PROFUNDIS:
Periodic Progress Report Year 1
Part B

January 8, 2003

This is Part B of the PROFUNDIS Periodic Progress Report Year 1 where we explain in more detail the technical achievements of each work package.

Contents

1 WP1: Models

1.1 Overview

1.1.1 Objectives

The goal of this workpackage is to develop a comprehensive automata-like model that supports effective techniques to specify and verify properties of network applications.

The research activities are centered around three tasks:

- Task 1.1 Automata with operations and substitutions (Participants: Uppsala, Lisbon, Pisa)
- Task 1.2 Proof Techniques (Participants: Uppsala, Lisbon, Inria, Pisa)
- Task 1.3. Prototype and case studies (Participants Uppsala, Pisa)

The activities on Task 1.1. are expected to be completed by the end of the second year, while those on Tasks 1.2. and 1.3. will continue until the end of the project.

1.1.2 Summary of Scientific Achievements

Theoretical results have been established about HD automata modeled as coalgebras. Also coalgebraic models (not necessarily finite state) of mobile calculi have been defined, equipped with operations of parallel composition and restriction. Symbolic verification techniques have been developed and applied to security protocols. In particular, the control reachability problem has been studied. The theoretical results have provided the firm foundations needed for the experimental development, and they have driven the design and the prototype implementation. Tool prototypes have been developed for HD automata minimization wrt. bisimilarity. Verification toolkits exploiting symbolic techniques have been designed, implemented and tested. Also, the distributed infrastructure of the PROFUNDIS Verification Environment has been designed and a preliminary implementation has been developed.

1.2 Scientific Achievements

Tasks consist of several themes and in all cases the research activities have advanced well on several of them, providing already some results in terms of publications. Hereafter we will briefly summarize the results of the research activities of the first year.

1.2.1 Task 1.1: Automata with operations and substitutions

Automata and Coalgebras Pisa has developed a formulation of HD automata as coalgebras on a category of *named sets* and *named functions*. Set

elements are equipped with names which are defined up to specific groups of name permutations called *symmetries* [16]. This work exibits two main contributions. First, it provides a coalgebraic definition, which implies the existence of representatives, minimal up to bisimilarity. In previous versions of HD automata, not equipped with symmetries, bisimilarity was characterized in terms of spans of open maps, but no minimal realization was guaranteed to exist. Second, this work formally specifies a declarative procedure to perform effective finite state verification via semantic minimization. Indeed, a toolkit performing state minimization of labelled transition systems for name passing calculi has been implemented [17]. The software architecture of the toolkit is derived directly from the co-algebraic formulation of the partition-refinement minimization algorithm. The direct correspondence between the semantical structures and the implementation structures facilitates the proof of correctness of the implementation.

Automata, Coalgebras and Operations Instead than coalgebras on a category of named sets, HD automata can be defined as coalgebras on a category of algebras (bialgebras). The construction automatically guarantees that bisimilarity is a congruence wrt. the operations of the algebras. However, this view of HD automata gives up finiteness in all except the most trivial cases. In previous work, an algebra was defined, whose carrier consisted of π-calculus agents equipped with name permutations only. In [8], the algebra has been further extended with the operations of parallel composition and restriction. Bisimilarity there corresponds to early observational equivalence, which in fact is a congruence with respect to parallel composition and restriction, but not with respect to prefix. In the same paper a rather general theorem is proved, which gives sufficient conditions for performing the bialgebraic construction in the presence of structural axioms.

Automata, Coalgebras and Spatial Logics Lisbon [25] has developed a notion of automata in which the set of states has been endowed with a structure intended to describe the spatial organization of states in a broad sense. The innovative feature of the model is the treatment of space in coalgebraic terms, the main reason being that for the applications we have in mind we need to observe the structure of given states rather then construct new ones. Indeed, we are looking for general models for spatial logic, where typically we wish to state that if a state has a certain structure, then it satisfies some properties. An additional advantage is that we have a uniform treatment of space and time, since the dynamics of transition systems is naturally described in coalgebraic terms.

1.2.2 Task 1.2: Proof Techniques

Symbolic Verification Techniques INRIA [2] has developed a *symbolic* reduction system for cryptographic protocols where properties such as secrecy or authenticity are specified by inserting *logical assertions* in the processes. The

symbolic reduction system provides a flexible decision procedure for *finite* processes and a reference for sound implementations (the corresponding work on the prototype implementation is commented in the next section). The symbolic reduction system can be regarded as a variant of syntactic unification which is compatible with certain set-membership constraints. For a large class of cryptographic protocols (e.g. the so called ping-pong protocols) a *dag* implementation of the symbolic reduction system leads to an algorithm running in NPTIME, thus matching the lower bound of the problem.

On a related line of work, Pisa [5] has developed a symbolic operational semantics for the spi-calculus that relies on unification and provides finite and effective models of cryptographic protocols. A method to carry out trace analysis directly on the symbolic model has been also introduced and proved complete (under certain conditions on the cryptographic primitives). A tool named STA (Symbolic Trace Analyzer) [6] has been developed, which implements symbolic execution.

