A - Short description of the project

Distributed Open and Timed Systems - DOTS

- Acronym: DOTS
- Topics: safety of the computerized systems
- Type: research project – 4 years

A-1 Context and motivation

The scientific context of the DOTS project is specification, verification and design of information systems. The research domain we have in mind started about 25 years ago with seminal papers by Clarke, Pnueli and Sifakis. Since then the domain has witnessed an impressive growth: a comprehensive theory has been developed, efficient algorithms have been designed, and tools like model checkers have been developed. These tools allow to verify automatically that a model of a system satisfies its specification. The research results have also penetrated the industry world as model checkers are now used in an industrial context for numerous case studies which in turn provided some feedback to improve the theory and algorithms.

Complex systems, such as embedded systems that are widely used nowadays (telecommunication, transport, automation), are often *distributed* – composed of several components that communicate together –, *timed* – contain timing constraints –, and *open* – interact with their environment. Each of these aspects considered separately is now relatively well understood and corresponds to an active research area. The big challenge is to deal with systems which present several of these features.

The aim of the DOTS project is to associate researchers specialized in verification of different aspects mentioned above in order to tackle problems that emerge when considering several features simultaneously. In this way we plan to significantly advance both theory as well as algorithmics of design and verification of distributed, open and timed systems.

The area of formal verification covers now a wide range of problems that share a common theoretical basis, but require specific approaches and techniques. In addition to *model checking* – the classical problem that consists in deciding whether a given model satisfies a given specification – the DOTS project will mainly address two important verification problems: *control* and *non-interference.*

An important characteristic of the DOTS project is our choice of methods and tools to address the problems mentioned above. We plan to use *games* to cope with interactive aspects and *partial orders* to deal with the distributed aspect.

A-2 Expected scientific results

The DOTS project is organized in several workpackages corresponding to possible combinations of timed/open/distributed aspects. The last one deals with the design of systems exhibiting the three characteristics. For each of this four workpackages, the principal scientific results we expect are summarized below.

- Timed open systems:
  - Definition of *pertinent classes* of timed games for the design of timed open systems.
  - Design of algorithms for the synthesis of timed controllers for *quantitative* control objectives.
  - Synthesis of *implementable* controllers, i.e. that can be executed with some bounded imprecision.
  - Definition of suitable *non-interference conditions* for timed systems.
  - Synthesis of timed *non interferent controlled* systems.
- Distributed open systems:
  - Development of the theory of *distributed game* models.
  - Study of the distributed control problem for input/output and *robust* specifications.
  - Utilization of *causal memory* in synchronous systems to solve distributed games.
  - Study of the control problem in *asynchronous* communication models.
  - Definition of a realistic notion of *non interference for distributed* systems and characterization in term of games.
  - *Quantification of severity* of information leaks.
- Timed distributed systems:
  - Design of a *concurrent* semantics for distributed timed systems.
  - Design of *efficient algorithms* for analyzing distributed timed systems.
  - *Implementation* of our results in a prototype tool and validation of the approach on real case-studies.
- Distributed Open Timed systems:
  - Application of techniques elaborated in previous WPs to two examples issued from diagnostic of telecom protocols and control of embedded systems.
  - Proposal of an integrated DOTS method.

A-3 Expected industrial results

The DOTS project is clearly based on an exploratory scientific program. Hence we do not have the ambition to develop tools directly exploitable by the industry. Nevertheless, formal verification is nowadays a requirement in numerous industrial domains and the demand of the industry is to enlarge the class of systems that can be automatically verified. We believe that the results of the DOTS project, and in particular the treatment of realistic examples planned in the last workpackage, will contribute to our collective ability to handle distributed, open and timed industrial systems.

A - Description courte du projet

Distributed Open and Timed Systems - DOTS

- Acronyme : DOTS
- Champs thématiques : sûreté des systèmes informatisés
- Type : projet de recherche – 4 ans

A-1 Contexte et motivation du projet

Le projet DOTS porte sur la spécification formelle et la vérification automatique des systèmes informatiques. Plus particulièrement il s'intéresse aux systèmes complexes, comme les fameux systèmes embarqués, qui intègrent des aspects *temporisés* (des contraintes temps-réel), des aspects *distribués* (ces systèmes contiennent plusieurs composants qui communiquent entre eux) et des aspects *interactifs* avec leur environnement (on parle alors de systèmes *ouverts*). Si chacun de ces aspects pris séparément est aujourd'hui bien connu, la vérification de systèmes combinant plusieurs de ces caractéristiques reste un problème largement ouvert.

L'objectif du projet DOTS est de permettre des avancées théoriques et algorithmiques (validées par des prototypes) pour la conception de systèmes temporisés, distribués et ouverts. Une caractéristique essentielle du projet est de rassembler des spécialistes de chacun des trois aspects mentionnés ci-dessus pour confronter les techniques propres à chaque domaine et aborder conjointement la vérification des systèmes complexes.

Nous nous intéresserons principalement au *model checking* (est-ce qu'un système vérifie une propriété donnée ?), mais aussi au problème du *contrôle* (comment superviser un système pour qu'il vérifie une propriété de correction donnée ?) et au problème de la *non-interférence* (comment s'assurer qu'un système ne communique pas certaines informations sensibles à un observateur extérieur ?).

Pour attaquer ces problèmes, nous travaillerons autour de deux approches particulières. D'une part, nous utiliserons le cadre des jeux pour modéliser et analyser les interactions des systèmes. Et d'autre part, nous utiliserons les méthodes à base d'ordres partiels pour appréhender les aspects distribués. Combiner ces techniques en y intégrant aussi des contraintes temps-réel est une perspective novatrice et constitue une spécificité importante de notre projet.

A-2 Retombées scientifiques et techniques attendues

Le projet DOTS est organisé en plusieurs sous-projets, chacun correspond à une combinaison possible des trois dimensions (temps, distribution, interaction) mentionnées ci-dessus. Nous pouvons décrire les résultats attendus de la manière suivante :

- Systèmes temporisés et ouverts (SP1) :
  - Définition de jeux temporisés adaptés à la vérification de systèmes temporisés et ouverts.
  - Algorithmes pour la synthèse de contrôleurs temporisés pour des objectifs de contrôle *quantitatifs.*

  - Algorithmes pour synthétiser des contrôleurs *implémentables*, c'est-à-dire pouvant supporter de légères perturbations dues à des imprécisions de la plateforme d'exécution.
  - Définition d'une notion pertinente de la non-interférence pour des systèmes temporisés.
  - Algorithmes pour le contrôle de systèmes non-interférents.

- Systèmes distribués et ouverts (SP2) :
  - Contribution à la théorie des jeux distribués.
  - Etude du problème du contrôle pour des spécifications particulières en fonction de la structure du système (par ex. restreintes à certaines données ou interdisant d'énoncer certaines propriétés).
  - Usage de la mémoire *causale* pour la résolution des jeux distribués.
  - Définition d'une bonne notion de non-interférence distribuée et sa caractérisation en terme de jeux.
  - Intégration d'aspects quantitatifs sur la perte d'informations pour la non-interférence.

