INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team ARLES

Software Architectures and Distributed Systems

Rocquencourt

THEME COM

Activity Report

2006

Table of contents

1. Team

Head of project-team
Valérie Issarny [Research director, INRIA, HdR]

Administrative assistant
Emmanuelle Grousset [INRIA]

INRIA research scientist
Nikolaos Georgantas [Junior research scientist]
Pascal Poizat [Assistant Professor, University of Evry, France, *en délégation* since September 2006]

Visiting scientists
Liam McNamara [PhD Student, University College London, United-Kingdom, October-December 2006]
Davy Preuveneers [PhD Student, KU Leuven, Belgium, June-August 2006]
Apostolos Zarras [Lecturer, University of Ioannina, Greece, July 2006]

Project technical staff
Sébastien Bianco [*Ingénieur Associé*, INRIA, from September 2006]
Damien Charlet [*Ingénieur Expert*, INRIA-Alcatel, until August 2006]
Rafik Chibout [*Ingénieur Expert*, INRIA-UBISEC & Alcatel, until August 2006]
Pierre-Guillaume Raverdy [*Ingénieur Expert*, INRIA-UBISEC & PLASTIC]
Letian Rong [*Ingénieur Expert*, INRIA-Thalès, from December 2006]
Daniele Sacchetti [*Ingénieur Expert*, INRIA-Amigo, until October 2006]
Ferda Tartanoglu [*Ingénieur Expert*, INRIA-Amigo & Thalès, until October 2006]
Graham Thomson [*Ingénieur Expert*, INRIA-Amigo, from September 2006]

Postdoc Researchers
Mauro Caporuscio [INRIA, from February 2006]
Gaogang Xie [*Fondation Franco-Chinoise pour la Science et ses Applications* (FFCSA), Part time]
Thomas Zahn [INRIA, from September 2006]

Ph.D. Students
Sonia Ben Mokhtar [INRIA fellowship, University of Paris 6]
Raghav Bhaskar [INRIA fellowship, joint PhD with the CODES project-team, Ecole Polytechnique, until June 2006]
Yérom-David Bromberg [INRIA fellowship, UVSQ]
Damien Fournier [INRIA fellowship, UVSQ]
Manel Fredj [INRIA fellowship, University of Paris 6]
Jinshan Liu [INRIA fellowship, UVSQ, until July 2006]
Roberto Speicys Cardoso [INRIA fellowship, University of Paris 6]

2. Overall Objectives

2.1. Overall Objectives

The development of distributed software systems remains a complex task, which is not only due to the systems' inherent complexity (*e.g.*, heterogeneity, concurrency), but also to the systems' continuous evolution (*e.g.*, integration of new technologies, changing environment in the mobile context). It is thus necessary to offer solutions to the two following issues:

- Supporting the rigorous development of distributed software systems by providing languages for systems modeling together with associated methods and tools for reasoning about the systems' functional and non-functional properties, and further mechanizing as far as possible the development process;
- Offering middleware platforms for both alleviating the complexity associated with the management of distributed resources and dealing with the efficient integration of new technological developments.

The ARLES project-team addresses the above two issues, investigating languages, methods, tools and middleware architectures to assist the development of distributed software systems that are efficient (in terms of both resource usage and delivered quality of service) and dependable. Our approach relies on the development of distributed systems from their architectural description. This choice is motivated by two factors:

- Our experience in architecture-based development of distributed systems has convinced us about the benefit of the approach regarding the robustness and performance of the resulting systems. The robustness of the system comes from the ability to practically exploit formal methods for modeling the systems' architectures and hence to reason about the behavior of the systems. The systems' performance results from the possibility to specialize the composition of the system according to both the applications' requirements and the runtime environment, and hence to integrate only necessary functionalities within the system, and further tune their realization according to available resources.
- Practically, the emergence of standard architectures for distributed systems and in particular supporting middleware, leads to the definition of reusable COTS (Commercial Off The Shelf) components for the implementation of both application-related and middleware-related functionalities. In addition, a number of systems are built by integrating legacy systems, as in particular witnessed in the context of information systems. The development of distributed systems thus becomes oriented towards the composition of system components and/or running system instances, which may be conveniently addressed at the system's architecture level.

The research activities of the ARLES project-team are more specifically centered around the development of distributed systems enabling the ambient intelligence vision. *Ambient intelligence* is an emerging user-centric service provision paradigm that aims to enhance the quality of life by seamlessly offering relevant information and services to the individual, anywhere and anytime. Systemically, this is realized as a synergistic combination of intelligent-aware interfaces, ubiquitous computing and ubiquitous networking. The intelligent-aware property of interfaces enables: (i) support of natural ways of interaction, *e.g.*, through speech and gesture; (ii) automatic adaptation to user's personal preferences; and (iii) proactiveness, stimulated by the presence of people, their location and their activities, instead of simple reactiveness to conventional ways of interaction, such as a keystroke or a mouse click. The ubiquitous (alternatively called pervasive) property of both computing and networking implies a useful, pleasant and unobtrusive presence of the system everywhere – at home, en route, in public spaces, in the car, at work, and wherever else the electronic environment support exists. The computing and networking facilities are distributed and accessible in wide varieties, as needed. The ubiquitous computing and networking model incorporates the mature paradigms of mobile and nomadic computing, and distributed systems.

While a number of base enablers such as wearable and handheld computers, wireless communication, and sensing mechanisms are already commercially available for deploying base infrastructures supporting the ambient intelligence vision, the development of ambient intelligence software systems still raises numerous scientific and technical challenges due to the specifics of ambient intelligence. In addition to traditional requirements for the software systems like dependability, the software systems shall deal with: mobility of users, increasing heterogeneity in devices, networks and software platforms, varying user and application requirements, diverse contexts of service provision, and natural interaction integrating multi-modal interfaces and exploiting knowledge about the user and his/her environment. The above requirements reveal the highly dynamic character of ambient intelligence systems, which should be accommodated by the overall software system architecture. Specifically, ambient intelligence software systems must comprehensively offer the following features: being self-adaptive according to the combined user-centric and computer-centric context so that service delivery continuously adapts to the highly changing situation of users, being dependable, and providing multi-modal interfaces for natural interaction with users. Developing systems with such features has given rise to extensive research since the end of the nineties, following the concern of seamlessly and effectively combining the numerous existing technologies for the benefit of users, as opposed to putting increased burden on them for mastering the increasing complexity of technologies. This concern is key to the ambient intelligence vision, as well as the ones of pervasive and autonomic computing. Despite the large interest of the research community in addressing the challenges raised by these visions since their emergence, numerous open issues that arose at that time are yet to be addressed. Within the ARLES research team, we aim at contributing to the realization of the ambient intelligence vision by providing software systems solutions that ease the development of supporting applications. Our research then encompasses the software engineering and distributed systems domains, as outlined below.

- **Software architectures for pervasive computing systems:** The development of distributed systems from the description of their architecture is now recognized as a sound approach. Among advantages, this enables exploiting architecture description languages and associated methods and tools for the thorough analysis of the systems' functional behavior and quality, and promotes the reuse of component systems. However, solutions to architecture-based development of software systems are mainly aimed at static systems whose component instances are known at design time. These solutions are thus not applicable to pervasive computing/ambient intelligence systems that are dynamically composed according to the users' situations. Still, architecture-based development constitutes a sound approach to the development of pervasive computing systems, which is in particular due to the composition of software systems that it supports. Part of our research intends to offer solutions to architecture-based development of distributed systems for ambient intelligence. We are in particular investigating architectural styles and modeling of pervasive computing systems, so as to allow the situation-sensitive composition of systems, while enforcing the systems' correctness with respect to offered functional and non-functional properties. As part of this effort, we are studying the service-oriented architectural style towards its adaptation to the requirements of ambient intelligence systems. Indeed, service-oriented architectures naturally support dynamic evolution and openness of the software system. The ambient intelligence vision further requires systems to be adaptive to the evolution of the environment, regarding both the computer- (*e.g.*, available network bandwidth) and user-centric (*e.g.*, preference of the user) context. Dependability of the software systems must also be carefully examined and enforced because the openness of the networking environment raises challenging dependability issues (*e.g.*, trust-based security as a solution to safely interact with unknown parties). Further, the ambient intelligence vision calls for making the computing systems transparent to users, which requires highly dependable software systems.
- **Middleware architectures for ambient intelligence systems:** Middleware architectures are key to the development of dependable, ambient intelligence systems. Middleware provides reusable, generic solutions to the management of the pervasive computing environment, ranging from the discovery of networked resources that keep changing (in particular from the standpoint of nomadic users), to their access. Hence, middleware alleviates the complexity of the pervasive computing environment and further promotes dependability by offering solutions reusable across systems. In order to truly enable dynamic, open pervasive environments, we are studying interoperable middleware for service-oriented distributed systems, which shall allow networked software services that are based on heterogeneous middleware technologies to seamlessly interoperate. The interoperable middleware shall further be accommodated by wireless, resource-constrained devices, acting either as service providers or as consumers. Complementary to the middleware interoperability issue, we are concerned about providing middleware architectures that enable full exploitation of the rich pervasive computing environment, and in particular of the potential diversity of the wireless networks that are available at a location. Towards this objective, we are in particular investigating middleware support for Mobile Ad hoc NETworks (MANET), which we consider as a prime enabler of the ambient intelligence/pervasive computing vision. Similarly, multi-radio networking is a major enabler of this vision by potentially improving network connectivity towards ubiquitous networking. In general, numerous challenges remain for middleware architectures to enable the ambient intelligence vision. It is one of the objectives of ARLES to investigate thorough solutions to those issues.

3. Scientific Foundations

3.1. Introduction

Keywords: *ambient intelligence*, *dependability*, *distributed systems*, *middleware*, *mobile ad hoc networks*, *mobile computing*, *pervasive computing*, *service-oriented architecture*, *software architecture*, *software engineering*, *system composition*, *web services*, *wireless networks*.

Research undertaken within the ARLES project-team aims to offer comprehensive solutions to support the development of pervasive computing systems that are dynamically composed according to the environment. This leads us to investigate dedicated software architecture styles from which to derive:

- Architecture description languages for modeling mobile distributed software systems enabling pervasive computing, together with associated methods and tools for reasoning about the systems' behavior and automating the systems' composition at runtime, and
- Middleware platforms for alleviating the complexity of systems development, by in particular offering adequate network abstractions.

The next section provides a brief overview of the state of the art in the area of software architectures for distributed systems; we survey base architectural styles that we consider in our work and further discuss the benefits of architecture-based development of distributed systems. Section 3.3 then addresses middleware architectures for mobile systems, discussing the impact of today's wireless networks, and in particular ad hoc networks, on the software systems, and core requirements that we consider for the middleware, i.e., managing the network's dynamics and enforcing dependability for the mobile systems. Each section refers to results on which we build, and additionally discusses some of the research challenges that remain in the area and that we are investigating as part of our research.

3.2. Software Architectures for Distributed Systems

Architectural representations of systems have shown to be effective in assisting the understanding of broader system concerns by abstracting away from details of the system. This is achieved by employing architectural styles that are appropriate for describing systems in terms of *components*, the interactions between these components – *connectors* – and the properties that regulate the composition of components – *configurations*. Thus, components are units of computation or data store, while connectors are units of interaction among components or rules that govern the interactions. Defining notations for the description of software architectures has been one of the most active areas of research in the software architecture community since its emergence in the early 90's. Regarding the overall development process, *Architecture Description Languages* (ADLs) that have been proposed so far are mainly concerned with architecture modeling during the analysis and design phase. In addition, some existing ADLs enable deriving system implementation and deployment, provided that there is an available implementation of the system's primitive components and connectors. In general, a major objective in the definition of ADLs is to provide associated CASE tools, which enables tasks underpinning the development process to be automated. In this context, special emphasis has been put on the usage of formal methods and associated tools for the analysis of complex software systems by focusing on the system's architecture, which is abstract and concise. As a result, work in the software architecture community provides a sound basis towards assisting the development of robust distributed systems, which is further eased by middleware platforms.

3.2.1. Middleware-based and service-oriented software architectures

Available middleware can be classified into three main categories: transaction-oriented middleware that mainly aims at system architectures whose components are database applications; message-oriented middleware that targets system architectures whose component interactions rely on publish/subscribe communication schemes; and object-oriented middleware that is based on the remote procedure call paradigm and enables the development of system architectures complying with the object paradigm (*e.g.*, inheritance, state encapsulation), and, hence, enforces an object model for the system (i.e., the architectural components are objects). Development of middleware-based systems is now quite mature although middleware heterogeneity is still an open issue. In addition, dealing with middleware heterogeneity in the presence of dynamic composition raises the issue of dynamically integrating and possibly adapting the system's components, which is being investigated in the middleware community.

