Lecture Notes in Computer Science 2874

Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Hong Kong
London
Milan
Paris
Tokyo

Corrado Priami (Ed.)

Global Computing

Programming Environments, Languages, Security, and Analysis of Systems

IST/FET International Workshop, GC 2003
Rovereto, Italy, February 9-14, 2003
Revised Papers

Springer

Volume Editor

Corrado Priami
Università di Trento, Dipartimento di Informatica e Telecomunicazioni
Via Sommarive, 14, 38050 Povo (TN), Italy
E-mail: priami@dit.unitn.it

Cataloging-in-Publication Data applied for

A catalog record for this book is available from the Library of Congress.

Bibliographic information published by Die Deutsche Bibliothek
Die Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data is available in the Internet at <http://dnb.ddb.de>.

CR Subject Classification (1998): D.2, D.3, F.3, D.4, D.1, C.2

ISSN 0302-9743
ISBN 3-540-20583-7 Springer-Verlag Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Olgun Computergrafik
Printed on acid-free paper SPIN: 10969496 06/3142 5 4 3 2 1 0

Preface

The goal of the IST/FET proactive initiative on Global Computing is to obtain models, frameworks, methods, algorithms to build systems that are flexible, dependable, secure, robust and efficient. The dominant concerns are 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 controlled by different owners.
- The computational entities are mobile, due to the movement of the physical platforms or 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 behavior 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 analyzing such systems.

The workshop covered the aspects related to languages and programming environments as well as analysis of systems and resources involving nine projects (AGILE, DART, DEGAS, MIKADO, MRG, MYTHS, PEPITO, PROFUNDIS, SECURE) out of the 13 founded under the initiative. After a year from the start of the projects, the goal of the workshop was to determine the state of the art in the topics studied in the two clusters related to programming environments and the analysis of systems, and to devise strategies and new ideas to profitably continue the research effort towards the overall objective of the initiative.

Before starting the technical contribution, we gave a brief description of the nine projects involved in the meeting.

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

Rovereto,
15 September 2003

Corrado Priami

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 onto 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; and
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 capitalize 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 semiformal modelling techniques, graph-based semantics, and software development in business domains characterized 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 capitalizing 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 objects capable of changing their behavior, e.g., by changing class, as well as calculi that are environment adaptable, e.g., able to test the 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 into five workpackages (WPs). The first three WPs
1. Frameworks and Calculi for Dynamic Software Assembly,
2. Flexible and Compositional Type Systems, and
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, and
5. Flexible Dynamic Type-Checking for Dynamic Software Assembly
are downstream (their feasibility will be reassessed at the first review point at 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) from 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 (semiformal) 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 from 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 many formal details from the designer as possible. The static and dynamic analysis with case studies should lead to the definition of new linguistic constructs and new models to analyze and reason about the performance and security of global systems.

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

- WP3 (UML feasibility, modification and tool customization) customizes a tool to build the designer's interface and manipulate UML models.
- WP4 (extraction, reflection and integration) 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 the 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 also exploits fine-grain models in which security and quantitative issues coexist.
- WP6 (static analysis) is responsible for specifying analysis in the flow logic and abstract interpretation approaches to determine 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) a 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 and 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 more trustworthy systems, to demonstrate their conformance to specifications, and to analyze their behavior.

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 more trustworthy systems, to demonstrate their conformance to specifications, and to analyze their behavior; and
- 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 organized around three technical work-packages (WP1–WP3) and one organizational 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. These 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 behavioral specifications. The latter will require the definition of novel co-inductive techniques for comparing the distributed behavior 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:

- a virtual machine technology to support WP1's core programming model together with WP2 typing schemes; and
- language features and language extensions supporting WP1's model and WP2's type systems.

MRG

Full title: **Mobile Resource Guarantees**
Contact person: Sannella, Donald – 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 behavior. These certificates will be condensed and formalized 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 parameterizations allowing for mobile virtual machines.
Objective 2: Development of a notion of formalized 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 endeavor.

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 behavior. These certificates will be condensed and formalized 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 formalized resource policy (cost models);
2. define notions of independently verifiable certificates (resource- sensitive program logic with proof objects);
3. foundations for efficient generation of certificates (type systems, identification of useful programmer annotations); and
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 run-time 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 formalized certificates; and
4. parameterization by arbitrary run-time 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 – Department of Informatics, University of Sussex

Objectives: Global computing refers to computation via the sharing of an open-ended, distributed network of mobile resources by agents of all sorts. The systems range from large mainframes to mobile computers embedded in your cellphone or credit card, and agents are not tied to any specific geographical or logical network location.