- Systèmes distribués et temporisés (SP3) :
  - Définition d'une sémantique *concurrente* pour les systèmes distribués temporisés.
  - Conception d'algorithmes efficaces pour l'analyse des systèmes distribués temporisés.
  - Réalisation d'un prototype des algorithmes précédents et test de leur efficacité sur de vrais exemples.

- Systèmes distribués, temporisés et ouverts (SP4) :
  - Application (et combinaison) des algorithmes et méthodes mis au point dans les sous-projets 1, 2 et 3 sur deux exemples importants. On visera d'une part un problème de diagnostic pour un protocole de telecom, et d'autre part un problème de contrôle d'un système embarqué.
  - Proposition d'une méthode de conception (modélisation/vérification) de systèmes distribués, temporisés et ouverts.

A-3 Retombées industrielles et économiques escomptées

Le projet DOTS est un projet de recherche exploratoire. Nous n'avons donc pas l'ambition de développer des outils directement exploitables dans l'industrie. Néanmoins, la vérification formelle est aujourd'hui une exigence dans de nombreux domaines économiques et industriels, qui ont donc une forte attente de résultats permettant d'enrichir les classes de systèmes automatiquement vérifiables. Nous sommes convaincus que les résultats du projet DOTS, et en particulier le traitement des exemples réalistes du dernier WorkPackage, contribueront à des avancées significatives dans notre capacité collective à traiter de vrais systèmes industriels interactifs, ouverts et temps-réel.

B - Scientific description of the project

Distributed Open and Timed Systems - DOTS

B-1 Scientific objectives

The scientific context of the DOTS project is specification, verification and design of information systems. The research domain we have in mind started about 25 years ago with seminal papers by Clarke, Pnueli and Sifakis. Since then the domain has witnessed an impressive growth: a comprehensive theory has been developed, efficient algorithms have been designed, and tools like model checkers have been developed. These tools allow to verify automatically that a model of a system satisfies its specification. The research results have also penetrated the industry world as model checkers are now used in an industrial context for numerous case studies which in turn provided some feedback to improve the theory and algorithms.

Complex systems, such as embedded systems that are widely used nowadays (telecommunication, transport, automation), are often *distributed* – composed of several components that communicate together –, *timed* – contain timing constraints –, and *open* – interact with their environment. Each of these aspects considered separately is now relatively well understood and corresponds to an active research area. The big challenge is to deal with systems which present several of these features.

The aim of the DOTS project is to associate researchers specialized in verification of different aspects mentioned above in order to tackle problems that emerge when considering several features simultaneously. In this way we plan to significantly advance both theory as well as algorithmics of design and verification of distributed, open and timed systems. Figure 1 describes interactions between the three main aspects and points to the corresponding sections in part B3 where a more detailed description is presented.

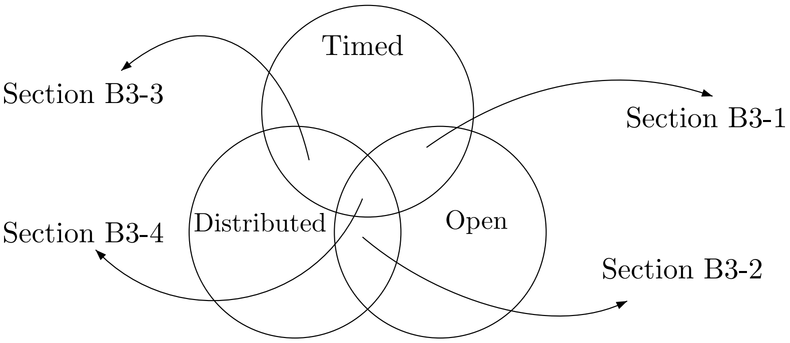

Figure 1: Overview of the DOTS project

The area of formal verification covers now a wide range of problems that share a common theoretical basis, but require specific approaches and techniques. In addition to *model checking* – the classical problem that consists in deciding whether a given model satisfies a given specification – the DOTS project will mainly address two important verification problems: *control* and *non-interference.*

Control: The control problem is to construct, when it is possible, a controller that can supervise a given system in order to satisfy a given specification. Thus the controlled system obtained this way is correct by construction. As explained further more in details, we will focus on the two, complementary, questions: *controllability* and *synthesis.* The first problem is to decide if a controller exists for a given system. The second demands to construct a controller, and in particular deals with the issue of its *implementability.* For both questions we will concentrate on *quantitative control objectives* such as: latency, service time, etc.

Non-interference: The problem addresses secrecy issues in information systems. These issues are modelled by the requirement that modifications made by high-level users should not be visible to users of lower level. This notion of non-interference has been widely studied for the last twenty years in the security community. More recently, several results have shown that model-checking approaches could apply to this category of problems. We propose to study non-interference problems for timed and distributed systems, formulate them as game problems and address the question of synthesis of non interferent systems.

An important characteristic of the DOTS project is our choice of methods and tools to address the problems mentioned above. We plan to use *games* to cope with interactive aspects and *partial orders* to deal with the distributed aspect.

- Games: Game playing is a powerful metaphor that fits many situations in which interaction between agents plays a central role. Various tasks in computer science such as design, synthesis, validation, testing, query evaluation, planning, etc. can be formulated in game-theoretic terms. Viewing them abstractly as games can often help to reveal the algorithmic essence of the problems, and clarify the relationships between problem domains. For example, two-player turn-based games are a natural framework to describe a system in interaction with an environment. In this framework, a winning strategy is a way to guarantee the required system behavior: constructing a strategy corresponds to synthesizing a reactive program that meets the specification. We plan to adapt this approach to distributed and timed settings.

- Partial orders: Parallel composition is known to be the main source of problems in verification. The approach via partial order semantics offers a way to alleviate the problem by proposing a more faithful way of modeling the parallel composition. This kind of semantics is well-known in the untimed setting both on the level of specifications as well as that of verification. For example, Message Sequence Charts are a very convenient way to describe scenarios of distributed systems. One can also mention verification techniques such as partial order reductions implemented in the model checker SPIN [26]. At present, it is not known how to apply these techniques to timed systems. For example, there is no efficient symbolic data-structure which handles both timing constraints and parallel composition. We plan to develop partial-order based methods to deal with timed and distributed systems.

Each of these two promising techniques has already proved its relevance. Bringing together specialists in both of them is a crucial aspect of our project.

B-2 Context

The teams involved in the DOTS project are widely recognized for their contributions in the design of timed, distributed and/or open systems: Members from each group together with their participation rate are given in Table 1 in page 26.

IRISA/Distribcom IRISA (Institut de Recherche en Informatique et Systèmes Aléatoires) is a joint research unit of CNRS, INSA Rennes, INRIA and University of Rennes 1. The DistribCom team[1] has developed an important body of techniques for *distributed observation and monitoring,* with applications to the distributed management of large scale telecommunication networks, services, and Web services [8]. These algorithms use models of true concurrency, in which time and states are only local, not global. Key structures handled by these algorithms are net unfoldings, event structures, and some special classes of graph grammars to address dynamicity. Time Petri Net has been chosen to address monitoring of real-time concurrent systems. A new definition of the unfolding of time Petri nets with dense time has been presented in [14].