Evolution of middleware and distributed system technologies has further led to the emergence of service-oriented system architectures to cope with the requirements of Internet-based systems. *Software services*, in particular in the form of XML *Web services*, offer a promising paradigm for software integration and interoperation. Simply stated, a service is an instantiated configured system, which may be composed with other services to offer a new system that actually realizes a system of systems. Although the definition of the overall Web services architecture is still incomplete, the base standards defining a core middleware for Web services have already been released by the W3C[1], partly building upon results from object-based and component-based middleware technologies. These standards relate to the specification of Web services and a supporting interaction protocol. SOAP (Simple Object Access Protocol) defines a lightweight protocol for information exchange that sets the rules of how to encode data in XML, as well as the SOAP mapping to an Internet transport protocol (*e.g.*, HTTP). The specification of Web service interfaces relies on the WSDL (Web Services Description Language) declarative language, which is used to specify: (i) the service's abstract interface that describes the messages exchanged with the service, and (ii) concrete binding information that contains specific protocol-dependent details including the network end-point address of the service. Complementary to the above core middleware for the integration of Web services is UDDI (Universal Description, Discovery and Integration); this specifies a registry for dynamically advertising and locating Web services. Composing Web services relates to dealing with the assemblage of existing services, so as to deliver a new service, given the corresponding published interfaces. Integration of Web services is then realized according to the specification of the overall process composing the Web services. The process specifying the composition must not solely define the functional behavior of the process in terms of interactions with the composed services, but also the process' non-functional properties, possibly exploiting middleware-related services. Various non-functional properties (*e.g.*, availability, extendibility, reliability, openness, performance, security, scalability) should be accounted for in the context of Web services. However, enforcing dependability of composite Web services is one of the most challenging issues, especially for supporting business processes, due to the fact that the composition process deals with the assemblage of loosely-coupled autonomous components.

Although Web services have been primarily designed for realizing complex business processes over the Internet, they are a promising architectural choice for pervasive computing. The pervasiveness of the Web allows anticipating the availability of Web services in most environments, considering further that they may be hosted on mobile devices. Hence, this serves as a sound basis towards dealing with the dynamic composition of services in the pervasive computing environment. However, this further requires specification of the Web services' functional and non-functional behavior that can be exploited for their dynamic selection and integration, which may in particular build upon work on the Semantic Web.

3.2.2. Architecture-based development of distributed systems

The building blocks of distributed software systems relying on some middleware platform, fit quite naturally with the ones of software architectures: the architectural components correspond to the application components managed by the middleware, and the architectural connectors correspond to the supporting middleware. Hence, the development of such systems can be assisted with an architecture-based development process in a straightforward way. This approach is already supported by a number of ADL-based development environments targeting system construction, such as the Aster environment that was previously developed by members of the ARLES project-team.

However, most of the work on the specification of connectors has focused on the characterization of the interaction protocols among components, whilst connectors abstracting middleware embed additional complex functionalities (*e.g.*, support for provisioning fault tolerance, security, transactions). The above concern has led the software architecture community to examine the specification of the non-functional properties offered by connectors. For instance, these may be specified in terms of logic formulae, which further enables synthesizing

[1] http://www.w3.org

middleware customized to the application requirements, as supported by the Aster ADL. Another issue that arises when integrating existing components, as provided by middleware platforms, results from assembling components that rely on distinct interaction patterns. This aspect is known as *architectural mismatch* and is one of the criteria substantiating the need for connectors as first-class entities in architecture descriptions. The abstract specification of connector behavior, as, for instance, supported by the Wright ADL, enables reasoning about the correctness of component and connector composition with respect to the interaction protocols that are used. However, from a more pragmatic standpoint, software development is greatly eased when supported by mechanisms for solving architectural mismatches, which further promotes software reuse.

Connectors that are implemented using middleware platforms actually abstract complex software systems comprising a broker, proxies, but also services for enhanced distribution management. Hence, middleware design deserves as much attention as the overall system design, and must not be treated as a minor task. Architecture-based design is again of significant assistance here. In particular, existing ADLs enable describing conveniently middleware architectures. In addition, given the fact that middleware architectures build upon well known solutions regarding the enforcement of non-functional properties, the synthesis of middleware architectures that comply with the requirements of given applications may be partly automated through a repository of known middleware architectures. In the same way, this *a priori* knowledge about middleware architectures enables one to deal with the safe dynamic evolution of the middleware architectures according to environmental changes, by exploiting both the *support for adaptation* offered by novel middleware platforms (*e.g.*, reflective middleware) and the *rigorous specification of software architectures* enabled by ADLs.

As briefly outlined above, results on software architectures for distributed systems primarily lie in the definition of ADLs that allow the rigorous specification of the elements composing a system architecture, which may be exploited for the system's design and, further, for the software system's assessment and construction. Ongoing work focuses on closer coupling with solutions that are used in practice for the development of software systems. This includes integration of ADLs with the now widely accepted UML standard for system modeling. Still in this direction, coupling with OMG's Model-Driven Architecture (MDA) should be much beneficial. Another area that has already deserved a great deal of attention in architecture-based development is the one of easing the design and construction of middleware underpinning the system execution out of existing middleware platforms. However, addressing all the features enabled by middleware within the architecture design is not yet fully covered. For instance, this requires reasoning about the composition of, possibly interfering, middleware services enforcing distinct non-functional properties. Another area of ongoing research work from the standpoint of architecture specification relates to handling needed architectural evolution as required by emerging applications, including those based on the Internet and/or aimed at mobile computing. In this context, it is mandatory to support the development of system architectures that can adapt to the environment. As a result, the system architecture shall serve dealing with the system evolution at runtime and further assessing the behavior of the resulting system.

3.3. Middleware Architectures for Mobile Systems

Advances in wireless networking combined with increasingly small-scale wireless devices are at the heart of the ambient intelligence (and pervasive computing) vision, as they together enable ubiquitous networking and computing. However, developing software systems such that they can actually be accessed anywhere, anytime, while supporting natural interaction with users, remains a challenge. Although solutions to mobile/nomadic computing have now been investigated for more than a decade following the emergence of wireless networks and devices, these have mostly concentrated on adapting existing distributed systems architectures, so that the systems can tolerate the occurrence of disconnection. Basically, this had led to applying replication strategies to the mobile environment, where computation and/or data are cached on mobile nodes and later synchronized with peer replicas when connection allows. Today's wireless networks enable dynamically setting up temporary networks among mobile nodes for the realization of some distributed function. However, this requires adequate development support, and in particular supporting middleware platforms for alleviating the complexity associated with the management of dynamic networks. In this context, ad hoc networking is amongst the most challenging network paradigm for distributed systems, due to its highly dynamic topology and the absence of any infrastructure. Moreover, it offers significant advantages towards the realization of ubiquitous networking and computing, still due to the absence of any infrastructure. The following section provides a brief overview of ad hoc networking, and is then followed by an overview of the key middleware functionalities that we are addressing for assisting the development of mobile systems. Such functionalities relate to the management of the network's dynamics and to enforcing system dependability.

3.3.1. Ad hoc networking

There exist two different ways of configuring a mobile network: infrastructure-based and ad-hoc-based. The former type of network structure is the most prominent, as it is in particular used in both Wireless LANs (*e.g.*, IEEE 802.11) and global wireless networks (*e.g.*, GSM, GPRS, UMTS). An infrastructure-based wireless network uses fixed network access points (known as base stations), with which mobile terminals interact for communicating, i.e., a base station forwards messages that are sent/received by mobile terminals. One limitation of the infrastructure-based configuration is that base stations constitute bottlenecks. In addition, it requires that any mobile terminal be in the communication range of a base station. The ad-hoc-based network structure alleviates this problem by enabling mobile terminals to cooperatively form a dynamic and temporary network without any pre-existing infrastructure.

The main issue to be addressed in the design of an ad hoc (network) routing protocol is to compute an optimal communication path between any two mobile terminals. This computation must minimize the number of control messages that are exchanged among mobile terminals, in order to avoid network congestion, but also to minimize energy consumption. There exist two base types of ad hoc routing protocols: proactive and reactive. Proactive protocols update their routing table periodically. Compared to proactive protocols, reactive protocols *a priori* reduce the network load produced by the traffic of control messages, by checking the validity of, and possibly computing, the communication path between any two mobile terminals only when communication is requested between the two. Hybrid routing protocols further combine the reactive and proactive modes. The design rationale of hybrid protocols is that it is considered advantageous to accurately know only the neighbors of any mobile terminal (i.e., mobile terminals that are accessible in a fixed number of hops). Since they are close to the terminal, communicating with neighbors is less expensive, and neighbors are most likely to take part in the routing of the messages sent from the terminal. Based on this, a hybrid protocol implements: (i) a proactive protocol for communication with mobile terminals in the neighborhood, and (ii) a reactive protocol for communication with the other terminals.

Spurred by the progress of technologies and deployment at low cost, the use of ad hoc networks is expected to be largely exploited for mobile computing, and no longer be restricted to specific applications (i.e., crisis applications as in military and emergency/rescue operations or disaster recovery). In particular, ad hoc networks effectively support ubiquitous networking, providing users with network access in most situations. However, we do not consider that pure ad hoc networks will be the prominent wireless networks. Instead, mobile distributed systems shall be deployed on hybrid networks, combining infrastructure-based and ad hoc networks, so as to benefit from their respective advantages. Development of distributed systems over hybrid wireless networks remains an open challenge, which requires dedicated middleware solutions for in particular managing the network's dynamics and resources.

3.3.2. Managing the network's dynamics

Trends in mobile computing have created new requirements for automatic configuration and reconfiguration of networked devices and services. This has led to a variety of protocols for lookup and discovery of networked resources. In particular, *discovery protocols* provide proactive mechanisms for dynamically discovering, selecting and accessing available resources. As such, resource discovery protocols constitute a core middleware functionality towards managing the network's dynamics in mobile computing systems. Resource discovery is a central component of distributed systems as it enables services and resources to discover each other on a network and evaluate potential interactions. Many academic and industry-supported protocols (*e.g.*, SLP, UDDI, SSDP) have been designed in different settings, and numerous are now in common usage, using either distributed or centralized approaches depending on assumptions about the underlying network and the environment. These design constraints have led to different, sometimes incompatible mechanisms for service advertisements, queries, security and/or access, while none of the existing resource discovery protocols is suitable for all environments.

The major structural difference between existing resource discovery protocols is the reliance (or not) on a central directory. A central directory stores all the information concerning resources available in the network, provided that resources advertise themselves to the central directory using a unicast message. Then, to access a resource, a client first contacts the central directory to obtain the resource's description, which is to be used for contacting the resource's provider. Prior to any resource registration or client request to the central directory, clients and resource providers must first discover the central directory by issuing broadcast or multicast requests. Centralized resource discovery is much suited to wireless infrastructure-based networks. However, this makes the discovery process dependent upon the availability of the central directory, which further constitutes a bottleneck. In order to support resource discovery in a wider network area, the use of a distributed set of fixed directories has been proposed. Directories are deployed on base stations (or gateways) and each one is responsible for a given discovery domain (*e.g.*, corresponding to a cell).

In the self-organizing wireless network model provided by ad hoc networks that use peer-to-peer communication and no fixed infrastructure, the use of fixed directories for resource discovery is no longer suitable. In particular, the selection of mobile terminals for hosting directories within an ad hoc network is a difficult task, since the network's topology frequently changes, and hence the connectivity is highly dynamic. Decentralized resource discovery protocols then appear more suitable for ad hoc networks. In this case, resource providers and clients discover each other directly, without interacting with a central directory. Specifically, when a client wants to access a resource, it sends a request to available providers using a broadcast message. However, this approach leads to the flooding of the network. An approach to disseminating information about network resources while not relying on the use of broadcast is to use geographic information for routing. Nodes periodically send advertisement along a geometric trajectory (basically north-south and west-east), and nodes located on the trajectory both cache and forward advertisements. Then, when a client seeks a resource, it sends a query that eventually intersects an advertisement path at a node that replies to the request. This solution assumes that the density of nodes is high enough, and further requires the replication of resource advertisements on a significant number of nodes. Hence, it incurs resource consumption that may not be accommodated by wireless, resource-constrained nodes. Resource consumption is further increased by the required support for geographical location (*e.g.*, GPS). Other solutions to decentralized resource discovery that try to minimize network flooding are based on local resource discovery. Broadcast is limited to the neighborhood, hence allowing only for resource discovery in the local area, as supported by base centralized resource discovery protocols. Discovery in the wider area then exploits solutions based on a hierarchy of discovery domains.