The main scientific and technological challenge in this setting is that agents must operate in environments about which they possess little information, and where no a priori trustworthy agents exist. Like Pinocchio, in such conditions it is all too easy to entrust your money to the cat and the fox. The global infrastructure can only be successful if it provides adequate security guarantees. Administrative domains will want to grant access only to selected agents, and these will need to protect themselves and their data from attacks while traversing potentially hostile environments or executing remotely outside the control of their originating locations.

The overall aim of MYTHS, in short, will be to develop type-based foundational theories of security for mobile and distributed systems in order to lay the foundations for the design of robust, high-level programming paradigms for global computing.

Description of the work: MYTHS is a three-year-long project involving three partners and is articulated in the three themes below:

- *Resource access control*, i.e., the control of access to and proper use by mobile agents of computational resources distributed on the network and, possibly, not centrally owned.
- *Information flow control*, i.e., the monitoring of how information flows inside systems and whether such flows comply with the set security policies and clearance levels, such as public, restricted, and top-secret.
- *Analysis of cryptographic protocols*, i.e., the study of the correctness of protocols designed to establish secure (encrypted) communication channels.

These are central, challenging issues for global computing, with far-reaching impact on the development of high-level, reliable, network-aware programming languages. To make the network useful at all, it is imperative to ensure privacy, confidentiality, integrity and authenticity of electronic interactions, and to be able to detect or build safeguards against unwanted flows of information.

The glue that weaves themes together is provided by the pivotal notions of *models* and *types*, whence the project's title. MYTHS will develop formal models for distributed and mobile code environments based on high-level process calculi and will develop type theories to control resources and information flow, and to undertake crypto analysis.

The work is organized into workpackages (WPs). At project start, *WP1*: *Core Models* will analyze existing models and extend them to MYTHS's purposes. The other first-phase activities focus on modelling agents' behaviors and interactions for the analysis and enforcement of security in each theme.

- *WP2*: *Typed Calculi of Capabilities.* Extends the notion of capability for resource access to global computing, and devises type systems to enforce capability management policies and detect violations.
- *WP3*: *Types for Information Flow Control.* Identifies the flow of information determined by mobility, communication and cryptoprimitives, investigates semantic characterizations of it, and develops type systems for noninterference.

– *WP4*: *Types for Protocol Analysis.* Defines typed calculi and analysis techniques for cryptographic protocols (especially for e-commerce), and applies type systems for noninterference to protocol analysis.

This work leads to the central phase of the project, *WP5*: *Typing with Partial Knowledge*, where MYTHS extends its results to networked environments with no centralized control and in which only partial knowledge of the components of the networks may be assumed. Also, *WP6*: *Mutable Trust and Security Levels* investigates the management of dynamic trust levels. The project concludes with *WP7: Programming-Level Applications*, focusing on the convergence of the results achieved in the three themes and how these can collectively be applied to high-level programming languages and paradigms.

Expected results. MYTHS's results fall under the following four captions.

– Type systems for resource access control and management of capabilities. These will help devise alternative programming paradigms for global computing and design the corresponding programming languages and applications.
– Type systems for information flow security. These will bring advances relevant to the design and production of security middleware.
– Type systems for protocol analysis. The results on protocol analysis will be beneficial for the design and production of cryptographic protocols, verification tools, and e-commerce and e-business applications.
– Programming-level applications. This research is explicitly concerned with pointing out programming-level constructs for secure programming for global computing; its impact if successful is thus obvious.

Most of the project's outcomes will be in the form of scientific papers. We expect, however, to deliver prototype implementations of type checkers and verification tools based on them.

Project's partners: University of Sussex, UK (Coordinator); École Normale Supérieure, Paris, France; Università "Ca' Foscari," Venice, Italy.

PEPITO

Full title: **Peer-to-Peer-Implementation-and-Theory**
Contact person: Sjöland, Thomas – Swedish Institute of Computer Science

Traditional centralized system architectures are ever more inadequate. We lack a good understanding of future decentralized peer-to-peer (P2P) models for collaboration and computing, of both how to build them robustly and what can be built. The PEPITO project will investigate completely decentralized 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 multiconsistent 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 devising 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 addresses);
- allowing individuals to publish information and services, and allowing individuals to collaborate while remaining anonymous.

Work description: PEPITO will assume a completely decentralized 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 (decentralized algorithms for diffusing information, and for coping with multiconsistent 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 the 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 – University of Uppsala

PROFUNDIS aims at developing methods to analyze the behavior 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 behavior 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 the correct behavior 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 behaviors, 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 behavioral

equivalences and preorder 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 the 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 – Trinity College 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 decentralized 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, including 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 decentralized 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, for instance, the trust lifecycle; how security policy can be expressed in terms of trust and how access control can be 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.

Table of Contents