IRCCyN IRCCyN (Institut de Recherche en Communication et CYbernétique de Nantes) is a joint research unit of CNRS, Ecole Centrale Nantes, Ecole des Mines Nantes and University of Nantes. The members of IRCCyN have a particular interest in modeling and verifying real-time and embedded systems. In that context, a large part of their work is focused on time Petri nets and their stopwatch extensions. For the analysis of these models they have developed the tool Romeo[2].
They have also made some significant contributions to the control of timed systems and have started recently to study the control problem for (timed) non-interference and proposed some pioneering problem statements and results in [3].

LaBRI LaBRI (Laboratoire Bordelais de Recherche en Informatique) is a joint research unit of CNRS, ENSEIR Bordeaux and University of Bordeaux 1. The members of this project participate in the work of the *Formal Methods* team within LaBRI. Their research interests range from theoretical research in automata theory, algebra and logic; through the theory of verification, infinite state model-checking, real-time model-checking; to the implementation of verification tools. Of particular interest to this project are the team's competence in: game theory and its applications, synthesis of distributed controllers, automata theory, verification of asynchronously communicating systems, theory of real-time systems, logics for concurrency, experience with industrial size verification and development of verification tools (AltaRica).

LAMSADE LAMSADE (Laboratoire d'Analyse et de Modélisation de Systèmes pour l'Aide à la DEcision) is a joint research unit of CNRS and University Paris-Dauphine. The research work of the team participating to the project is centered on design and evaluation of networks and computer systems. More precisely, the members of the project are interested in the verification of timed systems and the analysis of timed Web services, including client synthesis for such services which is closely related to control problems.

[1] http://www.irisa.fr/distribcom/

[2] http://romeo.rts-software.org

LSV LSV (Laboratoire Vérification et Vérification) is a joint research unit of CNRS and ENS Cachan (Ecole Normale Supérieure de Cachan). Research activities of the team involved in the project focus on the verification of critical software and systems. Its scientific program integrates fundamental long-term research together with applied activities, in cooperation with academic and industrial partners. The team's main competences include automata theory, model-checking, distributed and timed models. More recently, it starts to use these tools in the framework of control synthesis.

Existing cooperations. All the teams involved in DOTS project are also participating to several (inter)national research projects. An exhaustive list of contractual projects is given in Section C-2. Here we mention the main projects directly related to the topics of DOTS. Most of them are focused on one or two aspects (timed, open or distributed) considered in DOTS .

The French projects MeFoSyLoMa (with LAMSADE and other laboratories) and VERSYDIS (involving LaBRI, LSV but also LIAFA) are about verification and design of *distributed systems*. The CORTOS project (involving IRCCyN, LSV but also VERIMAG) is about timed and open systems. VERSYDIS and CORTOS have been founded by the ACI "Sécurité" and will end in December 2006.

The GAMES project (involving several European universities, among them LaBRI) is a European Research Training Network focused on the use of *games for the synthesis and verification*. The distributed games for the *open distributed systems* are studied in a cooperation between Warwick University and IRISA-Distribcom (this project is currently under submission to PAI Alliance).

The European network of Excellence ARTIST2 deals with the design of embedded systems. This is a large network where many aspects (from modeling to implementation of ES) are addressed. With more than 30 institutions in Europe, LSV is involved in ARTIST2. LSV is also involved in the MEDEA + project *Blueberries*.

Finally, the French-Indian project MODISTE-COVER (with LaBRI and LSV) is about *timed control* and *distributed control*. IRISA-Distribcom has also a cooperation about distributed control with the School of Computing National University of Singapore (NUS) via an INRIA associated team project (with P.S. Tiagarajan).

In addition to these projects, all the teams involved in DOTS have many informal cooperations with the international leader groups in verification of distributed, open and timed systems. For example, for emerging topics like unfolding or non-interference, cooperations exist between IRISA-Distribcom and V. Komenkho (Newcastle,UK) and Ottawa University and Ecole Polytechnique de Montréal, respectively.

As explained before an important characteristic of DOTS project is to bring together researchers working on different aspects with a common objective: combining and merging techniques in order to verify complex systems exhibiting timed, open and distributed aspects.

The DOTS project is structured with several workpackages corresponding to the combination of two or three aspects. For each of them, there will be a strong collaboration between the teams. The following array emphasizes the interactions between the teams:

	IRCCyN	IRISA	LaBRI	LAMSADE	LSV
Timed Open	X		X	X	X
Distributed Open		X	X		X
Distributed Timed	X	X		X	X
Distributed Open Timed	X	X	X	X	X

B-3 Detailed implementation plan

The DOTS project is organized in several workpackages corresponding to possible combinations of timed/open/distributed aspects. The last one deals with the design of systems exhibiting the three characteristics.

B-3.1 Timed open systems – (WP1)

In the untimed case, we understand quite well how to reduce control and synthesis problems to game solving problems. The situation is more difficult when real-time constraints have to be handled in the game. First there exist several possible choices for the semantics of such games: for example, one can allow each player to choose the moment of its moves independently of the other (thus they can surprise the other) or conversely ensure that all players act at the same moment (see [3, 15] for different choices of semantics). When the winning condition is simple and the timed constraints are those occurring in Timed Automata, the problem can be reduced to a finite game, using the standard region graph technique. Nevertheless these games are not always useful for control problems and sometimes we need to extend the framework a bit more but then we are faced with significant difficulties. For example the game solving problem is undecidable if there are at least three stop-watches or in the context of hybrid systems where clocks can have different rates. Finding pertinent definitions of games for the design of timed systems remains an important direction of work. We have identified three main directions. The first one concerns the theoretical framework of timed games (see Section B-3.1.1). The second one deals with the synthesis of timed controller (see Section B-3.1.2). The third one considers non-interference in timed systems (see Section B-3.1.3).

B-3.1.1 Quantitative objectives in timed games (T1.1)

Once time is explicit in the model, it becomes very interesting to consider *quantitative* winning objectives. For example, a specification of a real-time system typically involves optimizing some real-valued measure of QoS (Quality of Service), which can be latency, service time, etc. Such quantitative properties make the solving algorithms much more difficult (see for example [2, 11]). Thus we plan to work on the definition of pertinent classes of timed games for the design of timed open systems and study the associated algorithmic issues for quantitative objectives.

Another possible approach to state quantitative winning objectives in timed games could be the use of timed extensions of the Alternating Temporal Logic (ATL). Indeed ATL is a high-level specification language that allows us to express properties over multi-agents systems (or multi-players games). The existence of strategies for some (coalitions of) agents to ensure temporal properties can be easily stated with this formalism. Model checking ATL formulae gives then a natural way to determine whether a configuration of a game is winning or not for some player, that is whether a system is controllable in order to ensure some objective.

Extending ATL in order to handle timing constraints is clearly an interesting direction for expressing quantitative objectives for timed systems.

B-3.1.2 Synthesis of timed controllers (T1.2)

Controller synthesis for timed systems raise additional problems compared to the untimed setting because timed systems have infinitely many configurations, and "time elapsing" cannot be considered as a classical action: it is neither controllable nor uncontrollable. First results in this area have been obtained in the 1990's and this research domain has developed a lot in the last past years (see for example [3, 25, 13]). Thus a challenging topic is the *synthesis* of controllers. Indeed, deciding whether the system is controllable is not always equivalent to building a suitable controller for the system: the existence of a controller for a given system can sometimes be reduced to a model-checking problem [12], while the synthesis of a controller is often related to satisfiability problems which are usually much harder for timed systems. Design of algorithms for the controller synthesis for *quantitative control objectives* (for ex. for latency, service time, etc.) mostly remains an open question.

