UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

PROCEEDINGS OF THE INTERNATIONAL WORKSHOP "GOBAL COMPUTING: PROGRAMMING ENVIRONMENTS, LANGUAGES, SECURITY AND ANALYSIS OF SYSTEMS"

Editor: Corrado Priami

February 2003

Technical Report # DIT-03-006

Global Computing

Programming Environments, Languages, Security and Analysis of Systems

9-14 February, 2003
Rovereto, Trento

C. Priami Editor

Preface

According to the IST/ FET proactive initiative on GLOBAL COMPUTING, the goal is to obtain techniques (models, frameworks, methods, algorithms) for constructing systems that are flexible, dependable, secure, robust and efficient. The dominant concerns are not those of representing and manipulating data efficiently but rather those of handling the co-ordination and interaction, security, reliability, robustness, failure modes, and control of risk of the entities in the system and the overall design, description and performance of the system itself. Completely different paradigms of computer science may have to be developed to tackle these issues effectively. The research should concentrate on systems having the following characteristics:

- The systems are composed of autonomous computational entities where activity is not centrally controlled, either because global control is impossible or impractical, or because the entities are created or controlled by different owners.
- The computational entities are mobile, due to the movement of the physical platforms or by movement of the entity from one platform to another.
- The configuration varies over time. For instance, the system is open to the introduction of new computational entities and likewise their deletion. The behaviour of the entities may vary over time.
- The systems operate with incomplete information about the environment. For instance, information becomes rapidly out of date and mobility requires information about the environment to be discovered.

The ultimate goal of the research action is to provide a solid scientific foundation for the design of such systems, and to lay the groundwork for achieving effective principles for building and analysing such systems.

This workshop covers the aspects related to languages and programming environments as well as analysis of systems and resources involving 9 projects (AGILE , DART, DEGAS , MIKADO, MRG, MYTHS, PEPITO, PROFUNDIS, SECURE) out of the 13 founded under the initiative. After an year from the start of the projects, the goal of the workshop is to fix the state of the art on the topics covered by the two clusters related to programming environments and analysis of systems as well as to devise strategies and new ideas to profitably continue the research effort towards the overall objective of the initiative.

We acknowledge the Dipartimento di Informatica and Tlc of the University of Trento, the Comune di Rovereto, the project DEGAS for partially funding the event and the Events and Meetings Office of the University of Trento for the valuable collaboration.

Rovereto
28 january 2003 Corrado Priami

Contents

AGILE

Full Title: **Architectures for Mobility**
Contact Person: WIRSING, Martin - Ludwig-Maximilians-Universitaet Muenchen

Architecture-based approaches have been promoted as a means of controlling the complexity of system construction and evolution, namely for providing systems with the agility required to operate in turbulent environments and adapt very quickly to changes in the enterprise world. Recent technological advances in communication and distribution have made mobility an additional factor of complexity, one for which current architectural concepts and techniques are not prepared for. AGILE will provide means for addressing this new level of complexity by developing an architectural approach in which mobility aspects can be modelled explicitly and mapped on the distribution and communication topology made available at physical levels. The whole approach will be developed over a uniform mathematical framework based on graph-oriented techniques that will support sound methodological principles, formal analysis, and refinement.

Objectives: AGILE will develop an integrated architectural approach to the development of systems in which mobility is a key factor, including:

1. Primitives for explicitly addressing mobility within architectural models;
2. Algebraic models of the evolution processes that result from system reconfiguration caused by mobility of components;
3. Extensions to modelling languages like the UML that make the architectural primitives available to practitioners, together with tools for supporting animation and early prototyping;
4. Analysis techniques for supporting compositional verification of properties addressing evolution of computation, coordination and distribution;
5. Refinement techniques for relating logical modelling levels with the distribution and communication topology available at physical levels.

Work description: In order to meet the proposed goals, AGILE will capitalise on the experience that the members of the consortium have accumulated in the areas of formal software architectures, algebraic and logical development techniques, process calculi, concurrency, combination of formal and semi-formal modelling techniques, graph-based semantics, and software development in business domains characterised by a high volatility of requirements. More precisely, AGILE will follow three main strands of research:

1. the extension of our previous work on the development of a categorical framework supporting software architectures on the basis of the separation between 'computation' and 'coordination' with an additional dimension for 'distribution' and, consequently, 'mobility', providing primitives

-distribution contracts in line with the coordination contracts that we have been developing - with which the distribution topology can be explicitly modelled and refined across different levels of abstraction;

2. the definition of algebraic models for the underlying evolution processes, relating the reconfiguration of the coordination structure and the mobility of components across the distribution topology, again capitalising on our previous work in graph transformation techniques, and laying down the basis for logical analysis of evolution properties as well as tools for animation and early prototyping; and;

3. the extension of existing modelling languages and processes like the UML with the concepts and techniques that will have been developed in the other workpackages, including tools for animation and early prototyping. A fourth line of work consisting of case study development and prototyping will ensure that the project will develop a joint awareness of the problems and solutions to be developed, and that the three different technical strands will actually come together as part of a unified and effective architectural approach to mobility.

DART

Full Title: **Dynamic Assembly, Reconfiguration and Type-checking**
Contact Person: MOGGI, Eugenio - Università di Genova

The project will develop formalisms for dynamic assembly, reconfiguration and type-checking of complex distributed software systems, such as telephone and banking systems, that should be kept running as they evolve through patches or upgrades, and should be able to adapt to changes in the environment.

Such formalisms will advance the state of the art in modelling the "temporal" dimension of Global Computing (GC), where the ability to interleave meta-programming activities, like assembly and reconfiguration, with computational activities is a must.

The development of these calculi will rely on decisive progress in three areas: calculi for dynamic assembly, calculi for object evolution and adaptation, flexible and compositional type systems.

Objectives: The project aims to advance the state of the art in modelling and programming software evolution while retaining safety. More specifically:

- We will provide foundational calculi for dynamic assembly and reconfiguration which will be able to describe separate compilation, run-time code generation, dynamic linking and loading.

- We will design foundational calculi supporting object capable of changing their behaviour, e.g. by changing class, as well as calculi that are environment adaptable, e.g. able to test existence of objects in the execution environment.

- We will develop type systems that support "compositional analysis" through the existence of "principal typings", show how to use such type systems for separate compilation and incremental type inference, and address the issue of combining dynamic type-checking with dynamic assembly and reconfiguration.

Work description: The project is organized in 5 Workpackages (WPs). The first three WPs:
1. Frameworks and Calculi for Dynamic Software Assembly
2. Flexible and Compositional Type Systems
3. Calculi for Object Evolution aim to develop the calculi and type systems identified as key project objectives.

The main goal will be to carry out the foundational work, which will take the form of frameworks for dynamic assembly and reasoning about properties of different assembly strategies, calculi for object evolution and adaptation, type systems with properties that will make them particularly suitable for use in a dynamic context.

The other two WPs:
4. Applications to Prevalent Languages
5. Flexible Dynamic Type

Checking for Dynamic Software Assembly are downstream (their feasibility will be re-assessed at the first review point on month 12 of the project), their rationale is:

- to test the portability of innovative ideas expected from WPs 1 and 3, namely facilities supporting object evolution (i.e. allowing an object to change its class or the code of its methods) and environment-adaptable programming, to a major programming language. Such a language will be chosen (at the time of the first review) among the prevalent ones for GC. The emphasis on objects is motivated by the expectation that in any successful language for GC the object paradigm will play a major role.

- To test how the innovative ideas expected from WPs 1 and 2 (and developed fairly independently, but with portability in mind) can be merged in a unifying framework that will account for dynamic assembly, reconfiguration and type-checking.