Reachability and Security Protocols INRIA [3] has addressed the issue of decidability of the control reachability problem for various fragments of the asynchronous π-calculus in terms of *name generation*, *name mobility*, and *unbounded control*. It has been proved that the combination of name generation with either name mobility or unbounded control leads to an undecidable fragment. On the other hand, name generation with *unique receiver* and *bounded* input (a condition weaker than bounded control) is decidable by reduction to the coverability problem for Petri Nets. The control reachability problem has been also studied in the context of the Dolev-Yao model of cryptographic protocols [1]. A main result is the characterization of a new decidable class of cryptographic protocols with a complexity ranging from simple to double exponential.

1.2.3 Task 1.3. Prototype and case studies

The PROFUNDIS WEB The distinguished and innovative feature of the PROFUNDIS Verification environment, called *PROFUNDIS WEB*, is the idea of viewing the environment as a distributed infrastructure exploited as a *service distributor*. By service we do not mean a monolithic stand-alone verification toolkit, but rather a component available over the WEB that others might use to develop additional services. In the PROFUNDIS WEB each verification toolkit has an interface which is network accessible through standard network protocols and which describes the interaction capabilities of the verification toolkit. Hence, verification sessions over the PROFUNDIS WEB are developed by combining and integrating together the services available over the WEB. As a consequence, the PROFUNDIS WEB is highly portable (it may adapt to a variety of infrastructures) and supports interoperability and dynamic reconfiguration. Moreover, it supports the dynamic integration of several verification techniques. The design of the architecture of the PROFUNDIS WEB has been the result of an active collaboration among the research teams in PROFUNDIS, with the leading role played by Pisa and Uppsala. To illustrate the effective-

ness of the approach a prototype implementation of the PROFUNDIS WEB has been developed by Pisa [13]. This programming experiment has given us the opportunity of testing effectively advantages and weakness of the proposed approach at the very beginning of the project.

Verification Toolkits Several verification toolkits have been implemented in the first year of the project. INRIA [32, 31] has developed a verifier for cryptographic protocols called TRUST. The TRUST toolkit relies on an exact symbolic reduction method, combined with several techniques aiming at reducing the number of interleaving that have to be considered. Authentication and secrecy properties are specified in a very natural way, and whenever an error is found an intruder attacking the protocol is given.

Pisa [17, 26] has developed a toolkit performing state minimization of HD automata. The software architecture of the toolkit is directly suggested by the abstract, declarative, co-algebraic formulation of the partition-refinement minimization algorithm given in [16]. The direct correspondence between the semantical structures and the implementation structures has facilitated the proof of correctness of the implementation. The usefulness of the minimization toolkit has been shown in practice by performing finite state verification of π-calculus specifications.

During previous research efforts, Pisa has also developed a model checker called HAL to verify the satisfiability of properties of π-calculus specifications expressed by a suitable modal logic [12]. The construction of the model checker takes direct advantage of the finite representation of π-calculus in terms of HD-automata. In particular, the construction of the formula expressing properties of specifications is driven by the finite state representation of the system (i.e. the π-calculus process) to be verified. The version of HD-automata considered in this work is without symmetries, and thus it cannot be minimized directly. However under certain conditions, and at the expense of a possibly large increase of the number of states, HD automata can be translated into ordinary automata still preserving bisimilarity, and minimized as such. Both the minimization toolkit and the model checker have been integrated in the PROFUNDIS WEB [13].

Finally, a tool named STA (Symbolic Trace Analyzer) [6] has been developed in collaboration between Pisa (PROFUNDIS) and Firenze (Mikado), which implements symbolic execution of cryptographic protocols. A successful attack is reported in the form of an execution trace that violates the specified property. STA is written in ML. Currently, shared-key, public-key cryptography and hashing are supported.

1.3 Discussion

The tasks have proceeded in most cases according to plans in the TA. Thus the work of the first year for WP1 can be considered successful for all tasks. We comment below about each task separately.

Task 1.1: Automata with operations and substitutions In addition to the general goal of modeling HD-automata as coalgebras, the plans included: i) automata with name fusions; ii) automata with operations; iii) automata with substitutions; and iv) models for spatial logics. Progress was made mainly about the general goal [16] and about automata with operations [8]. About i) and iv), work in progress can be claimed in collaboration between Lisbon [25] and Pisa and between Pisa and Uppsala. Activity about iii) is in its intial stages in Pisa.

Task 1.2: Proof techniques The plans included: i) symbolic execution; ii) verification via proofs of semantic equivalence, including reachability problem for cryptographic protocols; iii) proving spatial properties; and iv) co-inductive techniques. Activity has focussed on symbolic execution [2, 5] and reachability [3, 1]. Some progress about iii) and iv) has also been made in connection with work mentioned above about respectively models for spatial logics and HD-automata as coalgebras.

