Information Societies Technology (IST) Programme

Future and Emerging Technologies - Open Domain

Project acronym: Advance

Project full title:

Advanced Validation Techniques
for Telecommunication Protocols

Contract no: IST-1999-29082

Deliverable 13

Progress Report

Year 2 : October 2001 - September 2002

Date of preparation: September 2002

Contributors: all groups

Edited by: Ahmed Bouajjani, University of Paris 7

Part A

Executive summary

1 Work progress and achievements

The main achievements obtained during the second year of the project are the following:

1.1 Verification methods and tools of extended automata

1. The tools LASH and TReX for symbolic analysis of infinite-state extended automata have been significantly improved. The improvements concern the used representation structures as well as the state space exploration techniques. Moreover, an integration of the NDD automata-based representation package of LASH into the TReX tool has been done.

2. The IF-tools have also been significantly improved and extended. Simulation and finite-state model-checking have been extended to the larger new version of the language IF (version IF2.0), especially in order to take into account dynamic creation of processes. Moreover, an important effort has been devoted to the improvement, the development, and the integration of abstraction and model-checking tools. In particular, static analysis reduction techniques for IF2.0 models have been developed and combined with efficient state space exploration techniques.

3. New algorithms have been developed for the verification of pushdown systems for various kinds of properties. An efficient model checker for pushdown systems, called MOPED, has been implemented and effectively applied to programs with recursive procedures. Several promising theoretical works on the extension of this framework to the case of parallel programs have been carried out.

1.2 Verification of parametrized networks and dynamic systems

1. The paradigm of regular model-checking have been extended through the design of more efficient and more general algorithms. In particular, regular model checking has been generalized to the case of regular tree relations. This generalization allows for instance to deal with parametrized networks with tree-like topologies, as well as many (infinite-state) systems modeled as term rewrite systems.

2. New methods and algorithms have been designed for checking liveness properties of parametrized networks. These methods apply several approaches: automatic inference of network invariants, automatic derivation of fairness conditions on finite abstract models, and probabilistic reasoning. These methods have been implemented in the TLV tool and effectively applied for the verification of nontrivial parametrized algorithms.

3. Decidability results have been obtained for general classes of parametrized systems: systems with shared resources, and networks of communicating processes with tree-like topologies,

4. Methods and algorithms for the analysis of classes of infinite-state dynamic systems have been developed. In particular, two important classes of systems which exhibit dynamic behavior have been considered, namely security protocols and multi-threaded programs.

5. Important results have been obtained on the verification of several complex and challenging case studies (PGM protocol, AXD take-over manager of Ericsson, TTP/C group membership algorithm used in the automotive industry). These results have

been obtained using novel methods and techniques allowing to consider for the first time different parametric aspects and sources of infinity.

1.3 Common specification and verification environment

1. The IF language has been extended in order to cover the case of parametrized networks of processes with different topologies (sequences, trees, hypercubes, and their combinations). For that, the extended version of IF allows the use of a new class of models based on automata with arrays having parametric bounds. Intuitively, each cell of an array contains the state of one of the processes in the network. The indices of the array cells can be vectors of integers or strings over a finite alphabet (strings are used in the case of tree-like topologies). The operations on the arrays can be guarded by formulas involving quantification over indices, which allow to express global conditions on the values in the arrays (states of all processes in the network).

2. A common (graphical) interface has been developed allowing to invoke all the tools developed in the project on models described in the common language IF. These tools have been integrated to this common environment through filters which use abstraction (slicing) in order to produce models in the fragments of IF which can be handled by each of the connected tools.

1.4 Cases studies and experimentations

An important effort has been devoted to the study and the verification of complex and challenging case studies. The obtained results are new and of very good quality. They are more general than existing ones concerning these case studies (the classes of systems to which these case studies belong): they take into account several aspects such as various dimensions of parameterization and infinity, which have never been considered in the automatic verification of such complex systems.

1. The PGM protocol has been verified by considering different parametric aspects: (1) arbitrary tree-like topology with an arbitrary number of components, (2) infinite number of different sent messages, (3) generic constraints on the parameters of the protocol: propagation delays, sizes of the windows, speed of the source of the sent messages, etc.

