathbf{v},t))$ are the characteristics of the Vlasov equation which are solution of the system of ordinary differential equations

$$\begin{aligned} \frac{d\mathbf{X}}{ds} &= \mathbf{V}, \\ \frac{d\mathbf{V}}{ds} &= \mathbf{E}(\mathbf{X}(s),s) + \mathbf{V}(s)\times\mathbf{B}(\mathbf{X}(s),s), \end{aligned} \tag{1}$$

with initial conditions $\mathbf{X}(t)=\mathbf{x}$, $\mathbf{V}(t)=\mathbf{v}$.

From this property, f^n being known one can induce a numerical method for computing the distribution function f^{n+1} at the grid points $(\mathbf{x}_i, \mathbf{v}_j)$ consisting in the following two steps:

1. For all i, j, compute the origin of the characteristic ending at $\mathbf{x}_i, \mathbf{v}_j$, i.e. an approximation of $\mathbf{X}(t_n; \mathbf{x}_i, \mathbf{v}_j, t_{n+1})$, $\mathbf{V}(t_n; \mathbf{x}_i, \mathbf{v}_j, t_{n+1})$.
2. As

$$f^{n+1}(\mathbf{x}_i, \mathbf{v}_j) = f^n(\mathbf{X}(t_n; \mathbf{x}_i, \mathbf{v}_j, t_{n+1}), \mathbf{V}(t_n; \mathbf{x}_i, \mathbf{v}_j, t_{n+1})),$$

f^{n+1} can be computed by interpolating f^n which is known at the grid points at the points $\mathbf{X}(t_n; \mathbf{x}_i, \mathbf{v}_j, t_{n+1}), \mathbf{V}(t_n; \mathbf{x}_i, \mathbf{v}_j, t_{n+1})$.

This method can be simplified by performing a time-splitting separating the advection phases in physical space and velocity space, as in this case the characteristics can be solved explicitly.

3.2.4. Adaptive semi-Lagrangian methods

Uniform meshes are most of the time not efficient to solve a problem in plasma physics or beam physics as the distribution of particles is evolving a lot as well in space as in time during the simulation. In order to get optimal complexity, it is essential to use meshes that are fitted to the actual distribution of particles. If the global distribution is not uniform in space but remains locally mostly the same in time, one possible approach could be to use an unstructured mesh of phase space which allows to put the grid points as desired. Another idea, if the distribution evolves a lot in time is to use a different grid at each time step which is easily feasible with a semi-Lagrangian method. And finally, the most complex and powerful method is to use a fully adaptive mesh which evolves locally according to variations of the distribution function in time. The evolution can be based on a posteriori estimates or on multi-resolution techniques.

3.2.5. Particle-In-Cell codes

The Particle-In-Cell method [56] consists in solving the Vlasov equation using a particle method, i.e. advancing numerically the particle trajectories which are the characteristics of the Vlasov equation, using the equations of motion which are the ordinary differential equations defining the characteristics. The self-fields are computed using a standard method on a structured or unstructured grid of physical space. The coupling between the field solve and the particle advance is done on the one hand by depositing the particle data on the grid to get the charge and current densities for Maxwell's equations and, on the other hand, by interpolating the fields at the particle positions. This coupling is one of the difficult issues and needs to be handled carefully.

3.2.6. Maxwell's equations in singular geometry

The solutions to Maxwell's equations are *a priori* defined in a function space such that the curl and the divergence are square integrable and that satisfy the electric and magnetic boundary conditions. Those solutions are in fact smoother (all the derivatives are square integrable) when the boundary of the domain is smooth or convex. This is no longer true when the domain exhibits non-convex *geometrical singularities* (corners, vertices or edges).

Physically, the electromagnetic field tends to infinity in the neighbourhood of the re-entrant singularities, which is a challenge to the usual finite element methods. Nodal elements cannot converge towards the physical solution. Edge elements demand considerable mesh refinement in order to represent those infinities, which is not only time- and memory-consuming, but potentially catastrophic when solving time dependent equations: the CFL condition then imposes a very small time step. Moreover, the fields computed by edge elements are discontinuous, which can create considerable numerical noise when the Maxwell solver is embedded in a plasma (e.g. PIC) code.

In order to overcome this dilemma, a method consists in splitting the solution as the sum of a *regular* part, computed by nodal elements, and a *singular* part which we relate to singular solutions of the Laplace operator, thus allowing to calculate a local analytic representation. This makes it possible to compute the solution precisely without having to refine the mesh.

This *Singular Complement Method* (SCM) had been developed [49] and implemented [48] in plane geometry.

An especially interesting case is axisymmetric geometry. This is still a 2D geometry, but more realistic than the plane case; despite its practical interest, it had been subject to much fewer theoretical studies [54]. The non-density result for regular fields was proven [58], the singularities of the electromagnetic field were related to that of modified Laplacians [45], and expressions of the singular fields were calculated [46]. Thus the SCM was extended to this geometry. It was then implemented by F. Assous (now at Bar-Ilan University, Israel) and S. Labrunie in a PIC–finite element Vlasov–Maxwell code [47].

As a byproduct, space-time regularity results were obtained for the solution to time-dependent Maxwell's equation in presence of geometrical singularities in the plane and axisymmetric cases [70], [46].

3.3. Large size problems

Parallelism, domain decomposition, code transformation

3.3.1. Introduction

The applications we consider lead to very large size computational problems for which we need to apply modern computing techniques enabling to use efficiently many computers including traditional high performance parallel computers and computational grids.

The full Vlasov-Maxwell system yields a very large computational problem mostly because the Vlasov equation is posed in six-dimensional phase-space. In order to tackle the most realistic possible physical problems, it is important to use all the modern computing power and techniques, in particular parallelism and grid computing.

3.3.2. Parallelization of numerical methods

An important issue for the practical use of the methods we develop is their parallelization. We address the problem of tuning these methods to homogeneous or heterogeneous architectures with the aim of meeting increasing computing resources requirements.

Most of the considered numerical methods apply a series of operations identically to all elements of a geometric data structure: the mesh of phase space. Therefore these methods intrinsically can be viewed as a data-parallel algorithm. A major advantage of this data-parallel approach derives from its scalability. Because operations may be applied identically to many data items in parallel, the amount of parallelism is dictated by the problem size.

Parallelism, for such data-parallel PDE solvers, is achieved by partitioning the mesh and mapping the submeshes onto the processors of a parallel architecture. A good partition balances the workload while minimizing the communications overhead. Many interesting heuristics have been proposed to compute near-optimal partitions of a (regular or irregular) mesh. For instance, the heuristics based on space-filing curves [73] give very good results for a very low cost.

Adaptive methods include a mesh refinement step and can highly reduce memory usage and computation volume. As a result, they induce a load imbalance and require to dynamically distribute the adaptive mesh. A problem is then to combine distribution and resolution components of the adaptive methods with the aim of minimizing communications. Data locality expression is of major importance for solving such problems. We use our experience of data-parallelism and the underlying concepts for expressing data locality [81], optimizing the considered methods and specifying new data-parallel algorithms.

As a general rule, the complexity of adaptive methods requires to define software abstractions allowing to separate/integrate the various components of the considered numerical methods (see [79] as an example of such modular software infrastructure).

Another key point is the joint use of heterogeneous architectures and adaptive meshes. It requires to develop new algorithms which include new load balancing techniques. In that case, it may be interesting to combine several parallel programming paradigms, i.e. data-parallelism with other lower-level ones.

Moreover, exploiting heterogeneous architectures requires the use of a run time support associated with a programming interface that enables some low-level hardware characteristics to be unified. Such run time support is the basis for heterogeneous algorithmics. Candidates for such a run time support may be specific implementations of MPI such as MPICH-G2 (a grid-enabled MPI implementation on top of the GLOBUS tool kit for grid computing [64]).

Our general approach for designing efficient parallel algorithms is to define code transformations at any level. These transformations can be used to incrementally tune codes to a target architecture and they warrant code reusability.

CORIDA Project-Team

3. Scientific Foundations

3.1. Analysis and control of fluids and of fluid-structure interactions

Participants: Thomas Chambrion, Antoine Henrot, Alexandre Munnier, Lionel Rosier, Jean-François Scheid, Takeo Takahashi, Marius Tucsnak, Jean-Claude Vivalda.

The problems we consider are modeled by the Navier-Stokes, Euler or Korteweg de Vries equations (for the fluid) coupled to the equations governing the motion of the solids. One of the main difficulties of this problem comes from the fact that the domain occupied by the fluid is one of the unknowns of the problem. We have thus to tackle a *free boundary problem*.

The control of fluid flows is a major challenge in many applications: aeronautics, pollution issues, regulation of irrigation channels or of the flow in pipelines, etc. All these problems cannot be easily reduced to finite dimensional models so a methodology of analysis and control based on PDE's is an essential issue. In a first approximation the motion of fluid and of the solids can be decoupled. The most used models for an incompressible fluid are given by the Navier-Stokes or by the Euler equations.

The optimal open loop control approach of these models has been developed from both the theoretical and numerical points of view. Controllability issues for the equations modeling the fluid motion are by now well understood (see, for instance, Imanuvilov [59] and the references therein). The feedback control of fluid motion has also been recently investigated by several research teams (see, for instance Barbu [54] and references therein) but this field still contains an important number of open problems (in particular those concerning observers and implementation issues). One of our aims is to develop efficient tools for computing feedback laws for the control of fluid systems.

In real applications the fluid is often surrounded by or it surrounds an elastic structure. In the above situation one has to study fluid-structure interactions. This subject has been intensively studied during the last years, in particular for its applications in noise reduction problems, in lubrication issues or in aeronautics. In this kind of problems, a PDE's system modeling the fluid in a cavity (Laplace equation, wave equation, Stokes, Navier-Stokes or Euler systems) is coupled to the equations modeling the motion of a part of the boundary. The difficulties of this problem are due to several reasons such as the strong nonlinear coupling and the existence of a free boundary. This partially explains the fact that applied mathematicians have only recently tackled these problems from either the numerical or theoretical point of view. One of the main results obtained in our project concerns the global existence of weak solutions in the case of a two-dimensional Navier–Stokes fluid (see [8]). Another important result gives the existence and the uniqueness of strong solutions for two or three-dimensional Navier–Stokes fluid (see [9]). In that case, the solution exists as long as there is no contact between rigid bodies, and for small data in the three-dimensional case.

3.2. Frequency domain methods for the analysis and control of systems governed by PDE's

Participants: Xavier Antoine, Bruno Pinçon, Karim Ramdani, Bertrand Thierry.

We use frequency tools to analyze different types of problems. The first one concerns the control, the optimal control and the stabilization of systems governed by PDE's, and their numerical approximations. The second one concerns time-reversal phenomena, while the last one deals with numerical approximation of high-frequency scattering problems.

3.2.1. Control and stabilization for skew-adjoint systems

The first area concerns theoretical and numerical aspects in the control of a class of PDE's. More precisely, in a semigroup setting, the systems we consider have a skew-adjoint generator. Classical examples are the wave, the Bernoulli-Euler or the Schrödinger equations. Our approach is based on an original characterization of exact controllability of second order conservative systems proposed by K. Liu [63]. This characterization can be related to the Hautus criterion in the theory of finite dimensional systems (cf. [58]). It provides for time-dependent problems exact controllability criteria **that do not depend on time, but depend on the frequency variable** conjugated to time. Studying the controllability of a given system amounts then to establishing uniform (with respect to frequency) estimates. In other words, the problem of exact controllability for the wave equation, for instance, comes down to a high-frequency analysis for the Helmholtz operator. This frequency approach has been proposed first by K. Liu for bounded control operators (corresponding to internal control problems), and has been recently extended to the case of unbounded control operators (and thus including boundary control problems) by L. Miller [64]. Using the result of Miller, K. Ramdani, T. Takahashi, M. Tucsnak have obtained in [5] a new spectral formulation of the criterion of Liu [63], which is valid for boundary control problems. This frequency test can be seen as an observability condition for packets of eigenvectors of the operator. This frequency test has been successfully applied in [5] to study the exact controllability of the Schrödinger equation, the plate equation and the wave equation in a square. Let us emphasize here that one further important advantage of this frequency approach lies in the fact that it can also be used for the analysis of space semi-discretized control problems (by finite element or finite differences). The estimates to be proved must then be uniform with respect to **both the frequency and the mesh size**.

In the case of finite dimensional systems one of the main applications of frequency domain methods consists in designing robust controllers, in particular of H^∞ type. Obtaining the similar tools for systems governed by PDE's is one of the major challenges in the theory of infinite dimensional systems. The first difficulty which has to be tackled is that, even for very simple PDE systems, no method giving the parametrisation of all stabilizing controllers is available. One of the possible remedies consists in considering known families of stabilizing feedback laws depending on several parameters and in optimizing the H^∞ norm of an appropriate transfer function with respect to this parameters. Such families of feedback laws yielding computationally tractable optimization problems are now available for systems governed by PDE's in one space dimension.

3.2.2. Time-reversal

The second area in which we make use of frequency tools is the analysis of time-reversal for harmonic acoustic waves. This phenomenon described in Fink [56] is a direct consequence of the reversibility of the wave equation in a non dissipative medium. It can be used to **focus an acoustic wave** on a target through a complex and/or unknown medium. To achieve this, the procedure followed is quite simple. First, time-reversal mirrors are used to generate an incident wave that propagates through the medium. Then, the mirrors measure the acoustic field diffracted by the targets, time-reverse it and back-propagate it in the medium. Iterating the scheme, we observe that the incident wave emitted by the mirrors focuses on the scatterers. An alternative and more original focusing technique is based on the so-called D.O.R.T. method [57]. According to this experimental method, the eigenelements of the time-reversal operator contain important information on the propagation medium and on the scatterers contained in it. More precisely, the number of nonzero eigenvalues is exactly the number of scatterers, while each eigenvector corresponds to an incident wave that selectively focuses on each scatterer.

Time-reversal has many applications covering a wide range of fields, among which we can cite **medicine** (kidney stones destruction or medical imaging), **sub-marine communication** and **non destructive testing**. Let us emphasize that in the case of time-harmonic acoustic waves, time-reversal is equivalent to phase conjugation and involves the Helmholtz operator.

In [2], we proposed the first far field model of time reversal in the time-harmonic case.

3.2.3. Numerical approximation of high-frequency scattering problems

This subject deals mainly with the numerical solution of the Helmholtz or Maxwell equations for open region scattering problems. This kind of situation can be met e.g. in radar systems in electromagnetism or in acoustics for the detection of underwater objects like submarines.

Two particular difficulties are considered in this situation

- the wavelength of the incident signal is small compared to the characteristic size of the scatterer,
- the problem is set in an unbounded domain.

These two problematics limit the application range of most common numerical techniques. The aim of this part is to develop new numerical simulation techniques based on microlocal analysis for modeling the propagation of rays. The importance of microlocal techniques in this situation is that it makes possible a local analysis both in the spatial and frequency domain. Therefore, it can be seen as a kind of asymptotic theory of rays which can be combined with numerical approximation techniques like boundary element methods. The resulting method is called the On-Surface Radiation Condition method.

3.3. Observability, controllability and stabilization in the time domain

Participants: Fatiha Alabau, Xavier Antoine, Thomas Chambrion, Antoine Henrot, Karim Ramdani, Marius Tucsnak, Jean-Claude Vivalda.

Controllability and observability have been set at the center of control theory by the work of R. Kalman in the 1960's and soon they have been generalized to the infinite-dimensional context. The main early contributors have been D.L. Russell, H. Fattorini, T. Seidman, R. Triggiani, W. Littman and J.-L. Lions. The latter gave the field an enormous impact with his book [61] , which is still a main source of inspiration for many researchers. Unlike in classical control theory, for infinite-dimensional systems there are many different (and not equivalent) concepts of controllability and observability. The strongest concepts are called exact controllability and exact observability, respectively. In the case of linear systems exact controllability is important because it guarantees stabilizability and the existence of a linear quadratic optimal control. Dually, exact observability guarantees the existence of an exponentially converging state estimator and the existence of a linear quadratic optimal filter. An important feature of infinite dimensional systems is that, unlike in the finite dimensional case, the conditions for exact observability are no longer independent of time. More precisely, for simple systems like a string equation, we have exact observability only for times which are large enough. For systems governed by other PDE's (like dispersive equations) the exact observability in arbitrarily small time has been only recently established by using new frequency domain techniques. A natural question is to estimate the energy required to drive a system in the desired final state when the control time goes to zero. This is a challenging theoretical issue which is critical for perturbation and approximation problems. In the finite dimensional case this issue has been first investigated in Seidman [66]. In the case of systems governed by linear PDE's some similar estimates have been obtained only very recently (see, for instance Miller [64]). One of the open problems of this field is to give sharp estimates of the observability constants when the control time goes to zero.

Even in the finite-dimensional case, despite the fact that the linear theory is well established, many challenging questions are still open, concerning in particular nonlinear control systems.

In some cases it is appropriate to regard external perturbations as unknown inputs; for these systems the synthesis of observers is a challenging issue, since one cannot take into account the term containing the unknown input into the equations of the observer. While the theory of observability for linear systems with unknown inputs is well established, this is far from being the case in the nonlinear case. A related active field of research is the uniform stabilization of systems with time-varying parameters. The goal in this case is to stabilize a control system with a control strategy independent of some signals appearing in the dynamics, i.e., to stabilize simultaneously a family of time-dependent control systems and to characterize families of control systems that can be simultaneously stabilized.

One of the basic questions in finite- and infinite-dimensional control theory is that of motion planning, i.e., the explicit design of a control law capable of driving a system from an initial state to a prescribed final one. Several techniques, whose suitability depends strongly on the application which is considered, have been and are being developed to tackle such a problem, as for instance the continuation method, flatness, tracking or optimal control. Preliminary to any question regarding motion planning or optimal control is the issue of controllability, which is not, in the general nonlinear case, solved by the verification of a simple algebraic criterion. A further motivation to study nonlinear controllability criteria is given by the fact that techniques developed in the domain of (finite-dimensional) geometric control theory have been recently applied successfully to study the controllability of infinite-dimensional control systems, namely the Navier–Stokes equations (see Agrachev and Sarychev [53]).

3.4. Implementation

This is a transverse research axis since all the research directions presented above have to be validated by giving control algorithms which are aimed to be implemented in real control systems. We stress below some of the main points which are common (from the implementation point of view) to the application of the different methods described in the previous sections.