Task 1.3: Prototype and case studies The plans included: i) verification environment and tool development; and ii) case studies. Remarkable progress can be claimed about environment architecture (PROFUNDIS WEB), which was agreed upon among the partners and which was implemented experimentally in [13]. Also tool development was rather successful, both in Pisa with MIHDA [17, 26], STA [6] and HAL [12] and in Sophia with TRUST[32, 31]. Case studies have been considered for exercising and benchmarking tool capabilities, for instance the GSM handover protocol for HAL and MIHDA.

Miscellaneous The paper on model checking [12] is the continuation of work initiated before PROFUNDIS. The present paper, however, contains refinements and new technical developments of the earlier works which justify the inclusion in the results of PROFUNDIS.

1.4 Future Work

The activity will proceed in according to what planned in the technical annex.

1.5 WP1 Publications

Papers accepted for publication in international journals

1. R. Amadio, D. Lugiez, and V. Vanackère. On the symbolic reduction of processes with cryptographic functions. *Theoretical Computer Science*, To appear.

2. R. Amadio and C. Meyssonnier. On decidability of the control reachability problem in the asynchronous pi-calculus. *Nordic Journal of Computing*, 9(2):70–101, 2002.

Papers under revision for publication in international journals

1. G. Ferrari, S. Gnesi, U. Montanari, and M. Pistore. A model checking verification environment for mobile processes. *Submitted to ACM TOSEM (under revision)*, 2002.

Papers accepted for publication in international conference and workshops

1. R. Amadio and W. Charatonik. On name generation and set-based analysis in Dolev-Yao model (extended abstract). In *Proc. CONCUR'02*, volume 2421 of *LNCS*. Springer Verlag, 2002.

2. M. Boreale and M. Buscemi. A framework for the analysis of security protocols. In *Proc. CONCUR'02*, volume 2421 of *LNCS*. Springer Verlag, 2002.

3. M. Buscemi and U. Montanari. A first order coalgebraic model of pi-calculus early observational equivalence. In *Proc. CONCUR'02*, volume 2421 of *LNCS*. Springer Verlag, 2002.

4. G. Ferrari, U. Montanari, and M. Pistore. Minimizing transition systems for name-passing calculi: A co-algbraic formulation. In *Proc. FOSSACS'02*, volume 2303 of *LNCS*. Springer Verlag, 2002.

5. V. Vanackère. The trust protocol analyser, automatic and efficient verification of cryptographic protocols. In *Verification Workshop - Verify02*, 2002.

On-line prototypes (downloadable) and user manuals

1. R. Raggi and E. Tuosto. *HD-Reducer (Online version)*. Dipartimento di Informatica, Università di Pisa, http://jordie.di.unipi.it:8080/mihda, 2002.

2. M. Boreale and M. Buscemi. *STA, a Tool for the Analysis of Cryptographic Protocols (Online version)*, Dipartimento di Sistemi ed Informatica, Università di Firenze, and Dipartimento di Informatica, Università di Pisa, http://www.dsi.unifi.it/ boreale/tool.html.

3. V. Vanackère. *The TRUST protocol analyser*. Lab. Informatique de Marseille, http://www.cmi.univ-mrs.fr/∼vvanacke/trust.html, 2002.

Technical reports

1. M. Buscemi and U. Montanari. π-calculus early observational equivalence: a first order coalgebraic model. Technical Report TR-02-14, Dipartimento di Informatica, Università di Pisa, 2002.

2. G. Ferrari, S. Gnesi, U. Montanari, R. Raggi, G. Trentanni, and E. Tuosto. Verification on the web. Technical Report TR-02-18, Dipartimento di Informatica, Università di Pisa, 2002.

3. G. Ferrari, U. Montanari, R. Raggi, and E. Tuosto. From coalgebraic specification to toolkit development. Technical Report TR-02-19, Technical Report, Dipartimento di Informatica Universita' di Pisa, 2002.

Drafts

1. L. Monteiro. Transition systems with spatial structures: A coalgebraic framework. Manuscript, 2002.

2 WP2: Specifications

2.1 Overview

2.1.1 Objectives

The objectives of this workpackage are to develop new logics to support the specification and verification of structural (spatial) and behavioural properties of concurrent mobile systems, and to develop proof systems for these logics based on sequent calculi.

The workpackage comprises three tasks:

- Task 2.1 Logics for systems with spatial and temporal structure (Participants: Lisbon, INRIA, Pisa)
- Task 2.2 Expressiveness (Participants: Uppsala, Lisbon, Inria, Pisa)
- Task 2.3. Tools and case studies (Participants Uppsala, Lisbon, Pisa)

The activities on Task 2.1. are expected to be completed by the end of the second year, and those on Tasks 2.2. and 2.3. will start in the second year and will continue until the end of the project.