The combination of dynamic type checking with dynamic assembly and reconfiguration is essential, since addressing these issues separately will either fail to guarantee safety and efficiency or be significantly less useful in the GC environment.

DEGAS

Full title: **Design Environments for Global ApplicationS**
Contact person: PRIAMI, Corrado - Università di Trento

DEGAS aims to combine structured (semi-formal) graphical methods for specification by picture and animation of global applications with formal methods for their analysis and verification. We will investigate to what extent UML is already suitable to model global applications and we will propose extensions. We will propose formal models of these applications based on the operational semantics of foundational process calculi for mobility. Static and dynamic analysis concentrate on two key-features of global computing: performance prediction and security. We will assess the foundational studies in a prototypical proof of concept environment that hides to the user as much as possible of the formal treatment. We will tune our development with case-studies on wireless telecommunication applications.

Objectives: DEGAS addresses foundational aspects for the design of global applications by enhancing the state of the art in scientific as well as engineering principles. The main concerns are the specification in UML and qualitative and quantitative analysis of global applications. We plan to define the key features of global (wireless) applications that should be exposed at an abstract level of specification and analysis. We provide formal relations between the (possibly richer or incomplete) UML models and the process calculi specifications to connect the specification and the verification environment by hiding as much formal details from the designer as possible. The static and dynamic analysis with case studies should bring to the definition of new linguistic constructs and new models to analyse and reason about performance and security of global systems.

Work description: DEGAS is organized into workpackages. Besides management and assessment of progress and results, we have (workpackage=WP):

- WP3 (UML feasibility, modification and tool customisation) that customizes a tool to build the designer's interface and manipulate UML models.

- WP4 (Extraction, reflection and Integration) that defines the interface between the specification part of the environment and the verification kernel. The extraction takes information from UML models and builds process calculi specifications; the reflection exposes to the user the results of the formal analysis in UML notation. The integration task is responsible for building a unique case tool out of the subtools developed during project lifetime.

- WP5 (Dynamic analysis) is responsible for defining new linguistic constructs and new models to carry out (quantitative and security) dynamic

analysis on transition-system based representations of global applications. The WP exploits also fine-grain models in which security and quantitative issues coeixst.

- WP6 (Static analysis) is responsible for specifying analysis in the flow logic and abstract interpretation approaches for determining the overall responsiveness of the system and to harden the design against denial of service attacks. We also investigate the usage of reachability information for controlling information leaks (to preserve confidentiality) and to ensure the correct authentication of devices.

- WP7 (Case studies) is responsible for validating the development of the project as well for providing experimental guidance to the foundational studies.

The services we selected as case studies are
(1) pilot service for mobile entertainment and
(2) mobile home banking.

MIKADO

Full title: **Mobile Calculi Based on Domains**
Contact person: STEFANI, Jean-Bernard - INRIA

Current middleware and programming language technologies are inadequate to meet the challenges posed by a global computing environment. In particular, they tend to support only a limited range of interactions, have a limited view of components and objects, fail to properly and uniformly support properties such as mobility, predictability, security, fault-tolerance, and they are not amenable to rigorous investigation for verification, validation and test purposes. The Mikado project intends to overcome these limitations by defining and prototyping new formal models for both the specification and programming of highly distributed and mobile systems, and to develop specification and analysis techniques which can be used to build safer and trustworthy systems, to demonstrate their conformance to specifications and to analyse their behaviour.

Objectives: The goal of the Mikado project is to construct a new formal programming model, based upon the notion of domain as a computing concept, which supports reliable, distributed, mobile computation, and provides the mathematical basis for a secure standard for distributed computing in open systems. Specifically, Mikado intends:

- To develop new formal models for both the specification and programming of large-scale, highly distributed and mobile systems;
- To develop new programming language features supporting such models, and to study their combination with functional and object-oriented programming;
- To develop specification and analysis techniques which can be used to build safer and trustworthy systems, to demonstrate their conformance to specifications, and to analyse their behaviour;
- To prototype new virtual machine technologies which can be used to implement in a "provably correct" way such models and languages.

Work description: The project is organised around three technical work-packages (WP1-WP3) and one organisational work-package (WP4):

- WP1: Core Programming Model;
- WP2: Specification and Analysis;
- WP3: Virtual Machine Technology and Language Support;
- WP4: Project Co-ordination and Dissemination

WP1 is concerned with the definition of a core programming model for global computing, based on the notion of domain. This work-package will provide the basis for the rest of the theoretical work taking place in WP2 and for the development work taking place in WP3.

WP2 is concerned with the definition of Specification and Analysis technologies for the project's programming model. This will range from the development of type systems and static analysis techniques for expressing constraints on concurrency, mobility and resource access for the underlying execution model, to providing proof technologies for assuring that mobile code, and more generally distributed systems, conform to predefined behavioural specifications. The latter will require the definition of novel co-inductive techniques for comparing the distributed behaviour of systems and the elaboration of new specification logics for expressing interesting partial views of systems and programming paradigms.

WP3 is concerned with the embodiment of the Mikado programming model developed in WP1 and WP2 in concrete programming technologies. Work in WP3 will be concerned with the development of several prototypes, including:

- Virtual machine technology to support WP1's core programming model together with WP2 typing schemes;

- Languages features and language extensions supporting WP1's model and WP2's type systems.

MRG

Full title: **Mobile Resource Guarantees**
Contact person: SANNELLA, Donald - The University of Edinburgh

The use of mobile code in a global environment aggravates existing security problems and presents altogether new ones, one of which is the maintenance of bounds on quantitative resources. Without some technological foundations for providing such guarantees, global computing will be confined to applications where malfunction due to resource bound violation is accepted as normal and has little consequence. With more serious applications, resource awareness will be a crucial asset. This project aims at developing the infrastructure needed to endow mobile code with independently verifiable certificates describing resource behaviour. These certificates will be condensed and formalised mathematical proofs of a resource-related property, which are by their very nature self-evident and unforgeable. Arbitrarily complex methods may be used to construct these certificates, but their verification will always be a simple computation.

Objectives:
Objective 1: Development of a framework for formal certificates of resource consumption, consisting of a cost model and a program logic for an appropriate virtual machine. In the first instance this will be a subset of the Java VM; later we will consider appropriate parameterisations allowing for mobile virtual machines.
Objective 2: Development of a notion of formalised and checkable proofs for this logic which will play the role of certificates, including the implementation of a proof checker.
Objective 3: Development of methods for machine generation of certificates for appropriate high-level code, either fully automatically or based on user-supplied annotations e.g. in the form of invariants. Type systems will be used as the underlying formalism for this endeavour.
Objective 4: Study relaxations of proof-based certificates based on several rounds of negotiations between supplier and user of code leading to higher and higher confidence that the resource policy is satisfied.

Work description: This project aims at developing the infrastructure needed to endow mobile code with independently verifiable certificates describing resource behaviour. These certificates will be condensed and formalised mathematical proofs of a resource-related property, which are by their very nature self-evident and unforgeable. Arbitrarily complex methods may be used to construct these certificates, but their verification will always be a simple computation.

The work plan consists of the following central tasks:

1. Define expressive formalised resource policy (cost models);
2. Define notions of independently verifiable certificate (resource sensitive program logic with proof objects);

3. Foundations for efficient generation of certificates (type systems, identification of useful programmer annotations);

4. Foundations for alternatives to generation of full certificates (proof-theoretic compression, probabilistically checkable proofs, game-theoretic approaches).

Where appropriate, each foundational task is accompanied by a prototype implementation and case studies. In addition, the project includes the following separate engineering-oriented tasks:

1. Design of runtime environment including virtual machine, bytecode, implemented program logic;

2. Design and implementation of a high-level programming language in which to write resource-certified code;

3. Generation and integrated use of formalised certificates;

4. Parameterisation by arbitrary runtime environment.

The deliverables are research papers describing our solutions to foundational problems and a working prototype which will be made available as free downloadable software.