For many infinite dimensional systems the use of co-located actuators and sensors and of simple proportional feed-back laws gives satisfying results. However, for a large class of systems of interest it is not clear that these feedbacks are efficient, or the use of co-located actuators and sensors is not possible. This is why a more general approach for the design of the feedbacks has to be considered. Among the techniques in finite dimensional systems theory those based on the solutions of infinite dimensional Riccati equation seem the most appropriate for a generalization to infinite dimensional systems. The classical approach is to approximate an LQR problem for a given infinite dimensional system by finite dimensional LQR problems. As it has been already pointed out in the literature this approach should be carefully analyzed since, even for some very simple examples, the sequence of feedbacks operators solving the finite dimensional LQR is not convergent. Roughly speaking this means that by refining the mesh we obtain a closed loop system which is not exponentially stable (even if the corresponding infinite dimensional system is theoretically stabilized). In order to overcome this difficulty, several methods have been proposed in the literature : filtering of high frequencies, multigrid methods or the introduction of a numerical viscosity term. We intend to first apply the numerical viscosity method introduced in Tcheougoue Tebou – Zuazua [67], for optimal and robust control problems.

TOSCA Project-Team

3. Scientific Foundations

3.1. Scientific Foundations

Most often physicists, economists, biologists, engineers need a stochastic model because they cannot describe the physical, economical, biological, etc., experiment under consideration with deterministic systems, either because of its complexity and/or its dimension or because precise measurements are impossible. Then they abandon trying to get the exact description of the state of the system at future times given its initial conditions, and try instead to get a statistical description of the evolution of the system. For example, they desire to compute occurrence probabilities for critical events such as the overstepping of a given thresholds by financial losses or neuronal electrical potentials, or to compute the mean value of the time of occurrence of interesting events such as the fragmentation to a very small size of a large proportion of a given population of particles. By nature such problems lead to complex modelling issues: one has to choose appropriate stochastic models, which require a thorough knowledge of their qualitative properties, and then one has to calibrate them, which requires specific statistical methods to face the lack of data or the inaccuracy of these data. In addition, having chosen a family of models and computed the desired statistics, one has to evaluate the sensitivity of the results to the unavoidable model specifications. The TOSCA team, in collaboration with specialists of the relevant fields, develops theoretical studies of stochastic models, calibration procedures, and sensitivity analysis methods.

In view of the complexity of the experiments, and thus of the stochastic models, one cannot expect to use closed form solutions of simple equations in order to compute the desired statistics. Often one even has no other representation than the probabilistic definition (e.g., this is the case when one is interested in the quantiles of the probability law of the possible losses of financial portfolios). Consequently the practitioners need Monte Carlo methods combined with simulations of stochastic models. As the models cannot be simulated exactly, they also need approximation methods which can be efficiently used on computers. The TOSCA team develops mathematical studies and numerical experiments in order to determine the global accuracy and the global efficiency of such algorithms.

The simulation of stochastic processes is not motivated by stochastic models only. The stochastic differential calculus allows one to represent solutions of certain deterministic partial differential equations in terms of probability distributions of functionals of appropriate stochastic processes. For example, elliptic and parabolic linear equations are related to classical stochastic differential equations, whereas nonlinear equations such as the Burgers and the Navier–Stokes equations are related to McKean stochastic differential equations describing the asymptotic behavior of stochastic particle systems. In view of such probabilistic representations one can get numerical approximations by using discretization methods of the stochastic differential systems under consideration. These methods may be more efficient than deterministic methods when the space dimension of the PDE is large or when the viscosity is small. The TOSCA team develops new probabilistic representations in order to propose probabilistic numerical methods for equations such as conservation law equations, kinetic equations, and nonlinear Fokker–Planck equations.

BIGS Project-Team

3. Scientific Foundations

3.1. Online data analysis

Participants: J-M. Monnez, R. Bar, P. Vallois. Generally speaking, there exists an overwhelming amount of articles dealing with the analysis of high dimensional data. Indeed, this is one of the major challenges in statistics today, motivated by internet or biostatistics applications. Within this global picture, the problem of classification or dimension reduction of online data can be traced back at least to a seminal paper by Mac Queen [61], in which the k-means algorithm is introduced. This popular algorithm, constructed for classification purposes, consists in a stepwise updating of the centers of some classes according to a stream of data entering into the system. The literature on the topic has been growing then rapidly since the beginning of the 90's.

Our point of view on the topic relies on the so-called *french data analysis school*, and more specifically on Factorial Analysis tools. In this context, it was then rapidly seen that stochastic approximation was an essential tool (see Lebart's paper [58]), which allows to approximate eigenvectors in a stepwise manner. A systematic study of Principal Component and Factorial Analysis has then been leaded by Monnez in the series of papers [64], [62], [63], in which many aspects of convergences of online processes are analyzed thanks to the stochastic approximation techniques.

3.2. Local regression techniques

Participants: S. Ferrigno, A. Muller-Gueudin. In the context where a response variable Y is to be related to a set of regressors X, one of the general goals of Statistics is to provide the end user with a model which turns out to be useful in predicting Y for various values of X. Except for the simplest situations, the determination of a good model involves many steps. For example, for the task of predicting the value of Y as a function of the covariate X, statisticians have elaborated models such as the regression model with random regressors:

$$Y = g(X, \theta) + \sigma(X)\epsilon.$$

Many assumptions must be made to reach it as a possible model. Some require much thinking, as for example, those related to the functional form of $g(\cdot, \theta)$. Some are made more casually, as often those related to the functional form of $\sigma(\cdot)$ or those concerning the distribution of the random error term ϵ. Finally, some assumptions are made for commodity. Thus the need for methods that can assess if a model is concordant with the data it is supposed to adjust. The methods fall under the banner of goodness of fit tests. Most existing tests are *directional*, in the sense that they can detect departures from only one or a few aspects of a null model. For example, many tests have been proposed in the literature to assess the validity of an entertained structural part $g(\cdot, \theta)$. Some authors have also proposed tests about the variance term $\sigma(\cdot)$ (cf. [59]). Procedures testing the normality of the ϵ_i are given, but for other assumptions much less work has been done. Therefore the need of a global test which can evaluate the validity of a global structure emerges quite naturally.

With these preliminaries in mind, let us observe that one quantity which embodies all the information about the joint behavior of (X, Y) is the cumulative conditional distribution function, defined by

$$F(y|x) = P(Y \leq y|X = x).$$

The (nonparametric) estimation of this function is thus of primary importance. To this aim, notice that modern estimators are usually based on the local polynomial approach, which has been recognized as superior to classical estimates based on the Nadaraya-Watson approach, and are as good as the recent versions based on spline and other methods. In some recent works [46], [47], we address the following questions:

- Construction of a global test by means of Cramer-von Mises statistic.
- Optimal bandwidth of the kernel used for approximation purposes.

We also obtain sharp estimates on the conditional distribution function in [48].

3.3. Stochastic modeling for complex and biological systems

In most biological contexts, mathematics turn out to be useful in producing accurate models with dual objectives: they should be simple enough and meaningful for the biologist on the one hand, and they should provide some insight on the biological phenomenon at stake on the other hand. We have focused on this kind of issue in various contexts that we shall summarize below.

Photodynamic Therapy: Photodynamic therapy induces a huge demand of interconnected mathematical systems, among which we have studied recently the following ones:

• The tumor growth model is of crucial importance in order to understand the behavior of the whole therapy. We have considered the tumor growth as a stochastic equation, for which we have handled the problem uncertainties on the measure times [31] as well as mixed effects for parameter estimation.

• Another important aspect to quantify for PDT calibration is the response to radiotherapy treatments. There are several valid mathematical ways to describe this process, among which we distinguish the so-called hit model. This model assumes that whenever a group of sensitive targets (chromosomes, membrane) in the cell are reached by a sufficient number of radiations, then the cell is inactivated and dies. We have elaborated on this scheme in order to take into account two additional facts: (i) The reduction of the cell situation to a two-state model might be an oversimplification. (ii) Several doses of radiations are inoculated as time passes. These observations have leaded us to introduce a new model based on multi-state Markov chains arguments [10], in which cell proliferation can be incorporated.

Bacteriophage therapy: Let us mention a starting collaboration between BIGS and the Genetics and Microbiology department at the Universitat Autònoma de Barcelona, on the modeling of bacteriophage therapies. The main objective here is to describe how a certain family of benign viruses is able to weaken a bacterium induced disease, which naturally leads to the introduction of a noisy predator-prey system of equations. It should be mentioned that some similar problems have been treated (in a rather informal way, invoking a linearization procedure) by Carletti in [39]. These tools cannot be applied directly to our system, and our methods are based on concentration and large deviations techniques (on which we already had an expertise [65], [68]) in order to combine convergence to equilibrium for the deterministic system and deviations of the stochastic system. Notice that A. Muller-Gueudin is also working with A. Debussche and O. Radulescu on a related topic [42], namely the convergence of a model of cellular biochemical reactions.

Gaussian signals: Nature provides us with many examples of systems such that the observed signal has a given Hölder regularity, which does not correspond to the one we might expect from a system driven by ordinary Brownian motion. This situation is commonly handled by noisy equations driven by Gaussian processes such as fractional Brownian motion or (in higher dimensions of the parameter) fractional fields.

The basic aspects of differential equations driven by a fractional Brownian motion (fBm) and other Gaussian processes are now well understood, mainly thanks to the so-called *rough paths* tools [60], but also invoking the Russo-Vallois integration techniques [67]. The specific issue of Volterra equations driven by fBm, which is central for the subdiffusion within proteins problem, is addressed in [43].

Fractional fields are very often used to model irregular phenomena which exhibit a scale invariance property, fractional Brownian motion being the historical fractional model. Nevertheless, its isotropy property is a serious drawback for instance in hydrology or in medecine (see [38]). Moreover, the fractional Brownian motion cannot be used to model some phenomena for which the regularity varies with time. Hence, many generalization (gaussian or not) of this model has been recently proposed, see for instance [32] for some Gaussian locally self-similar fields, [54] for some non-Gaussian models, [36] for anisotropic models.

Our team has thus contributed [41], [55], [54], [56], [66] and still contributes [35], [37], [36], [57], [49] to this theoretical study: Hölder continuity, fractal dimensions, existence and uniqueness results for differential equations, study of the laws to quote a few examples. As we shall see below, this line of investigation also has some impact in terms of applications: we shall discuss how we plan to apply our results to osteoporosis on the one hand and to fluctuations within protein molecules on the other hand.

3.4. Parameter identifiability and estimation

When one desires to confront theoretical probabilistic models with real data, statistical tools are obviously crucial. We have focused on two of them: parameter identifiability and parameter estimation.

Parameter identifiability [72] deals with the possibility to give a unique value to each parameter of a mathematical model structure in inverse problems. There are many methods for testing models for identifiability: Laplace transform, similarity transform, Taylor series, local state isomorphism or elimination theory. Most of the current approaches are devoted to *a priori* identifiability and are based on algebraic techniques. We are particularly concerned with *a posteriori* identifiability, *i.e.* after experiments or in a constrained experimental framework and the link with experimental design techniques. Our approach is based on statistical techniques through the use of variance-based methods. These techniques are strongly connected with global sensitivity approaches and Monte Carlo methods.

The parameter estimation for a family of probability laws has a very long story in statistics, and we refer to [33] for an elegant overview of the topic. Moving to the references more closely related to our specific projects, let us recall first that the mathematical description of photodynamic therapy can be split up into three parametric models : the uptake model (pharmacokinetics of the photosensitizing drug into cancer cells), the photoreaction model and the tumor growth model. (i) Several papers have been reported for the application of system identification techniques to pharmacokinetics modeling problems. But two issues were ignored in these previous works: presence of timing noise and identification from longitudinal data. In [31], we have proposed a bounded-error estimation algorithm based on interval analysis to solve the parameter estimation problem while taking into consideration uncertainty on observation time instants. Statistical inference from longitudinal data based on mixed effects models can be performed by the *Monolix* software (http://www.monolix.org) developed the Monolix group chaired by Marc Lavielle and France Mentré, and supported by Inria. In the recent past, we have used this tool for tumor growth modeling. (ii) According to what we know so far, no parameter estimation study has been reported about the photoreaction model in photodynamic therapy. A photoreaction model, composed of six stochastic differential equations, is proposed in [44]. The main open problem is to access to data. We currently build on an experimental platform which aims at overcoming this technical issue. Moreover, an identifiability study coupled to a global sensitivity analysis of the photoreaction model are currently in progress. (iii) Tumor growth is generally described by population dynamics models or by cell cycle models. Faced with this wide variety of descriptions, one of the main open problems is to identify the suitable model structure. As mentioned above, we currently investigate alternative representations based on branching processes and Markov chains, with a model selection procedure in mind.

A few words should be said about the existing literature on statistical inference for diffusion or related processes, a topic which will be at the heart of three of our projects (namely photodynamic and bacteriophage therapies, as well as fluctuations within molecules). The monograph [53] is a good reference on the basic estimation techniques for diffusion processes. The problem of estimating diffusions observed at discrete times, of crucial importance for applications, has been addressed mainly since the mid 90s. The maximum likelihood techniques, which are also classical for parameter estimation, are well represented by the contributions [45].

Some attention has been paid recently to the estimation of the coefficients of fractional or multifractional Brownian motion according to a set of observations. Let us quote for instance the nice surveys [30], [40]. On the other hand, the inference problem for diffusions driven by a fractional Brownian motion is still in its infancy. A good reference on the question is [69], dealing with some very particular families of equations, which do not cover the cases of interest for us.

CORTEX Project-Team

3. Scientific Foundations

3.1. Computational neuroscience

With regards to the progress that has been made in anatomy, neurobiology, physiology, imaging, and behavioral studies, computational neuroscience offers a unique interdisciplinary cooperation between experimental and clinical neuroscientists, physicists, mathematicians and computer scientists. It combines experiments with data analysis and functional models with computer simulation on the basis of strong theoretical concepts and aims at understanding mechanisms that underlie neural processes such as perception, action, learning, memory or cognition.

Today, computational models are able to offer new approaches for the understanding of the complex relations between the structural and the functional level of the brain, thanks to models built at several levels of description. In very precise models, a neuron can be divided in several compartments and its dynamics can be described by a system of differential equations. The spiking neuron approach (*cf.* § 3.2) proposes to define simpler models concentrated on the prediction of the most important events for neurons, the emission of spikes. This allows to compute networks of neurons and to study the neural code with event-driven computations.

Larger neuronal systems are considered when the unit of computation is defined at the level of the population of neurons and when rate coding and/or correlations are supposed to bring enough information. Studying Dynamic Neural Fields (*cf.* § 3.3) consequently lays emphasis on information flows between populations of neurons (feed-forward, feed-back, lateral connectivity) and is well adapted to defining high-level behavioral capabilities related for example to visuomotor coordination.

Furthermore, these computational models and methods have strong implications for other sciences (e.g. computer science, cognitive science, neuroscience) and applications (e.g. robots, cognitive prosthesis) as well (*cf.* § 4.1). In computer science, they promote original modes of distributed computation (*cf.* § 3.5); in cognitive science, they have to be related to current theories of cognition (*cf.* § 3.6); in neuroscience, their predictions have to be related to observed behaviors and measured brain signals (*cf.* § 3.4).

3.2. Computational neuroscience at the microscopic level: spiking neurons and networks

Computational neuroscience is also interested in having more precise and realistic models of the neuron and especially of its dynamics. We consider that the latter aspect cannot be treated at the single unit level only; it is also necessary to consider interactions between neurons at the microscopic scale.

On one hand, compartmental models describe the neuron at the inner scale, through various compartments (axon, synapse, cellular body) and coupled differential equations, allowing to numerically predict the neural activity at a high degree of accuracy. This, however, is intractable if analytic properties are to be derived, or if neural assemblies are considered. We thus focus on phenomenological punctual models of spiking neurons, in order to capture the dynamic behavior of the neuron isolated or inside a network. Generalized conductance based leaky integrate and fire neurons (emitting action potential, i.e. spike, from input integration) or simplified instantiations are considered in our group.

On the other hand, one central issue is to better understand the precise nature of the neural code. From rate coding (the classical assumption that information is mainly conveyed by the firing frequency of neurons) to less explored assumptions such as high-order statistics, time coding (the idea that information is encoded in the firing time of neurons) or synchronization aspects. At the biological level, a fundamental example is the synchronization of neural activities, which seems to play a role in, e.g., olfactory perception: it has been observed that abolishing synchronization suppresses the odor discrimination capability. At the computational

level, recent theoretical results show that the neural code is embedded in periodic firing patterns, while, more generally, we focus on tractable mathematical analysis methods coming from the theory of nonlinear dynamical systems.

For both biological simulations and computer science emerging paradigms, the rigorous simulation of large neural assemblies is a central issue. Our group is at the origin, up to our best knowledge, of the most efficient event-based neural network simulator (Mvaspike), based on well-founded discrete event dynamic systems theory, and now extended to other simulation paradigms, thus offering the capability to push the state of the art on this topic.

3.3. Computational neuroscience at the mesoscopic level: dynamic neural field

Our research activities in the domain of computational neurosciences are also interested in the understanding of higher brain functions using both computational models and robotics. These models are grounded on a computational paradigm that is directly inspired by several brain studies converging on a distributed, asynchronous, numerical and adaptive processing of information and the continuum neural field theory (CNFT) provides the theoretical framework to design models of population of neurons.

This mesoscopic approach underlines the fact that the number of neurons is very high, even in a small part of tissue, and proposes to study neuronal models in a continuum limit where space is continuous and main variables correspond to synaptic activity or firing rates in population of neurons. This formalism is particularly interesting because the dynamic behavior of a large piece of neuronal tissue can be studied with differential equations that can integrate spatial (lateral connectivity) and temporal (speed of propagation) characteristics and display such interesting behavior as pattern formation, travelling waves, bumps, etc.

The main cognitive tasks we are currently interested in are related to sensorimotor systems in interaction with the environment (perception, coordination, planning). The corresponding neuronal structures we are modeling are part of the cortex (perceptive, associative, frontal maps) and the limbic system (hippocampus, amygdala, basal ganglia). Corresponding models of these neuronal structures are defined at the level of the population of neurons and functioning and learning rules are built from neuroscience data to emulate the corresponding information processing (filtering in perceptive maps, multimodal association in associative maps, temporal organization of behavior in frontal maps, episodic memory in hippocampus, emotional conditioning in amygdala, selection of action in basal ganglia). Our aim is to iteratively refine these models, implement them on autonomous robots and make them cooperate and exchange information, toward a completely adaptive, integrated and autonomous behavior.