   The automatic verification of the PGM has been carried out using the tools developed in the project: IF-tools and LASH.

2. The AXD case study has been deeply studied. This work leads to the development of a new theoretical framework and a general methodology for the parametric verification of systems with shared resources. Interesting and nontrivial decidability results have been obtained for such systems and wide classes of (safety and liveness) properties including deadlock freeness, mutual exclusion, starvation freeness, etc.

3. The TTP/C group membership algorithm, used in automotive industry, have been studied and automatically verified for an arbitrary number of components and an arbitrary number of faults. The automatic verification of this algorithm has been carried out using the tool LASH.

2 Dissemination of results

2.1 Publications

Some of the obtained results during this year of the project have been published/presented (or have been accepted for publication) in the following international conferences (in chronological ordering):

- Intern. Conference on Tools and Algorithms for the Construction and Analysis of Systems *TACAS'02*, April 2002, France,
- Intern. Conference on Application and Theory of Petri Nets *ICATPN'02*, June 2002, Australia,
- Intern. Conference on Computer Aided Verification *CAV'02*, July 2002, Denmark,
- Intern. Conference on Concurrency Theory *CONCUR'02*, August 2002, Czech Republic,
- Intern. Workshop on Verification of Infinite-State Systems *Infinity'02*, August 2002, Czech Republic,
- Intern. Symposium on Formal Techniques in Real-Time and Fault Tolerant Systems *FTRTFT'02*, September 2002, Germany,
- Intern. Static Analysis Symposium *SAS'02*, September 2002, Spain,
- Intern. Conference on Logic for Programming, Artificial Intelligence, and Reasoning *LPAR'02*, October 2002, Georgia,
- Intern. Workshop on Verification, Model Checking, and Abstract Interpretation *VMCAI'03*, January 2003, USA,
- Intern. Symposium on Principles of Programming Languages *POPL'03*, January 2003, USA.

2.2 Workshop and conference organization

- The European Joint Conferences on Theory and Practice of Software (ETAPS) has been organized by VERIMAG (**UJF**) in Grenoble, April 2002. ETAPS is the primary European forum for academic and industrial researchers working on topics relating to Software Science. It consists of a confederation of several conferences (TACAS, ESOP, FASE, FOSSACS, and CC), and it also includes many satellite workshops on several specialized and emerging topics.
- The second International Workshop on Embedded Software (EMSOFT) has been organized by VERIMAG (**UJF**) in Grenoble, October 2002.

3 Coordination and Management

3.1 Internal meetings, visits, and cooperations

- Four project meetings have been organized in 6-7 December 2001 in Paris, 21-22 March 2002 in Edinburgh, 8-11 June in Nässlingen (Sweden), and 22-23 October 2002 in Paris.

- Javier Esparza from Edinburgh visited LIAFA (**UP7**) in May and July 2002. The collaboration between (**UEdin**) and (**UP7**) lead to a common publication.
- Ahmed Bouajjani and Marc Boyer from Paris (**UP7**) visited VERIMAG (**UJF**) February 2002. The groups of Paris and Grenoble had several working meetings around the PGM case study proposed by France-Telecom.
- Ahmed Bouajjani, Peter Habermehl, and Tomas Vojnar from Paris (**UP7**) visited Thomas Arts (**Ericsson-UAB**) in April 2002. They had several working meetings around the AXD case study proposed by Ericsson.
- The groups of Paris (**UP7**) and Liège (**Liège**) had many interactions on the integration of the NDD package of LASH in the TReX tool of Paris.
- The work on tool integration and the one on case studies (especially on PGM) was the result of several fruitful exchanges between all groups in the projects, and tight collaborations between them.
- The advice of the industrial partners/observers (**F-Telecom**) and (**Ericsson-UAB**) was very important. They have continuously guided the substantial fundamental research and experimental work which have been carried out this year.

3.2 Management of the project

The table below describes the manpower (Paid/Non paid persons-months effort) used by the partners of the project.