MYTHS

Full title: **Models and Types for Security in Mobile Distributed Systems**
Contact person: SASSONE, Vladimiro - The University of Sussex

Peer-to-peer computing on the Internet, mobile code downloading, and e-Commerce are already ubiquitous aspects of our computing environments. Casting a view to the immediate future, we see a global computational infrastructure emerging that will rely on the sharing of an open-ended distributed network of mobile resources among mobile computing agents. This infrastructure can only be successful insomuch as it provides adequate security guarantees of secrecy, integrity, availability, and more. MyThS seeks to develop type-based foundational theories of security for mobile and distributed systems. By relying on strong typing as the basic principle, MyThS addresses the foundations of programming languages and paradigms that allow static detection of security violations, and aims at developing type theoretic methods and tools that enable formal analyses of security guarantees appropriate for systems and applications on the global computing platform.

Work description: MyThS is undertaken by a small consortium of partners that reflects the very focused nature of the project and its objectives. The workplan of the project unfolds in three major themes: resource access control, information flow control, and analysis of cryptographic protocols. These are central, challenging themes for the global computing area, with far-reaching impact on the development of high-level, reliable, network-aware programming languages. Two notions will be pivotal throughout the themes: model and types. Based on high-level process calculi, MyThS will develop formal models for distributed and mobile code environments. Such models will be defined so as to address the diverse requirements for secure communication and mobility in open-ended networks with highly dynamic topologies, in which cryptography is a necessary prerequisite for security. The fundamental vehicle for ensuring security will be provided by typing systems, intended both as prescriptive and descriptive tools, capable of constraining and characterising agents' behaviours and interactions. They will be used to enforce and analyse security properties in each of our themes of investigation. By weaving together these hitherto independent contexts, MyThS will form a broad, coherent foundation for trustworthy communication in a global environment. The absence of central coordination, typical of the global computing network, will be a fundamental challenge for typing, as agents will not be able to trust that network objects comply with any given set of rules. MyThS will make provision for type systems to cope with this, by amalgamating techniques for static typing with new mechanisms for decentralised (dynamic) type-checking of distributed computing sites and migrating agents. Strong typing will provide formal guarantees of resilience against intended or accidental violations, and thus lay the foundations for the design of robust, high-level programming paradigms for global computing.

PEPITO

Full title: **Peer-To-Peer-Implementation-and-TheOry**
Contact person: SJÖLAND, Thomas - SICS

Traditional centralised system architectures are ever more inadequate. A good understanding is lacking of future decentralised peer-to-peer (P2P) models for collaboration and computing, both of how to build them robustly and of what can be built. The PEPITO project will investigate completely decentralised models of P2P computing.
It will:
(1) study the use-models of P2P systems, that is how they are perceived by users and what new applications are possible;
(2) develop the foundations of P2P computing, including formal foundations (calculi, proof techniques, security and resource models) and new distributed algorithms (for diffusing information and coping with multi-consistent views);
(3) provide a language-independent distribution subsystem tailored for P2P computing; and
(4) provide programming languages and platforms using this, showing that they are useful by implementing convincing demonstrator applications.

Objectives: Peer-to-peer computing (P2P) is a paradigm in which applications are connected to a shared network as peers, that is with the same capabilities and responsibilities. Current P2P applications are limited to information exchange. The objectives are to remove this limitation by:
- developing formal models to understand P2P computing;
- developing the distributed algorithms required for implementation;
- implementing a language-independent set of basic services;
- implementing languages and devise programming techniques and convincing demonstrator applications.

Further objectives are:
- better using resources at the network's edge;
- scaling better than server-centric computing;
- allowing device mobility (independence of IP address);
- allowing individuals to publish information and services, and allowing individuals to collaborate while remaining anonymous.

Work description: PEPITO will assume a completely decentralised architecture in which a peer can have four simultaneous roles: it may use services, provide services, forward requests, and provide caching of information. We also assume that peer nodes connect through a virtual network that is dynamic and intermittent, and that nodes do not possess a fixed IP address. To successfully deal with the complexity of P2P systems (in which failure, reconfiguration and security are central) it is important to pursue use-model analysis, theoretical work and prototyping in a closely linked style. The complementary expertise of the PEPITO partners makes this possible: the objectives will be addressed, but

enabling interaction between them is also crucial. Use-model analysis of this type of system will investigate how they are perceived by users, and what new applications are possible.

Theoretical work will study the foundational concepts of P2P systems. This includes mathematical models (calculi, proof techniques, security and resource models) and new distributed algorithms (decentralised algorithms for diffusing information, and for coping with multi-consistent computing - with simultaneous inconsistent views of entities). System Design and Prototyping will develop prototypes of programming languages and programming platforms (middleware) suitable for peer-to-peer computing (such platforms are lacking today; those existing are server-centric). One aspect will be a scalable and robust name/directory service based on our algorithms. Together, all these will enable development of applications that:

- handle dynamic connectivity and device mobility;
- allow individuals to become publishers of information and services;
- permit full use of existing network resources at the edge of the network;
- and permit applications to scale better than server-centric designs.

PROFUNDIS

Full title: **Proofs of Functionality for Mobile Distributed Systems**
Contact person: PARROW, Joachim - Kuglinga Tekniska Hogskolan

PROFUNDIS aims at developing methods to analyse the behaviour of distributed mobile systems, in order to ascertain that they function correctly. This involves modelling the systems in an abstract way and formulating rigorous correctness properties; it will be necessary to consider open and extensible systems with unknowable parts. For this purpose we shall develop operational models (based on automata), algebras, logical languages, and associated type systems. Analysis will be conducted through computer tools, both fully automatic and interactive. The novelty of the project lies in integrating several theoretical strands into one framework and one set of tools geared towards mobile distributed systems. In particular we shall consider security properties and systems used in electronic commerce.

Objectives: The objective of PROFUNDIS is to advance the state of the art of formal modelling and verification techniques to the point where key issues in mobile distributed systems, such as security protocols, authentication, access rights and resource management can be treated rigorously and with considerable automatic support. In particular we shall verify properties typical in so called open systems, where the behaviour of some parts (like intruders or adversaries) is unknowable, in extensible systems, where parts may be added or removed as the system executes, and in mobile systems where physical and logical connectivity between parts may change. We shall implement automatic and partly automatic analysis methods for ascertaining correct behaviour of such systems. For this purpose we shall integrate and focus several strands of ongoing theoretical work.

Work description: The work builds on recent advances in key theories for process behaviours, logics and types. We shall develop automata theoretic models suitable for our applications, with a particular interest in how they can be represented efficiently and used by automatic tools, and we shall determine how they are best used in connection with advanced forms of modal logics. The logics themselves will be developed, both in terms of their expressiveness for properties related to space and structure, and in terms of their accessibility and ease of use through suitable high-level representations. We shall identify and develop analysis techniques related to these models and logics. This involves traditional behavioural equivalences and pre-order checking, systematic simulation, and verification in interactive proof assistants. Here type systems will play an important role. Recent results show that types may themselves be used as crude but tractable correctness properties and therefore type inference is highly relevant, moreover, we shall explore how advanced type information can assist the other analysis techniques. The ideas will to a large extent be implemented in a common tool set. Key issues here will be development and adaption of algorithms for analysis, and determining the best way of using them for practical examples. We shall in particular consider examples on security properties in systems for electronic commerce.