Moreover when dealing with synthesis, we aim at designing a controller that can be actually *implemented* by a non-perfect (imprecise) hardware/software component: thus we have to look for a *robust* strategy that does not suffer from the imperfections of the target platform, *i.e.* a strategy that can be executed with some bounded precision (digital clocks, delays etc.). Recently, models have been proposed for characterizing classes of implementable systems [16, 1], which take into account several characteristics of platforms and processors on which they have to be implemented. Synthesizing implementable controllers instead of first synthesizing a controller and then checking if it is implementable is a new and promising direction. We also plan to develop those synthesis algorithms for subclasses of hybrid automata.

B-3.1.3 Timed non-interference (T1.3)

Non interference has been extensively studied in an untimed setting since its first introduction in [24]. It is well known that information can be retrieved in an implicit way from the variations of time intervals between message transmissions. Thus abstracting away time constraints may be too coarse to reveal information leaks. While timed extensions of discrete event systems have become a important area of research, very little work has yet been devoted to non-interference conditions for such timed systems [36, 5]. Moreover it does not directly address the questions of verification and control. For timed automata, we proved in [3] that the verification of a non-interference property based on trace equivalence can be reduced to the (undecidable) universality problem for timed automata. On the other hand, other non-interference properties based on reachability, simulation or bisimulation are decidable for TA (as for bounded time Petri nets).

Our main objectives are to study more deeply the notion of non interference in a timed framework and to propose some verification and controller synthesis algorithms for this type of properties.

First, at the semantics level, we need to define the notions of non interference for timed models, using the basic formalisms of timed transition systems and bisimulation-like equivalences. This will lead to a global theory of timed non interference, that could then be specialized with hybrid automata or time Petri nets, depending on how its analysis would adapt to practical problems.

Secondly, we will address the problem of synthesis of timed non interferent controlled systems. Indeed in the context of automated security system design, the problem is not to verify that some security policy holds for a given system, but to restrict the possible actions of this system to ensure that the policy is met.

Summary of the principal tasks of Workpackage 1

1. Quantitative objectives in timed games
2. Synthesis of timed controllers
3. Timed non-interference

B-3.2 Distributed open systems – WP2

The verification of distributed open systems is by definition more complex than for sequential systems. The major difficulty comes from the fact that no global view of the system can be used. Hence, no complete information is available directly and exchanges of data between the processes can be done only by using the given architecture.

It is therefore not a surprise that a lot of problems, decidable in the untimed setting, become undecidable in a distributed setting. It is in particular the case of the distributed synthesis problem where the goal is to satisfy a specification for a system of processes communicating by rendez-vous, by synchronizing it with *local* controllers. A distributed controller is divided into several components, each of which has to guess a correct behavior against the environment, knowing only part of the system's state. Pnueli and Rosner [35] have shown that this problem is undecidable for most classes of architectures and obtained decidability for a special class of hierarchical architectures called *pipelines*. Further works have shown, roughly, that pipelines are the only distributed architectures for which the problem is known to be decidable.

Similarly to other cases described earlier in this proposal, most of the variants of control in distributed setting can be modeled by games. In such *distributed games*, each controller is modeled by a player and there is also one player representing the environment. The control problem reduces to finding a strategy for a coalition of the players against the environment. This reduction permits to focus on essential aspects of the problem which are linked to incomplete information: players have only partial information about what has happened in the game.

Those distributed games are at present much less understood than the sequential games. One of the objectives of the DOTS projects will be to pursue their study (see Section B-3.2.1). This fundamental work should contribute to a better knowledge of distributed control and is aimed at finding suitable hypotheses to get decidable classes.

A promising direction is to define restrictions on the specifications that the system has to fulfill (see Section B-3.2.2).

Another avenue of research worth exploring is the control of *asynchronous* systems. Surprisingly, whereas these systems are in general more complex than synchronous ones, this particular complexity can here turn into advantage (see Section B-3.2.3).

Finally, we plan to study non-interference for distributed systems which is a quite new area of interest (see Section B-3.2.4).

B-3.2.1 Theory of distributed game models (T2.1)

In recent years the members of our project have proposed and studied two variants of distributed games [34, 20]. While in general the problem of finding winning strategies for players in these games is undecidable, several results on solving particular cases of games have been established. The first approach is sufficiently strong to prove all known decidability results mentioned above. The second variant introduced the promising notion of *causal* memory allowing the players to gain a causal view of the play while still being oblivious of the global state. Series-parallel architectures are decidable in the asynchronous setting when causal memory is allowed, departing strongly from the class of known (hierarchical) decidable architectures for *local* memories. A great challenge is to extend the decidability to all architectures, using causal memory for the controllers. We intend to study the precise relation between the two game models of [34] and [20]. This issue is not a mere issue of comparing formalisms. It is an instance of a more general problem of understanding the mechanisms of information flow between parties due to communication. We hope this will lead to new techniques to tackle the decidability of distributed games.

B-3.2.2 Distributed control for restricted specifications (T2.2)

Decidability of distributed control depends on several parameters: the architecture of the system defining how processes may communicate, the memory allowed for the controller (local, causal, . . .), the kind of specification used (restricted to input/output variables or not, . . .). As mentioned above, the distributed control problem with local memory and specification on all variables is decidable mainly for pipeline architectures only. We believe that specifications restricted to external variables (internal variables are left unconstrained) are more relevant and we aim at studying the decidability in this case. Also, we think that interesting specifications are *robust* meaning that variables that are “independent” in the architecture should not be linked by the specification. All known undecidable cases rely on the fact that a controller needs to guess some information to which it has no access and only holds for specifications that are non robust or that put constraints on the internal variables. We intend to study the distributed control problem for input/output and robust specifications. We also want to understand how the information flow “capacity” of the architecture influence the decidability of the distributed control. Finally, we intend to study how causal memory could be used in synchronous systems to help solving distributed games. We hope that the game techniques described in the previous paragraph will be helpful here.

B-3.2.3 Control of asynchronous communication models (T2.3)

Another track we want to follow is synthesis for asynchronous systems, like for example automata communicating via FIFO channels. Verification of such systems is considered to be very difficult as one buffer is sufficient to simulate any Turing machine. This difficulty can be in fact turned into advantage in the context of synthesis roughly because controllers need not be synchronized with the system and can react with some delay. We have already done some research in this direction by studying specification formalisms and implementability in asynchronous setting [21, 22, 23]. An important issue here is the quality of controller: one would like to avoid deadlocks and also to put some constraints on reaction time.

B-3.2.4 Distributed non-interference (T2.4)

Recent works [28, 27] have shown how to find information leaks, such as covert channels, in distributed protocols. The approach uses game theory, and defines a notion of iterated interference. The covert channel game is a multiplayer game, where malicious users play to create an information flow disallowed by the protocol, while the rest of the system tries to prevent it. A winning strategy for the users highlights the existence of a possible covert channel.