Participant	**WP1**	**WP2**	**WP3**	**Total**
(**UP7**)	13/4	14/6	1/1.5	28/11.5
(**F-Telecom**)	0	3	0	3
(**Liège**)	3.7/1.4	1.4 / 0.7	0/0.3	5.1/2.4
(**UEdin**)	8/4	4/2	0/0	12/6
(**Uppsala**)	15/4	3/2	0/1	18/7
(**UJF**)	9	9	1	19
(**Weizmann**)	6/1.5	4.6/1.3	0/0	10.6/2.8
	54.7/14.9	39/12	2/2.5	95.7/29.7

There was no major deviations from the expected effort defined in the original proposal. The existing deviations have the following explanations:

- France-Telecom (**F-Telecom**) planned an effort of 4 months for this year (instead of the 3 spent months). The deviation is due to the fact that France-Telecom has given priority to the work on PGM case study, and has not developed substantially the planned work on symbolic test case generation. It is planned to increase the effort next year on symbolic test case generation, especially due to the arrival of a new PhD student who will start his thesis on this topic. Hence, the manpower for next year is expected to be higher than 4 men/month.
- Paris used 4 more payed months (28 instead of 24). This corresponds in fact to the plans fixed at the end of last year: the 8 man-months planned for the first year of

the project were not used. Therefore, we decided to distribute these man-months on the remaining years, and use 4 additional months in each of the second and the third years (as a PhD student grant).

- The planned paid effort by Liège was 12 man-months. The deviation is due to the fact that one of the researchers has left the team in December 2001, and the group has not been able to replace him before July 2002.
- Uppsala has spent more paid man-months this year (18 instead of 13.2) due to the use of MSc and PhD students. Some of the additional months spent this year were remaining from last year, since Uppsala has only spent 11 paid man-months in the first year of the project.

Part B

Achievements per Working Package

1 WP1: Methods and tools

1.1 Task T1.1: Abstraction

In [BGM02] abstraction techniques based on various static analysis reduction techniques have been developed and implemented in the IF environment. These reduction techniques include slicing, live-variable analysis and dead-code elimination.

In [PXZ02], techniques combining abstraction techniques and automatic derivation of fairness conditions are introduced in order to verify liveness properties of parametrized systems. In [KPSZ02], another approach for proving liveness properties for parametrized systems is proposed. This approach is based on the notion of network invariants, and combines abstraction and deductive techniques, In [ZPK03], the verification of parametrized probabilistic systems is addressed using a reduction to the construction of abstractions with adequate fairness constraints.

In [BLP02] accurate abstraction techniques have been designed for the verification of cryptographic protocols.

1.2 Task T1.2: Symbolic analysis

1.2.1 Regular Model Checking

Regular model checking is a uniform framework for the verification of infinite-state systems. It consists in representing sets of configurations as regular languages and actions as finite-state transducers. Then, the verification problem is reduced to the problem of constructing the transitive closures of regular relations. The principle of regular model checking has been introduced and developed by several researchers in our project in the last few years.

This year, the paradigm of regular model checking has been extended by the teams of Paris and Uppsala through the design of more efficient and more general algorithms. In [AJNd02], a new and more efficient algorithm for regular model checking is proposed. In [BT02] and [AJMd02], regular model checking has been extended to tree languages and trees. These papers proposes acceleration techniques based on automatic procedures for computing transitive closure for regular tree relations, and on automatic widening techniques on tree automata. The extension of regular model checking to trees allow to deal with parametrized networks with tree-like topologies as well as many with other classes of systems which can be modeled as term rewrite systems, e.g., multi-threaded programs and cryptographic protocols.

1.2.2 Monotonic systems

A recent work by the Uppsala team addresses the problem of analyzing monotonic systems, which cover Petri nets, lossy channel systems, relational automata, broadcast and cache-coherence protocols, timed Petri nets, ... For such systems, the exact set of reachable states often has a more complicated structure than its downward closure, since it contains much more information about the system behavior. The paper [AMN02] proposes a novel formalism for representing downward-closed languages, called the *downward closed language generators (dlgs)*. It is shown that the standard set operations such that intersection, union and entailment checking, needed for performing symbolic reachability analysis, are computable for dlgs, which brings them close to their practical use in a tool. In [AN02] the model checking problem of LTL properties for timed Petri nets is shown to be undecidable.