SECURE

Full title: **Secure Environments for Collaboration among Ubiquitous Roaming Entities**
Contact person: CAHILL, Vinny - The Provost Fellows and Scholars of The College of The Holy Undivided Trinity of Queen Elizabeth Near Dublin

It is arguable whether the security mechanisms used to protect today's information systems are adequate. What is clear is that new approaches to security are needed for the infrastructure envisaged by the global computing initiative, which is characterized by decentralised control. The SECURE project will investigate a new approach to security founded on the notion of trust. The project aims to develop a model in which trust relationships are established from the record of interaction between entities, and a security mechanism expressed in terms of such trust. SECURE will also investigate how to specify access control policy based on trust. The project will formally define a computational trust model and a collaboration model capturing the dynamic aspects of the trust model; means to specify and to enforce security policies based on trust; means to evaluate security policies and implementations based on trust; and algorithms for trust management.

Objectives: The objectives of SECURE are the definition of a computational trust model allowing entities to reason about the trustworthiness of other entities for use in security related decisions; the definition of a collaboration model capturing the issues of trust formation, trust evolution, trust propagation and trust exploitation; the definition of means to specify and to enforce security policies based on trust including specifying the level of positive experiences required to allow a particular principal access to a specific resource; the definition of means to evaluate security policies and implementations based on trust while recognizing that there may be many different ways of establishing the required level of trust for collaboration to take place; the development of a framework encompassing algorithms for trust management include algorithms to handle trust formation, trust evolution and trust propagation; the validation of the approach in the context of the formal model.

Work description: The application of trust leads naturally to a decentralised approach to security management that can tolerate partial information albeit one in which there is an inherent element of risk for the trusting entity. Fundamentally, it is the ability to reason about trust that allows entities to accept risk when they are interacting with other entities and hence, the central problem to be addressed by SECURE is to provide entities with a basis for reasoning about trust. Thus, the heart of the SECURE workplan is the development of a computational model of trust that will provide the formal basis for reasoning about trust and for the deployment of verifiable security policies. The most important activity in the workplan is therefore the development of a formal computational trust model that captures human intuitions about trust, and must especially

allow computational entities to reason about the trustworthiness of other participants for use in security related decisions. We have planned to deliver two revisions of the model during the course of the project primarily because we expect the development of the model to be informed by the other activities in the project.

While the development of the computational trust model is at the heart of SECURE, it alone is not sufficient to allow us to deliver a feasible security mechanism for the global computing infrastructure. In this context it is equally important that we understand how trust is formed, evolves and is exploited in a system, e.g. the trust lifecycle; how security policy can be expressed in terms of trust and access control implemented to reflect policy; and how algorithms for trust management can be implemented feasibly for a range of different applications. Further activities address these issues based on an understanding of trust derived from the formal model but also contributing to the understanding of trust as a feasible basis for making security decisions to be embodied in the model.

A logic for reasoning about the resource usage of byte code programs

Lennart Beringer
University of Edinburgh, LFCS - lenb@inf.ed.ac.uk

Hans-Wolfgang Loidl
LM Universität München - hwloidl@informatik.uni-muenchen.de

In proving resource properties for byte-code programs we adopt the approach of foundational proof-carrying-code: we favour a very simple logic, thus minimising both the trusted code as well as the trusted theory base of our system. To obtain such a system for the Grail intermediate language, which abstracts over Java byte-code, we first present the main features of a resource-aware operational semantics for Grail. In order to facilitate proving resource properties over heap allocated data structures, we then adapt concepts of Reynold's Separation Logic to our framework and present proofs of standard example programs based on this level of abstraction. The presented work has been encoded in the Isabelle theorem prover and a first (demonstrator) version of this infrastructure for proof-carrying-code is available.

Grail: a functional intermediate language for resource-bounded computation

Kenneth MacKenzie
University of Edinburgh - kwxm@dcs.ed.ac.uk

The Mobile Resource Guarantees project aims to provide guarantees on the resource usage of mobile code. I will describe Grail, a small functional language which which we use to represent bytecode programs for the Java Virtual Machine

Extraction of complexity bounds from first-order functional programs.

Roberto Amadio
Université de Provence - amadio@cmd.univ-mrs.fr

Quasi-interpretations are a tool to bound the size of the values computed by a first-order functional program (or a term rewriting system) and thus a mean to extract bounds on its computational complexity. We study the synthesis of quasi-interpretations selected in the space of polynomials over the max-plus algebra determined by the non-negative rationals extended with -infinity and equipped with binary operations for the maximum and the addition. We prove that in this case the synthesis problem is NP-hard, and in NP for the particular case of multi-linear quasi-interpretations when programs are specified by rules of bounded size. The relevance of multi-linear quasi-interpretations is discussed by comparison to certain syntactic and type theoretic conditions proposed in the literature to control time and space complexity.

Typing with Conditions and Guarantees in LFPL

Michal Konecny
University of Edinburgh - mkonecny@inf.ed.ac.uk

Hofmann's LFPL is a functional language with constructs which can be interpreted as referring to heap locations. In this view, the language is suitable for expressing and verifying in-place update algorithms. Correctness of this semantics is achieved by a linear typing.

We review several non-linear typings of first-order LFPL programs which are more permissive than the linear typing. These systems have the traditional non-linear types extended with some annotations which can be efficiently inferred in an iterative process.

The annotations can be viewed as pre-conditions and rely-guarantees on the layout and change of the heap during evaluation. Their nature make the languages suitable for expressing and analysing in-place update algorithms with high levels of data sharing and reuse.

Capacity bounded computational ambients

Vladimiro Sassone
University of Sussex - vs@cogs.susx.ac.uk

We present a calculus of ambients where movements are constrained by the availability of space at destination. A type system guarantees precise bounds on the dimension of processes, so that ambients' capacities are never exceeded.

Capabilities for Mobility and Communication Control in Boxed Ambients

Michele Bugliesi
Università di Venezia - michele@dsi.unive.it

We study a variant of Boxed Ambients aimed at controlling interferences. Our calculus draws inspiration from Safe Ambients (SA) (with passwords) and modifies the communication mechanism of BA. Expressiveness is maintained through a new form of co-capability that at the same time registers incoming agents with the receiver ambient and performs access control. We show that new calculus has a rich semantics theory, including a sound labelled transition system characterisation, and an expressive, yet simple type system. Through a set of examples we characterise its expressiveness with respect to both BA and SA.

Principles for Entity Authentication

Riccardo Focardi Università di Venezia - focardi@dsi.unive.it

We study the roles of message components in authentication protocols. In particular, we investigate how a certain component contributes to the task of achieving entity authentication. To this aim, we isolate a minimal set of roles that enables us to extract general principles that should be followed to avoid attacks. We then formalize these principles in terms of rules for protocol parties and we prove that any protocol following these rules (i.e., designed according to the principles) will achieve entity authentication.

A Simple Language for Real-time Cryptographic Protocol Analysis

Roberto Gorrieri
Università di Bologna - gorrieri@cs.unibo.it

A real-time process algebra, enhanced with specific constructs for handling cryptogrpahic primitives, is proposed to model cryptographic protocols in a simple way. We show that some security properties, such as authentication, secrecy and integrity, can be re-formulated in this timed setting. Moreover, we hint that they can be seen as suitbale instances of a general information flow-like scheme, called tGNDC, parametric w.r.t. the observational semantics of interest. We show that, when considering time trace semantics, there exists a most powerful hostile environment (or enemy) that can try to compromise the protocol. Finally, we suggest a compositional proof principle that can help in proving the security of crypto-protocols.

Control Flow Analysis for Security Protocols

Flemming Nielson
Technical University of Denmark - nielson@imm.dtu.dk

Global applications often rely on communication through networks that cannot be considered secure. Consequently, they must rely on the security mechanisms offered by network protocols. Designing a correct protocol is, however, a difficult problem. In the talk we demonstrate how static analysis technology can be applied to validate the correctness of such protocols.