Starting from these ideas, the first objective is to define a realistic notion of non interference for distributed systems. So far, strong assumptions on the system made the results unlikely to be broadly applicable. We also plan to obtain characterizations of interference in a distributed system in terms of games. Former results only exhibited sufficient conditions. Further, only global strategies were considered, which did not make the approach complete since a winning strategy might not be distributable. A long term goal is to describe relevant classes of distributed systems where one can solve the resulting games with distributed strategies.

Finally, *quantification* of information leaks is an important aspect when searching for illegal flows: leaking only few bits of information may not be sufficient to create an exploitable flow. As already stated by G. Lowe [30], the current definition of flows through interference [24, 10, 38] can be mainly viewed as a reachability property, and does not embed the notion of liveness. As a consequence, it does not capture any quantification. We want to use extended formalisms to quantify how severe is a detected information leak.

Summary of the principal tasks of Workpackage 2

1. Development of the theory of distributed game models.
2. Finding more decidable cases of controllable distributed systems, in particular by restricting classes of specifications and allowing a causal memory.
3. Study of the control problem in asynchronous communication models.
4. Application of the developed techniques in other settings as noninterference and discovery of hidden channels.

B-3.3 Timed Distributed Systems – WP3

In the last decades, major advances in the analysis of distributed (or concurrent) systems were based on two paradigms related to the distribution: the *independence* and the *locality* of actions. Whereas *partial-order* methods mainly take advantage of the independence (see e.g. [37]), the *unfolding*-based methods rely on both concepts [17, 32]. Furthermore from a semantical point of view, system unfoldings are a theoretical well-founded alternative to the usual interleaving semantics. It must be emphasized that this (sophisticated) semantics is more discriminant than the classical one and may be applied for other purposes (than verification) like observation and diagnosis.

If the previous studies led to efficient tools and algorithms, no counterpart has so far been achieved for *timed systems*. The main reason being that the expression of time synchronizations between actions in the standard timed models is essentially *global* and consequently this yields numerous conceptual and technical difficulties in order to adapt or extend the previous methods. Furthermore experimentations of the existing methods on realistic examples for complexity reduction are not so concluding. We want to overcome these limitations by addressing the following issues:

- the design of a *concurrent* timed semantics framework by revisiting timed or untimed models for concurrency with a special emphasis on semantics and expressiveness,
- the extension of the efficient techniques developed for distributed discrete systems to distributed timed systems.

Note that the concurrent semantics for timed systems will be used to perform also *fault diagnosis*. Moreover the second task should provide algorithms and tools for verifying safety critical properties of large systems.

B-3.3.1 Concurrent Semantics for Distributed Timed Systems (T3.1)

The main models that include timing information and are used to specify distributed timed systems are based on widely used discrete models of concurrency, i.e. networks of automata, Petri nets and, more recently message sequence charts. Although *concurrent* semantics have been developed and extensively studied for these models, there is almost no result for the timed versions: Networks of Timed Automata (NTA), Time - or Timed - Petri Nets (TPN, TdPN) and Timed Message Sequence Charts (TMSC). We want to develop a concurrent semantics framework for these models which can give solid foundations to the area of distributed timed systems. This means:

1. designing a model for distributed timed systems, which may be based on a synthesis of the features of the aforementioned models; we want to go further than the sequential semantics used for timed systems and equip the distributed timed models with an explicit *concurrent* semantics (or partial order semantics).
2. *validating* this approach by demonstrating that it can be used on a wide range of practical examples or case-studies;
3. studying the *theoretical properties* of the model, e.g. comparing its expressiveness to the previous models of timed systems.

This is a challenging problem from a technical point of view: in the discrete case (automata, languages), concurrent semantics are much more theoretically founded and enjoy nicer properties than their timed counterparts. This foundation step is a key step towards building efficient tools and algorithms for distributed timed systems.

B-3.3.2 Efficient Algorithms for Analyzing Distributed Timed Systems (T3.2)

The second part of our work is the design of algorithms that take advantage of the concurrent semantics for timed systems. In the context of timed systems, and especially considering TPN and NTA, the theoretical corpus, including algorithms and software, has considerably matured these last ten years. There are even relatively efficient tools to analyze networks of Timed Automata and Time Petri nets. It is however surprising to see that almost all techniques proposed until now for these models, consist in eliminating the concurrency by the computation of a single large sequential automaton, equivalent in a certain sense (generally in terms of sequences of transitions) to the original model. This has the following drawbacks: it does not scale up when increasing the amount of concurrency but it also destroys the causality and concurrency information present in the original model (this one can be nevertheless crucial in some applications like control synthesis, testing and supervision). The different methods proposed until now can be classified as follows:

- Partial order method for TA: In [7, 33], the authors define an alternative semantics for product of TA based on local time elapsing. The efficiency of this method depends on two opposite factors: local time semantics generate more states but the independency relation restricts the exploration. In [31] (in fact a generalization of [6]), the independency between transitions of a TA is exploited in a different way: the key observation is that the occurrences of two independent transitions do no need to be ordered (and consequently nor the occurrences of the clock resets). The relative drawback of the method is that, before their exploration, the symbolic states include more variables than the clock variables.

- Partial order method for time Petri nets: In [39], the authors generalize the concept of stubborn set concept to time Petri nets calling it *a ready set* with additional constraints and applying it to the class graph construction of [9]. the efficiency of the method depends on whether the (dynamical) timing coupling between transitions is weak or not. Unfortunately the urgency semantics of this model entails a strong timing coupling.

- Process semantics for time Petri nets: The generalization of the unfoldings for time Petri nets has been developed by different researchers. From a semantical point of view, in [4] the authors have studied the realizability of a non branching process in a time Petri net. Their study reveals that the temporal mechanism of these nets is "incompatible" with the locality of the firing rule. Thus in [29], the author proposes a method which controls the class graph construction with an unfolding of the *untimed* net. A subtle drawback of this method is that the unfolding may be infinite whereas the time Petri net is bounded. In order to define a timed unfolding for such a net, in [18] the authors consider a unit time elapsing as a special transition of the net. Thus the global synchronization related to this transition decreases the locality of the unfolding. Furthermore, when the intervals associated with the transitions are great this method suffers the usual combinatorial explosion related to the discrete time approach.

This brief overview outlines that improving the complexity of the analysis of distributed timed systems is still an open and difficult topic. We claim that equipping the concurrent model with an explicit concurrent semantics (a partial order semantics) will definitely help here. Effectively taking advantage of it through new algorithms is our second objective. It can be stated along the following lines: design new and efficient algorithms for analyzing distributed timed systems, implement the algorithms we have designed within a prototype and validate it on real case studies.

Summary of the principal tasks of Workpackage 3

1. Design of a *concurrent* semantics for distributed timed systems;
2. Design of *efficient algorithms* for analyzing distributed systems;
3. *Implementation* of our results in a prototype tool and validation of the approach on real case-studies.

B-3.4 Design of distributed, timed and open systems – WP4

The design of safe complex systems, *i.e.* distributed, timed and open systems, is the aim of DOTS project.

All partners will be involved in this workpackage: in order to handle such systems, we will work for merging all techniques developed in previous cases. Of course combining these techniques is not an easy task and we will have to make an important effort in order to adapt them.