1.2.3 Pushdown specifications/systems

In [KPV02b], the problem of checking (nonregular) pushdown specifications, expressed as nondeterministic pushdown parity tree automata (PD-NPT), is addressed. The model-checking algorithm proposed in the paper involves a new solution to the nonemptiness problem of nondeterministic pushdown tree automata, where the best known upper bound is improved from a triple-exponential to a single exponential. In [KPV02a], an automata-theoretic framework is developed for reasoning about linear properties of infinite-state sequential systems. The framework is based on the observation that states of such systems, which carry a finite but unbounded amount of information, can be viewed as nodes in an infinite tree, and transitions between states can be simulated by finite-state automata. Checking that the system satisfies a temporal property can then be done by an alternating two-way automaton that navigates through the tree.

1.2.4 Multi-threaded programs, parallel recursive programs

In [BT02], regular model checking techniques are applied to the verification of multi-treaded programs. In the adopted approach, a configuration is a term representing the control structure of the program and actions of the program are modeled as term rewrite systems. Widening techniques on tree automata are used to compute the reachability sets of such systems. In [BET03], a general approach for the verification of parallel compositions of recursive programs is proposed. This approach is based on defining abstractions of communicating pushdown automata using an algebraic framework based on commutative Kleene algebras.

1.2.5 Symbolic representation structures

Significant progress has been made in the improvement, the development and the integration of packages for symbolic representation and manipulation of infinite sets of configurations (see deliverables D6, D9, and D11).

1.3 Task T1.3: Efficient state-space exploration

Efficient state-space exploration techniques have been developed and implemented in the IF-environment, especially in order to deal with the new features added to the IF language such as dynamic creation of processes (see deliverable D10).

Moreover, improved algorithms for state-space exploration and symbolic reachability analysis have been implemented in TReX and LASH tools (see deliverable D6).

2 WP2: Integration of tools and Experimentations

This year, our project has devoted an important effort to the work related to WP2. The work made on the integration of tools consists of the following points:

- Definition of an extension of the IF language in order to cover parametrized networks of processes, and the implementation of this extension in the IF environment,
- Definition and implementation of a common interface which allows to invoke all different tools developed in the project on models described in the common language IF,

- Connection of the tools to the IF environment through filters which allow to produce (using slicing techniques) an abstract model in the fragment of IF which can be handled by the corresponding tool.

Moreover, a first work on the integration of different packages of symbolic representation has been carried out. This work concerns the use of the NDD package (of Liège) in the TReX tool (of Paris).

As for experimentations, the work carried out this year consists of:

- A deep study and big modeling effort of the case studies (PGM protocol proposed by France-Telecom and the AXD system proposed by Ericsson),
- Parametric analysis and verification of the PGM protocol,
- Parametric verification of the TTP/C algorithm (used in automotive industry).

2.1 Task T2.1: Common language and interface

An extension of the IF language has been defined by Paris in order to cover the case of parametrized networks of processes with different topologies: sequences, trees, hypercubes, and their combinations (see deliverable D11). Parametrized networks are modeled using automata with arrays having parametric bounds. Intuitively, each cell of an array contains the state of one of the processes in the network. The indices of the array cells can be vectors of integers or strings over a finite alphabet (strings are used in the case of tree-like topologies). The operations on the arrays can be guarded by formulas involving quantification over indices, which allow to express global conditions on the values in the arrays (states of all processes in the network). The new features needed for the extension proposed by Paris have been implemented in the IF environment by Grenoble (see deliverable D11).

Moreover, a common (graphical) interface has been implemented by Grenoble allowing to invoke the tools developed in the project (IF-tools, LASH, TReX, Uppsala RMC, TLV, MC Kit) on models described in the common language IF.

2.2 Task T2.2: Integration of tools

2.2.1 Connecting tools to a common environment

The tools IF-tools, LASH, TReX, Uppsala RMC, TLV, MC Kit have been integrated to a common specification and verification environment. This environment has a common specification language which is IF, and a common interface which allows to invoke each of these tools. The integration of the tools has been carried out (by their developers) through filters which use the slicing tools of the IF-tools in order to produce models in the fragments of IF which can be handled by their corresponding tools (see deliverable D9).