Resource discovery protocols for hybrid networks that in particular suit ad hoc networks remains an open issue. Other fundamental limitations of the leading resource discovery protocols are: (i) reliance on syntactic matching of resource attributes included in the resource description, and (ii) unawareness of the environment where the resources are provided. The development of mobile/handheld devices, and wireless and ad hoc networks (*e.g.*, Wi-Fi, Bluetooth) have enabled the emergence of service-rich environments aimed at supporting users in their daily life. In these pervasive environments, a variety of infrastructure-based and/or infrastructure-less networks are available to the users at a location. Such heterogeneous environments bring new challenges to resource/service discovery. In such environments, we can identify the following challenges that a service discovery solution needs to address.

- **Context and semantic information**: In heterogeneous networks, the simple information used by existing service discovery protocols to define a service is not sufficient. Additional information needs to be collected about the networks' identity and characteristics (*e.g.*, bandwidth, cost, reliability), users, and devices. Semantic information is also necessary since service-rich environments may offer many similar services. Context and semantic information needs to be propagated along with service descriptions so that potential clients can evaluate available services and select the most appropriate one(s).
- **Protocol interoperability**: Many service discovery protocols have been proposed for different environments (Internet, home networks) and several have emerged as the leading protocol in their

target environment. A service discovery solution for heterogeneous networks needs to support or interoperate with these service discovery protocols. While discovery information can easily be collected from any protocol, converting service information between different protocols, or injecting information on services from a remote network may not be possible.

- **Network bridging**: The service discovery protocol for heterogeneous networks needs to learn about the different networks available at a location, and about the characteristics of the devices that can act as bridges to access other networks. Similar devices may both be technically able to bridge Wi-Fi and Bluetooth but may provide different QoS due to battery power, user mobility, cost, or installed software packages. As the heterogeneous network topology changes, links to some remote networks may become unavailable, and latency may change drastically as a new route will be used.
- **Information propagation**: The service discovery protocol for heterogeneous networks needs to filter the information that is propagated between networks, as the information usually collected by discovery protocols may not be completely relevant for remote hosts/networks. For example, services on remote networks may not be accessible (*e.g.*, security issues or minimum bandwidth unavailable). The discovery protocol should also evaluate how far discovery information should be propagated, and how it should be cached and managed at the bridges.
- **Remote service access**: Service discovery protocols collect information about available services, and provide this information back to requesting clients. Part of the information is the service providers location (*e.g.*, IP address of host and port number). It is usually assumed that clients can directly contact the service provider and request a service (that may be granted or not). In the case of heterogeneous networks however, it may not be possible to access the service provider due, for example, to IP network accessibility issues. The service bridges, which propagated the service information, may potentially be used to also propagate service request. Further, client and service providers must use semantically and syntactically matching protocols for service access, which cannot be guaranteed in highly open, pervasive networking environments.

While resource discovery constitutes a core middleware functionality towards easing the development of distributed software systems on top of dynamic networks, higher-level abstractions for dynamic networks need to be developed and supported by the middleware for easing the developers' task. The definition of such abstractions shall be derived from both features of the network and architectural principles elicited for mobile software systems, where we exploit our work in both areas. Related issues include characterizing and reasoning about the functional and non-functional behavior of the participating peer nodes, and in particular dealing with security requirements and resource availability that are crucial in the mobile environment.

3.3.3. Enforcing dependability

Dependability of a system is defined as the reliance that can justifiably be placed on the service that the system delivers. It decomposes into properties of availability, safety, reliability, confidentiality, integrity and maintainability, with security encompassing availability, confidentiality and integrity. Dependability affects the overall development process, combining four basic means that are fault prevention, fault removal, fault tolerance and fault forecasting. In the context of middleware architectures for mobile systems, we concentrate more specifically on fault tolerance means towards handling mobility-induced failures. Such failures affect most dependability properties. However, availability and security-related properties are the most impacted by the mobile environment due to changing connectivity and features of wireless networks that make them more prone to attacks. Security remains one of the key challenges for mobile distributed systems. In particular, the exploitation of ad hoc networks does not allow systematic reliance on a central infrastructure for securing the network, calling for decentralized trust management. Additionally, resource constraints of mobile devices necessitate the design of adequate cryptographic protocols to minimize associated computation and communication costs.

Enforcing availability in the mobile environment relies on adequate replication management so that data and/or services remain accessible despite the occurrence of disconnection. Such a concern has led to tremendous research work since the emergence of mobile computing. In particular, data replication over mobile nodes has led to novel coherency mechanisms adapted to the specifics of wireless networks. Solutions in the area relate to offering optimistic coherency protocols, so that data copies may be concurrently updated and later synchronized, when connectivity allows. In initial proposals, data copies were created locally on accessing nodes, since these proposals were aimed at global infrastructure-based networks, where the mobile node either has access to the data server or is isolated. However, today's wireless networks and in particular ad hoc networks allow for creating temporary collaborative networks, where peer nodes may share resources, provided they trust each other. Hence, this allows addressing replication of data and services over mobile nodes in accordance with their respective capabilities. Dually, peer-to-peer communication supported by ad hoc networks combined with decentralized resource discovery allow accessing various instances of a given resource, and hence may be conveniently exploited towards increasing availability. Today's wireless networks offer great opportunities towards availability management in mobile systems. However, providing effective solutions remains an open issue, as this must be addressed in a way that accounts for the constraints of the environment, including possible resource constraints of mobile nodes and changing network topology. Additionally, solutions based on resource sharing among mobile nodes require incentive mechanisms to avoid selfish behavior where nodes are trying to gain but not provide resource access.

4. Application Domains

4.1. Application Domains

Keywords: *ambient intelligence*, *distributed systems*, *information systems*, *mobile systems*, *web services*.

The ARLES project-team targets development support for applications relevant to the ambient intelligence vision, with a special focus on consumer-oriented applications. Architecture-based development of systems of systems is further directly relevant to enterprise information systems, whose composition is mainly static and relates to the integration of legacy systems. In addition, by building upon the Web services architecture for dealing with the dynamic composition of (possibly mobile) autonomous systems, our work is of direct relevance to e-business applications, providing specific solutions for the mobile context.

Our application domain is voluntarily broad since we aim at offering generic solutions. However, we examine exploitation of our results for specific applications, as part of the experiments that we undertake to validate our research results through prototype implementation. Applications that we consider in particular include demonstrators developed in the context of the European and Industrial projects to which we contribute (§ 7.1 & 7.3).

As illustration of applications investigated within ARLES, the COCOA semantic service middleware together with the INMIDIO interoperable middleware (§ 5.5 and 5.6) support the *networked home* environment [29]. The networked home specifically seeks to combine the home automation, consumer electronics, mobile communications and personal computing domains to provide new user applications that exploit the fluid integration of these traditionally strictly separated domains, and to lay a solid foundation towards realizing the ambient intelligence vision. The applicability of the COCOA semantic service middleware to the networked home environment has in particular been illustrated at the Amigo 2nd Annual Review Meeting and Open Day, at the Philips Research High Tech Campus, Eindhoven, November 2006 (§ 7.1.2). A demonstrator application was presented, which enabled users' personal wireless devices to discover, negotiate permission to use, and control home consumer electronic and personal computing devices.

5. Software

5.1. Introduction

In order to validate our research results, our research activities encompass the development of related prototypes, which we present here in chronological order of release.

5.2. WSAMI: A Middleware Based on Web Services for Ambient Intelligence

WSAMI (Web Services for AMbient Intelligence) is based on the Web services architecture and allows for the deployment of services on wireless handheld devices like smartphones and PDAs. WSAMI further supports the dynamic composition of distributed services over hybrid wireless networks. Moreover, WSAMI takes in charge the customization of the network's path through the dynamic integration of middleware-related services, in order to enforce quality of service with respect to offered dependability and performance properties.

The WSAMI middleware prototype is available since 2004. It is a Java-based implementation of the WSAMI core middleware, which builds upon IEEE 802.11b as the underlying WLAN and integrates the following components: (i) the WSAMI SOAP-based core broker, including the CSOAP[2] SOAP container for wireless, resource-constrained devices; and (ii) the Naming & Discovery service, including support for connector customization. The memory footprint of our CSOAP implementation is 90kB, as opposed to the 1100kB of the Sun's reference SOAP implementation. The overall memory footprint of our Web services platform is 3.9MB, dividing into 3MB for the CVM and 815kB for the Xerces XML parser, in addition to the CSOAP implementation. The WSAMI middleware prototype is an open-source software freely distributed under the terms of the GNU Lesser Public License (LGPL) at http://www-rocq.inria.fr/arles/download/ozone/index.htm.

Our prototype is being used for the implementation of demonstrator applications in the field of ambient intelligence, as well as a core for service-oriented middleware platforms aimed at advanced wireless networking environments, like Ariadne (§ 5.3).

5.3. Ariadne: A Protocol for Scalable Service Discovery in MANETs

The Ariadne service discovery protocol for MANETs has been designed to support decentralized Web service discovery in a hybrid network composed of multi-hop mobile ad hoc and/or infrastructure-based and/or wired networks. Ariadne enables small and resource-constrained mobile devices to seek and find complementary, possibly mobile, Web services needed to complete specified tasks, while minimizing the traffic generated and tolerating intermittent connectivity. The Ariadne protocol further enables service requesters to differentiate services instances according to non-functional properties. Specifically, the Ariadne protocol is based on the homogeneous and dynamic deployment of cooperating directories within the MANET. Scalability is achieved by limiting the generated traffic related to service discovery, and by using compact directory summaries (i.e., Bloom filters) to efficiently locate the directory that most likely has the description of a given service.

The prototype of the Ariadne service discovery protocol has been implemented in Java, and provides an application programming interface (API) so as to be easily integrated in a Web service-oriented middleware such as WSAMI (§ 5.2). The Ariadne prototype is an open source software freely distributed since 2005 under the terms of the GNU Lesser Public License (LGPL) at http://www-rocq.inria.fr/arles/download/ariadne/index.html.

Our prototype is being used for the implementation of demonstrator applications exploiting MANETs in the field of ambient intelligence.

5.4. MUSDAC: A Middleware for Service Discovery and Access in Pervasive Networks

The MUlti-protocol Service Discovery and ACcess (MUSDAC) middleware platform enables the discovery and access to services in the pervasive environment, which is viewed as a loose and dynamic composition of independent networks. MUSDAC manages the efficient dissemination of discovery requests between the different networks and relies on specific plug-ins to interact with the various middleware used by the networked services [20].

[2] Compact SOAP

We have implemented a first prototype of MUSDAC in Java (J2SE 1.4.2 and 1.5), which currently includes support for 5 different service discovery protocols, and remote access for SOAP-based services. The different plug-ins enable us to experiment with both repository-based (Ariadne, OSGi) and multicast-based (SLP, UPnP) protocols. The MUSDAC prototype is an open source software freely distributed since 2005 under the terms of the GNU Lesser Public License (LGPL) at http://www-rocq.inria.fr/arles/download/ubisec/index.html.

The MUSDAC middleware in particular serves as a base building block in the development of a middleware aimed at effectively enabling service-oriented computing in Beyond 3rd Generation (B3G) networks. This further leads to make evolve MUSDAC-based service discovery and access to multi-radio and multi-protocol networking environments.

5.5. INMIDIO: An Interoperable Middleware for Ambient Intelligence

In the pervasive computing environment, devices from various application domains, *e.g.*, home automation, consumer electronics, mobile and personal computing domains, need to dynamically interoperate irrespectively of the heterogeneity of their underlying hardware and software. Middleware has been introduced in order to overcome this issue by specifying a reference interaction protocol enabling compliant software systems to interoperate. However the emergence of different middleware platforms to address the requirements of specific application domains leads to a new heterogeneity issue among interaction protocol. Thus, at a given time and/or at a specific place, devices hosting the wrong middleware become isolated. In order to overcome this issue, we have developed a system called INMIDIO (INteroperable MIddleware for service Discovery and service InteractiOn) that dynamically resolves middleware mismatch [28]. More particularly, INMIDIO identifies the interaction middleware and also the discovery protocols that execute on the network and translates the incoming/outgoing messages of one protocol into messages of another, target protocol. Specifically, the system parses the incoming/outgoing message and, after having interpreted the semantics of the message, it generates a list of semantic events and uses this list to reconstruct a message for the target protocol, matching the semantics of the original message. The INMIDIO middleware acts in a transparent way with regard to discovery and interaction middleware protocols and with regard to the services running on top of them.

The service discovery protocols supported by the current INMIDIO prototype are UPnP, SLP and WS-Discovery, while the supported service interaction protocols are SOAP and RMI. The INMIDIO prototype is publicly available since 2006 and released under the GNU Lesser Public License (LGPL) at http://www-rocq.inria.fr/arles/download/inmidio/index.html.