First, we perform a systematic expansion of protocol narrations into terms of a process algebra in order to make precise some of the detailed checks that need to be made in a protocol.

Next, we develop a *control flow analysis* of the process algebra that tracks message flow on the network and specify a "hardest attacker" in the style of Dolev and Yao. The combination of the control flow analysis of the protocol and the attacker leads to a polynomial-time validation procedure for security protocols.

The techniques suffice for identifying a number of authentication flaws in symmetric key protocols such as Needham-Schroeder, Otway-Rees, Yahalom and Andrew Secure RPC and also to guarantee the validity of suitably amended versions of the protocols.

KLAIM: a Framework for Reliable Network Aware Programming

Rocco De Nicola
Università di Firenze - denicola@dsi.unifi.it

We describe the original Klaim model. Klaim is a language that supports a programming paradigm where processes, like data, can be moved from one computing environment to another. The language consists of a core Linda with multiple tuple spaces and of a set of operators for building processes. Klaim naturally supports programming with explicit localities. Localities are first-class data (they can be manipulated like any other data), but the language provides coordination mechanisms to control the interaction protocols among located processes.

Hierarchical and Dynamic Nets in Klaim

Lorenzo Bettini
Università di Firenze - bettini@dsi.unifi.it

We present some extensions of the original Klaim model: the new hierarchical and open net based network model and the object-oriented extensions of Klaim. We also present the implementation of Klaim that consists in the Java package Klava, providing the run-time system for Klaim operations and the programming language X-Klaim that extends the kernel language with high-level linguistic constructs.

Resource Access and Mobility Control with Dynamic Privileges Acquisition

Rosario Pugliese
Università di Firenze - pugliese@dsi.unifi.it

We present muKlaim, a process calculus with distribution and mobility whose main features have been derived from the language Klaim. We introduce a type system that controls the activity of the processes in a net and deals with dynamic variations of security policies. Privileges can be acquired and consumed both by fixed and mobile entities, i.e. network nodes and agents. By using a combination of static and dynamic type checking, and of in-lined reference monitoring, our system guarantees the absence of run-time errors due to lack of privileges. We state two type soundness results: one involves whole nets, the other is relative to subnets of larger nets.

MetaKLAIM and beyond

Eugenio Moggi
Università di Genova - moggi@disi.unige.it

We describe the main integration issues addressed in MetaKlaim and the solution adopted for combining staging, dynamic type checking and Klaim's primitives. We will also discuss directions for further integration, such as: more flexible types (dynamics), more refined types (monadic types and effects), mobile ambients instead of Klaim's localities, dynamic checks at administrative boundaries (guardians)

Joint work with Gianluigi Ferrari and rosario Pugliese.

The MIDHA toolkit and the PROFUNDIS Distributed Verification Environemnt

Gianluigi Ferrari
Dipartimento di Informatica, Università di Pisa - giangi@di.unipi.it

The architecture of a toolkit performing state minimization of labelled transition systems for name passing calculi is presented. The structure of the toolkit, called MIHDA, is directly suggested by the abstract semantical structure of the coalgebraic specification of the partition-refinement minimization algorithm. The proof of correctness of the implementation directly follows from the semantical structures.

The MIHDA toolkit is one of the components of the PROFUNDIS distributed verification environment. 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 (users and toolkits) 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 toolkit interaction capabilities. Hence, verification sessions over the Profundis WEB are developed by combining and integrating together the services available over the WEB. The Profundis WEB is highly portable (it may adapt to a variety of infrastructures) and supports interoperability, dynamic reconfiguration and dynamic integration of several verification techniques.

The design and the prototype implementation of the infrastructure of the Profundis WEB has been the result of an active collaboration among Pisa and Uppsala.

Architectural Primitives for Distribution and Mobility

Jose Fiadeiro
University of Leicester - jose@fiadeiro.org

Antonia Lopes
University of Lisbon - mal@di.fc.ul.pt

We address the integration of a distribution dimension in an architectural approach to system development and evolution based on the separation between coordination and computation. This third dimension allows us to separate key concerns raised by mobility, thus contributing to our ability to handle the complexity that is inherent to developing systems that are required to be location aware or operate in networks for which the distribution topology can change in time.

Mutatis Mutandis: Formalising Dynamic Software Updating

Gareth Stoyle and Peter Sewell
University of Cambridge - {gareth.stoyle,Peter.Sewell}@cl.cam.ac.uk

We identify key features of Dynamic Software Updating (DSU) by formalising an Erlang-like language in an operational semantics. We show the uses of such a language (through examples) and discuss the limitations of the calculus. We identify some open problems and talk about our current work which attempts to address these issues.

Lexically scoping distribution: what you see is what you get

Antonio Ravara
DMIST - amar@math.ist.utl.pt

We define a lexically scoped, asynchronous and distributed pi-calculus with local communication and process migration. This calculus follows a simple model of distribution for mobile calculi, with a lexical scoping discipline guaranteeing that each free channel belongs to a specific site, fixed throughout computation, and regardless of where a process is running. This discipline provides both for remote communication and for process migration, making explicit migration primitives superfluous.

Interpreting mobile ambients into pi-calculus

Pierpaolo Degano
Università di Pisa - degano@di.unipi.it

In order to compare π-calculus like and ambients like formalisms, we define a general abstract machine that can interpret both π-languages and ambient languages which allow to study the relations between different calculi. We implement this abstract machine by mirroring the transition system of Ambient-like languages on the transition system of the π-calculus. The technique is mainly based on a run time support, defined by the enhanced operational semantics of the π-calculus that enriches labels of transitions with encodings of their proofs. The enhanced labels allow to retrieve many different but coherent views of the same system via relabeling of transitions. In the current study, we use enhanced labels to manage the nesting of the administrative domains, forbidding those actions which do not respect the domain configuration defined in the source language. Technically, we add side conditions to the semantic rules implementing the checks above. One of the most interesting feature of our technique is the defining of two levels of abstractions: the one represented by the input/output primitives which encodes the message exchanges and the other one represented by the information handled by the enhanced labels that encodes the dynamic hierarchical tree structure of ambients.

Joint work with Linda Brodo and Corrado Priami.

PEPA nets: A dynamic stochastic modelling formalism

Stephen Gilmore
LFCS, Edinburgh - stg@dcs.ed.ac.uk

The DEGAS project developed the PEPA nets modelling language to investigate the performance of global computing applications via the analysis of stochastically timed process algebra models of mobile code systems.

PEPA nets extend the PEPA stochastic process algebra by allowing PEPA components to be used as the tokens of a coloured stochastic Petri net. Communication between tokens at different places is not allowed; for tokens to communicate they must first meet at one of the places of the net. While they are together in the same place they may communicate but when one of them moves to another place their communication must cease until another firing of the net brings them together again in the same place.

In addition to the PEPA components which are used as tokens, a PEPA net contains *static components* which are resident in one of the places of the net and are unable to move. Static components may communicate with tokens while they are present or they can communicate with other static components in

the same place. Communication between static components at different places is not allowed.

Restricting communication according to these rules provides a credible formalism for representing mobile code systems where the tokens of a PEPA net represent stateful mobile objects under a system of dynamic binding of names. A PEPA component has local state which can be modified by performing timed actions, either individually or in cooperation with another component. These activities make up an interface which is analogous to the interface which is provided by the methods which can be invoked on an object.

The static components in a PEPA net represent immobile parts of the system which may be either hardware (such as servers) or software (such as databases). Modern real-world systems are typically made up of a mixture of mobile and immobile components in this way.

The PEPA stochastic process algebra is supported by a range of tools including the PEPA Workbench, the Möbius Modelling Framework, and PRISM. We have extended the PEPA Workbench to implement PEPA nets. We will describe how the other PEPA tools can be used to analyse PEPA net models.

