Project Number:	IST-2001-33100
Project Acronym:	PROFUNDIS
Title:	Proofs of Functionality for Mobile Distributed Systems

Deliverable N. :	2
Due Date :	MM 24
Delivery Date :	MM 24

Short Description:
Automata, Domains and Coalgebras for name passing calculi. Automata models and calculi with explicit fusions. Finite state verification techniques for mobility. Symbolic verification techniques for security protocols. Prototype tool development, Web services, directory services and case studies.

Partners Owning:	Dipartimento di Informatica Univ. Pisa, Italy
Partners contributed :	Uppsala, FFCT Lisbon, INRIA, Pisa
Made available to:	

Contents

1 Overview 3

2 Scientific Contributions 4

2.1 Task 1.1: Automata with Operations and Substitutions 4

2.2 Task 1.2: Proof Techniques 6

2.3 Task 1.3: Prototype and Case Studies 8

3 Appendix: WP1 Scientific Contributions 11

1 Overview

The overall goal of this Work Package is to develop a comprehensive automata-like model that supports effective techniques to specify and verify properties of network applications. For year 2 the specific objective for Task 1.1 was the definition of an extended version of History Dependent (HD) automata for handling names and substitutions in their full generality, while for Task 1.2 we intended to develop proof techniques (typically of security protocols) based on symbolic execution. Task 1.3. was concerned with extending the existing tools to handle the enriched versions of HD-automata.

This deliverable includes the scientific contributions of the second year for WP1. The deliverable consists of a short presentation of the contributions and of several appendices with the contributed papers. The scientific contributions can be summarized as follows.

The structure of HD-automata and the state-space minimization algorithm have been formally specified using a polymorphic λ-calculus with dependent types. This specification closely corresponds to the ML implementation of the Mihda toolkit. It has been shown that the different categorical formulation of HD- automata, namely the category of named sets and named functions, the category of permutation algebras and certain categories of presheaves, are essentially equivalent. Finally, the HD-automata approach has been extended to Fusion Calculus, and the bialgebra approach to Fusion Calculus with explicit fusions. Fusion Calculus has also been also mapped to logic programming clauses. A new calculus with two binders, called DFusion, which is an extension of both open π-calculus and Fusion Calculus, has been defined.

Symbolic verification techniques for security protocols have been further developed. The symbolic technique of *history-dependent scheduling* has been formally defined and implemented in the TRUST verifier. A "low level" protocol description language has been introduced, together with a notion of symbolic execution. A verification methodology based on the notion of symbolic state-space exploration has been developed and implemented in the verification environment called ASPASYA.

Additional results have been obtained for relevant verification models: a translation of the spi calculus into a basic dialect of π has been defined and implemented which is sound with respect to may-testing; automatic techniques for reachability analysis of infinite-state systems have been studied; the verification problem for a substantial fragment of the ntcc calculus with infinite processes and the negation-free fragment of LTL have been shown to be decidable; and several CCS variants differing for infinite behavior and for name scoping have been compared.

The development of PROFUNDIS WEB has progressed and several toolkits have been made available on the the PROFUNDIS WEB service directory. Some of the services available on the PWeb have been exploited in the verification of properties of mobile systems.

2 Scientific Contributions

We present the contributions according to the tasks they belong to.

2.1 Task 1.1: Automata with Operations and Substitutions

Automata, Domains and Coalgebras The formulation of HD-automata as the basic model for name passing process calculi has been further developed in Pisa. In the work of the first year, HD automata were defined as coalgebras on a category of named sets and named functions, where set elements are equipped with names defined up to specific groups of permutations called symmetries. An alternative definition, useful for defining operators like parallel composition and restriction, was also given in terms of coalgebras on a category of permutation algebras (bialgebras). While the latter definition is simpler, its automata are infinite in all except the most trivial cases. The former definition, instead, provides models suitable for finite state verification in the usual case of finite control. Moreover, the coalgebraic formulation of the partition-refinement minimization algorithm has directly suggested the software architecture of the Mihda toolkit. During the second year, two main contributions were given. In [13] (but see also the PhD thesis of Emilio Tuosto [20]) the automata structure and the minimization algorithm have been formally specified using a polymorphic λ-calculus with dependent types. This provides a detailed formal definition for which important properties can be proved (e.g. the convergence of the algorithm) and which is very close to the actual implementation. The second contribution [9] relates the category of named sets and named functions with the category of permutation algebras (defined à la Lawvere as a functor category) and compares them with the categories of presheaves (e.g. $\mathbf{Set}^{\mathbf{I}}$ or the Schanuel topos) considered by Moggi, Stark, Hofmann, Fiore and Turi among others as suitable domains for languages with name generation and passing. The technical details are nontrivial but the final result is that the three approaches are essentially equivalent. While the work is not completed (coalgebras are not considered yet), it looks now feasible to draw a full formal correspondence between the HD-automata approach and the classical domain-theoretical approaches, with advantages also in terms of the associated frameworks for nominal logics.