In order to evaluate the proposed techniques, we plan to consider particular examples. First we plan to consider a problem of diagnosis in telecom protocols. Secondly we want to consider the control of an embedded system issued from industry. These systems will exhibit the different aspects mentioned above. Working together on these examples and considering the different facets of these systems will allow us to confront the approaches, the aim is then to propose an integrated DOTS methodology.

Clearly this workpackage will strongly depend on the previous results obtained in Workpackages 1, 2 and 3.

Summary of the principal tasks of Workpackage 4

1. Specification of realistic DOTS examples;
2. Formal analysis of the different facets of the examples;
3. Proposal for an integrated DOTS methodology.

B-3.5 Project management

Coordination. Each partner team has a coordinator:

- LSV: François Laroussinie
- IRCCyN: Didier Lime
- IRISA-Distribcom: Claude Jard
- LaBRI: Igor Walukiewicz
- LAMSADE: Béatrice Bérard

In addition, there is one leader for each workpackage. They will be responsible for the detailed coordination and planning of the workpackages and the monitoring of the corresponding tasks.

- WP 1 – timed & open : François Laroussinie
- WP 2 – distributed & open : Igor Walukiewicz
- WP 3 – timed & distributed : Claude jard
- WP 4 – distributed, timed & open: Franck Cassez

The steering committee (SC) of DOTS project will include all team coordinators and all workpackage leaders.

Kick-off meeting. The DOTS project will start by a kick-off meeting with all participants. All WP leaders will present a detailed program for their workpackage.

Review meeting. The steering committee will prepare the annual meeting with all participants. The aim of these meetings will be to present the work done in every WP and to discuss the possible necessary adaptations of the program in order to meet the goals of the project. Moreover every WP leader will organize specific meetings of the corresponding WP.

Communication. A web site of DOTS project will be created. It will be used both for the communication within the project and for the dissemination of results. All internal information (activity reports, minutes of meeting, etc.) will be accessible for all participants. A wiki area will be used for this aim. Moreover all publications will be available. We also plan to present our results in well recognized international conferences. A workshop could also be organized by DOTS project at the end of the project.

B-4 Deliverables

The person in charge of a deliverable is the leader of the corresponding workpackage.

Notation: R = report; P = prototype; D = demonstrator

Nb	Title	Nature	Resp.	Participants	$t_0 + \ldots$
D0.1	DOTS website	D	LSV	all	3
D0.2	Activity report - progess & eval.	R	LSV	all	6
D0.3	Activity report - progess & eval.	R	LaBRI	all	12
D0.4	Activity report - progess & eval.	R	IRISA	all	18
D0.5	Activity report - progess & eval.	R	LAMSADE	all	24
D0.6	Activity report - progess & eval.	R	IRCCyN	all	30
D0.7	Activity report - progess & eval.	R	LaBRI	all	36
D0.8	Activity report - progess & eval.	R	IRISA	all	42
D0.9	Final activity report	R	LSV	all	48
D1.1	Quantitative object. in timed games	R	LSV	LaBRI, LSV	18
D1.2a	Synthesis of timed controllers	R/P	-	IRCCyN, LSV	24
D1.2b	Implementability of timed cont.	R	-	IRCCyN, LSV	36
D1.3a	Spec. of timed non-interference	R	-	IRCCyN, LAMSADE	12
D1.3b	Verif. of timed non-interference	R/P	-	IRRCyN, LAMSADE	36
D1.3c	Control for timed non-interfe.	R/P	-	IRCCyN, LAMSADE	42
D2.1	Theory of distributed games	R	LaBRI	LaBRI, LSV	18
D2.2	Distributed control for restricted specification	R	-	LaBRI, LSV	24
D2.3	Control of asynchronous systems	R	-	IRISA, LaBRI	36
D2.4a	Notion of distributed non-interf.	R	-	IRISA, LaBRI	24
D2.4b	Quantitative specif. for non-interf.	R	-	IRISA, LaBRI	36
D3.1a	Model for distributed timed systems	R	IRISA	IRISA, LSV	18
D3.1b	Validation over case studies	R	-	IRISA, LSV	24
D3.2	Efficient algorithms for distributed timed systems	R	-	IRISA, IRCCyN, LSV, LAMSADE	24
D3.3	Implementation in a prototype	P	-	IRISA, IRCCyN, LSV, LAMSADE	42
D4.1	Specif. of realistic DOTS examples	R	IRCCyN	all	12
D4.2	Formal analysis of the different facets of the examples	R	-	all	42
D4.3	Proposal for an integrated DOTS methodology	R	-	all	48

B-5 Expected results

As we claimed in the preliminaries, our scientific objectives are to significantly advance both theory and algorithmics of design and verification of distributed, open and timed systems.

Therefore, the first output of the DOTS project will be the publication of articles in international journals and communications in international conferences in the domains covered by the project. Among them, the number of works involving several teams of the project will be a sign of success of the collaborations. The quality of the prototypes we plan to develop should also be evaluated. It will in particular be attested by the “size” of the examples we will be able to treat.

As we explained in our financial demands, the main part is devoted to PhD and post-docs positions. The success of the project will thus be judged also on our ability to attract good PhD and post-docs students and the quality of their studies.

Finally, the teams participating to the DOTS project have numerous international scientific relations with some of the very best groups all over the world. The project should be also the opportunity to develop these cooperations and in particular to encourage exchanges of researchers and students between the DOTS teams and these groups.

More precisely, we list below, for each workpackage, the scientific results we would like to obtain.

- Timed open systems:
  - Definition of *pertinent classes* of timed games for the design of timed open systems.
  - Design of algorithms for the synthesis of timed controllers for *quantitative* control objectives.
  - Synthesis of *implementable* controllers i.e. that can be executed with some bounded imprecision.
  - Definition of suitable *non-interference conditions* for timed systems.
  - Synthesis of timed *non interferent controlled* systems.
- Distributed open systems:
  - Development of the theory of *distributed game* models.
  - Study of the distributed control problem for input/output and *robust* specifications.
  - Utilization of *causal memory* in synchronous systems to solve distributed games.
  - Study of the control problem in *asynchronous* communication models.
  - Definition of a realistic notion of *non interference for distributed* systems and characterization in term of games.
  - *Quantification of severity* of information leaks.
- Timed distributed systems:
  - Design of a *concurrent* semantics for distributed timed systems.
  - Design of *efficient algorithms* for analyzing distributed timed systems.
  - *Implementation* of our results in a prototype tool and validation of the approach on real case-studies.
- Distributed Open Timed systems:
  - Application of techniques elaborated in previous WPs to two examples issued from diagnostic of telecom protocols and control of embedded systems.
  - Proposal of an integrated DOTS method.

C - Justifications scientifiques des moyens demandés

C-1 Moyens financiers demandés au GIP ANR dans le cadre du présent AAP

La majeure partie de notre budget est consacrée aux allocations de thèse et de postdoctorat. Notre projet de recherche est un projet de sur quatre ans, il nous permettra ainsi d'encadrer des thésards durant la totalité de leur thèse. Nous souhaitons recruter un doctorant pour chacun des trois premiers sous-projets (SP1 "temporisé-interactif" ; SP2 "distribué-intéractif" ; SP3 "distribué-temporisé"). De plus, nous sollicitons deux allocations de postdoctorat pour la seconde phase du projet (années 3 et 4) pour le SP4, lors de l'intégration des techniques élaborées dans les SP1, SP2 et SP3.