3.4. Brain Signal Processing

The observation of brain activity and its analysis with appropriate data analysis techniques allow to extract properties of underlying neural activity and to better understand high level functions. This study needs to investigate and integrate, in a single trial, information spread in several cortical areas and available at different scales (MUA, LFP, ECoG, EEG).

One major problem is how to be able to deal with the variability between trials. Thus, it is necessary to develop robust techniques based on stable features. Specific modeling techniques should be able to extract features investigating the time domain and the frequency domain. In the time domain, template-based unsupervised models allows to extract graphic-elements. Both the average technique to obtain the templates and the distance used to match the signal with the templates are important, even when the signal has a strong distorted shape. The study of spike synchrony is also an important challenge. In the frequency domain, features such as phases, frequency bands and amplitudes contain different pieces of information that should be properly identified using variable selection techniques. In both cases, compression techniques such as PCA or ICA can reduce the fluctuations of the cortical signal. Then, the designed models have to be able to track the dynamic evolution of these features over the time.

Another problem is how to integrate information spreading in different areas and relate this information in a proper time window of synchronization to behavior. For example, feedbacks are known to be very important to better understand the closed-loop control of a hand grasping movement. However, from the preparatory signal and the execution of the movement to the visual and somatosensory feedbacks, there is a delay. It is thus necessary to use stable features to build a mapping between areas using supervised models taking into account a time window shift.

Several recoding techniques are taken into account, providing different kinds of information. Some of them provide very local information such as multiunit activities (MUA) and local field potential (LFP) in one or several well-chosen cortical areas. Other ones provide global information about close regions such as electrocorticography (ECoG) or the whole scalp such as electroencephalography (EEG). If surface electrodes allow to easily obtain brain imaging, it is more and more necessary to better investigate the neural code.

3.5. Connectionist parallelism

Connectionist models, such as neural networks, are among the first models of parallel computing. Artificial neural networks now stand as a possible alternative with respect to the standard computing model of current computers. The computing power of these connectionist models is based on their distributed properties: a very fine-grain massive parallelism with densely interconnected computation units.

The connectionist paradigm is the foundation of the robust, adaptive, embeddable and autonomous processings that we aim at developing in our team. Therefore their specific massive parallelism has to be fully exploited. Furthermore, we use this intrinsic parallelism as a guideline to develop new models and algorithms for which parallel implementations are naturally made easier.

Our approach claims that the parallelism of connectionist models makes them able to deal with strong implementation and application constraints. This claim is based on both theoretical and practical properties of neural networks. It is related to a very fine parallelism grain that fits parallel hardware devices, as well as to the emergence of very large reconfigurable systems that become able to handle both adaptability and massive parallelism of neural networks. More particularly, digital reconfigurable circuits (e.g. FPGA, Field Programmable Gate Arrays) stand as the most suitable and flexible device for low cost fully parallel implementations of neural models, according to numerous recent studies in the connectionist community. We carry out various arithmetical and topological studies that are required by the implementation of several neural models onto FPGAs, as well as the definition of hardware-targetted neural models of parallel computation.

This research field has evolved within our team by merging with our activities in behavioral computational neuroscience. Taking advantage of the ability of the neural paradigm to cope with strong constraints, as well as taking advantage of the highly complex cognitive tasks that our behavioral models may perform, a new research line has emerged that aims at defining a specific kind of brain-inspired hardware based on modular and extensive resources that are capable of self-organization and self-recruitment through learning when they are assembled within a perception-action loop.

3.6. The embodiment of cognition

Recent theories from cognitive science stress that human cognition emerges from the interactions of the body with the surrounding world. Through motor actions, the body can orient toward objects to better perceive and analyze them. The analysis is performed on the basis of physical measurements and more or less elaborated emotional reactions of the body, generated by the stimuli. This elicits other orientation activities of the body (approach and grasping or avoidance). This elementary behavior is made possible by the capacity, at the cerebral level, to coordinate the perceptive representation of the outer world (including the perception of the body itself) with the behavioral repertoire that it generates either on the physical body (external actions) or on a more internal aspect (emotions, motivations, decisions). In both cases, this capacity of coordination is acquired from experience and interaction with the environment.

The theory of the situatedness of cognition proposes to minimize representational contents (opposite to complex and hierarchical representations) and privileges simple strategies, more directly coupling perception and action and more efficient to react quickly in the changing environment.

A key aspect of this theory of intelligence is the Gibsonian notion of affordance: perception is not a passive process and, depending on the current task, objects are discriminated as possible "tools" that could be used to interact and act in the environment. Whereas a scene full of details can be memorized in very different and costly ways, a task-dependent description is a very economical way that implies minimal storage requirements. Hence, remembering becomes a constructive process.

For example with such a strategy, the organism can keep track of relevant visual targets in the environment by only storing the movement of the eye necessary to foveate them. We do not memorize details of the objects but we know which eye movement to perform to get them: The world itself is considered as an external memory.

Our agreement to this theory has several implications for our methodology of work. In this view, learning emerges from sensorimotor loops and a real body interacting with a real environment are important characteristics for a learning protocol. Also, in this view, the quality of memory (a flexible representation) is preferred to the quantity of memory.

MASAIE Project-Team

3. Scientific Foundations

3.1. Description

Our conceptual framework is that of Control Theory : the system is described by state variables with inputs (actions on the system) and outputs (the available measurements). Our system is either an epidemiological or immunological system or a harvested fish population. The control theory approach begins with the mathematical modeling of the system. When a “satisfying" model is obtained, this model is studied to understand the system. By “satisfying", an ambiguous word , we mean validation of the model. This depends on the objectives of the design of the model: explicative model, predictive model, comprehension model, checking hypotheses model. Moreover the process of modeling is not sequential. During elaboration of the model, a mathematical analysis is often done in parallel to describe the behavior of the proposed model. By behavior we intend not only asymptotic behavior but also such properties as observability, identifiability, robustness ...

3.2. Structure and modeling

Problems in epidemiology, immunology and virology can be expressed as standard problems in control theory. But interesting new questions do arise. The control theory paradigm, input-output systems built out of simpler components that are interconnected, appears naturally in this context. Decomposing the system into several sub-systems, each of which endowed with certain qualitative properties, allow the behavior of the complete system to be deduced from the behavior of its parts. This paradigm, the toolbox of feedback interconnection of systems, has been used in the so-called theory of large-scale dynamic systems in control theory [33]. Reasons for decomposing are multiple. One reason is conceptual. For example connection of the immune system and the parasitic systems is a natural biological decomposition. Others reasons are for the sake of reducing algorithmic complexities or introducing intended behavior ...In this case subsystems may not have biological interpretation. For example a chain of compartments can be introduced to simulate a continuous delay [27], [29]. Analysis of the structure of epidemiological and immunological systems is vital because of the paucity of data and the dependence of behavior on biological hypotheses. The issue is to identify those parts of models that have most effects on dynamics. The concepts and techniques of interconnection of systems (large-scale systems) will be useful in this regard.

In mathematical modeling in epidemiology and immunology, as in most other areas of mathematical modeling, there is always a trade-off between simple models, that omit details and are designed to highlight general qualitative behavior, and detailed models, usually designed for specific situations, including short-terms quantitative predictions. Detailed models are generally difficult to study analytically and hence their usefulness for theoretical purposes is limited, although their strategic value may be high. Simple models can be considered as building blocks of models that include detailed structure. The control theory tools of large-scale systems and interconnections of systems is a mean to conciliate the two approaches, simple models versus detailed systems.

3.3. Dynamic Problems

Many dynamical questions addressed by Systems Theory are precisely what biologist are asking. One fundamental problem is the problem of equilibria and their stability. To quote J.A. Jacquez

> A major project in deterministic modeling of heterogeneous populations is to find conditions for local and global stability and to work out the relations among these stability conditions, the threshold for epidemic take-off, and endemicity, and the basic reproduction number

The basic reproduction number $\mathcal{R}_0$ is an important quantity in the study in epidemics. It is defined as the average number of secondary infections produced when one infected individual is introduced into a host population where everyone is susceptible. The basic reproduction number $\mathcal{R}_0$ is often considered as the threshold quantity that determines when an infection can invade and persist in a new host population. To the problem of stability is related the problem of robustness, a concept from control theory. In other words how near is the system to an unstable one ? Robustness is also in relation with uncertainty of the systems. This is a key point in epidemiological and immunological systems, since there are many sources of uncertainties in these models. The model is uncertain (parameters, functions, structure in some cases), the inputs also are uncertain and the outputs highly variable. That robustness is a fundamental issue and can be seen by means of an example : if policies in public health are to be taken from modeling, they must be based on robust reasons!

3.4. Observers

The concept of observer originates in control theory. This is particularly pertinent for epidemiological systems. To an input-output system, is associated the problem of reconstruction of the state. Indeed for a given system, not all the states are known or measured, this is particularly true for biological systems. This fact is due to a lot of reasons : this is not feasible without destroying the system, this is too expensive, there are no available sensors, measures are too noisy ...The problem of knowledge of the state at present time is then posed. An observer is another system, whose inputs are the inputs and the outputs of the original system and whose output gives an estimation of the state of the original system at present time. Usually the estimation is required to be exponential. In other words an observer, using the signal information of the original system, reconstructs dynamically the state. More precisely, consider an input-output nonlinear system described by

$$\begin{cases} \dot{x} = f(x,u) \\ y = h(x), \end{cases} \tag{2}$$

where $x(t) \in \mathbb{R}^n$ is the state of the system at time t, $u(t) \in U \subset \mathbb{R}^m$ is the input and $y(t) \in \mathbb{R}^q$ is the measurable output of the system.
An observer for the the system (1) is a dynamical system

$$\dot{\widehat{x}}(t) = g(\widehat{x}(t), y(t), u(t)), \tag{3}$$

where the map g has to be constructed such that: the solutions $x(t)$ and $\widehat{x}(t)$ of (1) and (2) satisfy for any initial conditions $x(0)$ and $\widehat{x}(0)$

$$\|x(t) - \widehat{x}(t)\| \leq c\, \|x(0) - \widehat{x}(0)\|\, e^{-a\,t}\ ,\ \ \forall t > 0.$$

or at least $\|x(t) - \widehat{x}(t)\|$ converges to zero as time goes to infinity.
The problem of observers is completely solved for linear time-invariant systems (LTI). This is a difficult problem for nonlinear systems and is currently an active subject of research. The problem of observation and observers (software sensors) is central in nonlinear control theory. Considerable progress has been made in the last decade, especially by the “French school", which has given important contributions (J.P. Gauthier, H. Hammouri, E. Busvelle, M. Fliess, L. Praly, J.L. Gouze, O. Bernard, G. Sallet) and is still very active in this area. Now the problem is to identify relevant class of systems for which reasonable and computable observers can be designed. The concept of observer has been ignored by the modeler community in epidemiology, immunology and virology. To our knowledge there is only one case of use of an observer in virology (Velasco-Hernandez J. , Garcia J. and Kirschner D. [38]) in modeling the chemotherapy of HIV, but this observer, based on classical linear theory, is a local observer and does not allow to deal with the nonlinearities.

3.5. Delays

Another crucial issue for biological systems is the question of delays. Delays, in control theory, are traditionally discrete (more exactly, the delays are lags) whereas in biology they usually are continuous and distributed. For example, the entry of a parasite into a cell initiates a cascade of events that ultimately leads to the production of new parasites. Even in a homogeneous population of cells, it is unreasonable to expect that the time to complete all these processes is the same for every cell. If we furthermore consider differences in cell activation state, metabolism, position in the cell cycle, pre-existing stores of nucleotides and other precursors needed for the reproduction of parasites, along with genetic variations in the parasite population, such variations in infection delay times becomes a near certainty. The rationale for studying continuous delays are supported by such considerations. In the literature on dynamical systems, we find a wealth of theorems dealing with delay differential equations. However they are difficult to apply. Control theory approaches (interconnections of systems), is a mean to study the influence of continuous delays on the stability of such systems. We have obtained some results in this direction [5].

SHACRA Project-Team

3. Scientific Foundations

3.1. Biomechanical Modeling

3.1.1. Biomechanical modeling of solid structures

Soft tissue modeling holds a very important place in medical simulation. A large part of the realism of a simulation, in particular for surgery or laparoscopy simulation, relies upon the ability to describe soft tissue response during the simulated intervention. Several approaches have been proposed over the past ten years to model soft-tissue deformation in real-time (mainly for solid organs), usually based on elasticity theory and a finite element approach to solve the equations. We were among the first to propose such an approach [24], [27] using different computational strategies. Although significant improvements were obtained later on (for instance with the use of co-rotational methods to handle geometrical non-linearities) these works remain of limited clinical use as they rely on linearized constitutive laws.

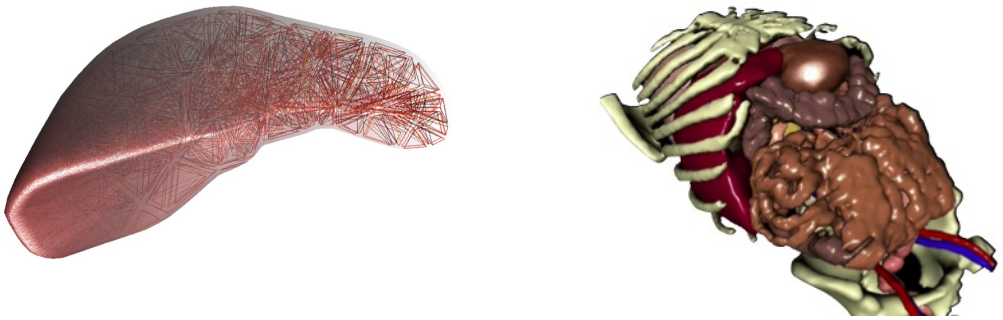

Figure 1. Biomechanical models of organs, based on the Finite Element Method and elasticity theory. Left: a model of the liver based on tetrahedral elements and small strain elasticity. Right: several organ models from a patient dataset combined to create a realistic abdominal anatomy.

An important part of our research is dedicated to the development of new, more accurate models that remain compatible with real-time computation. Such advanced models will not only permit to increase the realism of future training systems, but they will act as a bridge toward the development of patient-specific preoperative planning as well as augmented reality tools for the operating room. Yet, patient-specific planning or per-operative guidance also requires the models to be parametrized with patient-specific biomechanical data. Very little work has been done in this area, in particular when tissue properties need to be measured in vivo non-invasively. New imaging techniques, such as Ultrasound Elastography or Magnetic Resonance Elastography, could be used to this end [23]. We are currently studying the impact of parametrized patient-specific models of the liver in the context of the PASSPORT european project. This will be used to provide information about the deformation, tissue stiffness and tumor location, for various liver pathologies.

3.1.2. Biomechanical modeling of hollow structures

A large number of anatomical structures in the human body are vascularized (brain, liver, heart, kidneys, ...) and recent interventions (such as interventional radiology) rely on the vascular network as a therapeutical pathway. It is therefore essential to model the shape and deformable behavior of blood vessels. This will be

done at two levels. Global deformation of a vascular network: we have demonstrated previously [9] that we could recover the shape of thousands of vessels from medical images by extracting the centerline of each vessel (see Figure 2). The resulting vascular skeleton can be modeled as a deformable (tree) structure which can capture the global aspects of the deformation. More local deformations can then be described by considering now the actual local shape of the vessel. Other structures such as aneurysms, the colon or stomach can also benefit from being modeled as deformable structures. For this we will rely on shell or thin plate theory. We have recently obtained very encouraging results in the context of the Ph.D. thesis of Olivier Comas [26]. Such local and global models of hollow structures will be particularly relevant for planning coil deployment or stent placement, but also in the context of a new laparoscopic technique called NOTES which uses a combination of a flexible endoscope and flexible instruments. Obtaining patient-specific models of vascular structures and associated pathologies remains a challenge from an image processing stand point, and this challenge is even greater once we require these models to be adapted to complex computational strategies. To this extend we will pursue our collaboration with the MAGRIT team at Inria (through a PhD thesis starting in January 2010) and the Massachusetts General Hospital in Boston.

3.1.3. Blood Flow Simulation

Beyond biomechanical modeling of soft tissues, an essential component of a simulation is the modeling of the functional interactions occurring between the different elements of the anatomy. This involves for instance modeling physiological flows (blood flow, air flow within the lungs...). We particularly plan to study the problem of fluid flow in the context of vascular interventions, such as the simulation of three-dimensional turbulent flow around aneurysms to better model coil embolization procedures. Blood flow dynamics is starting to play an increasingly important role in the assessment of vascular pathologies, as well as in the evaluation of pre- and post-operative status. While angiography has been an integral part of interventional radiology procedures for years, it is only recently that detailed analysis of blood flow patterns has been studied as a mean to assess complex procedures, such as coil deployment. A few studies have focused on aneurysm-related hemodynamics before and after endovascular coil embolization. Groden et al. [31] constructed a simple geometrical model to approximate an actual aneurysm, and evaluated the impact of different levels of coil packing on the flow and wall pressure by solving Navier-Stokes equations, while Kakalis et al. [33] relied on patient-specific data to get more realistic flow patterns, and modeled the coiled aneurysm as a porous medium. As these studies aimed at accurate Computational Fluid Dynamics simulation, they rely on commercial software, and the computation times (dozens of hours in general) are incompatible with interactive simulation or even clinical practice. Generally speaking, accuracy and efficiency are two significant pursuits in numerical calculation, but unfortunately very often contradictory.

With the Ph.D. thesis of Yiyi Wei, we have recently started the development of a new technique for accurately computing, in near real-time, the flow of blood within an aneurysm, as well as the interaction between blood and coils. In this approach we rely on the Discrete Exterior Calculus method to obtain an ideal trade-off between accuracy and computational efficiency. Although still at an early stage, these results show that our approach can accurately capture the main characteristics of the complex blood flow patterns in and around an aneurism. The model also takes into account the influence of the coil on the blood flow within the aneurysm. The main difference between our approach and many other work done by internationally renowned teams (such as REO team at Inria or the Computer Vision Laboratory at ETH) comes from the importance we place in the computational efficiency of the method. To some extent our approach is similar to what has been done to obtain real-time finite element methods. We are essentially trying to capture the key characteristics of the behavior for a particular application. This is well illustrated by the work we started on flow modeling, which received an award in September 2009 at the selective conference on Medical Image Computing and Computer Assisted Interventions [10]. We will pursue this direction to accurately model the local flow in a closed domain (blood vessel, aneurysm ventricle, ...) and combine it with some of our previous work describing laminar flow across a large number of vessels [38] in order to define boundary conditions for the three-dimensional model.