2.2.2 Combining representation packages

Another aspect of tool integration has also been considered. Indeed, the team of Paris has implemented a version of parametric DBMs (the structures used in TReX) which uses the NDDs of LASH to represent linear constraints on parameters. The obtained implementation has been compared to the previous one which uses a logic-based representation of the constraints. The experimentations show the complementarity of these two styles of representations and suggest the development of an integrated version of LASH and TReX which uses both kinds of representations, an automata-based one as well as a logic (or polyhedra)-based one.

2.3 Task T2.3: Experimentations and case studies

2.3.1 PGM protocol

The PGM protocol have been deeply studied by all groups of the project. An important effort has been devoted to building formal models of the protocol. This effort allowed to clarify a lot of ambiguities concerning the specification of the protocol, and to write a general and clean automata-based model in the IF language.

The verification of the protocol has focused on the following questions:

1. is the protocol *faithful*, i.e., it is deadlock free, and for each sent message, each receiver should either deliver the message itself or a notification of definite loss,

2. under which conditions the protocol is *reliable*, i.e., all messages are transmitted without any definite loss (lost messages can be recovered).

These questions have been addressed by considering different parametric and infinity aspects. Mainly, the protocol has been shown to be faithful for (1) an arbitrary number of components receivers and intermediary nodes (on a time-abstracted version of the protocol), and for (2) an infinite number of sent messages (on a model restricted to one sender, one intermediary node, and one receiver). Moreover, generic constraints on the parameters of the protocol (size of the sender and receiver windows, speed of the source of sent messages, delays of propagation, rate of losses, etc) under which the protocol is reliable have been derived (and proved manually), and automatically checked using the IF-tools (see deliverable D12-1).

2.3.2 Systems with shared resources

In the AXD system, an unbounded number of clients must share resources under the control of lockers who can apply various kinds of policies (e.g., FIFO, FIFO with priorities, etc) to manage the attribution of these resources. Essential properties to show are deadlock freeness, mutual exclusion, starvation freeness, etc. The problem here is to be able to verify such properties for an arbitrary number of clients and different kinds of locker policies.

The study of the AXD system lead to the definition of a general theoretical framework for the verification of parametrized systems with shared resources. Several interesting and nontrivial decidability results have been obtained for such systems and wide classes of (safety and liveness) properties including all properties of interest mentioned above. These results establish cut-off bounds for the parametric verification of such properties and reduce this verification to finite-state model checking problems (see deliverable D12-2).

2.3.3 TTP/C group membership algorithm

The group membership algorithm is a very subtle mechanism which allows to stations which communicate (periodically) through a ring network to realize that they are faulty, to withdraw themselves in such a case from the group, and to join later the group. The main problem is to avoid that the emitting stations form two (or more) disconnected groups (so-called clique avoidance problem).

The verification of this algorithm is a real challenge since it has a very complex and nontrivial behavior which involves many parameterization and infinity sources.

We have been able to reduce, through a nontrivial abstraction, the verification problem of this algorithm with an arbitrary number of station to the reachability problem of counter automata. Using this approach, we have been able to verify automatically that if a fault occurs, the algorithm stabilizes after two rounds in a correct state (i.e., with only one group of emitting stations). For that, we used the tools ALV (developed at Santa Barbara university) using polyhedra-based representations of arithmetical constraints as well as LASH which uses an automata-based representations of such constraints (see [BM02]).

3 WP3: Project management / assessment

3.1 Task T3.1: Management and coordination

See the Executive Summary (Part A of the report).

3.2 Task T3.2: Assessment and quality control

3.2.1 Self-assessment criteria

We consider the self-assessment criteria defined in the technical annex describing the work in the project.

1. **Production of theoretical and applied results on methodology and validation techniques:** The evaluation criteria used for this work are the technical level and its scientific value, (selectivity of the conferences and journals), the relevance to the project and the proved and/or potential wide range of applications, and the quality of the dissemination (scientific quality and level of diffusion of the conferences and journals, variety of addressed communities). We consider that a minimal contribution of each (academic) participant should be at least one high quality publication per year.