Nous insistons sur le fait que ces recrutements auront une dimension multi-site : les thésards recrutés auront à travailler en étroite collaboration avec l'ensemble des chercheurs impliqués dans les SP correspondants, cela impliquera des séjours longs dans les différents sites et pourra aussi se traduire par des coencadrements de thèse.

Le budget "fonctionnement" sera réparti au prorata des investissements des différentes équipes impliquées. Il servira à l'organisation des réunions internes du projet (réunions plénières et réunions des sous-projets), aux visites entre sites (par exemple pour de longs séjours), ainsi qu'à des missions à l'étranger pour assister à des conférences internationales du domaines etc.

	Nombre	Coût	Total
Alloc. doctorat	3	90 k€ (pour 3 ans)	270 k€
Alloc. postdoctorat	2	45 k€ (par an)	90 k€
Equipement des CDD	5	5 k€	25 k€
Fonctionnement			200 k€
Total			**585 k€**

Ce qui donne la répartition suivante par partenaire :

	IRCCyN	IRISA	LaBRI	LAMSADE	LSV
Alloc. doctorants		90 k€	90 k€		90 k€
Alloc. postdoc	45 k€			45 k€	
Equipement pour les (post)doctorants	5 k€	5 k€	5 k€	5 k€	5 k€
Budget fonctionnement	42 k€	39 k€	46 k€	24 k€	49 k€
Total	92 k€	134 k€	141 k€	74 k€	144 k€

C-2 Autres actions contractuelles dans lesquelles les partenaires sont engagés

Nous présentons ici les projets institutionnels dans lesquels des personnes participant au projet sont investies.

DISTRIBCOM – IRISA :

- PERSIFORM – Ingénierie de Performances basée sur la Simulation à partir de Modèles Fonctionnels Formels
  Financement : projet RNRT Partenaires : France Telecom R&D, INT, Orpheus, VERIMAG
  Responsable : VERIMAG – Responsable local : C. Jard
  Durée : 2004–2007
- CO2 (Composition de Scénarios)
  Financement : Contrat France Telecom.
  Partenaires : France Telecom.
  Responsable : France Telecom, Responsable local : Loïc Hélouet
  Durée : 2004–2006.
- SWAN – (Supervision dynamique et autonomes de systèmes distribués)
  Financement : projet RNRT
  Partenaires : LORIA INRIA Lorraine, LIPN, QOSMETRIX, ALCATEL CIT, France Telecom
  Responsable : C. Jard
  Durée : 2004–2006
- CASDS (Contrôle et diagnostic de systèmes communicants distribués)
  Financement : Equipe associée INRIA, NUS Grant.
  Partenaires : School of Computing NUS, S4 IRISA.
  Responsables : P.S. Thiagarajan – Responsable Disribcom : Loïc Hélouet.
  Durée : 2006–2008

- (En cours de soumission) Games and concurrency
  Financement : PAI Alliance, MAE et British council
  Partenaires : Université de Warwick
  Responsables : M. Jurdzinski, D. Peled, B. Genest
  Durée : 2006–2008

LaBRI :

- GAMES (Jeux et automates pour synthèse et la validation)
  Financement : Research Training Network, Union Européenne
  Partenaires : Aachen, Bordeaux, Edinburgh, Paris 7, Rice, Uppsalla, Warsaw, Wien.
  Responsable : E. Graedel (Aachen), Responsable local : David Janin
  Durée : 2002–2006
- Versydis (Vérification de systèmes distribués)
  Financement : MENRT (ACI SI)
  Partenaires : LSV, LIAFA
  Responsable : P. Gastin (LSV), Responsable local : Igor Walukiewicz
  Durée : 2003–2006
- MODISTE-COVER Modèles distribués et temporisés pour le contrôle et la vérification
  Financement : MAE et CNRS (Projets de recherche en réseau), projet Franco-Indien
  Partenaires : LaBRI, Liafa, LSV, Chennai Mathematical Institute, Institute of Mathematical Sciences (Chennai), Indian Institute of Science (Bangalore)
  Responsables : P. Weil (LaBRI) et M. Mukund (CMI, Chennai)

Durée : 2005–2008
- Automata, profinite semigroups and symbolic dynamics
  Financement : cooperation PESSOA (PAI Portugal), projet Franco-Portugais
  Partenaires : LIAFA, LaBRI et Centro de Matematica Universidade do Porto Responsables : M. Zeitun (LaBRI) et Silva Da (Univ do Porto)
  Durée :2006–2007

LSV :

- European Network of Excellence ARTIST2 – Conception de systèmes embarqués.
  Financement : Information Society Technologies
  Partenaires (du cluster "Testing and Verification") : Univ. d'Aalborg (DK), VERIMAG, Univ. de Twente (PB), Centre Fédéré en Vérification (B), INRIA. IRISA
  Responsable : J. Sifakis (VERIMAG) – Responsable local : Ph. Schnoebelen
  Durée : 2004–2008
- CORTOS – contrôle de systèmes temporisés
  Financement : MENRT (ACI SI) Partenaires :IRCCyN, VERIMAG
  Responsable : P. Bouyer (LSV)
  Durée : 2003–2006
- Versydis (Vérification de systèmes distribués)
  Financement : MENRT (ACI SI)
  Partenaires : LaBRI, LIAFA
  Responsable : P. Gastin (LSV)
  Durée : 2003–2006
- MODISTE-COVER Modèles distribués et temporisés pour le contrôle et la vérification
  Financement : MAE et CNRS (Projets de recherche en réseau), projet Franco-Indien
  Partenaires : LaBRI, Liafa, LSV, Chennai Mathematical Institute, Institute of Mathematical Sciences (Chennai), Indian Institute of Science (Bangalore)
  Responsables : P. Weil (LaBRI) et M. Mukund (CMI, Chennai) –
  Responsable local : Paul Gastin
  Durée : 2005–2009
- (En cours de soumission) MORSE-2 – Vérification de systèmes embarqués critiques temps-réel
  Financement : projet RNTL
  Partenaires : LIP6, ENST, Sagem, AONIX
  Responsable : F. Kordon (LIP6) – Responsable local : F. Laroussinie (LSV)
  Durée : 2006–2009 (si accepté)

IRCCyN :

- CORTOS – contrôle de systèmes temporisés
  Financement : MENRT (ACI SI) Partenaires :LSV, VERIMAG
  Responsable : P. Bouyer (LSV) – Responsable local : F. Cassez
  Durée : 2003 – 2006

References

[1] K. Altisen and S. Tripakis. Implementation of timed automata: An issue of semantics or modeling? In *Proc. 3rd International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS'05)*, volume 3829 of *Lecture Notes in Computer Science*, pages 273–288. Springer, 2005.

[2] R. Alur, M. Bernadsky, and P. Madhusudan. Optimal reachability in weighted timed games. In *Proc. 31st International Colloquium on Automata, Languages and Programming (ICALP'04)*, volume 3142 of *Lecture Notes in Computer Science*, pages 122–133. Springer, 2004.

[3] E. Asarin, O. Maler, A. Pnueli, and J. Sifakis. Controller synthesis for timed automata. In *Proc. IFAC Symposium on System Structure and Control*, pages 469–474. Elsevier Science, 1998.