2.1.2 Summary of Scientific Achievements

The general contributions of the work developed in WP2 in Year 1 are:

- Spatial logics for pi-calculi, that can describe not only behavioural properties, but also other key features of modern distributed systems (*e.g.,* resource control, distribution, and secrecy).
- Study of fundamental meta-theoretic properties of spatial logics for ambient and pi-calculi (*e.g.,* expressiveness, separation).
- Logics for semi-structured data and related decision procedures.
- Logical formalization of the secure composition of web services.

2.2 Scientific Achievements

The tasks comprise several themes, not all expected to start in the first year. The central task for Year 1 was Task 2.1, but results in other tasks have also been obtained.

2.2.1 Task 2.1: Logics for systems with spatial and temporal structure

This task concerns the development of logics to specify and verify, in an integrated way, both behavioural and structural properties of concurrent systems based on mobile process calculi.

Base logics In collaboration with Cardelli at Microsoft Research Cambridge, Lisbon introduced a spatial logic to describe and verify properties of concurrent systems specified in the pi-calculus [10]. This logic allows the specification of spatial and behavioural properties by induction and co-induction (for instance, properties related to resource usage), and includes freshness and hidden name quantifiers (important to define secrecy and non-interference properties). A related logic and model-checker for the pi-calculus with recursion, based on a small set of structural behavioural and spatial observations, is currently under development in Lisbon [9].

The introduction of spatial logics as a foundation for query and programming languages for semistructured data (XML-like) has also been advocated recently; Inria (through the subsite LIF, Marseille), introduced new logics TL and SL to describe and query semi-structured data represented by multitrees [22, 23], and can embed XML Schema as a plain subset.

Pisa has applied a logic that combines modalities with a notion of type to formalize the secure composition of web services [7]. By its use of types, this work is also part of the WP3 deliverable.

Proof systems In collaboration with Cardelli at Microsoft Research Cambridge, Lisbon defined a sequent calculus based proof-system [11] for the spatial logic of [10]. This proof system combines good proof-theoretic properties (*e.g.*, cut-elimination) and direct applicability to concurrency, it was proposed for the asynchronous pi-calculus, but the general techniques adopted are easily generalizable to the case of other nominal calculi.

Inria (through the subsite LIF, Marseille), introduced decision procedures for satisfaction and model-checking the query languages logics TL and SL againts formal representations of semi-structured documents [22, 23], that also studies the complexity of the associated problems. This work builds upon new notions of tree- and sheaves-automata, which are tailored versions of automata for unranked trees with both associative and associative-commutative symbols.

Verification framework This topic was not expected to be substantially addressed during Year 1. Lisbon developed some preliminary work on the development of techniques of equational reasoning on pi-algebras, having in mind the development of specific techniques for (partially) mechanizing theorem proving in spatial logics.

2.2.2 Task 2.2: Expressiveness

This task focus on accessing the expressiveness of the base logics and in the identification of suitable high-level extensions. Although scheduled to start on the second year, some results have already been obtained in the first year.

Inria (through the subsite ENS, Lyon), have developed a work started by Sangiorgi on the expressive power of the Ambient Logic [20]. They established several expressiveness and separation results in the case where we also consider

in the underlying calculus the presence of the replication operator, that leads to possibly infinite behaviours and spatial decompositions.

Ongoing work also by the Lyon team currently studies the spatial logic for the pi-calculus developed in [10], trying to establish meta-theoretical properties about the expressive power of the spatial adjuncts, that, as the work of [20] on the Ambient Logic testifies, turns out to be quite powerful.

2.2.3 Task 2.3: Tools and case studies

This task concerns the implementation of verification tools for the logics, no significant results can be reported here at the present moment.

2.3 Discussion

The work in WP2 has progressed according to the schedule in the TA and the SEP, despite some difficulties encountered by Lisbon in recruiting personnel. We comment below about each task separately.

Task 2.1: Logics for systems with spatial and temporal structure We expected to have a first version of the syntax and semantics of the spatial logics as well as proof systems for spatial and temporal properties. These objectives have been attained. A spatial logic to describe and verify properties of concurrent systems specified in the pi-calculus has been proposed, including a sequent calculus based proof-system that combines good proof-theoretic properties (*e.g.*, cut-elimination) and direct applicability to concurrency. Other contributions include logics TL and SL to describe and query semi-structured data represented by multitrees, and a logical formalization of the secure composition of web services.

Task 2.2: Expressiveness Several expressiveness and separation results were established in the case where we also consider in the underlying calculus the presence of the replication operator, that leads to possibly infinite behaviours and spatial decompositions.

Task 2.3: Tools and case studies In the TA this task was scheduled to start in the second year only, but in the SEP it was stated that a first version of the prototype theorem-proving tool for the spatial logic might be available at the end of the first year. It turned out that this was not possible due to the difficulty experienced by Lisbon in hiring personnel. This delay is not serious and can be recovered in the second year.

2.4 Future Work

The activity will proceed according to what was planned in the technical annex.