3.2. Biomechanical Systems

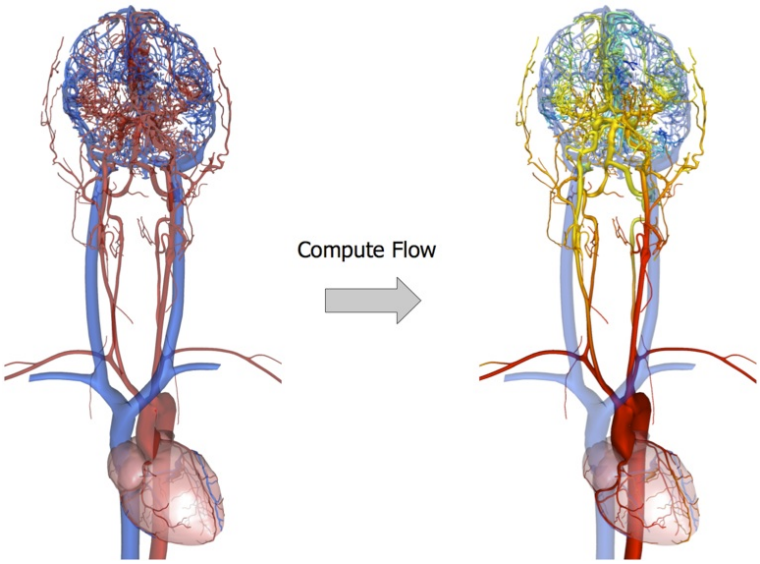

Figure 2. Blood flow and pressure distribution in the cerebrovascular system. The arterial vascular network is composed of more than 3,000 vessels, yet the computation is performed in real-time.

3.2.1. Constraint models and boundary conditions

To accurately model soft tissue deformations, the approach must account for the intrinsic behavior of the target organ, but also for its biomechanical interactions with surrounding tissues or with medical devices. While the biomechanical behavior of important organs (such as the brain or liver) has been well studied, few work exists regarding the mechanical interactions between the anatomical structures. For tissue-tool interactions, most approaches rely on a simple contact models, and rarely account for friction. While this simplification can produce plausible results in the case of an interaction between the end effector of a laparoscopic instrument and the surface of an organ, it is generally an incorrect approximation. As we move towards simulations for planning or rehearsal, accurately modeling contacts will take an increasingly important place. We have recently shown in [28] and [29] that we could compute, in real-time, complex interactions between a coil and an aneurysm, or between a flexible needle and soft-tissues. In laparoscopic surgery, the main challenge lies in the modeling of interactions between anatomical structures rather than between the instruments and the surface of an organ. During the different steps of a procedure organs slides against each other, while respiratory, cardiac and patient motion also generate contacts. Modeling these multiple interactions becomes even more complex when different biomechanical models are used to characterize the various soft tissues of the anatomy. Consequently, our objective is to accurately model resting contacts with friction, in a heterogeneous environment (spring-mass models, finite element models, particle systems, rigid objects, etc.). When different time integration strategies are used, a challenge lies in the computation of contact forces in a way that integrity and stability of the overall simulation are maintained. Our objective is to work on the definition of these various boundary conditions and on new resolution methods for such heterogeneous simulations. In particular we will investigate a simulation process in which each model continues to benefit from its own optimizations while taking into account the mechanical couplings due to interactions between objects.

3.2.2. Vascularized anatomy

From a clinical standpoint, several procedures involve vascularized anatomical structures such as the liver, the kidneys, or the brain. When a therapy needs to be applied on such structures, it is currently possible to perform

a procedure surgically or to use an endovascular approach. This requires to characterize and model the behavior of vessels (arteries and veins) as well as the behavior of soft tissue (in particular the parenchyma). Another challenge of this research will be to model the interactions between the vascular network and the parenchyma where it is embedded. These interactions are key for both laparoscopic surgery and interventional radiology as they allow to describe the motion of the vessels in a vascularized organ during the procedure. This motion is either induced by the surgical manipulation of the parenchymal tissue during surgery or by respiratory, cardiac or patient motion during interventional radiology procedures. From a biomechanical standpoint, capillaries are responsible for the viscoelastic behavior of the vascularized structures, while larger vessels have a direct impact on the overall behavior of the anatomy. In the liver for instance, the apparent stiffness of the organ changes depending on the presence or absence of large vessels. Also, the relatively isotropic nature of the parenchyma is modified around blood vessels. We propose to model the coupling that exists between these two different anatomical structures to account for their respective influence. For this we will initially rely on the work done during the Ph.D. thesis of Christophe Guebert (see ([32] for instance) and we will also investigate coupling strategies based on degrees of freedom reduction to reduce the complexity of the problem (and therefore also computation times). Part of this work is already underway in the context of the PASSPORT european project with IRCAD and soft tissue measurements will be performed in collaboration with the biomechanics laboratory at Strasbourg University.

3.2.3. Parallel Computation

Although the past decade has seen a significant increase in complexity and performance of the algorithms used in medical simulation, major improvements are still required to enable patient-specific simulation and planning. Using parallel architectures to push the complexity of simulated environments further is clearly an approach to consider. However, interactive simulations introduce new constraints and evaluation criteria, such as latencies, multiple update frequencies and dynamic adaptation of precision levels, which require further investigation. New parallel architectures, such as multi-cores CPUs, are now ubiquitous as the performances achieved by sequential units (single core CPUs) stopped to regularly improve. At the same time, graphical processors (GPU) offer a massive computing power that is now accessible to non-graphical tasks thanks to new general-purposes API such as CUDA and OpenCL. GPUs are internally parallel processors, exploiting hundreds of computing units. These architectures can be exploited for more ambitious simulations, as we already have demonstrated in a first step by adding support for CUDA within the SOFA framework. Several preliminary results of GPU-based simulations have been obtained, permitting to reach speedup factors (compared to a single core GPU) ranging from 16x to 55x. Such improvements permit to consider simulations with finer details, or new algorithms modeling biomechanical behaviors more precisely. However, while the fast evolution of parallel architectures is useful to increase the realism of simulations, their varieties (multi-core CPUs, GPUs, clusters, grids) make the design of parallel algorithm challenging. An important effort needs to be made is to minimize the dependency between simulation algorithms and hardware architectures, allowing the reuse of parallelization efforts on all architecture, as well as simultaneously exploiting all available computing resources present in current and future computers. The largest gains could be achieved by combining parallelism and adaptive algorithms. The design and implementation of such a system is a challenging problem, as it is no longer possible to rely on pre-computed repartition of datas and computations. Thus, further research is required in highly adaptive parallel scheduling algorithms, and highly efficient implementation able to handle both large changes in computational loads due to user interactions and multi-level algorithms, and new massively parallel architectures such as GPUs. A direction that we are also investigating is to combine multi-level representations and locally adaptive meshes. Multi-level algorithms are useful not only to speedup computations, but also to describe different characteristics of the deformation at each level. Combined with local change of details of the mesh (possibly using hierarchical structures), the simulation can reach a high level of scalability.

ALGORILLE Project-Team

3. Scientific Foundations

3.1. Structuring Applications

Computing on different scales is a challenge under constant development that, almost by definition, will always try to reach the edge of what is possible at any given moment in time: in terms of the scale of the applications under consideration, in terms of the efficiency of implementations and in what concerns the optimized utilization of the resources that modern platforms provide or require. The complexity of all these aspects is currently increasing rapidly:

3.1.1. Diversity of platforms.

Design of processing hardware is diverging in many different directions. Nowadays we have SIMD registers inside processors, on-chip or off-chip accelerators (GPU, FPGA, vector-units), multi-cores and hyperthreading, multi-socket architectures, clusters, grids, clouds... The classical monolithic architecture of one-algorithm/one-implementation that solves a problem is obsolete in many cases. Algorithms (and the software that implements them) must deal with this variety of execution platforms robustly.

As we know, the "*free lunch*" for sequential algorithms provided by the increase of processor frequencies is over, we have to go parallel. But the "*free lunch*" is also over for many automatic or implicit adaption strategies between codes and platforms: e.g the best cache strategies can't help applications that accesses memory randomly, or algorithms written for "simple" CPU (von Neumann model) have to be adapted substantially to run efficiently on vector units.

3.1.2. The communication bottleneck.

Communication and processing capacities evolve at a different pace, thus the *communication bottleneck* is always narrowing. An efficient data management is becoming more and more crucial.

Not many implicit data models have yet found their place in the HPC domain, because of a simple observation: latency issues easily kill the performance of such tools. In the best case, they will be able to hide latency by doing some intelligent caching and delayed updating. But they can never hide the bottleneck for bandwidth.

HPC was previously able to cope with the communication bottleneck by using an explicit model of communication, namely MPI. It has the advantage of imposing explicit points in code where some guarantees about the state of data can be given. It has the clear disadvantage that coherence of data between different participants is difficult to manage and is completely left to the programmer.

Here, our approach is and will be to timely request explicit actions (like MPI) that mark the availability of (or need for) data. Such explicit actions ease the coordination between tasks (coherence management) and allow the platform underneath the program to perform a pro-active resource management.

3.1.3. Models of interdependence and consistency

Interdependence of data between different tasks of an application and components of hardware will be crucial to ensure that developments will possibly scale on the ever diverging architectures. We have up to now presented such models (PRO, DHO, ORWL) and their implementations, and proved their validity for the context of SPMD-type algorithms.

Over the next years we will have to enlarge the spectrum of their application. On the algorithm side we will have to move to heterogeneous computations combining different types of tasks in one application. For the architectures we will have to take into account the fact of increased heterogeneity, processors of different speed, multi-cores, accelerators (FPU, GPU, vector units), communication links of different bandwidth and latency, memory and generally storage capacity of different size, speed and access characteristics. First implementations using ORWL in that context look particularly promising.

The models themselves will have to evolve to be better suited for more types of applications, such that they allow for a more fine-grained partial locking and access of objects. They should handle e.g collaborative editing or the modification of just some fields in a data structure. This work has already started with DHO which allows the locking of *data ranges* inside an object. But a more structured approach would certainly be necessary here to be usable more comfortably in applications.

3.1.4. Frequent IO

A complete parallel application includes I/O of massive data, at an increasing frequency. In addition to applicative input and output data flow, I/O is used for checkpointing or to store traces of execution. These then can be used to restart in case of failure (hardware or software) or for a post-mortem analysis of a chain of computations that led to catastrophic actions (for example in finance or in industrial system control). The difficulty of frequent I/O is more pronounced on hierarchical parallel architectures that include accelerators with local memory.

I/O has to be included in the design of parallel programming models and tools. ORWL will be enriched with such tools and functionalities, in order to ease the modeling and development of parallel applications that include data IO, and to exploit most of the performance potential of parallel and distributed architectures.

3.1.5. Algorithmic paradigms

Concerning asynchronous algorithms, we have developed several versions of implementations, allowing us to precisely study the impact of our design choices. However, we are still convinced that improvements are possible in order to extend its application domain, especially concerning the detection of global convergence and the control of asynchronism. We are currently working on the design of a generic and non-intrusive way of implementing such a procedure in any parallel iterative algorithm.

Also, we would like to compare other variants of asynchronous algorithms, such as waveform relaxations. Here, computations are not performed for each time step of the simulation but for an entire time interval. Then, the evolution of the elements at the frontiers between the domain that are associated to the processors are exchanged asynchronously. Although we have already studied such schemes in the past, we would like to see how they will behave on recent architectures, and how the models and software for data consistency mentioned above can be helpful in that context.

3.1.6. Cost models and accelerators

We have already designed some models that relate computation power and energy consumption. Our next goal is to design and implement an auto-tuning system that controls the application according to user defined optimization criteria (computation and/or energy performance). This implies the insertion of multi-schemes and/or multi-kernels into the application such that it will able to adapt its behavior to the requirements.

3.2. Transparent Resource Management for Clouds

During the next years, we will continue to design resource provisioning strategies for cloud clients. Given the extremely large offer of resources by public or private clouds, users need software assistance to make provisioning decisions. Our goal is to gather our strategies into a **cloud resource broker** which will handle the workload of a user or of a community of users as a multi-criteria optimization problems. The notions of resource usage, scheduling, provisioning and task management have to be adapted to this new context. For example, to minimize the makespan of a DAG of tasks, usually a fixed number of resources is assumed. On IaaS clouds, the amount of resources can be provisioned at any time, and hence the scheduling problem must be redefined: the new prevalent optimization criterion is the financial cost of the computation.

3.2.1. Provisioning strategies

Future work includes the design of new strategies to reuse already leased resources, or switch to less powerful and cheaper resources. On one hand, some economic models proposed by cloud providers may involve a complex cost-benefit analysis for the client which we want to address. On the other hand, these economic models incur additional costs, e.g for data storage or transfer, which must be taken into account to design a comprehensive broker.

3.2.2. User workload analysis

Another possible extension of the capability of such a broker, is user workload analysis. Characterizing the workload may help to anticipate the resource provisioning, and hence improve the scheduling.

3.2.3. Experimentations

Given the very large consumption of CPU hours, the above strategies will first be tested mostly through simulation. Therefore, we will closely work with the members of the Experimental methodologies axis to co-design the cloud interface and the underlying models. Furthermore, we will assess the gap between the performances on simulation and both public and private cloud. This work will take place inside the Cloud work package of the SONGS ANR project.

3.2.4. HPC on clouds

Clouds are not suitable to run massive HPC applications. However, it might be interesting to use them as cheap HPC platform for occasional or one shot executions. This will be investigated with the Structuring Applications axis and in collaboration with the LabEx IRMIA and the CALVI team.

3.3. Experimental methodologies for the evaluation of distributed systems

We strive at designing a comprehensive set of solutions for experimentation on distributed systems by working on several methodologies (simulation, direct execution on experimental facilities, emulation) and by leveraging the convergence opportunities between methodologies (shared interfaces, validation combining several methodologies).

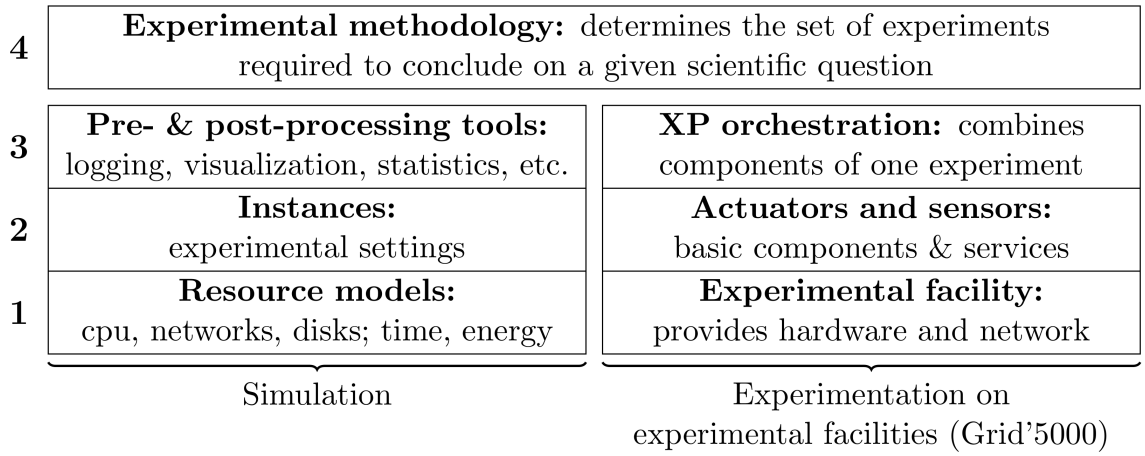

Figure 2. Our experimentation methodology, encompassing both simulation and experimental facilities.

3.3.1. Simulation and dynamic verification

Our team plays a key role in the SimGrid project, a mature simulation toolkit widely used in the distributed computing community. Since more than ten years, we work on the validity, scalability and robustness of our tool. Recent, we increased its audience to target the P2P research community in addition to the one on grid scheduling. It now allows **precise simulations of millions of nodes** using a single computer.

In the future, we aim at extending further the applicability to **Clouds and Exascale systems**. Therefore, we work toward disk and memory models in addition to the already existing network and CPU models. The tool's scalability and efficiency also constitutes a permanent concern to us. **Interfaces** constitute another important work axis, with the addition of specific APIs on top of our simulation kernel. They provide the "syntactic sugar" needed to express algorithms of these communities. For example, virtual machines are handled explicitly in the interface provided for Cloud studies. Similarly, we pursue our work on an implementation of the MPI standard allowing to study real applications using that interface. This work may also be extended in the future to other interfaces such as OpenMP or the ones developed in our team to structure applications, in particular ORWL. In the near future, we also consider using our toolbox to give **online performance predictions to the runtimes**. It would allow these systems to improve their adaptability to the changing performance conditions experienced on the platform.

We recently integrated a model checking kernel in our tool to enable **formal correctness studies** in addition to the practical performance studies enabled by simulation. Being able to study these two fundamental aspects of distributed applications within the same tool constitutes a major advantage for our users. In the future, we will enforce this capacity for the study of correctness and performance such that we hope to tackle their usage on real applications.

3.3.2. Experimentation using direct execution on testbeds and production facilities.

Our work in this research axis is meant to bring major contributions to the **industrialization of experimentation** on parallel and distributed systems. It is structured through multiple layers that range from the design of a testbed supporting high-quality experimentation, to the study of how stringent experimental methodology could be applied to our field, see Figure 3 ,

During the last years, we have played a **key role in the design and development of Grid'5000** by leading the design and technical developments, and by managing several engineers working on the platform. We pursue our involvement in the design of the testbed with a focus on ensuring that the testbed provides all the features needed for high-quality experimentation. We also collaborate with other testbeds sharing similar goals in order to exchange ideas and views. We now work on **basic services supporting experimentation** such as resources verification, management of experimental environments, control of nodes, management of data, etc. Appropriate collaborations will ensure that existing solutions are adopted to the platform and improved as much as possible.

One key service for experimentation is the ability to alter experimental conditions using emulation. We work on the **Distem emulator**, focusing on its validation and on adding features such as the ability to emulate faults, varying availability, churn, load injection, ...and investigate if altering memory and disk performance is possible. Other goals are to scale the tool up to 20000 virtual nodes and to improve its usability and documentation.

We work on **orchestration of experiments** in order to combine all the basic services mentioned previously in an efficient and scalable manner. Our approach is based on the reuse of lessons learned in the field of Business Process Management (BPM), with the design of a workflow-based experiment control engine. This is part of an ongoing collaboration with EPI SCORE (INRIA Nancy Grand Est), which has already yield promising preliminary results [15], [28].