2. **Construction of validation tools and their integration:** We consider as minimal success criterion the production of a validation environment where validation techniques for *infinite-state extended automata* (e.g. automata with counters, clocks, and FIFO-channels) are available. Further steps we intend to reach consists in the extension of this environment in order to deal also with (1) *parametrized systems* (e.g., networks with unbounded number of components, parametric timed automata, etc), and (2) *dynamic networks* and *mobile processes*. Those are highly exploratory (and hence risky) steps. We will consider as a success any progress toward the development of new methods, techniques, and prototype tools for dealing with these kinds of systems.

3. **Applications and experimentations:** We consider as minimal success criterion showing the application of our techniques and tools to nontrivial cases. The main success criterion we consider here is the power of our methods, techniques, and tools from the *qualitative* point of view: showing that our approach allows to tackle in a systematic way classes of systems that are, by nature, beyond the scope of the existing approaches. The second criterion we consider is the power of our techniques from the *quantitative* point of view: showing that our techniques allow to deal with cases of important sizes. The main objective of our work on case studies is indeed to experiment and to evaluate our techniques and tools according to the criteria above.

3.2.2 Achievements in the second period of the project

Results on methodology and validation techniques: The research has produced both theoretical and applied works with high technical level and scientific value. The main results obtained this year concern (1) new general methods and techniques for the verification of parametrized networks of processes, (2) decidability results for parametrized networks of systems with shared resources, (3) decidability results for parametrized networks with tree-like topologies, (3) new algorithms for the verification of extended automata, especially of pushdown automata, (4) methods and techniques for the verification of cryptographic protocols, multi-threaded programs, and concurrent recursive programs.

We consider that the quality of the obtained results is very satisfactory, and that our objectives for this year concerning the production of verification methods and algorithms were realized. This work has also raised many open questions which will be considered in the next period of the project.

Construction of validation tools and their integration: We have implemented new abstraction and verification techniques for extended automata and parametrized networks of processes. Moreover, we have connected all the tools developed in the project to the IF environment through a new version of the common language IF which has been extended in order to cover parametrized networks of processes. We consider that this work is quite satisfactory, and constitutes an important step toward our objective of building a verification environment offering various abstraction and verification tools.

We have also considered the integration of different symbolic representation packages. The NDD package has been used in TReX in combination of parametric DBMs. This work lead to several interesting research questions and to ideas for improving both tools LASH and TReX.

Applications and experimentations: An important and massive effort has been devoted to the study of different cases of complex systems and to experimental verification work. We consider that this effort has been successful and that the obtained results by the partners in the project are very satisfactory. We have indeed addressed nontrivial verification problems of industrial case studies and we have been able to provide novel and general solutions to these problems. We have also been able to use our tools to solve automatically these problems, taking into account, for the first time, various kinds of parameterization and infinity sources.

Moreover, these experimental works were very useful since they lead to a better understanding of the limitations of our tools, and to new ideas on how to improve them. Actually, the tools such as IF-tools, TReX and LASH have already taken advantage this year of the feedback from the experimental work.

3.2.3 Goals for the third period of the project

Results on methodology and validation techniques: We will continue the work on developing verification algorithms and methods for parametrized networks. In particular, we will consider the extension of regular model checking techniques to more general graph structures. The verification of liveness properties of parametrized networks will also be continued by considering various approaches (automatic generation of fairness constraints, probabilistic reasoning, etc).

We will also consider the verification problem of complex quantitative properties such as those which appear in the analysis of systems like the PGM protocol. A typical example of such properties relates the number of definitively loss (unrecoverable) messages with the rate of losses in the channels, the propagation delays, sizes of windows, and the speed of the source of sent messages.

Dynamic systems will also be an important subject of investigation. We will address in particular the specific problems which appear in the analysis of such systems: complex structure of the memory, communication between threads, etc.

Construction of validation tools and their integration: The work on case studies, especially the PGM and the TTP/C, have shown some limitations of our tools that we plan to overcome.