This is joint work with Jane Hillston, Leïla Kloul and Marina Ribaudo.

Control of resources in Ambients

David Teller
ENS Lyon - david.teller@ens-lyon.fr

Current software and hardware systems, being parallel and reconfigurable, raise new safety and fiability problems, and the resolution of these problems requires new tools. Numerous formal or technical methods attempt at reducing these threats. In this talk, we present an enrichment of the calculus of Mobile Ambients to obtain a formalism and a type system for this model, which make resource control possible for distributed and mobile systems. Such a control may be used to prevent numerous bugs and avoid Denial of Service attacks.

Towards a Formal Model for Trust

Marco Carbone and Mogens Nielsen
University of Aarhus - {carbone,mn}@brics.dk

We present a proposal for a formal model of trust in the Global Computing setting. The model focuses on the aspects of trust formation, evolution, and propagation. The model is based on partial order structures for trust, building on concepts and techniques from domain theory. Two important partial orderings are identified: the information ordering and the trust ordering, and a particular generic construction of models is being investigated, the so-called

interval construction. Various reasoning techniques are presented, and the expressive power of the model is illustrated on a few examples.

The essence of principal typings

Joe Wells
Heriot-Watt University - jbw@macs.hw.ac.uk

We present a new general definition of principal typings which does not depend on the details of any particular type system. We show that the new general definition correctly generalizes previous system-dependent definitions. We explain why the new definition is the right one. Furthermore, the new definition is used to prove that certain polymorphic type systems using forall-quantifiers, namely System F and the Hindley/Milner system, do not have principal typings.

Dynamic object re-classification: design and implementation of Fickle

Davide Ancona, Elena Zucca
Università di Genova - {davide,zucca}@disi.unige.it

Christopher Anderson and Sophia Drossopoulou
Imperial College-DART - {cla97,scd}@doc.ic.ac.uk

Mariangiola Dezani-Ciancaglini and Ferruccio Damiani
Università di Torino - {dezani,damiani}@di.unito.it

Re-classification changes at run-time the class membership of an object while retaining its identity. We suggest language features for object re-classification, which could extend an imperative, typed, class-based, object-oriented language. We present our proposal through the language Fickle. The imperative features combined with the requirement for a static and safe type system provided the main challenges. We develop a type and effect system for Fickle and prove its soundness with respect to the operational semantics. In particular, even though objects may be re-classified across classes with different members, they will never attempt to access non-existing members. To demonstrate that an extension of Java supporting dynamic object re-classification could be fully compatible with the existing Java environment, we present a translation from Fickle into a subset of Fickle without re-classification (that is a core Java-like language). The translation is proved to preserve static and dynamic semantics; moreover, it is shown to be effective, in the sense that the translation of a Fickle class does not depend on the implementation of used classes, so it can be done without having their sources, as it happens for Java compilation.

The problem of separate typechecking, separate code generation and principal typings in Java-like languages

Davide Ancona, Giovanni Lagorio and Elena Zucca
Università di Genova - {davide,lagorio,zucca}@disi.unige.it

As pointed out by Cardelli, separate compilation can be expressed by a typing judgment E —- S : t , where E is a type environment, S is a single source fragment, t is the inferred type, and other fragments used in t are not required to be available, but only requirements on them are specified in E. Though Java and C-sharp are usually considered typical examples of concrete programming environments supporting a mechanism of separate compilation, this mechanism does not correspond to true separate compilation in the sense above. Indeed, in Java and C-sharp used fragments (classes) must be available either in source or binary form (type information in E is extracted from code), and compilation is propagated to some of them. Moreover, requirements in E are far from being the minimal necessary in order to typecheck S, as it would be desirable in a context of dynamic assembly, where we want be able to safely replace a fragment by another if they intuitively satisfy the same assumptions (formally, have the same typings), without any need of re-compilation or even, in mode advanced scenarios, during system execution. We present an alternative compilation mechanism for Java-like languages supporting true separate compilation and weaker requirements on used fragments than those used by standard compilers, and describe a prototype implementation. Moreover, we analyze the possibility of finding principal typings for languages such as Java and C-sharp, and the related problem of generating code with type annotations needed for dynamic typechecking.

M^3 – Mobility Types for Mobile Processes in Mobile Ambients

Mario Coppo, Mariangiola Dezani-Ciancaglini, Elio Giovannetti and Ivano Salvo
Università di Torino - {coppo,dezani}@di.unito.it

We present an ambient calculus in which some of the most known calculi of concurrency and mobility can be encoded in a natural way. The calculus comes equipped with an incremental type system which allows typing components in possibly incomplete environments. Types are intended as a way of controlling both the kind of values exchanged in communications and the access and mobility properties of processes. A type inference procedure determines the "minimal" requirements to accept a system or a component as well typed. This

gives a kind of principal type.

A family of low communication, scalable and fault tolerant infrastructures for P2P applications

Luc Onana and Seif Haridi
IMIT KTH - {onana,seif}@imit.kth.se

Sameh El Ansary and Per Brand
Swedish Institute of Computer Science AB - sameh,perbrand@sics.se

We present DKS(N,k,f), a family of infrastructures for building Peer-to-Peer applications. Each instance is a fully decentralized overlay network characterized by three parameters: N the maximum number of nodes that can be in the network; k the search arity within the network and f the degree of fault-tolerance. Once these parameters are instantiated, the resulting network has several desirable properties. The first property is that there is no separate procedure for maintaining routing tables; instead out-of-date or erroneous routing tables is eventually corrected on the fly, thereby eliminating unnecessary bandwidth consumption. The second property is that lookup requests are resolved in at most logk(N) overlay hops under normal operations. Third, each node maintains only (k-1)logk(N)+1 addresses of other nodes for routing purposes. Fourth, new nodes can join and leave at will with negligible disturbance to the ability to resolve lookups in logk(N) hops on average. Fifth the probability of getting failure for a pair key/value that was inserted in the system is negligible, even if f consecutive nodes fail simultaneously.

Extending UML to Model Mobile Systems

Martin Wirsing and Hubert Baumeister
LM Universität München - {wirsing,baumeist}@lmu.de

In this talk we present the UML-extensions to model mobility developed within the AGILE project. The pure UML notation can be hardly used for specification of mobility. Therefore extensions of UML class diagrams, activity diagrams, sequence, and statechart diagrams have been proposed which allow one to conveniently specify mobile objects systems. These extensions use stereotypes to model mobile objects, locations, and activities like moving or cloning.

For activity diagrams we offer two notational variants for modeling mobility. One variant is location centered and focuses on the topology of locations. The other one focuses on the actor responsible for an activity.

Sequence diagrams for mobility model the behavior of mobile objects using a generalized version of lifelines. For different kinds of actions like creating, entering or leaving a mobile object stereotyped messages are used. The notation

provides also a zoom-out, zoom-in facility which allows one to abstract from specification details.

The extension of UML-statecharts is based on relaxing the unique association between a UML-statechart and its input-queue. The resulting communication paradigm is much more flexible than the original, asymmetric one and is well suited for the modeling of mobility-oriented as well as fault tolerant systems.

UML profiles for global computing

Perdita Stevens
University of Edinburgh - perdita@inf.ed.ac.uk

In order to meet the DEGAS project's aim of permitting software designers to use formal tools in conjunction with standard UML tools to model and verify performance and security aspects of global applications, we need to consider the expressiveness of UML. Standard UML as taken "off the shelf" does not have the ability to record all the information that will be required in the global applications domain. This talk will explain the relevance of the "UML profile" mechanism, and go on to discuss the specific aspects we have considered in drafting a DEGAS profile for UML. Wherever possible, we build on preexisting work; we also collaborate with the AGILE project.