Fusion Calculi Fusion calculus was introduced by Parrow and Victor more than five years ago. It is a variant of the π -calculus, where existing names can be fused; input and output are symmetric operations; and there is only one binder, scope restriction (in the π -calculus also input prefix is a binder). An asynchronous version with explicit fusions, where fusions behave like messages, was defined later by Gardner and Wischik. Fusions are appealing in several variants of name mobility: e.g. the *open* construct of Abadi and Cardelli's *Ambient Calculus* can be conveniently modeled as the fusion of two locations. Three contributions related to Fusion Calculus have been given in the second year by Pisa and Uppsala. In [14, 21] the HD-automata approach has been extended to Parrow and Victor's Fusion Calculus. While in [21] a complex, not implemented version of HD-automata was considered (HD-automata *with negative transitions*), in [14] a version was chosen instead which requires limited changes in the implementation of Mihda (fusions appear only in transition labels). The second contribution [8], extends the bialgebra approach to Fusion Calculus with explicit fusions. Interestingly enough, representing fusions as messages allows to keep essentially the same algebraic structure we had for the π-calculus model of the first year, i.e. only injective renamings are allowed. However, besides parallel composition, restriction and recursion, the fusion calculus is here equipped also with prefix operations, thus yielding a rather satisfactory model. As usual, the bialgebra approach automatically guarantees that (hyper-) bisimulation is a congruence. The third contribution, [6], analyzes the expressiveness of Fusion Calculus when compared with (open) π-calculus, and concludes that neither of them is more expressive than the other. The reason is that while the former has powerful fusion constructs which are missing in the π-calculus, the restriction operator of the latter cannot be simulated by scope restriction of the Fusion Calculus. In fact, names extruded by Fusion Calculus can still be fused with existing names, while π-calculus fresh names are guaranteed to be fresh forever. Paper [6] considers this as a significant lack of expressiveness for those applications, like security protocols, where fresh names are used for modeling private channels, session keys and nonces. A new calculus with two binders, called DFusion, which is an extension of both open π-calculus and Fusion Calculus, is then presented and it is given an operational and a bisimulation semantics which is shown to be a congruence. It is too early to say if DFusion will be actually useful for interesting applications. However we think that the issue of the existence of two binders (call them λ and ν) we raised in this context is rather general, and it can be relevant for most existing models for mobility. In particular, we think that HD-automata could be easily equipped with both of them.

Other Extensions With respect to models for spatial logic, Lisbon has developed in the first year 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 work has progressed in the second year, and two notes with additional results have been produced [18, 17]. With respect to models equipped, like logic programming, with general substitutions, a significant step forward is presented in paper [16], where Fusion Calculus is mapped to logic programming clauses via an intermediate step defined in terms of synchronized graph rewriting. The intermediate step is needed since interaction is realized in logic programming via shared variables which must be consistently bound, while Fusion Calculus agents interact via rendez-vous synchronizations on channels. Synchronized graph rewriting is a convenient formalism to implement Fusion channels via routers able to create *remote* synchronizations. It is interesting to notice that the mapping from Fusion Calculus to logic programming is correct since Fusion Calculus extrusion mechanism (see the discussion above) is consistent with logic programming unification. In fact, also in logic programming fresh variables - introduced by clauses with variables in the body which to not appear in the head - can be freely unified in subsequent steps. Had we wanted to map π-calculus instead (or D-Fusion!), the variable handling mechanism of logic programming should have been modified, possibly introducing a suitable constraint handler.