The improvements and extensions we are planing include:

- the improvement of the capabilities of our tools. Indeed, we have observed on our experimentations that in their current versions, tools such as TReX and LASH can hardly deal with models with more than 20 counters or clocks. Our work will address the design of more efficient representation structures, and finer state-space exploration techniques.
- a deeper integration of representation packages used in LASH and TReX. In particular, we think that it should be very useful to have mixed representations based on both automata and logic (or polyhedra).
- the improvement of the interfaces between tools and the design of automatic strategies for combining abstraction, model-checking and reachability analysis tools. The current version of our environment allows already to invoke different tools using a common model description language. We intend to improve this environment by allowing the user to invoke a sequence of tools depending on the nature of his model.
- the consideration of more general classes of models. For instance, the verification of complex quantitative properties, it could be necessary to consider models combining counting, timing, and probabilistic aspects.

Applications and experimentations: We will continue our work on the two case studies provided by France-Telecom and Ericsson.

The work on PGM will mainly concern

- the automatic synthesis of the parametric constraints for the reliability property. Indeed, the work which has been done concerning this parameterization aspects was to derive this constraint (manually) and to verify it by adopting an experimental approach which consists in checking automatically (using model-checking tools) that the property holds for various (significant) values of the parameters. We intend to automatize this experimental phase by considering more and more unfixed values of the parameters, and using tools such as LASH and TReX to synthesis automatically the constraints on these parameters ensuring the correctness of the protocol.
- the consideration of complex quantitative analysis of the protocol, e.g., (1) the verification of the reliability up to some percentage and (2) synthesis of parameter values (timing, loss rates, etc) in order to minimize the number of control messages per data messages.

As for the AXD systems of Ericsson, our work will address mainly:

- the definition of better cut-off bounds, for each locker policy, for the parametric verification of properties of interest, for an arbitrary number of clients,
- the definition of methods and algorithms for the verification that a given locker implements correctly some given policy, regardless of the number of the clients and their behavior,
- apply these results for performing automatic experiments on the verification of the AXD system.

3.3 Task T3.3: Dissemination and transfer activities

3.3.1 Second period of the project

Publications: We have published and/or presented our results in major conferences related to the topics of the project: CAV, TACAS, SAS, CONCUR, ICATPN, FTRTFT, LPAR, Infinity, VMCAI, and POPL. These conferences target several different research communities such as computer-aided verification, concurrency theory, abstract interpretation, static analysis, software validation, programming languages, etc. All the articles produced in our project can be found at:

`http://www.liafa.jussieu.fr/~haberm/ADVANCE/main.html`

Workshop and conference organization:

- The European Joint Conferences on Theory and Practice of Software (ETAPS) has been organized by VERIMAG (**UJF**) in Grenoble, April 2002. ETAPS is the primary European forum for academic and industrial researchers working on topics relating to Software Science. It consists of a confederation of several conferences (TACAS, ESOP, FASE, FOSSACS, and CC), and it also includes many satellite workshops on several specialized and emerging topics.
- The second International Workshop on Embedded Software (EMSOFT) has been organized by VERIMAG (**UJF**) in Grenoble, October 2002.

Relations with industrial partners and observers:

- The partner **F-Telecom** (Yves-Marie Quemener) has continuously contributed to the work of the project by guiding and evaluating the work on the case study of the PGM protocol. His participations to the working meetings was very fruitful.
- The observer **Ericsson-UAB** (Thomas Arts) has actively participated to the project meetings and has guided the work on the AXD take-over manager case study. His help for understanding the case study and his advice toward the relevant aspects to address were crucial and very fruitful.

3.3.2 Plan for the third period of the project

- We will continue our publication activity by targetting different communities and wide-audiance conferences and symposia. In particular, we have carried out this year several important works which has not yet been published, e.g., parametric verification of the PGM, general decidability results for networks with tree-like topologies, and networks of processes with shared resources, verification techniques for security protocols, etc.
- We plan the organization of a meeting with industrials around the PGM and its automatic verification. This workshop will be organized with France-Telecom (next spring), in order to present the approaches and tools developed in our project, and the results we have obtained to industrials interested in the PGM protocol, especially members of the France-Telecom departments involved in the development of this protocol.