3.3.3. Exploring new scientific objects.

We aim at addressing different kinds of distributed systems (HPC, Cloud, P2P, Grid) using the same experimental approaches. Thus a key requirement for our success is to build sufficient knowledge on target distributed systems to discover and understand the final research questions that our solutions should target. In the framework of ANR SONGS (2012-2016), we are working closely with experts from HPC, Cloud, P2P and Grid. We are also collaborating with the production grids community, e.g. on using Grid'5000 to evaluate the gLite middleware, and with Cloud experts in the context of the OpenCloudWare project.

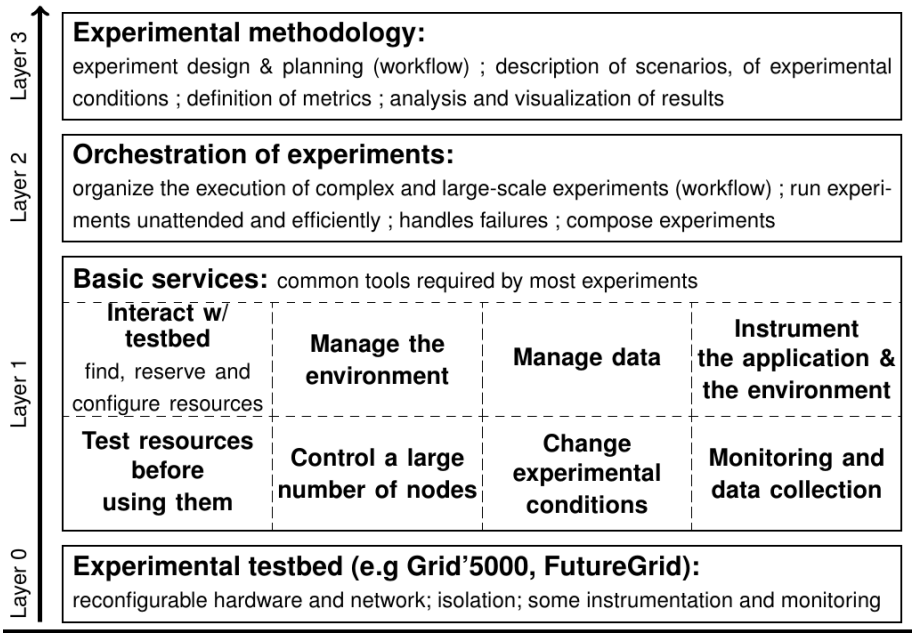

Figure 3. General structure of our project: We plan to address all layers of the experimentation stack.

MADYNES Project-Team

3. Scientific Foundations

3.1. Evolutionary needs in network and service management

The foundation of the MADYNES research activity is the ever increasing need for automated monitoring and control within networked environments. This need is mainly due to the increasing dependency of both people and goods towards communication infrastructures as well as the growing demand towards services of higher quality. Because of its strategic importance and crucial requirements for interoperability, the management models were constructed in the context of strong standardization activities by many different organizations over the last 15 years. This has led to the design of most of the paradigms used in today's deployed approaches. These paradigms are the Manager/Agent interaction model, the Information Model paradigm and its container, together with a naming infrastructure called the Management Information Base. In addition to this structure, five functional areas known under Fault, Configuration, Accounting, Performance and Security are associated to these standards.

While these models were well suited for the specific application domains for which they were designed (telecommunication networks or dedicated protocol stacks), they all show the same limits. Especially they are unable:

1. to deal with any form of dynamicity in the managed environment,
2. to master the complexity, the operating mode and the heterogeneity of the emerging services,
3. to scale to new networks and service environments.

These three limits are observed in all five functional areas of the management domain (fault, configuration, accounting, performance and security) and represent the major challenges when it comes to enable effective automated management and control of devices, networks and services in the next decade.

MADYNES addresses these challenges by focusing on the design of management models that rely on inherently dynamic and evolving environments. The project is centered around two core activities. These activities are, as mentioned in the previous section, the design of an autonomous management framework and its application to three of the standard functional areas namely security, configuration and performance.

3.2. Autonomous management

3.2.1. Models and methods for a self-management plane

Self organization and automation are fundamental requirements within the management plane in today's dynamic environments. It is necessary to automate the management processes and enable management frameworks to operate in time sensitive evolving networks and service environments. The automation of the organization of devices, software components, networks and services is investigated in many research projects and has already led to several solution proposals. While these proposals are successful at several layers, like IP auto-configuration or service discovery and binding facilities, they did not enhance the management plane at all. For example, while self-configuration of IP devices is commonplace, no solution exists that provides strong support to the management plane to configure itself (e.g. finding the manager to which an agent has to send traps or organizing the access control based on locality or any other context information). So, this area represents a major challenge in extending current management approaches so that they become self-organized.

Our approach is bottom-up and consists in identifying those parameters and framework elements (manager data, information model sharing, agent parameters, protocol settings, ...) that need dynamic configuration and self-organization (like the address of a trap sink). For these parameters and their instantiation in various management frameworks (SNMP, Netconf, WBEM, ...), we investigate and elaborate novel approaches enabling fully automated setup and operation in the management plane.

3.2.2. Design and evaluation of P2P-based management architectures

Over the last years, several models have emerged and gained wide acceptance in the networking and service world. Among them, the overlay networks together with the P2P paradigms appear to be very promising. Since they rely mainly on fully decentralized models, they offer excellent fault tolerance and have a real potential to achieve high scalability. Mainly deployed in the content delivery and the cooperation and distributed computation disciplines, they seem to offer all features required by a management framework that needs to operate in a dynamic world. This potential however needs an in depth investigation because these models have also many characteristics that are unusual in management (e.g. a fast and uncontrolled evolution of the topology or the existence of a distributed trust relationship framework rather than a standard centralized security framework).

Our approach envisions how a complete redesign of a management framework is done given the characteristics of the underlying P2P and overlay services. Among the topics of interest we study the concept of management information and operations routing within a management overlay as well as the distribution of management functions in a multi-manager/agent P2P environment. The functional areas targeted in our approach by the P2P model are network and service configuration and distributed monitoring. The models are to be evaluated against highly dynamic frameworks such as ad-hoc environments (network or application level) and mobile devices.

3.2.3. Integration of management information

Representation, specification and integration of management information models form a foundation for network and service management and remains an open research domain. The design and specification of new models is mainly driven by the appearance of new protocols, services and usage patterns. These need to be managed and exposed through well designed management information models. Integration activities are driven by the multiplication of various management approaches. To enable automated management, these approaches need to inter-operate which is not the case today.

The MADYNES approach to this problem of modeling and representation of management information aims at:

1. enabling application developers to establish their management interface in the same workspace, with the same notations and concepts as the ones used to develop their application,
2. fostering the use of standard models (at least the structure and semantics of well defined models),
3. designing a naming structure that allows the routing of management information in an overlay management plane, and
4. evaluating new approaches for management information integration especially based on management ontologies and semantic information models.

3.2.4. Modeling and benchmarking of dynamic networks

The impact of a management approach on the efficiency of the managed service is highly dependent on three factors:

- the distribution of the considered service and their associated management tasks,
- the management patterns used (e.g. monitoring frequency, granularity of the management information considered),
- the cost in terms of resources these considered functions have on the managed element (e.g. method call overhead, management memory footprint).

MADYNES addresses this problem from multiple viewpoints: communication patterns, processing and memory resources consumption. Our goal is to provide management patterns combining optimized management technologies so as to optimize the resources consumed by the management activity imposed by the operating environment while ensuring its efficiency in large dynamic networks.

3.3. Functional areas

3.3.1. Security management

Securing the management plane is vital. While several proposals are already integrated in the existing management frameworks, they are rarely used. This is due to the fact that these approaches are completely detached from the enterprise security framework. As a consequence, the management framework is "managed" separately with different models; this represents a huge overhead. Moreover the current approaches to security in the management plane are not inter-operable at all, multiplying the operational costs in a heterogeneous management framework.

The primary goal of the research in this activity is the design and the validation of a security framework for the management plane that will be open and capable to integrate the security services provided in today's management architectures. Management security interoperability is of major importance in this activity.

Our activity in this area aims at designing a generic security model in the context of multi-party / multi-technology management interactions. Therefore, we develop research on the following directions:

1. Abstraction of the various access control mechanisms that exist in today's management frameworks. We are particularly interested in extending these models so that they support event-driven management, which is not the case for most of them today.
2. Extension of policy and trust models to ease and to ensure coordination among managers towards one agent or a subset of the management tree. Provisional policies are of great interest to us in this context.
3. Evaluation of the adequacy of key distribution architectures to the needs of the management plane as well as selecting reputation models to be used in the management of highly dynamic environments (e.g. multicast groups, ad-hoc networks).

A strong requirement towards the future generic model is that it needs to be instantiated (with potential restrictions) into standard management platforms like SNMP, WBEM or Netconf and to allow interoperability in environments where these approaches coexist and even cooperate. A typical example of this is the security of an integration agent which is located in two management worlds.

Since 2006 we have also started an activity on security assessment. The objective is to investigate new methods and models for validating the security of large scale dynamic networks and services. The first targeted service is VoIP.

3.3.2. Configuration: automation of service configuration and provisioning

Configuration covers many processes which are all important to enable dynamic networks. Within our research activity, we focus on the operation of tuning the parameters of a service in an automated way. This is done together with the activation topics of configuration management and the monitoring information collected from the underlying infrastructure. Some approaches exist today to automate part of the configuration process (download of a configuration file at boot time within a router, on demand code deployment in service platforms). While these approaches are interesting they all suffer from the same limits, namely:

1. they rely on specific service life cycle models,
2. they use proprietary interfaces and protocols.

These two basic limits have high impacts on service dynamics in a heterogeneous environment.

We follow two research directions in the topic of configuration management. The first one aims at establishing an abstract life-cycle model for either a service, a device or a network configuration and to associate with this model a generic command and programming interface. This is done in a way similar to what is proposed in the area of call control in initiatives such as Parlay or OSA.

In addition to the investigation of the life-cycle model, we work on technology support for distributing and exchanging configuration management information. Especially, we investigate policy-driven approaches for representing configurations and constraints while we study XML-based protocols for coordinating distribution and synchronization. Off and online validation of configuration data is also part of this effort.

3.3.3. Performance and availability monitoring

Performance management is one of the most important and deployed management function. It is crucial for any service which is bound to an agreement about the expected delivery level. Performance management needs models, metrics, associated instrumentation, data collection and aggregation infrastructures and advanced data analysis algorithms.

Today, a programmable approach for end-to-end service performance measurement in a client server environment exists. This approach, called Application Response Measurement (ARM) defines a model including an abstract definition of a unit of work and related performance records; it offers an API to application developers which allows easy integration of measurement within their distributed application. While this approach is interesting, it is only a first step toward the automation of performance management.

We are investigating two specific aspects. First we are working on the coupling and possible automation of performance measurement models with the upper service level agreement and specification levels. Second we are working on the mapping of these high level requirements to the lower level of instrumentation and actual data collection processes available in the network. More specifically we are interested in providing automated mapping of service level parameters to monitoring and measurement capabilities. We also envision automated deployment and/or activation of performance measurement sensors based on the mapped parameters. This activity also incorporates self-instrumentation (and when possible on the fly instrumentation) of software components for performance monitoring purpose.

SCORE Team

3. Scientific Foundations

3.1. Introduction

Our scientific foundations are grounded on distributed collaborative systems supported by sophisticated data sharing mechanisms and an service oriented computing with an emphasis on orchestration and on non functional properties.

Distributed collaborative systems enable distributed group work supported by computer technologies. Designing such systems require an expertise in Distributed Systems and in Computer-supported collaborative activities research area. Besides theoretical and technical aspects of distributed systems, design of distributed collaborative systems must take into account the human factor to offer solutions suitable for users and groups. The Score team vision is to move away from a centralized authority based collaboration towards a decentralized collaboration where users have full control over their data that they can store locally and decide with whom to share them. The Score team investigated the issues related to the management of distributed shared data and coordination between users and groups.

Service oriented Computing [29] is an established domain on which the ECOO and now the Score team has been contributing for a long time. It refers to the general discipline that studies the development of computer applications on the web. A service is an independent software program with a specific functional context and capabilities published as a service contract (or more traditionally an API). A service composition aggregates a set of services and coordinate their interactions. The scale, the autonomy of services, the heterogeneity and some design principles underlying Service Oriented Computing open new research questions that are at the basis of our research. They spans the disciplines of distributed computing, software engineering and CSCW. Our approach to contribute to the general vision of Service Oriented Computing and more generally to the emerging discipline of Service Science has been and is still to focus on the question of the efficient and flexible construction of reliable and secure high level services through the coordination/orchestration/composition of other services provided by distributed organizations or people.

3.2. Consistency Models for Distributed Collaborative Systems

Collaborative systems are distributed systems that allow users to share data. One important issue is to manage consistency of shared data according to concurrent access. Traditional consistency criteria such as locking, serializability, linearizability are not adequate for collaborative systems.

Causality, Convergence and Intention preservation (CCI) [32] are more suitable for developing middleware for collaborative applications.

We develop algorithms for ensuring CCI properties on collaborative distributed systems. Constraints on the algorithms are different according to the type of distributed system and type of data. The distributed system can be centralized, decentralized or peer-to-peer. The type of data can include strings, growable arrays, ordered trees, semantic graphs and multimedia data.

3.3. Optimistic Replication

Replication of data among different nodes of a network allows improving reliability, fault-tolerance, and availability. When data are mutable, consistency among the different replicas must be ensured. Pessimistic replication is based on the principle of single-copy consistency while optimistic replication allows the replicas to diverge during a short time period. The consistency model for optimistic replication [31] is called eventual consistency, meaning that replicas are guaranteed to converge to the same value when the system is idle.

Our research focuses on the two most promising families of optimistic replication algorithms for ensuring CCI:

- the operational transformation (OT) algorithms [27]
- the algorithms based on commutative replicated data types (CRDT) [30]

Operational transformation algorithms are based on the application of a transformation function when a remote modification is integrated into the local document. Integration algorithms are generic, being parametrized by operational transformation functions which depend on replicated document types. The advantage of these algorithms is their genericity. These algorithms can be applied to any data type and they can merge heterogeneous data in a uniform manner.

Commutative replicated data types is a new class of algorithms initiated by WOOT [28] a first algorithm designed WithOut Operational Transformations. They ensure consistency of highly dynamic content on peer-to-peer networks. Unlike traditional optimistic replication algorithms, they can ensure consistency without concurrency control. CRDT algorithms rely on natively commutative operations defined on abstract data types such as lists or ordered trees. Thus, they do not require a merge algorithm or an integration procedure.

3.4. Business Process Management

Business Process Management (BPM) is considered as a core discipline behind Service Management and Computing. BPM, that includes the analysis, the modelling, the execution, the monitoring and the continuous improvement of enterprise processes is for us a central domain of studies.

A lot of efforts has been devoted in the past years to established standards business process models founded on well grounded theories (e.g. Petri Nets) that meet the needs of both business analyst but also of software engineers and software integrators. This has lead to heated debate as both points of view are very difficult to reconciliate between the analyst side and the IT side. On one side, the business people in general require models that are easy to use and understand and that can be quickly adapted to exceptional situations. On the other side, IT people need models with an operational semantic in order to be able transform them into executable artifacts. Part of our work has been an attempt to reconcile these point of views, leading on one side to the Bonita product and more recently on our work in crisis management where the same people are designing, executing and monitoring the process as it executes. But more generally, and at a larger scale, we have been considering the problem of process spanning the barriers of organisations. This leads us to consider the more general problem of service composition as a way to coordinate inter organisational construction of application providing value based on the composition of lower level services [26].

3.5. Service Composition

More and more, we are considering processes as piece of software whose execution traverse the boundaries of organisations. This is especially true with service oriented computing where processes compose services produced by many organisations. We tackle this problem from very different perspectives, trying to find the best compromise between the need for privacy of internal processes from organisations and the necessity to publicize large part of them, proposing to distribute the execution and the orchestration of processes among the organisations themselves, and attempting to ensure non-functional properties in this distributed setting [25].

Non functional aspects of service composition relate to all the properties and service agreements that one want to ensure and that are orthogonal to the actual business but that are important when a service is selected and integrated in a composition. This includes transactional context, security, privacy, and quality of service in general. Defining and orchestrating services on a large scale while providing the stakeholders with some strong guarantees on their execution is a first class problem for us. For a long time, we have proposed models and solutions to ensure that some properties (e.g. transactional properties) were guaranteed on process execution, either through design or through the definition of some protocols. Our work has also been extended to the problems of security, privacy and service level agreement among partners. These questions are still central in our work. Then, one major problem of current approaches is to monitor the execution of the

compositions, integrating the distributed dimension. This problem can be tackled using event-based algorithms and techniques. Using our previous results an event oriented composition framework DISC, we have obtain new results dedicated to the runtime verification of violations in services choreographies [6], [7], [12]

ALICE Project-Team

3. Scientific Foundations

3.1. Introduction

Computer Graphics is a quickly evolving domain of research. These last few years, both acquisition techniques (e.g., range laser scanners) and computer graphics hardware (the so-called GPU's, for Graphics Processing Units) have made considerable advances. However, despite these advances, fundamental problems still remain open. For instance, a scanned mesh composed of hundred million triangles cannot be used directly in real-time visualization or complex numerical simulation. To design efficient solutions for these difficult problems, ALICE studies two fundamental issues in Computer Graphics:

- the representation of the objects, i.e., their geometry and physical properties;
- the interaction between these objects and light.

Historically, these two issues have been studied by independent research communities. However, we think that they share a common theoretical basis. For instance, multi-resolution and wavelets were mathematical tools used by both communities [25]. We develop a new approach, which consists in studying the geometry and lighting from the *numerical analysis* point of view. In our approach, geometry processing and light simulation are systematically restated as a (possibly non-linear and/or constrained) functional optimization problem. This type of formulation leads to algorithms that are more efficient. Our long-term research goal is to find a formulation that permits a unified treatment of geometry and illumination over this geometry.

3.2. Geometry Processing for engineering

Participants: Laurent Alonso, Dobrina Boltcheva, Alejandro Galindo, Phuong Ho, Samuel Hornus, Thomas Jost, Bruno Lévy, David Lopez, Romain Merland, Vincent Nivoliers, Jeanne Pellerin, Nicolas Ray, Dmitry Sokolov, Rhaleb Zayer.

Mesh processing, parameterization, splines

Geometry processing recently emerged (in the middle of the 90's) as a promising strategy to solve the geometric modeling problems encountered when manipulating meshes composed of hundred millions of elements. Since a mesh may be considered to be a *sampling* of a surface - in other words a *signal* - the *digital signal processing* formalism was a natural theoretic background for this subdomain (see e.g., [26]). Researchers of this domain then studied different aspects of this formalism applied to geometric modeling.