2.2 Task 1.2: Proof Techniques

Symbolic Verification Techniques for Security Protocols The two research items in the previous PPR - symbolic verification techniques and security protocols - are now fully interdependent. We have here four lines of research, all aiming at the development of models and reasoning techniques for the analysis of security protocols. In the first line, developed at INRIA and already present in first year activity with the TRUST cryptographic protocol verifier, a novel technique called *history-dependent scheduling*, has been introduced [25]. This symbolic technique reduces the search space in the verification of cryptographic protocols. The technique has been implemented in the TRUST verifier.

The second line of work, developed at Pisa, was also active in the first year with a symbolic operational semantics of the spi-calculus and with the STA symbolic interpreter. In this line, a protocol description language along with its semantics has been introduced [5] together with a notion of symbolic execution. The language is able to express low-level features of cryptographic algorithms, like those needed to program the Diffie-Hellman protocol for exchanging a secret key over an insecure medium, without prior sharing of any secret. The PhD thesis [7] of Marzia Buscemi covers activity about this line and about bialgebras in Task 1.1.

The third line, also developed at Pisa, is new. A verification methodology based on the notion of symbolic state-space exploration has been developed [4]. The proposed methodology is supported by a verification environment called ASPASYA.

The last line of research developed at Uppsala [3] had the goal of translating the spi-calculus into a basic dialect of π. The translation has been proven sound with respect to may-testing and is has been implemented in a prototype tool. As a consequence, protocols can be described in the spi calculus and analysed with the emerging flora of tools already available for π. The translation also entails a more detailed operational view of the semantics of spi since the high level constructs are encoded in a well known lower level.

Verification of Infinite-state Processes Automatic techniques for reachability analysis have been developed for well-structured infinite-state systems. A line of research developed in Uppsala [2] aims at making this technique applicable also to non-well-structured systems for which reachability is undecidable in general. The idea of the approach is, first, to construct a well-quasi-ordered overapproximation of the given system in the form of a (generalised) Petri net. Then one can compute an overapproximation of $Pre^*(U)$, given an upward-closed set

U of configurations. If the initial configuration is not in this overapproximation of $Pre^*(U)$, then the answer to the reachability problem is surely NO. Otherwise, one can, in a second step, do an exact forward search which always stays inside the already computed overapproximation of $Pre^*(U)$, thus curbing the search-space. Due to the restricted search space, this forward search is more efficient than a normal forward search. Altogether, our method yields answers of the form YES, NO, or UNKNOWN. We have shown how to apply this technique to relabelling free CCS with finite summation, which is already a process calculus for which reachability is undecidable.

Another line of research , also developed at Uppsala [23], shows that the issue of establishing whether a given process satisfies a given LTL NTCC specification, is decidable for a substantial fragment of the ntcc calculus and the negation-free fragment of LTL. The noteworthy aspect of this result is that such an ntcc fragment allows *infinite-state processes*. Another interesting aspect is that as an application of the above result, also shows a new decidability result for Pnueli's First-Order LTL. The review papers [24, 22] illustrate the exploitation of TCCP for modeling and analysing reactive systems.

The last line of research [15] aims at providing a classification of several CCS variants found in the literature differing in the way of specifying infinite behavior (*replication, constant definitions, or recursive expressions)* and the scope of names *(dynamic* or *static).* The calculi were classified wrt to their expressiveness and decidability of process divergence. We regarded any two calculi as being *equally expressive* iff for every process in each calculus, there exists a *weakly bisimilar* process in the other. We proved that (1) the variant with recursive expressions and dynamic scoping is equally expressive to that with constant definitions while (2) the variant with recursive expressing and static scoping is equally expressive to that using replication. Moreover the divergence problem is undecidable for the calculi in (1) but decidable for those in (2).

Constraint Solving Recently constraint satisfaction techniques have been exploited to reason about properties of security protocols. A Constraint satisfaction problems (CSP) is any problem that can be expressed as that of finding, from a finite set of possibilities, a collection of values satisfying some given constraints. In its general setting the constraint satisfaction problem is known to be NP-complete. Nevertheless, in many real world instances a solution can be found with reasonable time and space efficiency when appropriate techniques are applied. The most frequently used are backtracking and arc-consistency techniques. Backtracking techniques are used to search for solutions. Arc-consistency (AC) techniques are typically used to reduce the problem before (or during) the search for solutions. In [19], we proposed a *new desirable property* for AC computation. We showed analytically and experimentally that the new property provides a further substantial reduction on the number of constraint checks.