2.5 WP2 Publications

Papers accepted for publication in international journals

1. L. Caires and L. Cardelli. A Spatial Logic for Concurrency (Part I). Accepted for publication in *Information and Computation*, 2002.

Papers presented at international conferences and workshops with publication in its proceedings

Note: 1. below is joint with work package 3.

1. A. Bracciali, A. Brogi, G. Ferrari and E. Tuosto. Security and dynamic compositions of open services. In *Proc. Int. Conference on Parallel and Distributed Processing Techniques and Applications (PDTA'02)*. CSREA Press, USA, 2002.

2. L. Caires and L. Cardelli. A Spatial Logic for Concurrency (Part II). In *CONCUR 2002: Concurrency Theory (13th International Conference)*, Lecture Notes in Computer Science. Springer-Verlag, 2002.

3. D. Hirschkoff, E. Lozes, and D. Sangiorgi. Separability, Expressiveness and Decidability in the Ambient Logic. In *17th Annual Symposium on Logic in Computer Science*, Copenhagen, Denmark, 2002. IEEE Computer Society.

Submitted papers

1. Denis Lugiez and Silvano Dal Zilio. XML Schema, Tree Logic and Sheaves Automata. Research report 4631, INRIA, November 2002 (submitted). http://www.inria.fr/rrrt/rr-4631.html.

Research reports

1. Luis Caires. Model-Checking of Spatial Properties in the pi-calculus. Research report 3, DI/FCT/UNL, December 2002.

2. L. Caires and L. Cardelli. A Spatial Logic for Concurrency (Part II). Technical Report 3/2002/DI/PLM/FCTUNL, DI/PLM FCT Universidade Nova de Lisboa, 2002.

3. Denis Lugiez and Silvano Dal Zilio. Multitrees Automata, Presburger's Constraints and Tree Logics. Research report 08-2002, LIF, Marseille, France, June 2002. http://www.lim.univ-mrs.fr/Rapports/08-2002-Lugiez-DalZilio.html.

3 WP3: Types

3.1 Overview

3.1.1 Objectives

The objectives are to develop new type systems to control interferences among processes and the resources used by the processes; to integrate the type techniques with operational and logic techniques; to investigate the robustness of the type techniques and their algorithmic definitions; to assess the applicability of the techniques by means of case studies, and to implement some of the type algorithms and proof techniques.

3.1.2 Summary of Scientific Achievements

New type systems have been introduced that, we think, significantly enlarge the collection of properties of mobile processes that can be handled with types. For other properties in which types alone seemed to be insufficient, we have developed techniques and models that combine ideas from types with ideas from other fields (modal logics, logical relations). The study of the impact of type systems on implementations, and the transfer to types developed in calculi of mobile processes to other languages, closer to high-level programming languages, has begun.

3.2 Scientific achievements

3.2.1 Task 3.1: interference and access control

In task is about the design of novel type systems for mobile code. The emphasis is on type systems that allows us to control interferences among processes and their access to resources.

This was the main task for Year 1 in WP3. A few strands of work have advanced well; some have already given public results (in form of papers). Following the description of WP3 in the TA, these strands are summarised below.

Classification of interference In a collaboration, Inria and Pisa [21] have obtained a formal and uniform definition of the classification of grave and plain interferences, and developed basic properties of them (these are forms of interference among processes that are specific of formalisms for mobile distributed systems like Ambients).

Access control Lisbon [24] has developed a (first version of a) type system for a distributed version of the pi-calculus that controls migration of processes and the resources on the host site that a migrated process can use.

On a related line of work, Pisa [14] has developed a model (based on Ambients) where the security guarantees, on the access to local resources and migration, is obtained by separate typed components (the *guardians*). Advantages of

the approach include: flexibility in the range of policies allowed, modularity in the design of complex systems.

Behavioural types Lisbon [27] has formalised types systems for mobile processes that express guarantees on the services that a process, or a concurrent object, can offer. Here types are highly dynamic: as processes evolve and change, so types can evolve and change too.

Resource allocation Inria [29] has studied the problem of controlling the number of processes that can migrate at the same time on a given site, on a version of the Safe Ambient calculus. For instance, a type can specify a number n for a given ambient, and then typing ensures that at any time this ambient will never host more than n subambients.

Inria [4] has also developed a type system that ensures "message-deliverability": that is, the fact that every emitted message has a chance of being received. Adopting this discipline requires a style of programming where resources are persistent. The basic properties of the resulting computational model (based on the π-calculus) have been studied.

Secure Composition of processes Pisa [15, 7] has studied the problem of safe composition of components in distributed systems. Precisely, the target of the study have been MetaKlaim (the distributed version of a coordination language equipped with staging mechanisms), and Web services. The work [15] has led to the design of new type systems. Relevant features of these type systems are: security policies can be dynamically enforced; trustness guarantees of wide area network applications are maintained whenever computations interoperate with potential untrusted components. In contrast, the work on web services [7] uses more standard types and uses modal logics, and is commented in WP2.