5.6. COCOA: A Semantic Service Middleware

COCOA is a comprehensive approach to semantic service description, discovery, composition, adaptation and execution, which enables the integration of heterogeneous services of the pervasive environment into complex user tasks based on their abstract specification. Using COCOA, abstract user tasks are realized by dynamically composing the capabilities of services that are currently available in the environment. The capabilities that a service provides are presented as a *conversation* – a workflow that specifies data and control dependencies for its capabilities. Similarly, an abstract task is presented as an orchestration of required capabilities. The conversations of the provided service capabilities are integrated to realize the orchestration of the abstract task, while guaranteeing that the dependencies of each of the provided capabilities are preserved. This allows complex user tasks to be created and reliably composed, while offering fine-grained control over the placement of capabilities in the task. This is especially important for the pervasive environment, where user tasks may frequently involve interaction with the user(s). In addition, the service composition can be optimized based on quality of service and context-aware parameters. To accommodate the inherent heterogeneity of services in the pervasive environment, capabilities are described and matched semantically, and adapted when necessary. Furthermore, different service groundings are supported, allowing diverse SOA platforms to be incorporated; interoperability at service grounding, *i.e.*, middleware level may then be ensured by employing INMIDIO (§ 5.5). Once a new service realizing the user task has been created, it is automatically deployed and executed.

The first integrated COCOA prototype was successfully demonstrated at the Amigo 2nd Annual Review Meeting and Open Day, at the Philips Research High Tech Campus, Eindhoven, November 2006 (§ 7.1.2). The next prototype version will be released under LGPL as open source software in 2007 on the Amigo Open Source Repository http://amigo.gforge.inria.fr/home/index.html.

5.7. MR_SDP: Nomadic Context-awareness in Multi-Radio Networks

The objective of MR_SDP (Multi-Radio Service Discovery Protocol) is to support context awareness based on effective service discovery in multi-radio networks. MR_SDP supports an adaptive service discovery protocol scheme optimized for multi-radio networks, which allows the client device to discover networked services over the embedded network interfaces, by using the relevant functionalities of the multi-radio wireless device while interfacing with the various Service Discovery Protocols (SDPs) that are available at the specific location and time. Further and as importantly, this is realized in a way that both minimizes resource-consumption on the device and offers a response time comparable to related SDPs.

We have implemented a first prototype of MR_SDP in C#/.Net, for mobile devices running Pocket PC 2003 and Windows Mobile 5. This prototype works along with the CSOAP middleware, that we ported for our target software platform (§ 5.2), and offers to mobile devices support for effective service discovery in multi-radio networks. This prototype includes support for Wi-Fi, Bluetooth and GPRS radio interfaces, and UPnP and Bluetooth SDPs. Other service discovery protocols may easily by integrated thanks to the plug-in-based architecture of MR_SDP.

6. New Results

6.1. Introduction

The ARLES project-team investigates solutions in the forms of languages, methods, tools and supporting middleware, to assist the development of distributed systems, with a special emphasis on mobile distributed systems enabling the ambient intelligence vision. Towards that goal, we undertake an approach that is based on the architectural description of software systems, further allowing to deal with the dynamic composition of systems according to the environment. Our research activities thus subdivide into two core activities:

(*i*) Software architectures for pervasive computing systems, where we investigate architectural styles dedicated to the target systems from which to derive languages for system modeling and related methods and tools for supporting system development (§ 6.2).

(*ii*) Middleware architectures for ambient intelligence systems, building upon architectural styles elicited for pervasive computing systems and further investigating solutions that meet constraints associated with today's wireless networks and devices for their effective exploitation (§ 6.3).

6.2. Software Architectures for Pervasive Computing Systems

Participants: Sonia Ben Mokhtar, Sébastien Bianco, Mauro Caporuscio, Damien Fournier, Nikolaos Georgantas, Valerie Issarny, Pascal Poizat, Letian Rong, Roberto Speicys Cardoso, Ferda Tartanoglu, Graham Thomson.

Building upon our past work on modeling software architectures of closed distributed systems for supporting the systems' analysis and synthesis, we are investigating solutions to architecture-based development of distributed systems for ambient intelligence/pervasive computing. This leads to more specifically investigate the service-oriented architectural style for pervasive computing, studying the modeling of services so that services can be effectively deployed, discovered, accessed and composed in the heterogeneous, open pervasive computing environment. In that context, our research activities over the year 2006 have focused on a number of complementary areas, as presented in the following sections.

6.2.1. Foundations of service interoperability in pervasive computing

Among the features characterizing pervasive computing systems are the highly dynamic character of the computing and networking environment and the high heterogeneity of the integrated technologies in terms of networks, devices and software infrastructures. To cope with such dynamics and diversity, pervasive computing systems should have the capacity to be deployed and to execute in an open, ad hoc way, integrating the available hardware and software resources at the specific time and place [5]. This is facilitated by rendering such resources autonomous, networked components that may be incorporated in a larger system. Then, supporting interoperability between heterogeneous system components becomes a key issue. A recent computing paradigm, which can provide solutions to ad hoc system composition is Service-Oriented Architectures (SOA), where system components are abstracted as services. However, the interoperability problem remains, as SOA enforce a specific middleware platform and a standard syntactic service description, for which a common agreement is not possible in the open pervasive environment. Therefore, a promising approach towards application-level functional interoperability relies on semantic modeling of services as introduced by the Semantic Web and Semantic Web Services paradigms. Nevertheless, the interoperability requirements of pervasive systems are wider, concerning functional and non-functional interoperability that spans both the middleware and application levels. Thus, interoperability in pervasive systems is a major and still open issue.

We have been studying in depth the architectural and behavioral features of pervasive systems that need to be interoperable, and we have been working on solutions towards ad hoc interoperability in pervasive environments. In previous work within this context, we proposed to introduce semantics in the service-oriented system architectural design, thus establishing a common architectural description at a higher, technology-independent level, based on semantic concepts. Building on this principle, we have implemented middleware-level interoperability as discussed in Section 6.3.1 and have worked on application-level service interoperability and interoperable, semantics-based service composition, as detailed in Section 6.2.2.

Overall, our approach to service composition aims at enabling distributed, dynamic, ad hoc, complex user applications to be deployed and executed in pervasive environments by exploiting the services available in the given environment. We have decomposed this system functionality in a number of functional steps, namely, *Service Description, Discovery, Composition, Adaptation* and *Execution*, which we call collectively with the acronym SD-SDCAE. We have been working on a comprehensive approach to SD-SDCAE. Specifically, we have introduced a reference system architecture to support SD-SDCAE, and we have produced results concerning the description, discovery, composition, and execution aspects [27].

In our approach to semantics-based service interoperability, we further consider context and QoS as key concerns, as they decisively affect the user's experience of the pervasive environment. We have, in particular, studied key context-aware system concepts that need to be incorporated in the SOA style to enable context-aware pervasive services. This aims at ad hoc networking of context sources and of context-aware applications to optimally exploit the available, possibly heterogeneous, contextual resources, which will enable anytime, anywhere context-awareness [16].

6.2.2. Semantic service description, discovery and composition

To enable semantic service description, we have introduced a declarative language that models both application-level and middleware-level service features and supports both functional and non-functional properties. This language has been formalized as a high-level OWL ontology, reusing base classes from OWL-S [26]. We have applied this language to describe both a composite service-based application – which we call *task* – and its component services. Specifically, services are associated with concrete provided descriptions, while tasks have abstract required descriptions that have not yet been mapped on concrete service realizations [8].

Based on semantic description, we have identified SDCAE as a middleware-level functionality that is located on user devices, on other devices hosting services, and on a selected device hosting a service repository, where the latter abstraction of a single service repository is only logical and may have various physical realizations. We have developed an API that may be exploited by services and tasks for: registering a service with the repository; and, querying the service repository for provided service capabilities, based on the required capabilities identified in the abstract task description. Further, we have been working on an efficient solution to service discovery, which we call *COCOA-SD* (COnversation-based service COmposition middlewAre-Service Discovery). We have particularly focused on enhancing the efficiency of COCOA-SD, targeting pervasive environments, which involve resource-constrained devices and require acceptable response times for interactive applications. First, we have introduced adequate matching relations for the semantic matching of service capabilities, which can not only decide matching but also evaluate the matching degree, as exact matching of the task's required capabilities with available service capabilities is seldom the case. Second, we have developed optimizations for service discovery and its underlying semantic matching. Applying matching relations is based on the use of existing ontologies and semantic reasoning tools; the latter are the object of intensive research within the knowledge representation community. However, such reasoning is particularly costly both in terms of computing power and time; the most expensive part concerns classifying ontologies, i.e., producing expanded forms of them required for reasoning. Moreover, existing reasoning tools are not aimed at the resource-constrained, interactive pervasive computing environments. We have carried out thorough analysis of the cost of semantic reasoning, employing popular semantic reasoners. Based on the outcome of this analysis, we have developed an optimization technique, which includes classifying ontologies offline and encoding ontologies and, accordingly, task and service descriptions with numeric values; this reduces reasoning to numeric comparison. Further, we group "similar" service descriptions in the service repository, so that, when matching within a service repository, a task description will not need to be compared against all registered service descriptions. We have implemented and evaluated our efficient semantic service discovery in terms of performance; achieved results are highly encouraging [9], [10].

By employing our semantic language, we further described both services (their conversation) and tasks (the supported orchestration) as workflows of high-level capabilities. As discussed above, COCOA-SD deals with the discovery of services offering appropriate capabilities that will be integrated for realizing a task. Then, COCOA-CI (COCOA-Conversation Integration) deals with the composition of the conversations of the selected services to reconstruct the tasks orchestration. To perform such a composition, COCOA introduces an abstraction of service and task workflows in terms of finite state automata. This makes possible the translation of the conversation integration problem into an automata analysis problem. The distinctive feature of COCOA is the ability to compose services that expose complex behaviors to realize a user task that itself has a complex behavior. We have particularly focused on enhancing the flexibility of COCOA-CI. By considering workflows of capabilities, a task may flexibly be rebuilt from different services in various environments or at distinct times; these services may provide different low-level implementations of the required high-level capabilities. Furthermore, COCOA can effectively reconstruct a tasks orchestration by interleaving conversations of the selected services. We have elaborated a prototype implementation and evaluation of COCOA-CI, which shows that it can efficiently be employed in the demanding pervasive environment [8] (§ 5.6).

The outcome of COCOA-CI is a concrete task description, where the selected service conversations have been incorporated. To run the composite service-based application, this task needs to be executed. We have developed a transcription tool that transforms a task's orchestration from our semantic language, based on the OWL-S process model, into BPEL. Even though BPEL does not support semantics, it supports executable workflow descriptions and is an industrial standard. We have employed an existing BPEL engine to execute the resulting BPEL task, which orchestrates the composed services [27].

6.2.3. Model-based software adaptation

Incompatibilities between the behavioral interfaces (protocols) of interacting components or services may prevent their composition or lead to incorrect composition behaviors (*e.g.*, deadlocks or QoS loss). The objective of software adaptation is to compensate such incompatibilities building as automatically as possible corrective connectors or components called adaptors. In previous work, we developed a model-based approach to deadlock-free component adaptation and a dedicated tool for the synthesis of adaptor protocols from user-defined adaptation contracts, which we call Adaptor. Although this approach works system-wide (adaptation of any number of components in an architecture), it is based on a closed system hypothesis and is thus not suited to environments where the system architecture needs to evolve, *e.g.*, service-oriented pervasive computing where services may enter or leave the system at any time. We have therefore extended our approach to support: (i) adaptation of open systems; and (ii) incremental adaptation [19], where adaptors are distributed over the architecture and where only parts of the adaptors have to be recomputed when the architecture evolves. Composition or adaptation processes build on automata operations, which can yield large state-space models, or models with redundant or useless parts. This may limit the applicability of these processes at run-time on resource-constrained devices. Based on the observation that services have a transactional nature (to ensure a given service, components rely on several basic event exchanges and/or service requests to other components), we have developed transaction-oriented reduction and simulation algorithms for composition or adaptation models. A perspective is to make these reduction techniques applicable "on-the-fly" during a composition or adaptation process such as the one enabled by semantic services discussed in the previous section.

Our perspectives concern the application of model-based adaptation to service-oriented architectures, *e.g.* Web Services and Semantic Web Services, first by relating adaptor models and Web Services composition languages such as BPEL4WS. Adaptation may enhance solutions to the dynamic ad hoc composition of services, as integrated in the COCOA middleware: (i) to support non one-to-one relations between services required by user tasks and services provided in pervasive environments, hence enhancing the set of discovered and composed services, and (ii) to relate high-level (user related) and low-level (implementation related) service descriptions. Dually, the efficient service discovery algorithm of COCOA, supporting semantic and QoS service descriptions may successfully help the adaptation process in obtaining automatically relevant correspondences between services in user adaptation contracts.