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*, or *PWeb*, is the idea of viewing the environment as a distributed infrastructure exploited as a *service distributor*. 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. One main idea of our approach is to make semantic-based verification toolkits available as Web services, using standards such as WSDL and SOAP. Another main idea is to establish service directories for publishing such Web services. Verification web services plus service directories thus provide a platform for distributed, deeply integrated and therefore coordinated verification activities. PWeb is developed at Pisa and Uppsala.

The core of the PWeb is a *directory service*. A PWeb directory service is a component that maps the description of the Web services represented by suitable XML types into the corresponding network addresses. Moreover, it performs the binding of services by exploiting the `trader` engine. The trader engine manipulates pools of services distributed over several PWeb directory services. It can be used to obtain a Web service of a certain type and to bind it inside the application. The `trader` engine gives to the PWeb directory service the ability of finding and binding at run-time web services without "hard-coding" the name of the web service inside the application code. The description of the PWeb directory service is reported in [1]. The work reported in [11] illustrates the exploitation of some of the services of the PWeb in the verification of mobile systems.

Verification Toolkits The verification toolkits implemented in the first year of the project have been extended and/or re-engineered to accommodate new facilities, suggested by the theoretical investigations. In particular, the notion of history-dependent scheduling [25] has been implemented in the TRUST toolkit to reduce the dimension of the state space. We have already remarked on the development of the ASPASYA toolkit [4]. The minimization tool MIHDA has been described in [12]. Also, as already mentioned, its minimization algorithm has been formally specified using a polymorphic λ-calculus with dependent types [13]. Finally, the work reported in [10] illustrates the effectiveness and usability of the HAL Model Checker, a service available in the PWeb. A previous version of [10], submitted for publication, was part of the deliverable of the first year. The revised version, now accepted for publication, includes some new case studies which allow us to demonstrate several common patterns arising frequently when reasoning about π-calculus specifications. They concern the use of π-logic for verification of mobility and security properties.

References

[1] M. Baldamus, J. Bengtson, G. Ferrari, and R. Raggi. Web-Services as a New Approach to Distributing and Coordinating Semantics-Based Verification Toolkits. In *Web Services and Formal Methods*, 2004. WS-FM workshop proceedings, to appear.

[2] M. Baldamus, R. Mayr, and G. Schneider. A Backward/Forward Strategy for Verifying Safety Properties of Infinite-State Systems. Technical Report 2003-065, Department of Information Technology, Uppsala University, Sweden, 2003.

[3] M. Baldamus, J. Parrow, and B. Victor. Spi Calculus Translated to π-Calculus Preserving May Testing. Technical Report 2003-063, Department of Information Technology, Uppsala University, Sweden, 2003.

[4] G. Baldi, A. Bracciali, G. Ferrari, and E. Tuosto. ASPASyA: an automated tool for security protocol analysis based on a symbolic approach. Submitted for publication, 2003.

[5] M. Boreale and M. Buscemi. Symbolic analysis of crypto-protocols based on modular exponentiation. In *Proc. of MFCS 2003*, Lecture Notes in Computer Science 2747. Springer-Verlag, 2003. An extended version appears in Proc. of FCS'03.

[6] M. Boreale, M. Buscemi, and U. Montanari. The distinctive fusion calculus. Unpublished draft, 2003.

[7] M. Buscemi. *Models and Security Verification of Mobile Systems.* PhD thesis, Dipartimento di Matematica, University of Neaples "Federico II", 2003.

[8] M. Buscemi and U. Montanari. A compositional coalgebraic model of explicit fusions. Unpublished draft, 2003.

[9] M. Miculan F. Gadducci and U. Montanari. Some characterization results for permutation algebras. In *Workshop COMETA*, ENTCS. Elsevier, 2003. to appear.

[10] G. Ferrari, S. Gnesi, U. Montanari, and M. Pistore. A model checking verification environment for mobile processes. *ACM Transactions on Software Engineering and Methodologies (TOSEM)*, To appear, 2004.