- We will pursue our collaboration with France-Telecom and Ericsson on the case studies of PGM and AXD301. France-Telecom is also intending to develop methods and tools for symbolic test case generation. This will be an opportunity to intensify our scientific cooperation with France-Telecom, and our technology transfer activities,
- We intend the organization of a workshop at the end of our project where leading researchers in the domain of computer-aided verification, software verification, and program analysis will be invited to participate.
- We also intend to participate actively to the animation of the research activity in our scientific community par organizing meetings, workshops, and schools around topics closely related to the work in our project: verification of infinite-state systems, real-time systems, software verification, verification of embedded systems, etc.

References

[AJMd02] Parosh Aziz Abdulla, Bengt Jonsson, Pritha Mahata, and Julien d'Orso. Regular tree model checking. In *Proc. 14^{th} Int. Conf. on Computer Aided Verification (CAV'02)*, volume 2404 of *Lecture Notes in Computer Science*, 2002.

[AJNd02] Parosh Aziz Abdulla, Bengt Jonsson, Marcus Nilsson, and Julien d'Orso. Regular model checking made simple and efficient. In *Proc. 13^{th} Int. Conf. on Concurrency Theory (CONCUR'02)*, volume 2421 of *Lecture Notes in Computer Science*, 2002.

[AMN02] P. A. Abdulla, P. Mahata, and A. Nylén. Downward closed language generators. Technical report, Uppsala University, Sweden, 2002.

[AN02] P. A. Abdulla and A. Nylén. Undecidability of ltl for timed petri nets. In *Proceedings of 4th International Workshop on Verification of Infinite-State Systems*, 2002.

[BET03] Ahmed Bouajjani, Javier Esparza, and Tayssir Touili. A Generic Approach to the Static Analysis of Concurrent Programs with Procedures. In *30th Symp. on Principles of Programming Languages (POPL'03)*, New Orleans, USA, January 2003. ACM.

[BGM02] M. Bozga, S. Graf, and L. Mounier. IF-2.0: A Validation Environment for Component-Based Real-Time Systems. In K.G. Larsen Ed Brinksma, editor, *Proceedings of CAV'02 (Copenhagen, Denmark)*, volume 2404 of *LNCS*, pages 343–348. Springer, July 2002.

[BLP02] L. Bozga, Y. Lakhnech, and M. Périn. Pattern-based Abstraction for Verifying Secrecy in Protocols. Technical report, Verimag, University of Grenoble, 2002.

[BM02] Ahmed Bouajjani and Agathe Merceron. Parametric Verification of a Group Membership Algorithm. In *7th Intern. Symp. on Formal Techniques in Real-Time and Fault Tolerant Systems (FTRTFT'02)*, Oldenburg, Germany, September 2002. Lecture Notes in Computer Science, Springer.

[BT02] Ahmed Bouajjani and Tayssir Touili. Extrapolating Tree Transformations. In *Proc. 14^{th} Int. Conf. on Computer Aided Verification (CAV'02)*, volume 2404 of *Lecture Notes in Computer Science*, 2002.

[KPSZ02] Y. Kesten, A. Pnueli, E. Shahar, and L. Zuck. Network invariants in action. In *Proc. 13^{th} Int. Conf. on Concurrency Theory (CONCUR'02)*, volume 2421 of *Lecture Notes in Computer Science*, 2002.

[KPV02a] O. Kupferman, N. Piterman, and M. Vardi. Model checking linear properties of prefix-recognizable systems. In *Proceedings of 14th International Conference on Computer-Aided Verification*, 2002.

[KPV02b] O. Kupferman, N. Piterman, and M. Vardi. Pushdown specifications. In *Proceedings of 9th International Conference on Logic for Programming Artificial Inteligence and Reasoning*, 2002.

[PXZ02] A. Pnueli, J. Xu, and L. Zuck. Liveness with (0,1,infinity)-counter abstraction. In *Proc. 14^{th} Int. Conf. on Computer Aided Verification (CAV'02)*, volume 2404 of *Lecture Notes in Computer Science*, 2002.

[ZPK03] L. Zuck, A. Pnueli, and Y. Kesten. Automatic verification of free choice. In *3th Intern. Workshop on Verification, Model Checking, and Abstract Interpretation*, Lecture Notes in Computer Science, 2003.