[4] T. Aura and J. Lilius. A causal semantics for time Petri nets. *Theor. Comp. Sci.*, 243(1–2):409–447, 2000.

[5] R. Barbuti and L. Tesei. A decidable notion of timed non-interference. *Fundamenta Informaticae*, 54:137–150, 2003.

[6] W. Belluomini and C. J. Myers. Verification of timed systems using POSETs. In *Proc. 10th Int. Conf. Computer Aided Verif. (CAV'98)*, volume 1427 of *LNCS*, pages 403–415. Springer, 1998.

[7] J. Bengtsson, B. Jonsson, J. Lilius, and W. Yi. Partial order reductions for timed systems. In *Proc. 9th Int. Conf. Concurrency Theory (CONCUR'98)*, volume 1466 of *LNCS*, pages 485–500. Springer, 1998.

[8] A. Benveniste, E. Fabre, C. Jard, and S. Haar. Diagnosis of asynchronous discrete event systems, a net unfolding approach. *IEEE Transactions on Automatic Control*, 48(5):714–727, May 2003.

[9] B. Berthomieu and M. Diaz. Modeling and verification of time dependent systems using time Petri nets. *IEEE Trans. Softw. Engineering*, 17(3):259–273, 1991.

[10] G. Boudol and I. Castellani. Non-interference for concurrent programs and thread systems. *Journal of Theoretical Computer Science*, 281(1):109–130, 2002.

[11] P. Bouyer, F. Cassez, E. Fleury, and K. G. Larsen. Optimal strategies in priced timed game automata. In *Proc. 24th Conference on Foundations of Software Technology and Theoretical Computer Science (FST&TCS'04)*, volume 3328 of *Lecture Notes in Computer Science*, pages 148–160. Springer, 2004.

[12] P. Bouyer, F. Cassez, and F. Laroussinie. Modal logics for timed control. In *Proc. 16th International Conference on Concurrency Theory (CONCUR'05)*, volume 3821 of *Lecture Notes in Computer Science*, pages 81–94. Springer, 2005.

[13] P. Bouyer and F. Chevalier. On the control of timed and hybrid systems. *EATCS Bulletin*, 89, June 2006. To appear.

[14] T. Chatain and C. Jard. Time supervision of concurrent systems using symbolic unfoldings of time Petri nets. In *FORMATS*, volume 3829 of *LNCS*, pages 196–210, 2005. Extended version available in INRIA Research Report RR-5706.

[15] L. de Alfaro, M. Faella, T. A. Henzinger, R. Majumdar, and M. Stoelinga. The element of surprise in timed games. In R. M. Amadio and D. Lugiez, editors, *CONCUR*, volume 2761 of *Lecture Notes in Computer Science*, pages 142–156. Springer, 2003.

[16] M. De Wulf, L. Doyen, and J. Raskin. Almost ASAP semantics: From timed models to timed implementations. In *Proc. 7th International Workshop on Hybrid Systems: Computation and Control (HSCC'04)*, volume 2993 of *Lecture Notes in Computer Science*, pages 296–310. Springer, 2004.

[17] J. Esparza, S. Römer, and W. Vogler. An improvement of McMillan's unfolding algorithm. *Formal Methods in Syst. Design*, 20(3):285–310, 2002.

[18] H. Fleischhack and C. Stehno. Computing a finite prefix of a time Petri net. In *Proc. 23rd Int. Conf. Application and Theory of Petri Nets (ICATPN'02)*, volume 2369 of *LNCS*, pages 163–181. Springer, 2002.

[19] G. Gardey, J. Mullins, and O. H. Roux. Non-interference control synthesis for security timed automata. In *3rd International Workshop on Security Issues in Concurrency (SecCo'05)*, Electronic Notes in Theoretical Computer Science, San Francisco, USA, Aug. 2005. Elsevier.

[20] P. Gastin, B. Lerman, and M. Zeitoun. Distributed games and distributed control for asynchronous systems. In *Proc. of LATIN'04*, volume 2976 of *lncs*, pages 455–465. springer, 2004.

[21] B. Genest. *The odyssey of MSC-graphs*. PhD thesis, Paris 7, 2005.

[22] B. Genest. On implementation of global concurrent systems with local asynchronous controllers. In *CONCUR*, volume LNCS 3653, pages 443–457, 2005.

[23] B. Genest, D. Kuske, and A. Muscholl. A kleene theorem for a class of communicating automata with effective model-checking algorithms. *Information and Computation*, 204:920–956, 2006.

[24] J. Goguen and J. Meseguer. Security policy and security models. In *Proc.of IEEE Symposium on Security and Privacy*, pages 11–20. IEEE Computer Society Press, 1982.

[25] Th. A. Henzinger and P. W. Kopke. Discrete-time control for rectangular hybrid automata. *Theoretical Computer Science*, 221:369–392, 1999.

[26] G. Holzmann. *The SPIN model checker*. Addison-Wesley, 2003.

[27] L. Hélouët. Finding covert channels in protocols with message sequence charts: the case of rmtp2. In *SAM'04, Conference on SDL and MSCs*, 2004.

[28] L. Hélouët, M. Zeitoun, and A. Degorre. Scenarios and covert channels: another game. In *Proc of Games in design and Verification*, 2004.

[29] J. Lilius. Efficient state space search for time Petri nets. In *ENTCS*, volume 18, 1998.

[30] G. Lowe. Quantifying information flow. In *IEEE Computer Security Foundations Workshop*, pages 18–31, June 2002.

[31] D. Lugiez, P. Niebert, and S. Zennou. A partial order semantics approach to the clock explosion problem of timed automata. In *Proc. 10th Int. Conf. Tools and Algo. for the Construction and Analysis of Syst. (TACAS'04)*, volume 2988 of *LNCS*, pages 296–311. Springer, 2004.

[32] K. McMillan. A technique of state space search based on unfolding. *Formal Methods in Syst. Design*, 6(1):45–65, 1995.

[33] M. Minea. Partial order reduction for model checking of timed automata. In *Proc. 10th Int. Conf. Concurrency Theory (CONCUR'99*, volume 1664 of *LNCS*, pages 431–446. Springer, 1999.

[34] S. Mohalik and I. Walukiewicz. Distributed games. In *FSTTCS'03*, volume 2914 of *Lect. Notes Comp. Sci.*, pages 338–351, 2003.

[35] A. Pnueli and R. Rosner. Distributed reactive systems are hard to synthetize. In *Proceedings of 31th IEEE Symp. FOCS*, pages 746–757, 1990.

[36] R. G. R. Focardi and F. Martinelli. Real-time information flow analysis. *IEEE Journal on Selected Areas in Communications*, 21(1):20–35, 2003.

[37] A. Valmari. Stubborn sets for reduced state space generation. In *Proc. 10th Int. Conf. Applications and Theory of Petri Nets*, volume 483 of *LNCS*, pages 491–515. Springer, 1989.

[38] Volpano and Smith. Eliminating covert flows with minimum typings. In *PCSFW: Proc. 10th Computer Security Foundations Workshop.* IEEE Computer Society Press, 1997.

[39] T. Yoneda and B.-H. Schlingloff. Efficient verification of parallel real-time systems. *Formal Methods in Syst. Design*, 11(2):187–215, 1997.