6.2.4. Privacy-protecting architectures for pervasive computing environments

Future pervasive environments have the potential of collecting and correlating a greater quantity of data, thus increasing the menace to privacy posed by computer systems. This risk is closely related to the system architecture, as the architecture defines how data flows from users to applications. In our vision, pervasive computing systems and applications will follow the service-oriented architecture paradigm as it is well suited to handle the heterogeneity, dynamism and mobility inherent to pervasive environments.

The use of SOAs in pervasive computing creates new threats to privacy. Service information concerning access, discovery and service descriptions is sensitive and must be protected, since it can reveal information about the activity of a user, to an external observer. Moreover, contextual information may be used in service publishing and location to improve service discovery performance, thus increasing the sensitivity of service information. In order to address these concerns, we have proposed a taxonomy of privacy invasions in pervasive computing and an architecture based on that taxonomy that provides mechanisms to control disclosure of personal information and enables users to reveal only the information strictly necessary to perform a service access [23]. Our architecture is designed as a complement for existing service-oriented context-aware pervasive system architectures. It leverages existing components of such architectures and introduces privacy-related modules that interact to provide multi-level protection of personal data. Building on the proposed architecture, we are designing a service discovery protocol that enables users to semantically discover services, while still protecting personal data contained in service discovery requests and service publications.

6.2.5. System architectures for multi-radio networks

Recently, Beyond 3rd Generation (B3G) networks have gained importance as an effective way to realize pervasive computing, where convergence is pursued between wireless telecommunication networks and wireless IP networks. Indeed, B3G networks offer broad connectivity through various network technologies at once, and now become available to the vast majority of end-users, thanks to the newest multi-radio devices like smart phones embedding, *e.g.*, UMTS, Wi-Fi and Bluetooth networking. Still, most applications running on those devices do not exploit the multi-radio networking capability in an integrated manner, and are commonly developed with a single network in mind. In this context, B3G networks specifically aim at exploiting the notion of (minimal) integration of different connectivity standards, still preserving the heterogeneity of the various networking systems and their qualitative and quantitative characteristics. Taking benefit from the rich B3G networking environment is raising tremendous challenges, related to the effective management of this complex environment. A B3G-enabled user, having a multi-radio capable device, benefits from such an ambient network by increasing the perimeter of reachable service providers, at the expense of a higher network management complexity. As B3G-capable devices hold several radio interfaces, the possibility to switch from one to another upon disconnection (vertical handover) appears. For those reasons, deployment of B3G-enabled applications, distributed on several heterogeneous radio networks, is an important issue. This complexity, induced by the heterogeneity of the wireless technologies, should be hidden to the user and, to be effective, abstracted to the application by a middleware.

Exploiting the SOA style in the context of B3G poses new requirements on the underlying middleware [11]. In fact, the middleware has to provide the proper abstractions concerning both the network characteristics and the service-oriented computing paradigm. In particular, such abstractions provide the support for making the application aware about the network diversities, in terms of properties and offered functionalities. Further, they provide the support for seamlessly choosing the most appropriate network(s) according to the actual networking context, relevant network properties and the interaction behavior. As a consequence, this requires to model the different (sub-)networks and the context. Various performance factors are eligible, with selection criteria being antagonistic for most of them [13]: (i) optimizing response time by selecting the radio network with the highest bandwidth possibly at the expense of power consumption, (ii) optimizing energy consumption by selecting the radio network with the lowest power consumption, (iii) optimizing dependability by selecting the radio network that has the largest coverage and guaranteed quality of service, at the expense of financial cost, and (iv) optimizing financial cost by selecting free of charge radio networks, often at the expense of quality. Due to the above characteristics, B3G networks can no longer be considered as "passive" entities which only transport data between end-points, but they must be considered as "active" parties that have their own behavior and provide services. This creates a completely new application domain where applying current software engineering design tools, such as software architectures, fails. In fact, dealing with B3G networks requires to explicit low-level details usually abstracted by the architectural descriptions. B3G-oriented application modeling and the use of proper standard tools (*e.g.*, UML and ADLs) are emerging as important research challenges [14].

6.3. Middleware Architectures for Ambient Intelligence Systems

Participants: Raghav Bhaskar, Yérom-David Bromberg, Damien Charlet, Rafik Chibout, Manel Fredj, Nikolaos Georgantas, Valérie Issarny, Jinshan Liu, Liam McNamara, Davy Preuveneers, Pierre-Guillaume Raverdy, Daniele Sacchetti, Gaogang Xie, Thomas Zahn, Apostolos Zarras.

In order to ease the development of distributed systems enabling the ambient intelligence vision, we are investigating supporting core middleware platform and associated services. Our work in this area over the year 2006 has concentrated on middleware enabling collaboration among mobile nodes by addressing issues of interoperability and service discovery and access, in the rich pervasive networking environment. We have further pursued our study on middleware for MANETs.

6.3.1. Solving middleware heterogeneity in pervasive computing environments

In the pervasive computing environment, communication among networked software services involve the use of specific middleware protocols, making applications tightly coupled to middleware. Additionally, to overcome the constraints of wireless networks (*e.g.*, limited bandwidth, poor network quality of service and either voluntary or forced frequent disconnection), several communication models have arisen. Thus, as there exist many styles of communication and consequently many styles of middleware, we have to deal with middleware heterogeneity: an application implemented upon a specific middleware cannot interoperate with services developed upon another. Similarly, we cannot predict at design time the requirements needed at runtime since the execution environment is not known. However, no matter which underlying communication protocols are present, mobile nodes must both discover and interact with the services available in their vicinity. The above issue calls for an interoperability system that enables inter-networking between heterogeneous middleware platforms. This interoperability system must at least overcome both the heterogeneity of discovery protocols [12] and the one of communication protocols among services. As part of our research work, we have elaborated such an interoperability system [2]. Specifically, this system interposes at the network level (deployed whether on the client, a gateway or the service host) and takes care of efficiently and dynamically translating messages from one middleware protocol to another so that networked services may effectively both discover and interact with each other. Protocol translation relies on the translation of protocol messages from and to semantic events. The translation is further instantiated dynamically according to the protocols active in the network. Our system is further designed so that it can be accommodated by wireless, resource-constrained devices, whether service providers or consumers. We have developed a prototype to validate our approach in terms of performance (§ 5.5). Our results comply with the strict requirements of the pervasive environment. Overall, our contribution lies in providing an interoperability layer that hides middleware heterogeneity to the applications. It also supports efficient interoperability between networked devices, including resource-constrained ones, without requiring any change to applications/services and related middleware.

6.3.2. Context-aware service discovery and access in pervasive networks

As part of our research on achieving middleware interoperability, we have further designed the MUlti-Protocol Service Discovery and ACcess (MUSDAC) middleware to overcome both the limited interconnectivity between the different networks available at a location, and the lack of interoperability between the existing discovery and access protocols. Indeed, the use of various wireless technologies (*e.g.*, cellular networks, Wi-Fi, or Bluetooth) and network deployment models (*e.g.*, ad hoc or infrastructure-based) results in many independent networks being available to users at a location. As users can only be connected to a limited number of networks at the same time (often a single one), many services may not be accessible (i.e., not IP reachable). The use of various middleware platforms (*e.g.*, UPnP, Jini, Web services) by mobile users further limits the number of accessible services due to the incompatibilities between the different discovery and access protocols.

With MUSDAC, the environment is viewed as a dynamic composition of independent networks in which services use different protocols for discovering and accessing services [21]. To enable clients to access all the networked services, the platform manages the efficient dissemination of discovery requests between the different networks and relies on specific plug-ins to interact with existing middleware. The originality of our approach is that MUSDAC [22]: (i) is itself provided as a service through existing discovery protocols; and (ii) uses context information not only to provide context-aware service discovery but also to control the dissemination of service information. An instance of MUSDAC is started in each independent network and is composed of: (i) a Manager that provides service discovery and access to clients within the network; (ii) Service Discovery and Access (SDA) Plug-ins and Transformers that interact with specific discovery domains to collect service information and perform service access; and (iii) a set of Bridges that assist Managers in providing service discovery and service access in the pervasive environment. In MUSDAC, services are described using a generic and modular description format able to record any information available in an SD-specific service definition. SDA Plug-ins and Transformers generate these descriptions based on the SD-specific service descriptions they receive.

A client looking for services in the pervasive environment first discovers their MUSDAC service, and then issues a discovery request that is forwarded to the Manager, which then forwards it to the local SDA Plug-ins for processing, and to the active Bridges in the network for propagation throughout the global network. Bridges and Managers perform advanced filtering based on context information (*e.g.*, service and client profiles, network status) to only return relevant instances of the requested service. For service access, MUSDAC creates a communication channel along the Bridges leading to the service's network, and sends the client's service access requests on this communication channel. The Manager at the other end of the communication channel then executes the service access requests on behalf of the client and returns the results.

We have implemented a first prototype of MUSDAC in Java (J2SE 1.4.2 and 1.5) that currently includes support for five different service discovery protocols, and remote access for SOAP-based services based on WSAMI (§ 5.4). The different SDA Plug-ins enable us to experiment with both repository-based (Ariadne, OSGi) and multicast-based (SLP, UPnP) protocols. We have evaluated the performance of service discovery and context management in the MUSDAC platform [20]. While the overhead is significant compared to simple, network-level protocols such as SLP, it becomes negligible when interacting with more complex protocols such as UDDI. Simple context information also enables a significant reduction of the network traffic and to only return the most relevant services to the client. We started an evaluation of the scalability of the MUSDAC platform through simulations in order to evaluate adaptive policies for Bridge elections and information dissemination. We now plan to evolve the MUSDAC middleware towards B3G networks. A first evolution is to decouple the B3G communication overlay and the discovery process on this overlay. Another evolution is to integrate security and trust for the interconnection of networks and the access to services.

6.3.3. Energy-efficient service discovery in multi-radio networks

The wide deployment of mobile networks and the emergence of powerful portable devices have provided a powerful basis for pervasive computing. As particularly addressed by B3G networks, the recent evolution of mobile networks introduces the convergence of wireless technologies, where several radio interfaces are to be used concurrently. Thus, B3G-aware applications shall make the most effective use of this connectivity. In pervasive environments, mobile users may discover and access services offered on the networks using Service Discovery Protocols (SDPs). Several SDPs are currently in use, each one designed for a specific target network architecture and setting. Thus, in a multi-radio environment, each SDP does not equally suit each radio interface.

In order to provide effective service discovery in multi-radio networks, the most resource-efficient interface shall be chosen with respect to two main criteria: the adequacy of the radio interface against the SDP to be used, and energy saving, which is crucial for battery-powered devices. Toward this goal, we have assessed how to exploit multiple radio interfaces from the standpoint of service discovery and access with respect to energy consumption as well as the adequacy of legacy SDPs to the various networks, so as to classify the most appropriate networks for each SDP. We have shown in a first study [15], which measured the power consumption of wireless interfaces during service discovery and access, that the most appropriate case would consist in discovering services via the Bluetooth interface, and access them via the Wi-Fi interface. This perfect case, however, is not always feasible. Thus, in another study, we have taken into account the specifics of legacy SDPs and discussed how they behave when operating on each wireless interface. In order to translate these qualitative observations into parameters usable by an adaptive algorithm, we inferred quantitative values reflecting the adequacy of use of each SDP against each radio interface. Finally, we have studied how each SDP can be energy-efficiently combined and configured on each radio interface by taking into account the other SDPs and applications running on the device.

6.3.4. Tolerating Connectivity Loss in Pervasive Computing Environments

Pervasive computing provides an attractive vision for the future of computing. Network access and computational devices will be available everywhere, providing networked services. Mobile and stationary devices will be dynamically connected and coordinated to seamlessly help people in accomplishing their tasks, via composition of networked services as presented in Section 6.2.2. However, pervasive computing environments are characterized by high dynamics due to the mobility of devices that populate these environments. This mobility leads to the frequent loss of connectivity between devices of the pervasive computing environments. One of the most challenging issues for pervasive computing is then to deal with the connectivity loss in order to guarantee the continuity of composite services, including maintaining consistency of composite and component services.

More specifically, in pervasive computing environments, the availability of specific networked service instances cannot be guaranteed over time as users move and services leave and join the network accordingly. Furthermore, connectivity loss cannot be anticipated in general. We have therefore revisited the problem of dynamic reconfiguration of composite services, towards the elaboration of dynamic reconfiguration techniques that account for the aforementioned observations [7]. The main purpose of the proposed techniques is to tolerate the negative effects of dynamic changes rather than preventing them from happening. In effect, change prevention introduces constraints on the environment (*e.g.*, limiting user mobility or services that can be invoked by users to guarantee connectivity in all situations), while we aim at dealing with the environment's dynamics and heterogeneity rather than constraining them [17]. To this aim, we have investigated fault tolerance techniques for distributed systems as they present effective solutions for dealing with process failure while preserving distributed system consistency. We focused on applying these techniques to the specifics of the pervasive computing environment. However, as these techniques do not originally target pervasive computing environments, they present several inadequacies. Therefore, we have proposed an adaptation of fault tolerance techniques to pervasive computing environments, together with their integration into a middleware architecture. The proposed architecture is conceptualized as an extension to the WSAMI middleware (§ 5.2).