Software architectures, Global Computing and Graph Transformations

Ugo Montanari
Dipartimento di Informatica, Università di Pisa - ugo@di.unipi.it

Work in collaboration with Gianluigi Ferrari, Dan Hirsch, Ivan Lanese and Emilio Tuosto.

Graphical notations are widely accepted as an expressive and intuitive working tool for system specifications and design. This talk outlines a declarative approach based on synchronized graph transformations to deal with the modeling of Wide Area Network applications. In Synchronized Hyperedge Replacement (SHR) systems [1], productions rewrite hyperarcs into graphs, provided the actions they offer on the adjacent nodes satisfy certain synchronization conditions. Further work on SHR added mobility [4], namely the ability of passing nodes while synchronizing. This made possible to develop graphical calculi in the pi-calculus [6] and ambient calculus [2] style, possibly equipped with a compositional bisimilarity semantics [7], with applications to software architecture [4,5].

Ongoing work on the subject studies how to express the structure of wide area networks in the SHR framework [3], and how to handle difficult issues

concerning security and access control, routing, and network reconfiguration. Also it has been shown that SHR can be translated into logic programming [8], allowing for the application of tools and techniques available in that framework.

References

[1] Degano, P. and Montanari, U. A Model of Distributed Systems Based on Graph Rewriting, Journal of the ACM Vol. 34, N2, April 1987, pp. 411-449.
[2] Ferrari, G.L., Montanari, U. and Tuosto, E., A LTS Semantics of Ambients via Graph Synchronization with Mobility, in: Antonio Restivo, Simona Ronchi Della Rocca and Luca Roversi, Eds., ICTCS 2001, Springer LNCS 2202, October 2001, pp. 1-16.
[3] Ferrari, G., Montanari, U. and Tuosto, E., Graph-based Models of Internetworking Systems, Proc. 10th Anniversary UNU/IIST Colloquium, Springer LNCS. To appear.
[4] Hirsch, D., Inverardi, P. and Montanari, U., Reconfiguration of Software Architecture Styles with Name Mobility, in: Antonio Porto, Gruia-Catalin Roman, Eds., Coordination 2000, Springer LNCS 1906, pp. 148-163.
[5] Hirsch, D. and Montanari, U., A Graphical Calculus for Name Mobility, Proc. Workshop on Software Engineering and Mobility, co-located with ICSE 2001, May 2001, Toronto.
[6] Hirsch, D. and Montanari, U., Synchronized Hyperedge Replacement with Name Mobility, in: Kim Larsen and Mogens Nielsen, Eds., CONCUR 2001 - Concurrency Theory, Springer LNCS 2154, pp.121-136.
[7] Koenig, B. and Montanari, U., Observational Equivalence for Synchronized Graph Rewriting, in: Naoki Kobayashi and Benjamin Pierce, Eds., TACS 2001, Springer LNCS 2215, pp. 145–164.
[8] Lanese, I. and Montanari, U., Software Architectures, Global Computing and Graph Transformation via Logic Programming, in: Leila Ribeiro, Ed., Proc SBES'2002 - 16th Brazilian Symposium on Software Engineering, October 2002, Anais, p. 11-35.

Specification and Analysis of Mobile and Distributed Systems using Graph Transformation

Andrea Corradini
Dipartimento di Informatica - Università di Pisa - andrea@di.unipi.it

In this talk we first summarize the basics of the algebraic approaches to Graph Transformation and of their concurrent semantics that was developed in the recent years. The relationship with the theory of Petri nets is emphasized.

Next we discuss some advantages and drawbacks of the use of Graph Transformation for specifying distributed systems, showing that even if some features of the formalism match perfectly some typical properties of such systems, still a hard work is necessary for tailoring a specific approach to graph transformation to the kind of systems which are of interest in the AGILE project. We indicate

some directions of future work around the Object-Based Graph Transformation approach, which looks promising in this perspective.

Finally we describe some verification techniques for graph transformation systems developed recently, and based on finite approximations of the (usually infinite) unfolding.

Coalgebraic models for mobile systems and systems with spatial structures

Ugo Montanari
Dipartimento di Informatica, Università di Pisa - ugo@di.unipi.it

Luis Monteiro
Departamento de Informática Universidade Nova De Lisboa, lm@fct.unl.pt

In the talk we shortly introduce the ordinary notion of labeled transition systems (LTS) as coalgebras in the category Set, and then we describe two proposal about how to equip LTS with additional structures which are relevant for modeling compositional, mobile distributed systems.

In the first part (presented by Luis Monteiro) we describe an approach to the study of transition systems in which the set of states has been endowed with a structure intended to capture the spatial organization of states in a broad sense. The traditional approach has been to consider the set of states as an algebra for an appropriate set of operators. Instead, we propose to treat 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 to construct new states. 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. In the talk we discuss the so-called intensional character of spatial logic from the point of view of transition systems with spatial structure. We also point out that such systems can be viewed as non-interleaving models of concurrency.

In the second part (presented by Ugo Montanari) we consider *structured coalgebras* (previously studied by Turi and Plotkin, and by Corradini, Heckel and Montanari) which are algebras and coalgebras at the same time. When an ordinary LTS can be lifted to a structured coalgebra, not only a minimal realization exists, but it is also a structured coalgebra, i.e. bisimilarity is guaranteed to be a congruence. This means that abstract semantics is compositional. In the talk we describe rather intuitive sufficient conditions for the existence of the lifting, which can hold also in the presence of structural axioms. This framework is convenient for modeling mobile systems, since name substitutions can be described as operations of the algebra and suitably axiomatized, while name passing and name extrusion can be also easily expressed. Applications to mo-

bile calculi like the π calculus are described, where bisimilarity can be defined without side conditions on the generated names and minimal realizations can actually be computed in finite cases. Also possible extensions including name fusions and general functional substitutions will be discussed.

A Brief History of Authentication

Dieter Gollmann
Microsoft Cambridge - diego@microsoft.com

This talk will examine how the term 'authentication', a central concept in security, has developed and changed its meaning over time. We will use this example to illustrate some of the pitfalls and opportunities when applying formal methods for security analysis.

Probabilistic Multicast

Patrik Eugster and Rachid Guerraoui
Ecole Polytechnique Fdrale de Lausanne - {patrick.eugster,rachid.guerraoui}@epfl.ch

We present here a gossip-based multicast algorithm, called Probabilistic Multicast (pmcast), which deals with the cast of multicasting events only to subsets of processes in a large group. Our pmcast algorithm relies on specific orchestration of processes in the group, which can be viewed as a "superimposition" of spanning trees. The resulting compound "tree" exploits (1) the topology of the network, and at the same time (2) commonalities in the interests of processes, and adapts easily to variations in the composition of the group. The scalability in terms of network resources, inherit to gossip-based algorithms, is combined with (3) a notion of membership scalability, and further improved by (4) mainly involving those processes in the dissemination of an event that are interested in that event.

Application Scenarios

Giovanna Di Marzo Serugendo
University of Geneva - Giovanna.Dimarzo@cui.unige.ch

This talk presents the application scenarios that the SECURE consortium has identified for validating the project results. These applications are likely to be implemented by the SECURE partners. They are wide ranging in application domain, underlying technology, and trust-risk issues, and will serve to test the security model proposed by SECURE. Each application scenario is described together with its implementation status. The talk ends with some words on the Lana mobile agent platform that has been chosen for implementing most of the scenarios and for instantiating the security framework.

Towards a Framework for Assessing Trust-Based Admission Control in Collaborative Ad Hoc Applications

Christian Jensen
Technical University of Denmark- Christian.Jensen@imm.dtu.dk

The proliferation of mobile devices and the development of ad hoc networking technologies has introduced the possibility of a vast, networked infrastructure of diverse entities partaking in collaborative applications with each other. However, this may require interaction between users who may be marginally or completely unknown to each other, or interaction in situations where complete information is unavailable.