[11] G. Ferrari, S. Gnesi, U. Montanari, R. Raggi, G. Trentanni, and E. Tuosto. Verification on the web of mobile processes. Submitted for publication, 2003.

[12] G. Ferrari, U. Montanari, R. Raggi, and E. Tuosto. From co-algebraic specifications to implementation: The mihda toolkit. In *First International Symposium on Formal Methods for Components and Objects (FMCO)*, Springer Lecture Notes in Computer Science. Springer, 2003.

[13] G. Ferrari, U. Montanari, and E. Tuosto. Co-algebraic minimization of hd-automa in a polymorphic λ-calculus. Under Revision for Theoretical Computer Science, 2003.

[14] G. Ferrari, U. Montanari, E. Tuosto, B. Victor, and K. Yemane. Modelling and minimising the fusion calculus using hd-automata. Technical report, University of Pisa, 2003.

[15] P. Giambiagi, G. Schneider, and F. Valencia. On the expressiveness of ccs-like calculi. In *In Proceedings of FOSSACS'04*. LNCS, Springer-Verlag, 2004.

[16] I. Lanese and U. Montanari. Mapping fusion and synchronized hyperedge replacement into logic programming. Under Revision for Theory and Practice of Logic Programming, 2003.

[17] L. Monteiro. A note on a noninterleaving model of concurrency based on transition systems with spatial structure. Technical Note, DI-FCT/UNL, 2003.

[18] L. Monteiro. A note on models for spatial logic based on transition systems with spatial structure. Technical Note, DI-FCT/UNL, 2003.

[19] C. Rueda and F. Valencia. Non-viability deductions in arc-consistency computation. Technical report, 2003. WP1, Submitted for conference publication.

[20] E. Tuosto. *Non Functional Aspects of Wide area Network Programming.* PhD thesis, Dipartimento di Informatica, Univ. Pisa, 2003.

[21] E. Tuosto, B. Victor, and K. Yemane. Polyadic history-dependent automata for the fusion calculus. Technical Report 2003-062, Department of Information Technology, Uppsala University, December 2003.

[22] F. Valencia. Concurrency, time and constraints. In *Proceedings of the Nineteenth International Conference on Logic Programming (ICLP 2003).* LNCS, Springer-Verlag, 2003. WP2, Invited publication.

[23] F. Valencia. Timed concurrent constraint programming: Decidability results and their application to ltl. In *Proceedings of the Nineteenth International Conference on Logic Programming (ICLP 2003).* LNCS, Springer-Verlag, 2003.

[24] Frank Valencia. Notes on timed ccp. In *4th Advanced Course on Petri Nets ICPN'03.* LNCS, Springer-Verlag, 2004. WP2, Invited publication.

[25] V. Vanackere. History-dependent scheduling for cryptographic processes. In *Proc. VMCAI 2004*, Springer Lecture Notes in Computer Science. Springer, 2004. to appear.

3 Appendix: WP1 Scientific Contributions

This section lists the papers contributing to Work Package 1.

1. Appendix 1.2.1

   M. Baldamus, J. Bengtson, G. Ferrari, and R. Raggi. Web-Services as a New Approach to Distributing and Coordinating Semantics-Based Verification Toolkits, To appear in Proc. *First International Workshop on Web Services and Formal Methods*, ENTCS, 2004.

2. Appendix 1.2.2

   M. Baldamus, R. Mayr, and G. Schneider. A Backward/Forward Strategy for Verifying Safety Properties of Infinite-State Systems. Technical Report 2003-065, Department of Information Technology, Uppsala University, Sweden, 2003.

3. Appendix 1.2.3

   M. Baldamus, J. Parrow, and B. Victor. Spi Calculus Translated to π-Calculus Preserving May Testing. Technical Report 2003-063, Department of Information Technology, Uppsala University, Sweden, 2003.

4. Appendix 1.2.4

   G. Baldi, A. Bracciali, G. Ferrari, and E. Tuosto. ASPASyA: an automated tool for security protocol analysis based on a symbolic approach. Technical Report, 2004

5. Appendix 1.2.5

   M. Boreale, M. Buscemi, and U. Montanari. The distinctive fusion calculus. Unpublished draft, 2003.

6. Appendix 1.2.6

   M. Buscemi and U. Montanari. A compositional coalgebraic model of explicit fusions. Unpublished draft, 2003.

7. Appendix 1.2.7

   M. Miculan F. Gadducci and U. Montanari. Some characterization results for permutation algebras. In *Workshop COMETA*, ENTCS. Elsevier, 2003. To appear.