6.3.5. Efficient service access in MANETs

Technological advances in both wireless networking and portable device capabilities have met social popularity and led to an increasing number of devices and services that we use to accomplish our daily tasks. It will not be long before we see a large number of interconnected devices providing services and resources. Subsequently, an efficient way to access the various services and resources will be required [3]. The access will need to take care of the special challenges posed by MANET: (i) the lack of infrastructure; (ii) the limited resources on the terminals; and (iii) the terminal mobility. Moreover, for a broader application of this paradigm in civil applications, the service access will also need to take into account (iv) the autonomy and resulting selfishness of the interconnected hosts of services and resources.

To address the above challenges, our research first includes studying the specification of services, which concerns both their functional and non-functional properties. To address the limited resources (*e.g.*, computing power) on thin devices, our QoS specification incorporates the resource consumption of a service, which is attributed to the service cost. The next step of service access is how to discover remote services and how to select services if there are multiple qualified instances available. We exploit the signal strength to discover reliable services and apply Vickrey auction to select services in order to incentivize service provision. After service discovery and selection, services are invoked, with QoS monitoring carried out in the mean time. The latter provides input for comparing the actually provided to the advertised values, for the purpose of evaluating the trustworthiness of a terminal. And to facilitate the access of services on stranger terminals, we introduce a reputation mechanism that enforces reputation information sharing and honest recommendation elicitation [18]. Complementarily to the above, we have investigated cryptographic protocols for MANETs so as to enable secure communication among peer mobile nodes [4], [1].

6.3.6. Peer-to-peer computing in MANETs

Peer-to-peer networks and ad hoc networks share a good number of characteristics – for example, the lack of a central infrastructure, dynamic topologies or the need for self-organization. In the past, peer-to-peer systems have been used as efficient building blocks for distributed applications on the Internet such as event notification systems or messaging systems. Lately, large research efforts have been made to adapt such peer-to-peer techniques for employment in MANETs, that provide an attractive model for pervasive networking. Based on a peer-to-peer routing algorithm we have developed that provides efficient key-based routing for MANETs, we have demonstrated how range queries (i.e., queries that search the network not just for one particular value but for an entire range of values instead) in such MANETs can be efficiently implemented [25]. Furthermore, we have evaluated a peer-to-peer based approach to resource discovery in MANETs [24]. Using this approach, arbitrary resources can be located in MANETs very efficiently in a completely decentralized manner. These results indicate that peer-to-peer techniques can be successfully employed in MANETs as building blocks for distributed applications that one is accustomed to from the domain of the Internet.

We are currently pursuing a joint project with the HIPERCOM team. The focus of this project lies in combining efficient proactive ad hoc routing for the local network scope with global peer-to-peer key-based routing. The aim is, thus, to provide dynamic and seamless Internet-style routing for mobile ad hoc networks. For a different research project, we are also currently collaborating with the Max Planck Institute for Software Systems in Saarbrücken, Germany. This project addresses the more general question of which data search strategies are most appropriate in the plethora of varying conditions in mobile ad hoc environments. For example, certain networks environments might favor simple, broadcast-based search strategies whereas others might gain significantly from more complex (but harder to maintain) algorithms. Our goal is to identify according to the environment conditions which strategy is superior.

6.3.7. Quality of Service in MANETs

Increasingly, Quality of Service (QoS) sensitive applications, such as streaming media, high-bandwidth content distribution and Voice over IP (VoIP), will be deployed on MANETs as part of the pervasive computing realization. However, traditional QoS guarantee technologies cannot be used directly in MANETs due to the dynamic network environment. We have proposed a QoS management mechanism combining Caching and Backup Service Paths (CBSP) and an Enhanced CBSP (ECBSP) for soft QoS guarantee in MANETs. In CBSP, Service Provider Nodes (SPNs) with distinct Service Paths (SPs) providing the required service are found in the MANET during the service discovery phase. The client node then selects one SP to get the service and the other SPs as Backup Service Paths (BSPs). If the serving SP fails to serve the client node, the client node can handover quickly to a BSP and consume the resource in its cache during the handover operation in order to avoid service interruption. In ECBSP, the required data are further divided into several segments and transmitted concurrently to the client node through different SPs for enhanced availability. Simulation experiments using the ns2 simulator showed that CBSP/ECBSP can improve the performance of applications in MANETs effectively.

7. Other Grants and Activities

7.1. European Initiatives

7.1.1. IST FP6 STREP UBISEC

Participants: Rafik Chibout, Valérie Issarny, Pierre-Guillaume Raverdy.

- **Name:** IST UBISEC – *Ubiquitous Networks with Secure Provision of Services, Access and Content Delivery*
- **URL:** http://www.c-lab.de/ubisec/
- **Related activities:** §6.3.2
- **Period:** [January 2004 - February 2006]
- **Partners:** Siemens Business Services (Germany) – project coordinator, Orga Systems (Germany), France Télécom (France), INRIA (UR Rocquencourt), Universidad Carlos III de Madrid (Spain), Universidad de Malaga (Spain), Universidat Politecnica de Catalunya (Spain), Paderborn University (Germany).

UBISEC's mission is to address new business areas and technologies originating from the integration of public wide area networks (*e.g.*, cellular, Internet), and private corporate and home/SOHO local area networks. The new integrated networks will create new demands in terms of services and will improve quality of life for the users both in their private or professional environment. In order to address the related issues and technology challenges, UBISEC is aiming at an advanced platform for large-scale mobility and security based on SmartCard technologies for context-aware and personalized authorization and authentication services in heterogeneous networks. This requires advanced personalization and localization technologies with high security in order to keep privacy and to protect computing devices, their software components, and personal user data including user profiles. Automatic customization is provided through situation-dependent (context-aware) secure management and access control evolving user, device, and application profiles. Automatic SmartCard-based access control and authentication is preserved by a set of advanced distributed network services which guarantee personalized content delivery through efficient pre-fetching and caching. Flexible service announcement (directory services), discovery, provisioning, and delivery, support the mobile user while moving across heterogeneous networks. Final trials and validation based on the prototypes developed within the project were undertaken at the pervasive computing environment Laboratory from Telefonica Investigación y Desarrollo in Boecillo (Spain) and at the home network laboratory from Paderborn University and SBS in Paderborn (Germany) and demonstrated the feasibility of the UBISEC approach.

7.1.2. IST FP6 IP Amigo

Participants: Sonia Ben Mokhtar, Sébastien Bianco, Yérom-David Bromberg, Nikolaos Georgantas, Valérie Issarny, Daniele Sacchetti, Ferda Tartanoglu, Graham Thomson.

- **Name:** IST Amigo – *Ambient Intelligence for the networked home environment*
- **URL:** http://www.extra.research.philips.com/euprojects/amigo/index.htm
- **Related activities:** §6.2.1, §6.2.2, §6.3.1
- **Period:** [September 2004 - February 2008]
- **Partners:** Philips Research Eindhoven (The Netherlands) – project coordinator, Philips Design - Philips Consumer Electronics (The Netherlands), Fagor (Spain), France Télécom (France), Fraunhofer IMS (Germany), Fraunhofer IPSI (Germany), Ikerlan (Spain), INRIA (URs Rocquencourt, Futurs, Loraine, Rhône Alpes), Italdesign Giugiaro (Italy), Knowledge (Greece), Microsoft (Germany), Telin (The Netherlands), ICCS (Greece), Telefónica I+D (Spain), University of Paderborn (Germany), VTT (Finland).

Home networking has already emerged in specific applications such as PC to PC communication and home entertainment systems, but its ability to really change peoples lives is still dogged by complex installation procedures, the lack of interoperability between different manufacturers equipment and the absence of compelling user services. By focusing on solving these key issues, the Amigo project aims to overcome the obstacles to widespread acceptance of this new technology. The project will develop open, standardized, interoperable middleware and attractive user services, thus improving end-user usability. The project will show the end-user usability and attractiveness of such a home system by creating and demonstrating prototype applications improving everyday life, addressing all vital user aspects: home care and safety, home information and entertainment, and extension of the home environment by means of ambiance sharing for advanced personal communication. The Amigo project will further support interoperability between equipment and services within the networked home environment by using standard technology when possible and by making the basic middleware (components and platform) and basic user services available as open source software together with architectural rules for everyone to use.

7.1.3. IST FP6 STREP Plastic

Participants: Mauro Caporuscio, Manel Fredj, Valérie Issarny, Pierre-Guillaume Raverdy, Roberto Speicys Cardoso.

- **Name:** IST Plastic – *Providing Lightweight and Adaptable Service Technology for Pervasive Information and Communication*
- **URL:** http://www.ist-plastic.org/
- **Related activities:** §6.2.4, §6.2.5, §6.3.2, §6.3.4.
- **Period:** [February 2006 - July 2008]
- **Partners:** INRIA (UR Rocquencourt) – project coordinator, 4D Soft (Hungary), CNR (Italy), IBM (Belgium), Siemens Business Services (Germany), Telefónica I+D (Spain), UCL (United-Kingdom), Universitá di L'Aquila (Italy), Universitá della Svizzera Italiana (Switzerland), Virtual Trip (Greece), Pragmatica Technologies (Argentina).

The vision of PLASTIC is that users in the B3G era should be provided with a variety of application services exploiting the network's diversity and richness, without requiring systematic availability of an integrated network infrastructure. The success of the provided services then depends on the user perception of the delivered QoS. In particular, the network's diversity and richness must be made available and be exploitable at the application layer, where the delivered services can be most suitably adapted. This demands a comprehensive software engineering approach to the provisioning of services, which encompasses the full service life cycle, from development to validation, and from deployment to execution. The PLASTIC project aims to offer a provisioning platform for software services deployed over B3G networks. The platform will enable dynamic adaptation of services to the environment with respect to resource availability and delivered QoS, via a development paradigm based on Service Level Agreements and resource-aware programming. The middleware will be service oriented, to enable integration and composition of heterogeneous software services from both infrastructure-based and ad hoc networks. The middleware will further integrate key functions for supporting the management of adaptive services in the open wireless environment, dealing with resource awareness and dependability.

7.2. International Research Networks and Work Groups

7.2.1. ESF Scientific Programme MiNEMA

- **Name:** ESF Scientific Programme – *Middleware for Network Eccentric and Mobile Applications*
- **URL:** http://www.minema.di.fc.ul.pt/index.html
- **Period:** [September 2003 - August 2008]
- **Steering Committee:** University Klagenfurt (Austria), KU Leuven (Belgium), University of Cyprus (Cyprus), Aarhus University (Denmark), University of Helsinki (Finland), University of Ulm (Germany), TCD (Ireland), University of Lisboa (Portugal), CTH (Sweden), EPFL (Switzerland), Lancaster University (UK).

MiNEMA is a European Science Foundation (ESF) Scientific Programme aiming to bring together European groups from different communities working on middleware for mobile environment. The programme intends to foster the definition and implementation of widely recognized middleware abstractions for new and emerging mobile applications. The programme includes the following planned activities:

- Short term visit exchanges among the programme participants (PhD students).
- Organization of a "closed" workshop for programme participants, to allow the dissemination of early research results and experiences.
- Sponsoring of workshops and conferences in the area of MiNEMA.
- Organization of a summer school on the subjects covered by the programme.

7.2.2. ERCIM WG RISE

- **Name:** ERCIM Working Group – *Rapid Integration of Software Engineering Techniques*
- **URL:** http://rise.uni.lu/tiki/tiki-index.php
- **Period:** [Created 2004]
- **Participants:** CCLRC (UK), CNR (Italy), CWI (The Netherlands), FNR (Luxembourg), FORTH (Greece), Fraunhofer FOKUS & IPSI (Germany), INRIA (UR Rocquencourt), LIRMM (France), NTNU (Norway), SARIT (Switzerland), SICS (Sweden), SpaRCIM (Spain), SZTAKI (Hungary), University of Newcastle (UK), VTT (Finland).

The main aim of the RISE working group is to conduct research on providing new, integrated and practical software engineering approaches that are part of a methodological framework and that apply to new and evolving applications, technologies and systems. In order not to consider all the scope of software engineering, the RISE working group focuses on the following sub domains: Softwares/Systems Architectures, Reuse, Testing, Model Transformation/Model Driven Engineering, Requirement Engineering, Lightweight formal methods, and CASE tools.

The RISE working group limits also its researches to specific application domains for the problems and solutions it proposes. The starting application domains proposed are: Web Systems, Mobility in Communication Systems, High Availability Systems, and Embedded Systems.