Although many advances have been made in the geometry processing area, important problems still remain open. Even if shape acquisition and filtering is much easier than 30 years ago, a scanned mesh composed of hundred million triangles cannot be used directly in real-time visualization or complex numerical simulation. For this reason, automatic methods to convert those large meshes into higher level representations are necessary. However, these automatic methods do not exist yet. For instance, the pioneer Henri Gouraud often mentions in his talks that the *data acquisition* problem is still open. Malcolm Sabin, another pioneer of the "Computer Aided Geometric Design" and "Subdivision" approaches, mentioned during several conferences of the domain that constructing the optimum control-mesh of a subdivision surface so as to approximate a given surface is still an open problem. More generally, converting a mesh model into a higher level representation, consisting of a set of equations, is a difficult problem for which no satisfying solutions have been proposed. This is one of the long-term goals of international initiatives, such as the AIMShape European network of excellence.

Motivated by gridding application for finite elements modeling for oil and gas exploration, in the frame of the Gocad project, we started studying geometry processing in the late 90's and contributed to this area at the early stages of its development. We developed the LSCM method (Least Squares Conformal Maps) in cooperation with Alias Wavefront [5]. This method has become the de-facto standard in automatic unwrapping, and was adopted by several 3D modeling packages (including Maya and Blender). We experimented various applications of the method, including normal mapping, mesh completion and light simulation [2].

However, classical mesh parameterization requires to partition the considered object into a set of topological disks. For this reason, we designed a new method (Periodic Global Parameterization) that generates a continuous set of coordinates over the object [6]. We also showed the applicability of this method, by proposing the first algorithm that converts a scanned mesh into a Spline surface automatically [4].

We are still not fully satisfied with these results, since the method remains quite complicated. We think that a deeper understanding of the underlying theory is likely to lead to both efficient and simple methods. For this reason, we studied last year several ways of discretizing partial differential equations on meshes, including Finite Element Modeling and Discrete Exterior Calculus. This year, we also explored Spectral Geometry Processing and Sampling Theory (more on this below).

3.3. Computer Graphics

Participants: Sylvain Lefebvre, Samuel Hornus, Bruno Lévy, Vincent Nivoliers, Nicolas Ray, Dmitry Sokolov, Rhaleb Zayer.

texture synthesis, texture mapping,

Content creation is one of the major challenge in Computer Graphics. Modelling geometries and surface appearances which are visually appealing and at the same time enforce precise design constraints is a task only accessible to highly skilled and trained designers.

In this context the team focuses on methods for by-example content creation. Given an input example and a set of constraints, we design algorithms that can automatically generate a new shape (geometry+texture). We formulate the problem of content synthesis as the joint optimization of several objectives: Preserving the local appearance of the example, enforcing global objectives (size, symmetries, mechanical properties), reaching user defined constraints (locally specified geometry, contacts). This results in a wide range of optimization problems, from statistical approaches (Markov Random fields), to combinatorial and linear optimization techniques.

In addition to the core algorithm we also work on the representation of the content, so as to allow for its efficient manipulation. In this context we develop data-structures and algorithms targeted at massively parallel architectures, such as GPUs. These are critical to reach the interactive rates expected from a content creation technique. We also propose novel ways to store and access content stored along surfaces [7] or in volumes [1].

MAGRIT Project-Team

3. Scientific Foundations

3.1. Camera calibration and registration

One of the most basic problems currently limiting Augmented Reality applications is the registration problem. The objects in the real and virtual worlds must be properly aligned with respect to each other, or the illusion that the two worlds coexist will be compromised.

As a large number of potential AR applications are interactive, real time pose computation is required. Although the registration problem has received a lot of attention in the computer vision community, the problem of real-time registration is still far from being a solved problem, especially for unstructured environments. Ideally, an AR system should work in all environments, without the need to prepare the scene ahead of time, and the user should walk anywhere he pleases.

For several years, the Magrit project has been aiming at developing on-line and marker-less methods for camera pose computation. We have especially proposed a real-time system for camera tracking designed for indoor scenes [1]. The main difficulty with on-line tracking is to ensure robustness of the process. For off-line processes, robustness is achieved by using spatial and temporal coherence of the considered sequence through move-matching techniques. To get robustness for open-loop systems, we have developed a method which combines the advantage of move-matching methods and model-based methods by using a piecewise-planar model of the environment. This methodology can be used in a wide variety of environments: indoor scenes, urban scenes .... We are also concerned with the development of methods for camera stabilization. Indeed, statistical fluctuations in the viewpoint computations lead to unpleasant jittering or sliding effects, especially when the camera motion is small. We have proved that the use of model selection allows us to noticeably improve the visual impression and to reduce drift over time.

The success of pose computation largely depends on the quality of the matching stage over the sequence. Research are conducted in the team on the use of probabilistic methods to establish robust correspondences of features over time. The use of *a contrario* decision is under investigation to achieve this aim [3]. We especially address the complex case of matching in scenes with repeated patterns which are common in urban scenes. We also consider learning based techniques to improve the robustness of the matching stage.

Another way to improve the reliability and the robustness of pose algorithms is to combine the camera with another form of sensor in order to compensate for the shortcomings of each technology. Each technology approach has limitations: on the one hand, rapid head motions cause image features to undergo large motion between frames that may cause visual tracking to fail. On the other hand, inertial sensors response is largely independent from the user's motion but their accuracy is bad and their response is sensitive to metallic objects in the scene. In past works [1], we have proposed a system that makes an inertial sensor cooperate with the camera-based system in order to improve the robustness of the AR system to abrupt motions of the users, especially head motions. This work contributes to the reduction of the constraints on the users and the need to carefully control the environment during an AR application. Ongoing research on such hybrid systems are under consideration in our team with the aim to improve the accuracy of reconstruction techniques as well as to obtain dynamic models of organs in medical applications.

Finally, it must be noted that the registration problem must be addressed from the specific point of view of augmented reality: the success and the acceptance of an AR application does not only depend on the accuracy of the pose computation but also on the visual impression of the augmented scene. The search for the best compromise between accuracy and perception is therefore an important issue in this project. This research topic has been addressed in our project both in classical AR and in medical imaging in order to choose the camera model, including intrinsic parameters, which describes at best the considered camera.

3.2. Scene modeling

Modeling the scene is a fundamental issue in AR for many reasons. First, pose computation algorithms often use a model of the scene or at least some 3D knowledge on the scene. Second, effective AR systems require a model of the scene to support occlusion and to compute light reflexions between the real and the virtual objects. Unlike pose computation which has to be computed in a sequential way, scene modeling can be considered as an off-line or an on-line problem according to the application. Within the team we have developed interactive in-situ modeling techniques dedicated to classical AR applications. We also developed off-line multimodal techniques dedicated to AR medical applications.

In-situ modeling
Most automatic techniques aim at reconstructing a sparse and thus unstructured set of points of the scene. Such models are obviously not appropriate to perform interaction with the scene. In addition, they are incomplete in the sense that they may omit features which are important for the accuracy of the pose recovered from 2D/3D correspondences. We have thus investigated interactive techniques with the aim of obtaining reliable and structured models of the scene. The goal of our approach is to develop immersive and intuitive interaction techniques which allow for scene modeling during the application [7].

Multimodal modeling With respect to classical AR applications, AR in medical context differs in the nature and the size of the data which are available: A large amount of multimodal data are acquired on the patient or possibly on the operating room through sensing technologies or various image acquisitions. The challenge is to analyze these data, to extract interesting features, to fuse and to visualize this information in a proper way. Within the Magrit team, we address several key problems related to medical augmented environments. Being able to acquire multimodal data which are temporally synchronized and spatially registered is the first difficulty we face when considering medical AR. Another key requirement of AR medical systems is the availability of 3D (+t) models of the organ/patient built from images, to be overlaid onto the users's view of the environment.

Methods for multimodal modeling are strongly dependent on the image modalities and the organ specificities. We thus only address a restricted number of medical applications –interventional neuro-radiology and the Augmented Head project– for which we have a strong expertise and close relationships with motivated clinicians. In these applications, our aim is to produce realistic models and then realistic simulations of the patient to be used for surgeon's training or patient's re-education/learning.

One of our main applications is about neuroradiology. For the last 15 years, we have been working in close collaboration with the neuroradiology laboratory (CHU-University Hospital of Nancy) and GE Healthcare. As several imaging modalities are now available in a intraoperative context (2D and 3D angiography, MRI, ...), our aim is to develop a multi-modality framework to help therapeutic decision and treatment.

We have mainly been interested in the effective use of a multimodality framework in the treatment of arteriovenous malformations (AVM) and aneurysms in the context of interventional neuroradiology. The goal of interventional gestures is to guide endoscopic tools towards the pathology with the aim to perform embolization of the AVM or to fill the aneurysmal cavity by placing coils. An accurate definition of the target is a parameter of great importance for the success of the treatment. We have proposed and developed multimodality and augmented reality tools which make various image modalities (2D and 3D angiography, fluoroscopic images, MRI, ...) cooperate in order to help physicians in clinical routine. One of the successes of this collaboration is the implementation of the concept of *augmented fluoroscopy*, which helps the surgeon to guide endoscopic tools towards the pathology. Lately, in cooperation with the Shacra EPI, we have proposed new methods for implicit modeling of the aneurysms with the aim of obtaining near real time simulation of the coil deployment in the aneurysm [4]. Multi-modality techniques for reconstruction are also considered within the european ASPI project, the aim of which is to build a dynamic model of the vocal tract from various images modalities (MRI, ultrasound, video) and magnetic sensors.

MAIA Project-Team

3. Scientific Foundations

3.1. Sequential Decision Making

3.1.1. Synopsis and Research Activities

Sequential decision making consists, in a nutshell, in controlling the actions of an agent facing a problem whose solution requires not one but a whole sequence of decisions. This kind of problem occurs in a multitude of forms. For example, important applications addressed in our work include: Robotics, where the agent is a physical entity moving in the real world; Medicine, where the agent can be an analytic device recommending tests and/or treatments; Computer Security, where the agent can be a virtual attacker trying to identify security holes in a given network; and Business Process Management, where the agent can provide an auto-completion facility helping to decide which steps to include into a new or revised process. Our work on such problems is characterized by three main research trends:

(A) *Understanding how, and to what extent, to best model the problems.*

(B) *Developing algorithms solving the problems and understanding their behavior.*

(C) *Applying our results to complex applications.*

Before we describe some details of our work, it is instructive to understand the basic forms of problems we are addressing. We characterize problems along the following main dimensions:

(1) Extent of the model: full vs. partial vs. none. This dimension concerns how complete we require the model of the problem – if any – to be. If the model is incomplete, then learning techniques are needed along with the decision making process.

(2) Form of the model: factored vs. enumerative. Enumerative models explicitly list all possible world states and the associated actions etc. Factored models can be exponentially more compact, describing states and actions in terms of their behavior with respect to a set of higher-level variables.

(3) World dynamics: deterministic vs. stochastic. This concerns our initial knowledge of the world the agent is acting in, as well as the dynamics of actions: is the outcome known a priori or are several outcomes possible?

(4) Observability: full vs. partial. This concerns our ability to observe what our actions actually do to the world, i.e., to observe properties of the new world state. Obviously, this is an issue only if the world dynamics are stochastic.

These dimensions are wide-spread in the AI literature. We remark that they are not exhaustive. In parts of our work, we also consider the difference between discrete/continuous problems, and centralized/decentralized problems. The complexity of solving the problem – both in theory and in practice – depends crucially on where the problem resides in this categorization. In many applications, not one but several points in the categorization make sense: simplified versions of the problem can be solved much more effectively and thus serve for the generation of *some* – if possibly sub-optimal – action strategy in a more feasible manner. Of course, the application as such may also come in different facets.

In what follows, we outline the main formal frameworks on which our work is based; while doing so, we highlight in a little more detail our core research questions. We then give a brief summary of how our work fits into the global research context.

3.1.2. Formal Frameworks

3.1.2.1. Deterministic Sequential Decision Making

Sequential decision making with deterministic world dynamics is most commonly known as **planning**, or **classical planning** [51]. Obviously, in such a setting every world state needs to be considered at most once, and thus enumerative models do not make sense (the problem description would have the same size as the space of possibilities to be explored). Planning approaches support factored description languages allowing to model complex problems in a compact way. Approaches to automatically learn such factored models do exist, however most works – and also most of our works on this form of sequential decision making – assume that the model is provided by the user of the planning technology. Formally, a problem instance, commonly referred to as a **planning task**, is a four-tuple $\langle V, A, I, G\rangle$. Here, V is a set of variables; a value assignment to the variables is a world state. A is a set of actions described in terms of two formulas over V: their preconditions and effects. I is the initial state, and G is a goal condition (again a formula over V). A solution, commonly referred to as a **plan**, is a schedule of actions that is applicable to I and achieves G.

Planning is **PSPACE**-complete even under strong restrictions on the formulas allowed in the planning task description. Research thus revolves around the development and understanding of search methods, which explore, in a variety of different ways, the space of possible action schedules. A particularly successful approach is **heuristic search**, where search is guided by information obtained in an automatically designed **relaxation** (simplified version) of the task. We investigate the design of relaxations, the connections between such design and the search space topology, and the construction of effective **planning systems** that exhibit good practical performance across a wide range of different inputs. Other important research lines concern the application of ideas successful in planning to stochastic sequential decision making (see next), and the development of technology supporting the user in model design.

3.1.2.2. Stochastic Sequential Decision Making

Markov Decision Processes (**MDP**) [52] are a natural framework for stochastic sequential decision making. An MDP is a four-tuple $\langle S, A, T, r\rangle$, where S is a set of states, A is a set of actions, $T(s, a, s') = P(s'|s, a)$ is the probability of transitioning to s' given that action a was chosen in state s, and $r(s, a, s')$ is the (possibly stochastic) reward obtained from taking action a in state s, and transitioning to state s'. In this framework, one looks for a **strategy**: a precise way for specifying the sequence of actions that induces, on average, an optimal sum of discounted rewards $\mathrm{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$. Here, $(r_0, r_1, ...)$ is the infinitely-long (random) sequence of rewards induced by the strategy, and $\gamma \in (0, 1)$ is a discount factor putting more weight on rewards obtained earlier. Central to the MDP framework is the Bellman equation, which characterizes the **optimal value function** V^*:

$$\forall s \in S, \quad V^*(s) = \max_{a \in A} \sum_{s' \in S} T(s, a, s')[r(s, a, s') + \gamma V^*(s')].$$

Once the optimal value function is computed, it is straightforward to derive an optimal strategy, which is deterministic and memoryless, i.e., a simple mapping from states to actions. Such a strategy is usually called a **policy**. An **optimal policy** is any policy π^* that is **greedy** with respect to V^*, i.e., which satisfies:

$$\forall s \in S, \quad \pi(s) \in \arg\max_{a \in A} \sum_{s' \in S} T(s, a, s')[r(s, a, s') + \gamma V^*(s')].$$

An important extension of MDPs, known as Partially Observable MDPs (**POMDPs**) allows to account for the fact that the state may not be fully available to the decision maker. While the goal is the same as in an MDP (optimizing the expected sum of discounted rewards), the solution is more intricate. Any POMDP can be seen to be equivalent to an MDP defined on the space of probability distributions on states, called **belief states**. The Bellman-machinery then applies to the belief states. The specific structure of the resulting MDP makes it possible to iteratively approximate the optimal value function – which is convex in the **belief space** – by piecewise linear functions, and to deduce an optimal policy that maps belief states to actions. A further

extension, known as a DEC-POMDP, considers $n \geq 2$ agents that need to control the state dynamics in a decentralized way without direct communication.

The MDP model described above is enumerative, and the complexity of computing the optimal value function is **polynomial** in the size of that input. However, in examples of practical size, that complexity is still too high so naïve approaches do not scale. We consider the following situations: (i) when the state space is large, we study approximation techniques from both a theoretical and practical point of view; (ii) when the model is unknown, we study how to learn an optimal policy from samples (this problem is also known as Reinforcement Learning [57]); (iii) in factored models, where MDP models are a strict generalization of classical planning – and are thus at least **PSPACE**-hard to solve – we consider using search heuristics adapted from such (classical) planning.

Solving a POMDP is **PSPACE**-hard even given an enumerative model. In this framework, we are mainly looking for assumptions that could be exploited to reduce the complexity of the problem at hand, for instance when some actions have no effect on the state dynamics (**active sensing**). The decentralized version, DEC-POMDPs, induces a significant increase in complexity (**NEXP**-complete). We tackle the challenging – even for (very) small state spaces – exact computation of finite-horizon optimal solutions through alternative reformulations of the problem. We also aim at proposing advanced heuristics to efficiently address problems with more agents and a longer time horizon.

3.1.3. Project-team positioning

Within Inria, the most closely related teams are TAO and Sequel. TAO works on evolutionary computation (EC) and statistical machine learning (ML), and their combination. Sequel works on ML, with a theoretical focus combining CS and applied maths. The main difference is that TAO and Sequel consider particular algorithmic frameworks that can, amongst others, be applied to Planning and Reinforcement Learning, whereas we revolve around Planning and Reinforcement Learning as the core problems to be tackled, with whichever framework suitable.

In France, we have recently begun collaborating with the IMS Team of Supélec Metz, notably with O. Pietquin and M. Geist who have a great expertise in approximate techniques for MDPs. We have have links with the MAD team of the BIA unit of the INRA at Toulouse, led by R. Sabbadin. They also use MDP related models and are interested in solving large size problems, but they are more driven by applications (mostly agricultural) than we are. In Paris, the Animat Lab, that was a part of the LIP6 and is now attached to the ISIR, has done some interesting works on factored Markov Decision Problems and POMDPs. Like us, their main goal was to tackle problems with large state space.

In Europe, the IDSIA Lab at Lugano (Switzerland) has brought some interesting ideas to the field of MDP (meta-learning, subgoal discovery) but seems now more interested in a *Universal Learner*. In Osnabrück (Germany), the Neuroinformatic group works on efficient reinforcement learning with a specific interest in the application to robotics. For deterministic planning, the most closely related groups are located in Freiburg (Germany), Glasgow (UK), and Barcelona (Spain). We have active collaborations with all of these.