Humans use the concept of trust to help decide the extent to which they cooperate with others. It provides a mechanism for lowering access barriers and enables complex transactions between groups. Trust, however, takes many different forms and is difficult to stringently define or understand.

The aim of our work is to develop a trust framework that enables access control based on trust-based admission control policies that define the trust relationship between entities in collaborative ad hoc applications. We do this by integrating a trust-formation element into an admission control mechanism to manage interaction between previously unknown users. Initial evaluation of this framework is based on a simple distributed blackjack card game application, which implements the trust-based admission control system to assign roles to users according to their trust-based admission rights. Results of the evaluation show that the trust-based admission control system reacts correctly to user behaviour, i.e. the system adjusts trust values and implements admission policies accordingly.

Risk and Trust in Global Computing

Ken Moody
University of Cambridge - Ken.Moody@cl.cam.ac.uk

We present a risk model for SECURE and how it interacts with the computational trust model. The risk model, based on existing models used in safety-critical programming and the insurance industry, assesses risk for trust-mediated actions on a per-outcome basis. The risk of an outcome is defined as the likelihood of the outcome and the financial cost or benefit if it does occur. We have used the risk model in a number of applications including an augmented city, transparent PDA collaboration, trust-based routing, a P2P backup system and unified messaging. A selection will be presented.

Generalizing Chord: S-Chord and other approaches

Valentin Mesaros and Peter Van Roy
Université Catholique de Louvain - {valentin,pvr}@info.ucl.ac.be

Bruno Carton
CETIC asbl - bruno.carton@cetic.be

Efficient data location in peer-to-peer systems has become the subject of many research theses. Chord is one of the simplest peer-to-peer systems that addresses this issue. Despite its simplicity, one of the main limitations remains the asymmetric organization of its routing. This leads to problems like inability to make in-place notifications of routing changes, and incapacity to support symmetric applications and to exploit efficiently network proximity. As a solution to this limitation, we propose S-Chord, an extension of Chord. In S-Chord the routing table is organized in symmetric manner, and the circular search space can be walked through bidirectionally. The S-Chord results, for the worst case, in an improvement of lookup efficiency of 25%, compared to Chord with the same size routing table. Furthermore on average, assuming a uniform distribution of queries, S-Chord results in a 10% improvement.

DIT presentation

The Department of Information and Communication Technology (DIT) was established at the University of Trento in January 2002. It represents the point of aggregation of the skills on information and communication technology and intends to provide a dynamic and qualified response to the ever-increasing demand of such competences from the productive tissue at local, national or international level.

These targets can be achieved since people working at DIT are able to exploit the different skills of several research groups operating on various disciplines (computer science, telecommunications and electronics), traditionally connected to different scientific communities.

On the other hand, due to the wide range of knowledge provided by the different research groups, DIT can be thought as an important structure for the development and modernization of the traditional scientific disciplines. Such an interdisciplinary support is completed by a leading action in terms of services and of knowledge "management" in order to propose a national and international efficient model of technological transfer. The Department aims at being a model of integrated scientific innovation able to cope with the increasing demand of the so-called information technology society. Such a role requires excellence in both scientific research (papers) and its practical exploitation (industrial projects). In the starting phase such requirements have been reached thanks to a staff of 30 professors and researchers affiliated to several Faculties of the University of Trento.

International Graduate School in Information and Communication Technologies . The International Graduate School in Information and Communication Technologies is organized by DIT in collaboration with other research partners.

The educational offerings and the research opportunities are directed to the acquisition of the necessary skills for students to perform research and development at universities, public or private research institutes, and industry. In this way, a Faculty made up of internationally recognized Italian and foreign professors, who are responsible for the educational activities and take part in the organization of the doctoral program (the doctorate course has a minimum length of three years and consists of courses during the first two years, while the third is dedicated to the writing of the dissertation).

The ICT doctorate also involves research into integration with industry and technology transfer. The research topics are either theoretical or applied and the dissertations cover the continuum between innovative industrial and theoretical research. The involvement of industry guarantees the practical value of the research, while the University guarantees the innovative aspects as well as the necessary scientific background. This actual collaboration is also demonstrated by the research grants covered by research projects and other partners.

With regard to the numbers attracted to the course, in the 2002/2003 academic year more than eighty requests for admission were received by the school

of doctorate research. Students came from European and non-European countries, as well as from Italy. The official language of the course is English. The University offers Italian and non-Italian students an increased study grant. In addition, the agreements underwritten between the Autonomous Province of Trento, the Minister for Foreign Affairs, the Italian Embassy in India and the Italian Embassy in Bulgaria have provided, since the start, dedicated study grants which can be used by students from these countries.

There are currently 64 students registered, coming from European and non-European countries, as well as from Italy. In average more than 50% of students are from abroad.

In the near future, further integration will take place with Specialist Degrees whose courses (given in English) will make up the critical mass needed for the operation of the Doctorate in the ICT as a graduate school like many others in the US or the UK.

Research at DIT. Research in the Department will be on the basis of technological transfer with other international institutions whether public or private.

It is organized into Programs of Research, the objectives of which are to combine the skills present within DIT with joint strategic lines of development. The research programs last for three years and are renewable. They involve all the interdisciplinary skills necessary to reach the planned objectives.

The research programs of the current structure for the research plan for the three-year period 2002-2005 are described in more detail below:

- *Bioinformatics*: the objective of this program is the construction of tools for modelling, analysing, and simulating complex biomolecular systems thus establishing firm grounds to develop computational methods in systems biology. The classical algorithmic aspects of bioinformatics are studied as well.

- *Computer Networks and Distributed Algorithms*: the objective of this program is the design and analysis of algorithms for problems of combinatorial optimization applied to planning and management of computer networks. The current focus of research is wireless networks and optical networks.

- *Multimedia Communications and Networking*: the objective of this program is the analysis, processing and coding of audio and video signals and the analysis of the benefits of multimedia networks. The current focus is on the management of multimedia signals both in a fixed network (video retrieval, picture archives) and in mobile networks (wireless multimedia).

- *Remote Sensing for the Environment*: the objective of this program is the development of methodologies for the analysis and archival of remotely surveyed signals and images in Geographic Information Systems (GIS). The current focus of research is the development of advanced techniques for the analysis of data from a variety of sources (optics, radar, laser,

GIS) for the management of natural resources and the prevention and monitoring of environmental disasters.

- *Software Engineering and Formal Methods*: the objective of this program is the development of systems and methodologies for Software Engineering and the development of formally certified systems for specific needs and security requirement. The current focus of research is the development of tools for specifics, verification, validation and automation of software development techniques for distributive systems.

- *Evaluation and Testing in Industrial Processes*: the objective of this program is the development of technologies and systems for the evaluation and monitoring of industrial processes and products by means of non-invasive analysis. The current focus of research is the study of techniques based on electromagnetic diagnostics and signals analysis, as well as the definition of protocols for the quantitative evaluation of system parameters. It is planed to expand this program in 2002.

- *Knowledge Management and Distributed Information Systems*: the objective of this program is the study and design of distributed services and the development of methodologies, analysis and tools used in knowledge management in distributed systems. The current focus of research is the development of languages, algorithms and tools for this problem.

- *Electronic Sensors and Microsystems*: the objective of this program is the study and the design of electronic systems, in various scales of integration, for the acquisition and analysis of heterogenic signals. The current focus of research is large-scale system integration (VLSI), as well as the realization of electronic and optical-electronic devices, on the electronic elaboration of multi-dimensional signals.