7.2.3. ERCIM WG SESAMI

- **Name:** ERCIM Working Group – *Smart Environments and Systems for Ambient Intelligence*
- **URL:** http://www.ics.forth.gr/sesami/
- **Period:** [Created 2006]
- **Participants:** LORIA (France), INRIA (UR Rocquencourt), CNR (Italy), University of Luxemburg (Luxembourg), VTT (Finland), SpaRCIM (Spain), Fraunhofer FOKUS & IPSI (Germany), University College Dublin (UK), CWI (The Netherlands), Middlesex University (UK), KU Leuven (Belgium), IMAG (France), University of Linz (Austria), University of Graz (Austria), University of Thessaly (Greece), University of Salzburg (Austria), University of Kiel (Germany), University of Kaiserslauten (Germany), University of Munich (Germany).

Ambient Intelligence represents a vision of the (not too far) future where "intelligent" or "smart" environments and systems react in an attentive, adaptive, and active (sometimes even proactive) way to the presence and activities of humans and objects in order to provide intelligent/smart services to the inhabitants of these environments.

Ambient Intelligence technologies integrate sensing capabilities, processing power, reasoning mechanisms, networking facilities, applications and services, digital content, and actuating capabilities distributed in the surrounding environment. While a wide variety of different technologies is involved, the goal of Ambient Intelligence is to hide their presence from users, by providing implicit, unobtrusive interaction paradigms. People and their social situations, ranging from individuals to groups, be them work groups, families or friends and their corresponding environments (office buildings, homes, public spaces, etc) are at the center of the design considerations.

The ERCIM Working Group SESAMI aims to facilitate the continued collaboration of researchers and practitioners working on the design, implementation and evaluation of Ambient Intelligence systems and applications, on the grounds of ongoing, and potentially cross-domain, basic and applied, research and development. In this context, SESAMI will pursue novel insights on designing, implementing, managing and maintaining smart computational environments of any scale, in order to effectively enhance and go beyond traditional support of human activities for any given situation, context, role, mission, and task.

7.2.4. ERCIM WG STM

- **Name:** ERCIM Working Group – *Security and Trust Management*
- **URL:** http://www.iit.cnr.it/STM-WG/
- **Period:** [Created 2005]
- **Participants:** British Telecom, CLRC (UK), CNR (Italy), CETIC (Belgium), CWI (The Netherlands), DTU (Denmark), FORTH-ICS (Greece), FNR (Luxembourg), Fraunhofer-SIT (Germany), HP, IBM Research, INRIA (URs Rocquencourt & Sophia Antipolis), IUC (Ireland), L3S (Germany), Marasyk University (Czech Republic), Microsoft EMIC (Germany), NTNU (Norway), Politecnico Torino (Italy), SAP (Germany), SARIT (Switzerland), SICS (Sweden), Siemens Corporate Technology, SparCIM (Spain), SZTAKI (Hungary), VTT (Finland), Eindhoven University of Technology (The Netherlands), University of Milan (Italy), University of Roma Tor Vergata (Italy), University of Trento (Italy), University of Twente (The Netherlands), VCPC, W3C.

The pervasive nature of the emerging Information and Communication Technologies (ICT) expands the well known current security problems on ICT, due to the increased possibilities of exploiting existing vulnerabilities and creating new threats. On the other hand, it poses new problems in terms of possible attack scenarios, threats, menaces and damages. Moreover, the increased virtual and physical mobility of the users enhances their interaction possibilities. Thus, there is a demand for a reliable establishment of trust relationships among the users. Privacy is also a main concern in the current ambient intelligence paradigm: everywhere there are devices interacting with users and information about the users is possibly being gathered by the devices at anytime. All these problems are perceived at different levels of concern by users, technology producers, scientific and governance communities.

This ERCIM Working Group aims at focusing the research of the ERCIM institutions on a series of activities (*e.g.*, projects and workshops) for fostering the European research and development on security, trust and privacy in ICT. These will be among the main issues of current and future research efforts for "security" in a broad sense in Europe (http://www.cordis.lu/security/).

7.2.5. PLATON MobWS

- **Name:** PLATON Collaboration – *Context-aware mobile Web services for nomadic e-business*
- **URL:** http://www.egide.asso.fr/fr/programmes/pai/resultats/details.jhtml?id=6763&template=detail.htm
- **Period:** [January 2005 - December 2006]
- **Partners:** INRIA (UR Rocquencourt), University of Ioannina-DSLab.

A common research project of the ARLES team and the University of Ioannina-DSLab, aiming at QoS-aware mobile e-business systems. We contribute with an approach to dynamic reconfiguration of mobile systems.

7.3. Industrial Projects

7.3.1. MR_SDP: Service Discovery for Multi-radio Networks

Participants: Damien Charlet, Rafik Chibout, Nikolaos Georgantas, Valérie Issarny.

- **Name:** MR_SDP – *Service Discovery for Multi-radio Networks*
- **Related activities:** §6.3.3
- **Period:** [July 2005 - June 2006]
- **Partner:** Alcatel.

This is a research effort on the design and prototype implementation of a new service discovery protocol for multi-radio networks, which shall enable the discovery of heterogeneous resources that are accessible through the various radio networks offered by todays wireless devices, while minimizing resource consumption.

7.3.2. DYONISOS: Dynamic Organization and Instantiation of Systems-of-Systems

Participants: Sonia Ben Mokhtar, Sébastien Bianco, Nikolaos Georgantas, Valérie Issarny, Ferda Tartanoglu, Letian Rong.

- **Name:** Carroll Dyonisos – Phase 1 – *Dynamic Organization and Instantiation of Systems-of-Systems*
- **Related activities:** §6.2.2
- **Period:** [April 2006- February 2007]
- **Partner:** CEA, THALES.

DYONISOS takes place within the context of an ongoing R&D effort in Thalès addressing the engineering of Systems of Systems (SoSs). The first phase of the DYONISOS project is a concept assessment effort targeting advanced SOA concepts and technology for dynamic service composition and workflow execution. The industrial objective is to: (i) assess the relevance and applicability of the advanced SOA concepts for architecting and configuring SoSs across their life cycle (from definition time to operation time), and (ii) assess the feasibility and added value of the advanced SOA technology regarding the implementation of execution mechanisms for dynamic SoS instantiation. DYONISOS Phase 1 is a collaboration between the ARLES team, CEA, and three Thalès entities: DLJ, SC2 and TRT. The project is built around a case-driven approach, where a representative DLJ case study is conducted, which drives the conceptual and technical developments.

7.4. Ministry Grant

7.4.1. ACI CorSS

- **Name:** ACI CorSS – A formal approach to the composition and refinement of system services
- **URL:** http://www.irit.fr/CORSS
- **Related activities:** §6.2.2
- **Period:** [September 2003 - August 2006]
- **Partners:** SVF FERIA (Toulouse) – project coordinator, INRIA (UR Rocquencourt), OBASCO/LOAC at Ecole des Mines de Nantes (Nantes), COMPOSE at INRIA/LABRI (Bordeaux), MOSEL at LORIA (Nancy).

The CorSS project is a joint work between teams from the system community and teams from the formal methods community. Its aim is to study development mechanisms for ensuring the safety of the system services that are to be certified. The underlying development concepts are refinement and composition. The project in particular investigates specific formalisms, well suited for the development of systems, as well as their needs in terms of refinement and composition. More specifically, the project considers feature interaction for telecommunication software, the derivation of robust Web services, and the composition of basic OS kernel services for which it examines relevant composition techniques and proof methods.

7.4.2. Service-Oriented Multi-Radio Networks

- **Name:** Service-Oriented Multi-Radio Networks
- **Related activities:** §6.2.5
- **Period:** [2006]
- **Partners:** Laboratoire d'Informatique de l'Université de Franche-Comté, INRIA (UR Rocquencourt)

Based on the works realized on the subject of network convergence, this project has the goal of studying B3G networks as the coupling of available radio networks where different interfaces can be used concurrently. Those interfaces can be heterogeneous both in terms of provided capabilities and in terms of the network architecture to which they are connected. The use of multiple networks, however, does not guarantee better performance. Using two different networks at the same time can improve communication (by aggregating network bandwidth), degrade communication (as a result of higher energy consumption) or have unpredictable results (depending on the capabilities and the networks to which the interfaces are connected).

On a multi-network communication scenario, optimally defining which networks to use for a service access in such a way as to decrease global energy consumption is an NP-complete problem. This project aims at formally defining decision strategies that have close to optimal results but that are not resource-intensive so that they can be frequently used to adapt to the high dynamics of pervasive networks.

8. Dissemination

8.1. Involvement within the Scientific Community

8.1.1. Programme Committees

- S. Ben Mokhtar is PC member of MoSO 2006: Workshop on Mobile Services and Ontologies, at the 7th International Conference on Mobile Data Management (MDM'06).
- N. Georgantas is co-organizer of RSPSI'06: 1st International Workshop on Requirements and Solutions for Pervasive Software Infrastructures, at the 4th International Conference on Pervasive Computing (Pervasive'06)

- N. Georgantas is PC member of FRCSS'06: 1st International EASST-EU Workshop on Future Research Challenges for Software and Services, associated to ETAPS'06.
- N. Georgantas is co-organizer of FuncAAL'07: 1st International Workshop on Functional Architectures for Assisted Living Services, at the 5th International Conference on Pervasive Computing (Pervasive'07).
- N. Georgantas is PC member of ICSOFT'06: 1st International Conference on Software and Data Technologies.
- N. Georgantas is PC member of SIPE'06: 1st International Workshop on Services Integration in Pervasive Environments, at the IEEE International Conference on Pervasive Services (ICPS'06).
- V. Issarny is PC member of MP2P'06: International Workshop on Mobile Peer-to-Peer Computing.
- V. Issarny is PC member of iTrust'06: International Conference on Trust Management.
- V. Issarny is PC member of ICDCS'06: IEEE International Conference on Distributed Computing Systems.
- V. Issarny is Topic vice-chair of Euro-Par'06 – Topic on Mobile and Ubiquitous Computing.
- V. Issarny is PC member of PDP'2006 - Special session on mobile computing, 14th EUROMICRO Conference on Parallel, Distributed and Network-based Processing.
- V. Issarny is PC member of Sens'2006: International Workshop on Semantics-enabled Networks and Services.
- V. Issarny is PC member of MoSO 2006: Workshop on Mobile Services and Ontologies, at the 7th International Conference on Mobile Data Management (MDM'06).
- V. Issarny is PC member of MDM'2006 & 2007: IEEE International Conference on Mobile Data Management.
- V. Issarny is PC vice-chair of InterSense'2006: 1st International Conference on Integrated Internet Ad hoc and Sensor Networks.
- V. Issarny is PC member of WADS'06: DSN 2006 Workshop on Architecting Dependable Systems.
- V. Issarny is PC member of MIDAS'06: IEEE 1st International Workshop on Middleware for Mobile Ad hoc and Sensor Networks.
- V. Issarny is PC member of ECBS'2006 Workshop: Workshop on Concepts, Patterns, and Mechanisms for the self-organized Integration of Networked Systems.
- V. Issarny is PC member of FSE'2006: ACM 14th Symposium on Foundations of Software Engineering.
- V. Issarny is PC member of WS-MaTe'06: International Workshop on Web Services - Modeling and Testing.
- V. Issarny is PC member of EFTS'06: International Workshop on Engineering of Fault-Tolerant Systems.
- V. Issarny is PC member of TSPUC'2006: Second International Workshop on Trust, Security and Privacy for Ubiquitous Computing.
- V. Issarny is PC member of EuroMicro'2006: 32nd EUROMICRO CONFERENCE – Track on "Multimedia & Telecommunications: Dependable Adaptive Systems.
- V. Issarny is PC member of ATC'2006: The 3rd IFIP International Conference on Autonomic and Trusted Computing.
- V. Issarny is PC member of EWSA'2006: 3rd European Workshop on Software Architecture, co-located with the French-Speaking Conference on Software Architecture (CAL'2006).
- V. Issarny is PC member of CAL'2006: 1ère Conférence Francophone sur les Architectures Logicielles.