3.2.2 Task 3.2: Integration of types with operational and logic techniques

This task 3.2 is about integration: integration of types with other techniques (for instance, logical) integration of type systems; etc.

Operational and logical techniques Inria [28] has studied the problem of termination of mobile processes (the fact that a process never reaches a divergence, that is, a point in which an infinite sequence of internal steps can be produced). The termination of a non-trivial subset of the π-calculus has been proved using a combination of operational, logical techniques and of type systems.

To formalise the secure composition of web services, Pisa [7] has developped a model which uses constructs from modal logics and a (simple) notion of type. As here logic plays a more important role than types, this work is actually part

of the WP2 deliverable, but the combination of logics and types is also relevant for the theme of task 3.2 in WP3.

3.2.3 Task 3.3: Practicality and scalability

This task is about the scalability of the techniques based on types to practical programming languages, and the develpment of algorithms for the automatic verification.

Inria is working on the design of type systems that guarantee certain security properties (with emphasis on non-interference properties) on a sequential subset of the JVM. (This work is still at an early stage and has not yet led to public documents)

Lisbon [30, 18] has worked on the scalability of the idea of "session types", developped by a number of people in the π-calculus (Honda, Vasconcelos, Gay) to more advanced programming constructs. Two lines of work have been pursued. The first has led to a proposal for typing the behavior of objects in component models. The main motivation for this is that most component models, CORBA in particular, do not offer any support for expressing behavior al properties of objects beyond the "static" information provided by IDLs. The other line has focused on a language based on the lambda-calculus with side-effecting input/output operations and recursive functions. It is a step towards including session types in a concurrent imperative language (with references). The final aim is a type system to check the correctness of protocols.

3.2.4 Task 3.4: Case studies and tool development

This task is about development of implementations for the type systems and the techniques based on types, and to case studies.

INRIA has done some implementation work on non-standard type systems for guaranteeing security, and in particular confidentiality, of mobile code. Precisely, a type system for enforcing non-interference of concurrent programs, inspired from previous work by Boudol and Castellani, has been formalized and proved correct in the proof assistants Coq and Isabelle. (This work has not led to a paper yet.)

INRIA [19] (collaboration with Mikado and DARTS) has also investigated the correctness and the implementation of an abstract machines based on some of the type systems for interference control in task 3.1: One of the main objectives here was to show that the control of interferences guaranteed by types is also useful in implementations.

3.3 Discussion

The tasks have proceeded according to schedule. We comment below the differences wrt the TA, which are minor, and – we believe – quite normal for a FET project.

Task 1

We do not have contributions on the header "declassification" that appeared in the TA. The problem of designing type systems for declassification appears to be much harder than expected. We have invited at the first PROFUNDIS meeting in Sophia an external speaker (F. Pottier) who has worked on this problem for sequential languages. We have learnt from Pottier's talk and subsequent questions that in concurrency we have probably not reached yet a point in which the problem can be successfully tackled.

On the other hand, the work names "access control" and "behavioural types" cover aspects that had not been announced in the TA.

Other tasks

The work of Tasks 3.3 and 3.4, which was supposed to start only during the second year, has been anticipated. As a consequence, there has been less time for certain aspects of Task 3.2 that appeared in the TA. An example is the theme named "spatial types", where some work has been done but has not produced outcome yet.

Miscellaneous

The papers [21, 19, 4] are the continuation of work initiated before PROFUNDIS, summary of which had appeared at conferences. The papers contains refinements and new technical developments (in some cases quite substantial) over the earlier work, which we believe fully justify the inclusion in the PROFUNDIS output and deliverable.

3.4 Future work

Taks 3.1 We consider the header "Classification of interference" as complete; no further work here is expected. We consider the work in the other headers as satisfactory, but improvements or enhancements of some of the results obtained this year will be pursued. However most of the effort is expected to go into exploring the robustness of such results obtained (in particular the be transfer to other calculi or richer languages).

Task 3.2 This task will continue, according to the TA.

Task 3.3 and 3.4 We intend to pursue the machine-checked verification of non-interference for a subset of Java. On the basis of this formalisation, a study will be conducted to assess if and how special purposes tools (tactics) can be developed to validate efficiently type systems for mobile code.

The work on the design of type systems for non-interference properties on a sequential subset of the JVM will continue. In this setting we will also investigate declassification, with the aim of establishing a correspondence between a

reference defensive virtual machine that performs verifications at run-time, and may allow for declassification, and an implementation offensive virtual machine that relies on the bytecode verifier to perform verifications statically.

In the medium term, these two directions might be merged through a study of a certifying compiler that would ensure that non-interfering source programs are compiled to non-interfering bytecode programs.