In the rest of the world, the most important groups regarding MDPs can be found at Brown University, Rutgers Univ. (M. Littman), Univ. of Toronto (C. Boutilier), MIT AI Lab (L. Kaelbling, D. Bertsekas, J.Tsitsiklis), Stanford Univ., CMU, Univ. of Alberta (R. Sutton), Univ. of Massachusetts at Amherst (S. Zilberstein, A. Barto), *etc*. A major part of their work is aimed at making Markov Decision Process based tools work on real life problems and, as such, our scientific concerns meet theirs. For deterministic planning, important related groups and collaborators are to be found at NICTA (Canberra, Australia) and at Cornell University (USA).

3.2. Understanding and mastering complex systems

3.2.1. General context

There exist numerous examples of natural and artificial systems where self-organization and emergence occur. Such systems are composed of a set of simple entities interacting in a shared environment and exhibit complex collective behaviors resulting from the interactions of the local (or individual) behaviors of these entities.

The properties that they exhibit, for instance robustness, explain why their study has been growing, both in the academic and the industrial field. They are found in a wide panel of fields such as sociology (opinion dynamics in social networks), ecology (population dynamics), economy (financial markets, consumer behaviors), ethology (swarm intelligence, collective motion), cellular biology (cells/organ), computer networks (ad-hoc or P2P networks), etc.

More precisely, the systems we are interested in are characterized by :

- *locality*: Elementary components have only a partial perception of the system's state, similarly, a component can only modify its surrounding environment.
- *individual simplicity*: components have a simple behavior, in most cases it can be modeled by stimulus/response laws or by look-up tables. One way to estimate this simplicity is to count the number of stimulus/response rules for instance.
- *emergence*: It is generally difficult to predict the global behavior of the system from the local individual behaviors. This difficulty of prediction is often observed empirically and in some cases (e.g., cellular automata) one can show that the prediction of the global properties of a system is an undecidable problem. However, observations coming from simulations of the system may help us to find the regularities that occur in the system's behavior (even in a probabilistic meaning). Our interest is to work on problems where a full mathematical analysis seems out of reach and where it is useful to observe the system with large simulations. In return, it is frequent that the properties observed empirically are then studied on an analytical basis. This approach should allow us to understand more clearly where lies the frontier between simulation and analysis.
- *levels of description and observation*: Describing a complex system involves at least two levels: the micro level that regards how a component behaves, and the macro level associated with the collective behavior. Usually, understanding a complex system requires to link the description of a component behavior with the observation of a collective phenomenon: establishing this link may require various levels, which can be obtained only with a careful analysis of the system.

We now describe the type of models that are studied in our group.

3.2.2. Multi-agent models

To represent these complex systems, we made the choice to use reactive multi-agent systems (RMAS). Multi-agent systems are defined by a set of reactive agents, an environment, a set of interactions between agents and a resulting organization. They are characterized by a decentralized control shared among agents: each agent has an internal state, has access to local observations and influences the system through stimulus response rules. Thus, the collective behavior results from individual simplicity and successive actions and interactions of agents through the environment.

Reactive multi-agent systems present several advantages for modeling complex systems

- agents are explicitly represented in the system and have the properties of local action, interaction and observation;
- each agent can be described regardless of the description of the other agents, multi-agent systems allow explicit heterogeneity among agents which is often at the root of collective emergent phenomena;
- Multi-agent systems can be executed through simulation and provide good model to investigate the complex link between global and local phenomena for which analytic studies are hard to perform.

By proposing two different levels of description, the local level of the agents and the global level of the phenomenon, and several execution models, multi-agent systems constitute an interesting tool to study the link between local and global properties.

Despite of a widespread use of multi-agent systems, their framework still needs many improvements to be fully accessible to computer scientists from various backgrounds. For instance, there is no generic model to mathematically define a reactive multi-agent system and to describe its interactions. This situation is in contrast with the field of cellular automata, for instance, and underlines that a unification of multi-agent systems under a general framework is a question that still remains to be tackled. We now list the different challenges that, in part, contribute to such an objective.

3.2.3. Current challenges

Our work is structured around the following challenges that combine both theoretical and experimental approaches.

3.2.3.1. Providing formal frameworks

Currently, there is no agreement on a formal definition of a multi-agent system. Our research aims at translating the concepts from the field of complex systems into the multi-agent systems framework.

One objective of this research is to remove the potential ambiguities that can appear if one describes a system without explicitly formulating each aspect of the simulation framework. As a benefit, the reproduction of experiments is facilitated. Moreover, this approach is intended to gain a better insight of the self-organization properties of the systems.

Another important question consists in monitoring the evolution of complex systems. Our objective is to provide some quantitative characteristics of the system such as local or global stability, robustness, complexity, etc. Describing our models as dynamical systems leads us to use specific tools of this mathematical theory as well as statistical tools.

3.2.3.2. Controlling complex dynamical system

Since there is no central control of our systems, one question of interest is to know under which conditions it is possible to guarantee a given property when the system is subject to perturbations. We tackle this issue by designing exogenous control architectures where control actions are envisaged as perturbations in the system. As a consequence, we seek to develop control mechanism that can change the global behavior of a system without modifying the agent behavior (and not violating the autonomy property).

3.2.3.3. Designing systems

The aim is to design individual behaviors and interactions in order to produce a desired collective output. This output can be a collective pattern to reproduce in case of simulation of natural systems. In that case, from individual behaviors and interactions we study if (and how) the collective pattern is produced. We also tackle "inverse problems" (decentralized gathering problem, density classification problem, etc.) which consist in finding individual behaviors in order to solve a given problem.

3.2.4. Project-team positioning

Building a reactive multi-agent system consists in defining a set (generally a large number) of simple and reactive agents within a shared environment (physical or virtual) in which they move, act and interact with each other. Our interest in these systems is that, in spite of their simple definition at the agent level, they produce coherent and coordinated behavior at a global scale. The properties that they may exhibit, such as robustness and adaptivity explain why their study has been growing in the last decade (in the broader context of "complex systems").

Our work on such problems is characterized by five research trends: (A) *Defining a formal framework for describing and studying these systems*, (B) *Developing and understanding reactive multi-agent systems*, (C) *Analysing and proving properties*, (D) *Deploying these systems on typical distributed architectures such as swarms of robots, FPGAs, GPUs and sensor networks*, (E) *Transferring our results in applications.*

Multi-agent System is an active area of research in Artificial Intelligence and Complex Systems. Our research fits well into the international research context, and we have made and are making a variety of significant contributions both in theoretical and practical issues. Concerning multi-agent simulation and formalization, we compete or collaborate in France with S. Hassas in LIESP (Lyon), CERV (Brest), IREMIA (la Réunion), Ibisc (Evry), Lirmm (Montpellier), Irit (Toulouse), A. Drogoul (IRD, Bondy) and abroad with F. Zambonelli (Univ. Modena, Italy) A. Deutsch (Dresden, Germany), D. Van Parunak (Vector research, USA), P. Valkenaers, D. Weyns (Univ. Leuven, Belgium), etc. Regarding our work on swarm robotics we have common objectives with the DISAL [3] EPFL Laboratory, the Bristol Robotics Laboratory, the Distributed Robotics Laboratory at MIT, the team of W. & D. Spears at Wyoming university, the Pheromone Robotics project at HRL Lab. [4], the FlockBots project at GMU[5], the team of G. Théraulaz at CNRS-Toulouse and the teams of J.-L. Deneubourg and M. Dorigo at ULB (Bruxelles).

[3] Distributed Intelligent Systems and Algorithms Laboratory including EPFL Swarm-Intelligent Systems Group (SWIS) founded in 2003 and the Collective Robotics Group (CORO) founded in 2000 at California Institute of Technology USA

[4] HRL, Information and systems sciences Lab (ISSL), Malibu CA, USA (D. Payton)

[5] George Mason University, Eclab, USA (L. Panait, S. Luke)

ORPAILLEUR Project-Team

3. Scientific Foundations

3.1. From KDD to KDDK

knowledge discovery in databases, knowledge discovery in databases guided by domain knowledge, data mining methods

Knowledge discovery in databases is a process for extracting knowledge units from large databases, units that can be interpreted and reused within knowledge-based systems.

Knowledge discovery in databases is a process for extracting knowledge units from large databases, units that can be interpreted and reused within knowledge-based systems. From an operational point of view, the KDD process is performed within a KDD system including databases, data mining modules, and interfaces for interactions, e.g. editing and visualization. The KDD process is based on three main operations: selection and preparation of the data, data mining, and finally interpretation of the extracted units. The KDDK process –as implemented in the research work of the Orpailleur team– is based on *data mining methods* that are either symbolic or numerical:

- Symbolic methods are based on frequent itemsets search, association rule extraction, and concept lattice design (Formal Concept Analysis and extensions [95], [108]).
- Numerical methods are based on second-order Hidden Markov Models (HMM2, designed for pattern recognition [107]). Hidden Markov Models have good capabilities for locating stationary segments, and are mainly used for mining temporal and spatial data.

The principle summarizing KDDK can be understood as a process going from complex data units to knowledge units being guided by domain knowledge (KDDK or "knowledge with/for knowledge") [104]. Two original aspects can be underlined: (i) the KDD process is guided by domain knowledge, and (ii) the extracted units are embedded within a knowledge representation formalism to be reused in a knowledge-based system for problem solving purposes.

The various instantiations of the KDDK process in the research work of Orpailleur are mainly based on *classification*, considered as a polymorphic process involved in tasks such as modeling, mining, representing, and reasoning. Accordingly, the KDDK process may feed knowledge-based systems to be used for problem-solving activities in application domains, e.g. agronomy, astronomy, biology, chemistry, and medicine, and also for semantic web activities involving text mining, information retrieval, and ontology engineering [97], [81].

3.2. Methods for Knowledge Discovery guided by Domain Knowledge

knowledge discovery in databases guided by domain knowledge, lattice-based classification, formal concept analysis, frequent itemset search, association rule extraction, second-order Hidden Markov Models, stochastic process, numerical data mining method

knowledge discovery in databases guided by domain knowledge is a KDD process guided by domain knowledge ; the extracted units are represented within a knowledge representation formalism and embedded within a knowledge-based system.

Classification problems can be formalized by means of a class of individuals (or objects), a class of properties (or attributes), and a binary correspondence between the two classes, indicating for each individual-property pair whether the property applies to the individual or not. The properties may be features that are present or absent, or the values of a property that have been transformed into binary variables. Formal Concept Analysis (FCA) relies on the analysis of such binary tables and may be considered as a symbolic data mining technique to be used for extracting (from a binary database) a set of formal concepts organized within a concept lattice [95]. Concept lattices are sometimes also called Galois lattices [82].

The search for frequent itemsets and the extraction of association rules are well-known symbolic data mining methods, related to FCA (actually searching for frequent itemsets may be understood as traversing a concept lattice). Both processes usually produce a large number of items and rules, leading to the associated problems of "mining the sets of extracted items and rules". Some subsets of itemsets, e.g. frequent closed itemsets (FCIs), allow to find interesting subsets of association rules, e.g. informative association rules. This is why several algorithms are needed for mining data depending on specific applications [123], [122].

Among useful patterns extracted from a database, frequent itemsets are usually thought to unfold "regularities" in the data, i.e. they are the witnesses of recurrent phenomena and they are consistent with the expectations of the domain experts. In some situations however, it may be interesting to search for "rare" itemsets, i.e. itemsets that do not occur frequently in the data (contrasting frequent itemsets). These correspond to unexpected phenomena, possibly contradicting beliefs in the domain. In this way, rare itemsets are related to "exceptions" and thus may convey information of high interest for experts in domains such as biology or medicine [124], [125].

From the numerical point of view, a Hidden Markov Model (HMM2) is a stochastic process aimed at extracting and modeling a sequence of stationary distributions of events. These models can be used for data mining purposes, especially for spatial and temporal data as they show good capabilities to locate patterns both in time and space domains. A special research effort focuses on the combination of knowledge elicited by experts and time-space regularities as extracted by an unsupervised classification based on stochastic models [23].

3.3. Elements on Text Mining

knowledge discovery form large collection of texts, text mining, information extraction, document annotation, ontologies

Text mining is a process for extracting knowledge units from large collections of texts, units that can be interpreted and reused within knowledge-based systems.

The objective of a text mining process is to extract new and useful knowledge units in a large set of texts [80], [89]. The text mining process shows specific characteristics due to the fact that texts are complex objects written in natural language. The information in a text is expressed in an informal way, following linguistic rules, making the mining process more complex. To avoid information dispersion, a text mining process has to take into account –as much as possible– paraphrases, ambiguities, specialized vocabulary, and terminology. This is why the preparation of texts for text mining is usually dependent on linguistic resources and methods.

From a KDDK perspective, the text mining process is aimed at extracting new knowledge units from texts with the help of background knowledge encoded within an ontology and which is useful to relate notions present in a text, to guide and to help the text mining process. Text mining is especially useful in the context of semantic web for ontology engineering [86], [85], [84]. In the Orpailleur team, the focus is put on real-world texts in application domains such as astronomy, biology and medicine, using mainly symbolic data mining methods. Accordingly, the text mining process may be involved in a loop used to enrich and to extend linguistic resources. In turn, linguistic and ontological resources can be exploited to guide a "knowledge-based text mining process".

3.4. Elements on Knowledge Systems and Semantic Web

knowledge representation, ontology, description logics, classification-based reasoning, case-based reasoning, semantic web, knowledge-based information retrieval, web mining

Knowledge representation is a process for representing knowledge within an ontology using a knowledge representation formalism, giving knowledge units a syntax and a semantics. Semantic web is based on ontologies and allows search, manipulation, and dissemination of documents on the web by taking into account their contents, i.e. the semantics of the elements included in the documents.

Usually, people try to take advantage of the web by searching for information (navigation, exploration), and by querying documents using search engines (information retrieval). Then people try to analyze the obtained results, a task that may be very difficult and tedious. Semantic web is an attempt for guiding search for information with the help of machines, that are in charge of asking questions, searching for answers, classifying and interpreting the answers. However, a machine may be able to read, understand, and manipulate information on the web, if and only if the knowledge necessary for achieving those tasks is available. This is why ontologies are of main importance with respect to the task of setting up semantic web. Thus, there is a need for representation languages for annotating documents, i.e. describing the content of documents, and giving a semantics to this content. Knowledge representation languages are (the?) good candidates for achieving the task: they have a syntax with an associated semantics, and they can be used for retrieving information, answering queries, and reasoning.

Semantic web constitutes a good platform for experimenting ideas on knowledge representation, reasoning, and KDDK. In particular, the knowledge representation language used for designing ontologies is the OWL language, which is based on description logics (or DL [79]). In OWL, knowledge units are represented within concepts (or classes), with attributes (properties of concepts, or relations, or roles), and individuals. The hierarchical organization of concepts (and relations) relies on a subsumption relation that is a partial ordering. The inference services are based on subsumption, concept and individual classification, two tasks related to "classification-based reasoning". Furthermore, classification-based reasoning can be associated to case-based reasoning (CBR), that relies on three main operations: retrieval, adaptation, and memorization. Given a target problem, retrieval consists in searching for a source (memorized) problem similar to the target problem. Then, the solution of the source problem is adapted to fulfill the constraints attached to the target problem, and possibly memorized for further reuse.

In the trend of semantic web, research work is also carried on semantic wikis which are wikis i.e., web sites for collaborative editing, in which documents can be annotated thanks to semantic annotations and typed relations between wiki pages. Such links provide kind of primitive knowledge units that can be used for guiding information retrieval or knowledge discovery.

PAROLE Project-Team

3. Scientific Foundations

3.1. Introduction

Research in speech processing gave rise to two kinds of approaches:

- research that aims at explaining how speech is produced and perceived, and that therefore includes physiological aspects (vocal tract control), physical (speech acoustics), psychoacoustics (peripheral auditory system), and cognitive aspects (building sentences),
- research aiming at modeling the observation of speech phenomena (spectral analysis, stochastic acoustic or linguistic models).

The former research topic is motivated by the high specificity of speech among other acoustical signals: the speech production system is easily accessible and measurable (at least at first approach); acoustical equations are reasonably difficult from a mathematical point of view (with simplifications that are moderately restrictive); sentences built by speakers are governed by vocabulary and grammar of the considered language. This led acousticians to develop research aiming at generating artificial speech signals of good quality, and phoneticians to develop research aiming at finding out the origin of speech sound variability and at explaining how articulators are utilized, how sounds of a language are structured and how they influence each other in continuous speech. Lastly, that led linguists to study how sentences are built. Clearly, this approach gives rise to a number of exchanges between theory and experimentation and it turns out that all these aspects of speech cannot be mastered easily at the same time.

Results available on speech production and perception do not enable using an analysis by synthesis approach for automatic speech recognition. Automatic speech recognition thus gives rise to a second approach that consists in modeling observations of speech production and perception. Efforts focused onto the design of numerical models (first simple vectors of spectral shapes and now stochastic or neural models) of word or phoneme acoustical realizations, and onto the development of statistical language models.

These two approaches are complementary; the latter borrows theoretical results on speech from the former, which, in its turn, borrows some numerical methods. Spectral analysis methods are undoubtedly the domain where exchanges are most marked. The simultaneous existence of these two approaches is one of the particularities of speech research conducted in Nancy and we intend to enhance exchanges between them. These exchanges will probably grow in number because of new applications like: **(i)** computer aided foreign language learning which requires both reliable automatic speech recognition and fine acoustic and articulatory speech analysis, **(ii)** automatic recognition of spontaneous speech which requires robustness against noise and speaker variability.

3.2. Speech Analysis and Synthesis

Our research activities focus on acoustical and perceptual cues of speech sounds, speech modifications and acoustic-to-articulatory inversion. Our main applications concern the improvement of the oral component of language learning, speech synthesis and esophageal voices.

3.2.1. Oral comprehension

We developed tools to improve speech perception and production, and made perceptual experiments to prove their efficiency in language learning. These tools are also of interest for hearing impaired people, as well as for normally hearing people in noisy environments and also for children who learn to read (children who have language disabilities without cognitive deficit or hearing impairment and "normal" children).

3.2.1.1. Computer-assisted learning of prosody

We are studying automatic detection and correction of prosodic deviations made by a learner of a foreign language. This work implies three different tasks: (a) the detection of the prosodic entities of the learner's realization (lexical accent, intonative patterns), (b) the evaluation of the deviations, by comparison with a model, and (c) their corrections, both verbal and acoustic. This last kind of feedback is directly done on the learner's realization: the deviant prosodic cues are replaced by the prosodic cues of the reference. The identification and correction tasks use speech analysis and modification tools developed in our team.

Within the framework of a new project (see 6.1.6.2), we also investigate the impact of a language intonational characteristics on the perception and production of the intonation of a foreign language.