8. Appendix 1.2.8

   G. Ferrari, S. Gnesi, U. Montanari, and M. Pistore. A model checking verification environment for mobile processes. *ACM Transactions on Software Engineering and Methodologies (TOSEM)*, To appear, 2004.

9. Appendix 1.2.9

   G. Ferrari, S. Gnesi, U. Montanari, R. Raggi, G. Trentanni, and E. Tuosto. Verification on the web of mobile processes. Submitted for publication, 2003.

10. Appendix 1.2.10

    G. Ferrari, U. Montanari, R. Raggi, and E. Tuosto. From co-algebraic specifications to implementation: The mihda toolkit. In First International Symposium on Formal Methods for Components and Objects (FMCO), Springer Lecture Notes in Computer Science, 2852. Springer, 2003.

11. Appendix 1.2.11

    G. Ferrari, U. Montanari, and E. Tuosto. Co-algebraic minimization of hd-automa in a polymorphic λ-calculus. Under Revision for Theoretical Computer Science, 2003.

12. Appendix 1.2.12

    Ferrari,Gianluigi and Montanari,Ugo and Tuosto,Emilio and Victor,Björn and Yemane,Kidane, Modelling and Minimising the Fusion Calculus Using HD-Automata, Unpublished Draft, 2004.

13. Appendix 1.2.13

    I. Lanese and U. Montanari. Mapping fusion and synchronized hyperedge replacement into logic programming. Under Revision for Theory and Practice of Logic Programming, 2003.

14. Appendix 1.2.14

    M. Buscemi M. Boreale. Symbolic analysis of crypto-protocols based on modular exponentiation. In *Proc. of MFCS 2003*, Lecture Notes in Computer Science 2747. Springer-Verlag, 2003. An extended version appears in Proc. of FCS'03.

15. Appendix 1.2.15

    L. Monteiro. A note on a noninterleaving model of concurrency based on transition systems with spatial structure. Technical Note, DI-FCT/UNL, 2003.

16. Appendix 1.2.16

    L. Monteiro. A note on models for spatial logic based on transition systems with spatial structure. Technical Note, DI-FCT/UNL, 2003.

17. Appendix 1.2.17

    C. Rueda and F. Valencia. Non-viability deductions in arc-consistency computation. Technical report, 2003.

18. Appendix 1.2.18

    Tuosto,Emilio and Victor,Björn and Yemane,Kidane, Polyadic History-Dependent Automata for the Fusion Calculus, Tech, Report, Department of Information Technology, Uppsala, Sweden, 2003 (2003-62),

19. Appendix 1.2.19

    F. Valencia. Timed concurrent constraint programming: Decidability results and their application to ltl. In *Proceedings of the Nineteenth International Conference on Logic Programming (ICLP 2003).* LNCS, Springer-Verlag, 2003.

20. Appendix 1.2.20

    F. Valencia, P. Giambiagi, and G. Schneider. On the expressiveness of CCS-like Calculi calculi. In *In Proceedings of FOSSACS'04.* LNCS, Springer-Verlag, 2004.

21. Appendix 1.2.21 F. Valencia. Concurrency, time and constraints. In *Proceedings of the Nineteenth International Conference on Logic Programming (ICLP 2003).* LNCS, Springer-Verlag, 2003.

22. Appendix 1.2.22 F. Valencia. Notes on timed ccp. In *4th Advanced Course on Petri Nets ICPN'03.* LNCS, Springer-Verlag, 2004.

23. Appendix 1.2.23

    V. Vanackere. History-dependent scheduling for cryptographic processes. In *Proc. VMCAI 2004*, Springer Lecture Notes in Computer Science. Springer, 2004. To appear.

24. Appendix 1.2.24

    M. Buscemi. *Models and Security Verification of Mobile Systems.* PhD thesis, Dipartimento di Matematica, University of Neaples "Federico II", 2003.

25. Appendix 1.2.25

    E. Tuosto. *Non Functional Aspects of Wide area Network Programming.* PhD thesis, Dipartimento di Informatica, Univ. Pisa, 2003.