3.5 WP3 Publications

Papers accepted for publication in international journals

1. R. Amadio, G. Boudol and C. Lhoussaine On message deliverability and non-uniform receptivity. To appear in *Fundamenta Informaticae*

2. F. Levi and D. Sangiorgi. Mobile safe ambients. To appear in the journal *TOPLAS*.

Papers under revision for publication in international journals

1. G. Ferrari, E. Moggi, R. Pugliese MetaKlaim: A Type Safe Multi-stage Language for Global Computing Under revision for Mathematical Structure in Computer Science

Papers accepted for publication in international conferences and workshops

Note: 1. below is joint with work package 2.

1. A. Bracciali A. Brogi G. Ferrari and E. Tuosto Security and dynamic compositions of WEB Services In Proc. Int. Conference on Parallel and Distributed Processing Techniques and Applications (PDTA'02), CSREA Press, 2002.

2. G. Ferrari, E. Moggi, R. Pugliese. Guardians for Ambient Based Monitoring. In Proc. Foundations of Wide Area Network Programming, ENTCS 66 (3), 2002.

3. D. Teller, P. Zimmer, and D. Hirschkoff. Using Ambients to Control Resources. In *Proceedings of the 13th Int. Conf. in Concurrency Theory (CONCUR'02)*, volume 2421 of *LNCS*, pages 288–303. Springer Verlag, 2002.

4. D. Sangiorgi. Types, or: Where's the difference between CCS and π? In *Proc. CONCUR '02*, volume 2421, 2002. accompanying paper for an invited talk.

5. A. Vallecillo, V. T. Vasconcelos, and A. Ravara. Typing the behavior of objects and components using session types. In Antonio Brogi and Jean-Marie Jacquet, editors, *Electronic Notes in Theoretical Computer Science*,

volume 68. Elsevier Science Publishers, 2002. presented at FOCLASA'02 - 1st International Workshop on Foundations of Coordination Languages and Software Architectures.

Draft of papers and papers submitted

1. A. Ravara, P. Resende, and V. Vasconcelos. An algebra of behavioural types. Preprint, Section of Computer Science, Department of Mathematics, Instituto Superior Técnico, 1049-001 Lisboa, Portugal, 2002. Submitted for publication.

2. S. Gay, V. T. Vasconcelos, and A. Ravara. Session types for inter-process communication. Preprint, Department of Computer Science, University of Lisbon, Campo Grande, Edifcio C5, 1749-016 Lisboa, Portugal, 2002. Submitted for publication.

3. F. Martins and A. Ravara. Controling migration in lsdpi. Preprint, Section of Computer Science, Department of Mathematics, Instituto Superior Técnico, 1049-001 Lisboa, Portugal, 2002. In preparation.

4. P. Giannini, D. Sangiorgi, and A. Valente. A distributed abstract machine for Safe Ambients. Draft. 2002.

4 List of Publications

The following is a list of all PROFUNDIS publications year 1.

References

[1] R. Amadio and W. Charatonik. On name generation and set-based analysis in Dolev-Yao model (extended abstract). In *Proc. CONCUR'02*, volume 2421 of *LNCS*. Springer Verlag, 2002.

[2] R. Amadio, D. Lugiez, and V. Vanackere. On the symbolic reduction of processes with cryptographic functions. *Theoretical Computer Science*, To appear.

[3] R. Amadio and C. Meyssonnier. On decidability of the control reachability problem in the asynchronous pi-calculus. *Nordic Journal of Computing*, 9(2):70–101, 2002.

[4] R. M. Amadio, G. Boudol, and C. Lhoussaine. On message deliverability and non-uniform receptivity. Research report 05-2002, LIF, Marseille, France, May 2002. Accepted for pubblication in Fundamenta Informaticae.

[5] M. Boreale and M. Buscemi. A framework for the analysis of security protocols. In *Proc. CONCUR'02*, volume 2421 of *LNCS*. Springer Verlag, 2002.

[6] M. Boreale and M. Buscemi. *STA, a Tool for the Analysis of Cryptographic Protocols (Online version)*. Dipartimento di Sistemi ed Informatica, Università di Firenze, and Dipartimento di Informatica, Università di Pisa,, http://www.dsi.unifi.it/ boreale/tool.html, 2002.

[7] A. Bracciali, A. Brogi, G. Ferrari, and E. Tuosto. Security and dynamic composition of web services. In *Proc. Int. Conference on Parallel and Distributed Techniques and applications*. CSREA Press, USA, 2002.

[8] M. Buscemi and U. Montanari. A first order coalgebraic model of pi-calculus early observational equivalence. In *Proc. CONCUR'02*, volume 2421 of *LNCS*. Springer Verlag, 2002. Full version in Technical Report TR-02-14, Dipartimento di Informatica, Universitá di Pisa, August 2002.

[9] L. Caires. Model-checking of spatial properties in the pi-calculus. Research report 3, DI/FCT/UNL, December 2002.

[10] L. Caires and L. Cardelli. A spatial logic for concurrency (part i). *Information and Computation*, 2002. Accepted for publication.