3.2.1.2. Phonemic discrimination in language acquisition and language disabilities

We keep working on a project concerning identification of early predictors of reading, reading acquisition and language difficulties, more precisely in the field of specific developmental disabilities : dyslexia and dysphasia. A fair proportion of those children show a weakness in phonological skills, particularly in phonemic discrimination. However, the precise nature and the origin of the phonological deficits remain unspecified. In the field of dyslexia and normal acquisition of reading, our first goal was to contribute to identify early indicators of the future reading level of children.We based our work on the longitudinal study - with 85 French children - of [45], [46] which indicates that phonemic discrimination at the beginning of kindergarten is strongly linked to success and specific failure in reading acquisition. We study now the link between oral discrimination both with oral comprehension and written compréhension. Our analyses are based on the follow up of a hundred chrildren for 4 years from kindergarten to end of grade 2 (from age 4 to age 8). Publications in progress.

3.2.1.3. Esophageal voices

It is possible for laryngectomees to learn a substitution voice: the esophageal voice. This voice is far from being natural. It is characterized by a weak intensity, a background noise that bothers listening, and a low pitch frequency. A device that would convert an esophageal voice to a natural voice would be very useful for laryngectomees because it would be possible for them to communicate more easily. Such natural voice restitution techniques would ideally be implemented in a portable device.

3.2.2. Acoustic-to-articulatory inversion

Acoustic-to-articulatory inversion aims at recovering the articulatory dynamics from speech signal that may be supplemented by images of the speaker face. Potential applications concern low bit rate speech coding, automatic speech recognition, speech production disorders assessment, articulatory investigations of phonetics, talking heads and articulatory feedback for language acquisition or learning.

Works on acoustic-to-articulatory inversion widely rely on an analysis by synthesis approach that covers three essential aspects:

Solving acoustic equations. In order to solve the acoustic equations adapted to the vocal tract, one assumes that the sound wave is a plane wave in the vocal tract and that it can be unbend. There are two families of solving methods:

(i) frequency methods through the acoustical-electrical analogy,

(ii) spatio-temporal methods, through the direct solving of finite difference equations derived from Webster equations.

Measuring the vocal tract. This represents an important obstacle because there does not exist any reliable method enabling a precise measurement in time and dimension. MRI (Magnetic Resonance Imaging) enables 3D measurements but is not sufficiently fast and X-rays only allows a sagittal slice of the vocal tract to be captured while involving not acceptable health hazards.

Articulatory modeling. Articulatory models aim at describing all the possible vocal tract shapes with a small number of parameters, while preserving deformations observed on a real vocal tract. Present articulatory models often derive from data analysis of cineradiography moving pictures. One of the most widely used is the one built by Maeda [52].

One of the major difficulties of inversion is that an infinity of vocal tract shapes can give rise to the same speech spectrum. Acoustic-to-articulatory inversion methods are categorized into two families:

- methods that optimize a function generally combining speaker's articulatory effort and acoustical distance between natural and synthesized speech. They exploit constraints allowing the number of possible vocal tract shapes to be reduced.
- table look-up methods resting on an articulatory codebook of articulatory shapes indexed by their acoustical parameters (generally formant frequencies). After possible shapes have been recovered at each time, an optimization procedure is used to find an inverse solution in the form of an optimal articulatory path.

As our contribution only concerns inversion, we accepted widely used articulatory synthesis methods. We therefore chose Maeda's articulatory model, the acoustical-electrical analogy to compute the speech spectrum and the spatio-temporal method proposed by Maeda to generate the speech signal. As regards inversion, we chose Maeda's model to constrain vocal tract shapes because this model guarantees that synergy and compensation articulatory phenomena are still possible, and consequently, that articulatory deformations close to those of a human speaker may be recovered. The most important challenges in this domain are the inversion of any class of speech sounds and to perform inversion from standard spectral data, MFCC for instance. Indeed at present, only vowels and sequences of vowels can be inverted, and only some attempts concern fricatives sounds. Moreover, most of the inversion techniques use formant frequencies as input data although formants cannot be extracted from speech easily and reliably.

3.2.3. Strategies of labial coarticulation

The investigation of labial coarticulations strategies is a crucial objective with the view of developing a talking head which would be understandable by lip readers, especially deaf persons.

In the long term, our goal is to determine a method of prediction of labial coarticulation adaptable to a virtual speaker. Predicting labial coarticulation is a difficult problem that gave rise to many studies and models. To predict the anticipatory coarticulation gestures (see [42] for an overall presentation of labial coarticulation), three main models have been proposed: the look-ahead model, the time-locked model and the hybrid model.

These models were often compared on their performance in the case of the prediction of anticipation protrusion in VCV or VCCV sequences where the first vowel is unrounded, the consonant(s) is neutral with respect to labial articulation and the last vowel is rounded. There is no general agreement about the efficiency of these models. More recent models have been developed. The one of Abry and Lallouache [37] advocates for the theory of expansion movements: the movement tends to be anticipated when no phonological constraint is imposed on labiality. Cohen and Massaro [40] proposed dominance functions that require a substantial numerical training.

Most of these models derive from the observations of a limited number of speakers. We are thus developing a more explicative model, i.e., essentially a phonetically based approach that tries to understand how speakers manage to control labial parameters from the sequence of phonemes to be articulated.

3.2.4. Speech Synthesis

Data-driven speech synthesis is widely adopted to develop Text-to-Speech (TTS) synthesis systems. Basically, it consists of concatenating pieces of signal (units) selected from a pre-recorded sentence corpus. Our ongoing work on acoustic TTS was recently extended to study acoustic-visual speech synthesis (bimodal units).

3.2.4.1. Text-to-speech synthesis

Data-driven text-to-speech synthesis is usually composed of three steps to transform a text in speech signal. The first step is Natural Language Processing (NLP) which tags and analyzes the input text to obtain a set of features (phoneme sequence, word grammar categories, syllables...). It ends with a prosodic model which transforms these features into acoustic or symbolic features (F0, intensity, tones...). The second step uses a Viterbi algorithm to select units from a corpus recorded beforehand, which have the closest features to the prosodic features expected. The last step amounts to concatenate these units.

Such systems usually generate a speech signal with a high intelligibility and a naturalness far better than that achieved by old systems. However, building such a system is not an easy task [39] and the global quality mainly relies on the quality of the corpus and prosodic model. The prosodic model generally provides a good standard prosody, but, the generated speech can suffer from a lack of variability. Especially during the synthesis of extended passages, repetition of similar prosodic patterns can lead to a monotonous effect. Therefore, to avoid this problem due to the projection of linguistic features onto symbolic or acoustic dimensions (during NLP), we [41] proposed to perform the unit selection directly from linguistic features without incorporating any prosodic information. To compensate the lack of prosodic prediction, the selection needs to be performed with numerous linguistic features. The selection is no longer restrained by a prosodic model but only driven by weighted features. The consequence is that the quality of synthesis may drop in crucial instants. Our works deal to overcome this new problem while keeping advantage of the lack of prosodic model.

These works have an impact on the construction of corpus and on the NLP engine which needs to provide as much information as possible to the selection step. For instance, we introduced a chunker (shallow parser) to give us information on a potential rhythmic structure. Moreover, to perform the selection, we developed an algorithm to automatically weight the linguistic features given by the NLP. Our method relies on acoustic clustering and entropy information [41]. The originality of our approach leads us to design a more flexible unit selection step, constrained but not restrained.

3.2.4.2. Acoustic-visual speech synthesis

Audiovisual speech synthesis can be achieved using 3D features of the human face supervised by a model of speech articulation and face animation. Coarticulation is approximated by numerical models that describe the synergy of the different articulators. Acoustic signal is usually synthetic or natural speech synchronized with the animation of the face. Some of the audiovisual speech systems are inspired by recent development in speech synthesis based on samples and concatenative techniques. The main idea is to concatenate segments of recorded speech data to produce new segments. Data can be video or motion capture. The main drawback of these methods is that they focus on one field, either acoustic or visual. But (acoustic) speech is actually generated by moving articulators, which modify the speaker's face. Thus, it is natural to find out that acoustic and face movements are correlated. A key point is therefore to guarantee the internal consistency of the acoustic-visual signal so that the redundancy of these two signals acknowledged as a determining perceptive factor, can really be exploited by listeners. It is thus important to deal with the two signals (acoustic and visual) simultaneously and to keep this link during the whole process. This is why we make the distinction between audiovisual speech synthesis (where acoustic is simply synchronized with animation) and acoustic-visual speech where speech is considered as a bimodal signal (acoustic and visual) as considered in our work. Our long-term goal is to contribute to the fields of acoustic speech synthesis and audiovisual speech synthesis by building a bimodal corpus and developing an acoustic-visual speech synthesis system using bimodal unit concatenation.

3.3. Automatic speech recognition

Automatic speech recognition aims at reproducing the cognitive ability of humans to recognize and understand oral speech. Our team has been working on automatic speech recognition for decades. We began with knowledge-based recognition systems and progressively made our research works evolve towards stochastic approaches, both for acoustic and language models. Regarding acoustic models, we have especially investigated HMM (Hidden Markov Models), STM (Stochastic Trajectory Models), multi-band approach and BN (Bayesian Networks). Regarding language models, our main interest has concerned ngram approaches (word classes, trigger, impossible ngram, etc).

The main challenge of automatic speech recognition is its robustness to multiple sources of speech variability [44]. Among them, we have been focusing on acoustic environment, inter- and intra-speaker variability, different speaking styles (prepared speech, spontaneous, etc.) and non-native pronunciations.

Another specifity of automatic speech recognition is the necessity to combine efficiently all the research works (in acoustic modeling, langage modeling, speaker adaptation, etc.) into a core platform in order to evaluate them, and to go beyond pure textual transcriptions by enriching them with punctuation, syntax, etc., in order to make them exploitable by both humans and machines.

3.3.1. Acoustic features and models

The raw acoustic signal needs to be parameterized to extract the speech information it contains and to reduce its dimensionality. Most of our research and recognition technologies make use of the classical Mel Feature Cepstral Coefficients, which have proven since many years to be amongst the most efficient front-end for speech recognition. However, we have also explored alternative parameterizations to support some of our recent research progresses. For example, prosodic features such as intonation curves and vocal energy give important cues to recognize dialog acts, and more generally to compute information that relates to supra-phonemic (linguistic, dialog, ...) characteristics of speech. Prosodic features are developed jointly for both the Speech Analysis and Speech Recognition topics. We also developed a new robust front-end, which is based on wavelet-decomposition of the speech signal.

Concerning acoustic models, stochastic models are now the most popular approach for automatic speech recognition. Our research on speech recognition also largely exploits Hidden Markov Models (HMM). In fact, HMMs are mainly used to model the acoustic units to be recognized (usually triphones) in all of our recognition engines (ESPERE, ANTS...). Besides,we have investigated Bayesian Networks (BN) to explicitly represent random variables and their independence relationships to improve noise robustness.

3.3.2. Robustness and invariance

Part of our research activities about ASR aims at improving the robustness of recognizers to the different sources of variability that affect the speech signal and damage the recognition. Indeed, the issue of the lack of robustness of state-of-the-art ASR systems is certainly the most problematic one that still prevents the wide deployment of speech recognizers nowadays. In the past, we developed a large range of techniques to address this difficult topic, including robust acoustic models (such as stochastic trajectory and multi-band models) and model adaptation techniques (such as missing data theory). The following state-of-the-art approaches thus form our baseline set of technologies: MLLR (Maximum Likelihood Linear Regression), MAP (Maximum A Posteriori), PMC (Parallel Model Combination), CMN (Cepstral Mean Normalization), SAT (Speaker Adaptive Training), HLDA (Heteroscedastic Linear Discriminant Analysis), Spectral Subtraction and Jacobian Adaptation.

These technologies constitute the foundations of our recent developments in this area, such as non-native speaker adaptation, out-of-vocabulary words detection and adaptation to pronunciation variations. Handling speech variabilities may also benefit from exploiting additional external or contextual sources of information to more tightly guide the speech decoding process. This is typically the role of the language model, which shall in this context be augmented with higher-level knowledge, such as syntactic or semantic cues. Yet, automatically extracting such advanced features is very challenging, especially on imperfect transcribed speech.

The performance of automatic speech recognition (ASR) systems drastically drops when confronted with non-native speech. If we want to build an ASR system that takes into account non-native speech, we need to modify the system because, usually, ASR systems are trained on standard phone pronunciations and designed to recognize only native speech. In this way, three method categories can be applied: acoustic model transformation, pronunciation modeling and language modeling. Our contribution concerns the first two methods.

3.3.3. Segmentation

Audio indexing and automatic broadcast news transcription need the segmentation of the audio signal. The segmentation task consists in two steps: firstly, homogeneous segments are extracted and classified into speech, noise or music, secondly, speakers turns are detected in the extracted speech segments.

Speech/music segmentation requires to extract discriminant acoustic parameters. Our contribution concerns the MFCC and wavelet parameters. Another point is to find a good classifier. Various classifiers are commonly used: k-Nearest-Neighbors, Hidden Markov Models, Gaussian Mixture Models, Artificial Neural Networks.

As to detect speaker turns, the main approach consists of splitting the audio signal into segments that are assumed to contain only one speaker and then a hierarchical clustering scheme is performed for merging segments belonging to the same speaker.

3.3.4. Speech/text alignment

Speech/text alignment consists in finding time boundaries of words or phones in the audio signal knowing the orthographic transcription. The main applications of speech/text alignment are training of acoustic models, segmentation of audio corpus for building units for speech synthesis or segmentation of the sentence uttered by a learner of a foreign language. Moreover, speech/text alignement is a useful tool for linguistic researchers.

Speech/text alignment requires two steps. The first step generates the potential pronunciations of the sentence dealing with multiple pronunciations of proper nouns, liaisons, phone deletions, and assimilations. For that, the phonetizer is based on a phonetic lexicon, and either phonological rules or an automatic classifier as a decision tree. The second step finds the best pronunciation corresponding to the audio signal using acoustic HMM models and an alignment algorithm. The speech team has been working on this domain for a long time.

3.4. Speech to Speech Translation and Langage Modeling

Speech-to-Speech Translation aims at translating a source speech signal into a target speech signal. A sequential way to adress this problem is to first translate a text to another one. And after, we can connect a speech recognition system at the input and a text to speech synthesis system at the output. Several ways to adress this issue exist. The concept used in our group is to let the computer learning from a parallel text all the associations between source and target units. A unit could be a word or a phrase. In the early 1990s [38] proposes five statistical translation models which became inescapable in our community. The basic idea of the model 1 is to consider that any word of the target language could be a potential translation of any source word. The problem is then to estimate the distribution probability of a target word given a source one. The translation problem is similar to the speech recognition one. Indeed, we have to seek the best foreign sentence given a source one. This one is obtained by decoding a lattice translation in which a language and translation models are used. Several issues have to be supported in machine translation as described below.

3.4.1. Word translation

The first translation systems identify one-to-one associations between words of target and source languages. This is still necessary in the present machine translation systems. In our group we develop a new concept to learn the translation table. This approach is based on computing all the inter-lingual triggers inside a parallel corpus. This leads to a pertinent translation table [51]. Obviously, this is not sufficient in order to make a realistic translation because, with this approach, one word is always translated into one word. In fact, it is possible to express the same idea in two languages by using different numbers of words. Thus, a more general one-to-one alignement has to be achieved.

3.4.2. Phrase translation

The human translation is a very complex process which is not only word-based. A number of research groups developed phrase-based systems which are different from the baseline IBM's model in training. These methods, deals with linguistic units which consists in more than one word. The model supporting phrase-based machine translation uses reordering concept and additional feature functions. In order to retrieve phrases, several approaches have been proposed in the litterature. Most of them require word-based alignments. For example, Och and al. [54] collected all phrase pairs that were consistent with the word alignment provided by Brown's models.

We developed a phrase based algorithm which is based on finding first an adequate list of phrases. Then, we find out the best corresponding translations by using our concept of inter-lingual triggers. A list of the best translations of sequences is then selected by using simulated annealing algorithm.

3.4.3. Language model

A language model has an important role in a statistical machine translation. It ensures that the translated words constitute a valid linguistic sentence. Most of the community uses n-grams models, that is what we do also.

3.4.4. Decoding

The translation issue is treated as an optimization problem. Translating a sentence from English into a Foreign language involves finding the best Foreign target sentence f^* which maximizes the probability of f given the English source sentence e. The Bayes rule allows to formulate the probability $P(f|e)$ as follows:

$$f^* = \arg\max_f P(f|e) = \arg\max_f P(e|f)P(f)$$

The international community uses either PHARAOH [48] or MOSES [47] based on a beam search algorithm. In our group we started decoding by PHARAOH but we moved recently to MOSES.

SEMAGRAMME Team

3. Scientific Foundations

3.1. Fondation

The present proposal relies on deep mathematical foundations. We intend to develop models based on well-established mathematics. We seek two main advantages from this approach. On the one hand, by relying on mature theories, we have at our disposal sets of mathematical tools that we can use to study our models. On the other hand, developing various models on a common mathematical background will make them easier to integrate, and will ease the search for unifying principles.

The main mathematical domains on which we rely are formal language theory, symbolic logic, and type theory.

3.1.1. Formal language theory

studies the purely syntactic and combinatorial aspects of languages, seen as sets of strings (or possibly trees or graphs). Formal language theory has been especially fruitful for the development of parsing algorithms for context-free languages. We use it, in a similar way, to develop parsing algorithms for formalisms that go beyond context-freeness. Language theory also appears to be very useful in formally studying the expressive power and the complexity of the models we develop.

3.1.2. Symbolic logic

(and, more particularly, proof-theory) is concerned with the study of the expressive and deductive power of formal systems. In a rule-based approach to computational linguistics, the use of symbolic logic is ubiquitous. As we previously said, at the level of syntax, several kinds of grammars (generative, categorial...) may be seen as basic deductive systems. At the level of semantics, the meaning of an utterance is capture by computing (intermediate) semantic representations that are expressed as logical forms. Finally, using symbolic logics allows one to formalize notions of inference and entailment that are needed at the level of pragmatics.

3.1.3. Type theory and typed λ-calculus

Among the various possible logics that may be used, Church's simply typed λ-calculus and simple theory of types (a.k.a. higher-order logic) play a central part. On the one hand, Montague semantics is based on the simply typed λ-calculus, and so is our syntax-semantics interface model. On the other hand, as shown by Gallin, [28] the target logic used by Montague for expressing meanings (i.e., his intensional logic) is essentially a variant of higher-order logic featuring three atomic types (the third atomic type standing for the set of possible worlds).