[11] L. Caires and L. Cardelli. A spatial logic for concurrency (part ii). In *CONCUR 2002: Concurrency Theory (13th International Conference)*, Berlin, 2002. Lecture Notes in Computer Science. Springer-Verlag. Also as Technical Report 3/2002/DI/PLM/FCTUNL.

[12] G. Ferrari, S. Gnesi, U. Montanari, and M. Pistore. A model checking verification environment for mobile processes. *Submitted to ACM TOSEM (under revision)*, 2002.

[13] G. Ferrari, S. Gnesi, U. Montanari, R. Raggi, G. Trentanni, and E. Tuosto. Verification on the web. Technical Report TR-02-18, Dipartimento di Informatica, Università di Pisa, 2002.

[14] G. Ferrari, E. Moggi, and R. Pugliese. Guardians for ambient based monitoring. In *Proc. Foundations of Wide Area Network Programming*, ENTCS. Elseviers, 2002.

[15] G. Ferrari, E. Moggi, and R. Pugliese. Metaklaim: A type safe multistage language for global computing. Technical Report Under revision for Mathematical Structures in Computer Science, 2002.

[16] G. Ferrari, U. Montanari, and M. Pistore. Minimizing transition systems for name-passing calculi: A co-algbraic formulation. In *Proc. FOSSACS'02*, volume 2303 of *LNCS*. Springer Verlag, 2002.

[17] G. Ferrari, U. Montanari, R. Raggi, and E. Tuosto. From coalgebraic specification to toolkit development. Technical Report TR-02-19, Technical Report, Dipartimento di Informatica Universita' di Pisa, 2002.

[18] S. Gay, V. T. Vasconcelos, and A. Ravara. Session types for inter-process communication. Preprint, Department of Computer Science, University of Lisbon, Campo Grande, Edifcio C5, 1749-016 Lisboa, Portugal, 2002. Submitted for publication.

[19] P. Giannini, D. Sangiorgi, and A. Valente. A distributed abstract machine for Safe Ambients. Extended and refined version of a paper appeared in *Icalp.01*, 2002.

[20] D. Hirschkoff, E. Lozes, and D. Sangiorgi. Separability, Expressiveness and Decidability in the Ambient Logic. In *Proc. of LICS'02*, 2002.

[21] F. Levi and D. Sangiorgi. Mobile safe ambients. To appear in the TOPLAS journal. Extended and refined version of a paper appeared in *Proc. 27th POPL*, ACM Press, 2002.

[22] D. Lugiez and S. D. Zilio. Multitrees Automata, Presburger's Constraints and Tree Logics. Research report 08-2002, LIF, Marseille, France, June 2002. http://www.lim.univ-mrs.fr/Rapports/08-2002-Lugiez-DalZilio.html.

[23] D. Lugiez and S. D. Zilio. XML Schema, Tree Logic and Sheaves Automata. Research report 4631, INRIA, November 2002. http://www.inria.fr/rrrt/rr-4631.html.

[24] F. Martins and A. Ravara. Controling migration in lsdpi. Preprint, Section of Computer Science, Department of Mathematics, Instituto Superior Técnico, 1049-001 Lisboa, Portugal, 2002. In preparation.

[25] L. Monteiro. Transition systems with spatial structures: A coalgebraic framework. Manuscript, 2002.

[26] R. Raggi and E. Tuosto. *HD-Reducer (Online version)*. Dipartimento di Informatica, Universita' di Pisa, http://jordie.di.unipi.it:8080/mihda, 2002.

[27] A. Ravara, P. Resende, and V. Vasconcelos. An algebra of behavioural types. Preprint, Section of Computer Science, Department of Mathematics, Instituto Superior Técnico, 1049-001 Lisboa, Portugal, 2002. Submitted for publication.

[28] D. Sangiorgi. Types, or: Where's the difference between CCS and π? In *Proc. CONCUR '02*, volume 2421 of *LNCS*. Springer Verlag, 2002. accompanying paper for an invited talk.

[29] D. Teller, P. Zimmer, and D. Hirschkoff. Using Ambients to Control Resources. In *Proceedings of the 13th Int. Conf. in Concurrency Theory (CONCUR'02)*, volume 2421 of *LNCS*, pages 288–303. Springer Verlag, 2002.

[30] A. Vallecillo, V. T. Vasconcelos, and A. Ravara. Typing the behavior of objects and components using session types. In A. Brogi and J.-M. Jacquet, editors, *Electronic Notes in Theoretical Computer Science*, volume 68. Elsevier Science Publishers, 2002. presented at FOCLASA'02 - 1st International Workshop on Foundations of Coordination Languages and Software Architectures.

[31] V. Vanackère. *The TRUST protocol analyser*. Lab. Informatique de Marseille, http://www.cmi.univ-mrs.fr/∼vvanacke/trust.html, 2002.

[32] V. Vanackère. The TRUST protocol analyser, automatic and efficient verification of cryptographic protocols. In *Verification Workshop - Verify02*, 2002.