- V. Issarny is PC member of HPCC'2006: Second International Conference on High Performance Computing and Communications – Special Topic on Wireless and Mobile Computing.
- V. Issarny is PC member of AmI.d'2006: International Conference on Ambient Intelligence Developments.
- V. Issarny is PC member of STM'2006: 2nd International Workshop on Security and Trust Management.
- V. Issarny is PC member of CFSE'2006: 5ème Conférence Française sur les Systèmes d'Exploitation.
- V. Issarny is PC member of ICEBE'2006: IEEE International Conference on E-Business Engineering.
- V. Issarny is PC member of MPAC'2006: 4th International Workshop on Middleware for Pervasive and Ad-Hoc Computing at Middleware'2006.
- V. Issarny is PC member of MW4SOC'2006: Middleware for Service-Oriented Computing Workshop at Middleware'2006.
- V. Issarny is PC member of WICSA'2007: Sixth Working IEEE/IFIP Conference on Software Architecture.
- V. Issarny is PC member of WOSP'2007: ACM 6th Workshop on Software and Performance.
- V. Issarny is PC member of CAMGC'2007: International Workshop on Context-Awareness and Mobility in Grid Computing at 7th IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2007).
- V. Issarny is PC member of FASE'2007 & 2008: International Conference on Fundamental Approaches to Software Engineering.
- V. Issarny is PC member of DATE'2007 - Topic: Middleware and hardware dependent software for Embedded Systems.
- V. Issarny is PC member of Coordination'2007: 9th International Conference on Coordination Models and Languages.
- V. Issarny is PC member of TOOLS'2007: International Conference TOOLS EUROPE 2007 - Objects, Models, Components, Patterns.
- V. Issarny is TPC member of MobileSummit'2007: 16th IST Mobile & Wireless Communications Summit.
- V. Issarny is PC member of ESEC-FSE'07: 11th European Software Engineering Conference and the 15th ACM SIGSOFT Symposium on the Foundations of Software Engineering.
- P. Poizat is co-organizer and PC member of FOCLASA'07: 6th International Workshop on the Foundations of Coordination Languages and Software Architectures, at the 18th International Conference on Concurrency Theory (CONCUR'07);
- G. Xie is PC member of MSN 2006: 2nd International Conference on Mobile Ad-hoc and Sensor Networks
- G. Xie is PC member of CTC 2006: 4th Chinese Testing Conference

8.1.2. Other activities

- Y.-D. Bromberg was reporter of the OFTA working group on Pervasive Computing until April 2006.
- D. Fournier is reporter of the OFTA working group on Pervasive Computing since May 2006.
- N. Georgantas is member of the review committee of the special issue on "Adaptation et gestion du contexte" of the journal Ingénierie des systèmes d'information (Hermès).

- N. Georgantas is member of the jury of the Prix ASF (Association ACM SIGOPS de France) de la recherche en système 2006 (prize for best thesis in the systems domain).
- V. Issarny is associate editor for the software engineering area of the ACM Computing Surveys.
- V. Issarny is coordinator of the FP6 IST STREP PLASTIC project.
- V. Issarny is chair of the executive committee of the AIR&D consortium on Ambient Intelligence Research and Development (http://www-rocq.inria.fr/arles/AIRD/).
- V. Issarny is coordinator of the OFTA (http://www.ofta.net) working group on Pervasive Computing.
- V. Issarny coordinated the module on "Ubiquitous computing: State of the art and challenges for the software infrastructure" at Smart University'2006.
- P. Poizat is member of the recruiting commissions for Computer Science Assistant Professors in University of Evry, France and in CNAM Paris, France.

8.2. Teaching

- S. Ben Mokhtar gives a course on Distributed Objects Architectures (laboratory). Final year of the five-year computer engineering degree at the Institut des Sciences et Techniques des Yvelines of the University of Versailles Saint-Quentin en Yvelines.
- S. Ben Mokhtar gives a course on Java Programming. First year of the three-year engineering degree at École Polytechnique, Palaiseau.
- M. Fredj gives a course on Middleware Architectures (laboratory). Final year of the five-year computer engineering degree at the Ecole Supérieure d'Ingénierie Léonard de Vinci of the Pôle Universitaire Léonard de Vinci
- N. Georgantas gives a course on Middleware Architectures (lectures). Final year of the five-year computer engineering degree at the Ecole Supérieure d'Ingénierie Léonard de Vinci of the Pôle Universitaire Léonard de Vinci.
- N. Georgantas gives a course on Distributed Objects Architectures (lectures). Final year of the five-year computer engineering degree at the Institut des Sciences et Techniques des Yvelines of the University of Versailles Saint-Quentin en Yvelines.
- V. Issarny gives a course on Software Architectures for Distributed Systems (lectures), as part of the SAL course of the Master 2 COSY of the University of Versailles Saint-Quentin en Yvelines.

8.3. Internships

During year 2006, members of the ARLES project-team supervised the work of the following student interns:

- Ahmad Abdulwakeel, "Service Access in Ubiquitous Service Computing Environments", Master of Science in Software Engineering of Distributed Systems, Royal Institute of Technology, Stockholm, Sweden.
- Javed Ahmed, "Interoperable Web Services Discovery", Master of Science in Software Engineering of Distributed Systems, Royal Institute of Technology, Stockholm, Sweden.
- Sébastien Armand, "Study of the Scalability of the Musdac Platform", Summer internship, Ecole Centrale de Lyon.
- Feng Shi, "A Graphical Tool for Editing Semantic Service Specifications", 3ème année du cycle ingénieur, Institut des Sciences et Techniques des Yvelines of the University of Versailles Saint-Quentin en Yvelines, Versailles.
- Issam Jra, "Implementation of Service Discovery Protocol Plug-ins for Multi-radio Terminals", 2ème annee du cycle ingenieur, Institut des Sciences et Techniques des Yvelines of the University of Versailles Saint-Quentin en Yvelines, Versailles.

- Anupam Kaul, “On line Reasoning tool for Semantic Matching of Web Services in Ambient Intelligence Environments”, Master of Science in Software Systems Engineering, RWTH Aachen, Germany.
- Youssef Laarouchi, “Design of an Event-based Engine for Interoperability in Mobile Computing Environments”, 3ème année du cycle ingénieur, Ecole Nationale Superieure Electronique, Informatique et Radiocommunications, Bordeaux.
- Farouz Masood, “Design and Implementation of Middleware Components for Adaptive Mobile Services in Hybrid Networks”, Master of Science in Software Engineering of Distributed Systems, Royal Institute of Technology, Stockholm, Sweden.

8.4. Invited Talks

Members of the ARLES project-team gave presentations at conferences and workshops, as listed in the publication section. They also gave the following talks:

- S. Ben Mokhtar. “Efficient semantic service discovery in pervasive computing environments”, PhD Student Thematic Workshop on Service Description and Discovery for MANETS, University of Lancaster, September 2006, Lancaster, United-Kingdom.
- S. Ben Mokhtar. “Semantic service discovery and composition in pervasive computing environments”, Seminar, RMIT University of Melbourne, December 2006, Melbourne, Australia.
- S. Ben Mokhtar. “Semantic services discovery and composition in pervasive computing environments”, Seminar, University of New South Wales, December 2006, Sydney, Australia.
- M. Fredj. “Connectivity loss in pervasive computing environments: Open issues”, PhD session, Lugano Summer School on Dependable Software Systems, July 2006, Lugano, Switzerland.
- M. Fredj. “Solutions to connectivity loss in pervasive computing environments”, Seminar, Journée d’adaptation, October 2006, Perpignan, France.
- N. Georgantas. “Interoperable service-oriented architecture for ambient intelligence in the networked home”, Tutorial, SmartUniversity, Ubiquitous computing: State of the art and challenges for the software infrastructure, September 2006, Sophia Antipolis, France.
- V. Issarny. “A service-oriented approach to pervasive computing”, Keynote, International Workshop on Pervasive Systems (PerSys), October 2006, Montpellier, France.
- V. Issarny. “Interoperability in pervasive computing environments”, Seminar, Charles University, April 2006, Prague, Czech Republic.
- V. Issarny. “Engineering software systems for ubiquitous computing”, Tutorial, SmartUniversity, Ubiquitous computing: State of the art and challenges for the software infrastructure, September 2006, Sophia Antipolis, France.
- R. Speicys Cardoso. “Dynamic Adaptation using Mobile Agents in Ubiquitous Computing”, XIII Concurso Latinoamericano de Tesis de Maestría, August 2006, Santiago, Chile.
- G. Xie. “Traffic Flow Measurement at Application Level for High Speed Network”. Presentation at France Telecom R&D. July 2006, Paris, France.

8.5. Awards

- J. Liu. “Best Chinese Student on a Foreign Country 2006”, Chinese Embassy in France, May 2006.
- R. Speicys Cardoso. “Best Latin-American Master’s Thesis in Computer Science 2005 - 3rd place”, Centro Latinoamericano de Estudios en Informatica (CLEI), August 2006.

9. Bibliography

Year Publications

Doctoral dissertations and Habilitation theses

[1] R. BHASKAR. *Protocoles cryptographiques pour les réseaux ad hoc*, Ph. D. Thesis, Ecole Polytechnique, June 2006.

[2] Y.-D. BROMBERG. *Résolution de l'hétérogénéité des intergiciels d'un environnement ubiquitaire*, Ph. D. Thesis, University of Versailles-Saint Quentin en Yvelines, December 2006.

[3] J. LIU. *Supporting QoS-aware Service Discovery in Ubiquitous Computing Environments*, Ph. D. Thesis, University of Versailles-Saint Quentin en Yvelines, July 2006.

Articles in refereed journals and book chapters

[4] D. AUGOT, R. BHASKAR, V. ISSARNY, D. SACCHETTI. *A three round authenticated group key agreement protocol for ad hoc networks*, in "Pervasive and Mobile Computing Journal", to appear, January 2007.

[5] N. GEORGANTAS, P. INVERARDI, V. ISSARNY. *Software Platforms*, in "True Visions: Tales on the Realization of Ambient Intelligence", 2006.

[6] N. GEORGANTAS, V. ISSARNY, C. CERISARA. *Dynamic Synthesis of Natural Human-Machine Interfaces in Ambient Intelligence Environments*, in "Ambient Intelligence, Wireless Networking, Ubiquitous Computing", 2006.

[7] A. ZARRAS, M. FREDJ, N. GEORGANTAS, V. ISSARNY. *Rigorous Development of Complex Fault-Tolerant Systems*, Lecture Notes in Computer Science, chap. Engineering Reconfigurable Distributed Software Systems: Issues Arising for Pervasive Computing, Springer Berlin, 2006, p. 364-386.

Publications in Conferences and Workshops

[8] S. BEN MOKHTAR, N. GEORGANTAS, V. ISSARNY. *COCOA: COnversation-based Service COmposition in PervAsive Computing Environments*, in "Proceedings of the IEEE International Conference on Pervasive Services (ICPS'06)", June 2006.

[9] S. BEN MOKHTAR, A. KAUL, N. GEORGANTAS, V. ISSARNY. *Efficient Semantic Service Discovery in Pervasive Computing Environments*, in "Proceedings of the ACM/IFIP/USENIX 7th International Middleware Conference", November 2006.

[10] S. BEN MOKHTAR, A. KAUL, N. GEORGANTAS, V. ISSARNY. *Towards Efficient Matching of Semantic Web Service Capabilities*, in "Proceedings of WS-MaTe - International Workshop on Web Services - Modeling and Testing", June 2006.

[11] A. BERTOLINO, W. EMMERICH, P. INVERARDI, V. ISSARNY. *Softure: Adaptable, Reliable and Performing Software for the Future*, in "Proceedings of the 1st International EASST-EU Workshop on Future Research Challenges for Software and Services (FRCCS'06)", April 2006.

[12] Y.-D. BROMBERG, V. ISSARNY, P.-G. RAVERDY. *Interoperability of Service Discovery Protocols: Transparent versus Explicit Approaches*, in "Proceedings of the 15th IST Mobile & Wireless Communications Summit", June 2006.

[13] M. CAPORUSCIO, D. CHARLET, V. ISSARNY, A. NAVARRA. *Energetic Performance of Service-oriented Multi-radio Networks: Issues and Perspectives*, in "Proceedings of 6th International Workshop on Software and Performance (WOSP07)", to appear, February 2007.

[14] M. CAPORUSCIO, V. ISSARNY. *A UML 2.0 Profile for Architecting B3G Applications*, in "Proceedings of the 3rd International Workshop on Rapid Integration of Software Engineering techniques (RISE 2006)", September 2006.

[15] D. CHARLET, R. CHIBOUT, V. ISSARNY. *Service Discovery in Multi-radio Networks: An Assessment of Existing Protocols*, in "Proceedings of MSWIM'06 - The 9th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems", October 2006.

[16] D. FOURNIER, S. BEN MOKHTAR, N. GEORGANTAS, V. ISSARNY. *Towards Ad hoc Contextual Services for Pervasive Computing*, in "Proceedings of the Middleware for Service Oriented Computing (MW4SOC) Workshop of the 7th International Middleware Conference 2006", November 2006.

[17] M. FREDJ, A. ZARRAS, N. GEORGANTAS, V. ISSARNY. *Adaptation to Connectivity Loss in Pervasive Computing Environments*, in "Proceedings of the 4th MiNEMA Workshop", July 2006.

[18] J. LIU, V. ISSARNY. *An Incentive Compatible Reputation Mechanism for Ubiquitous Computing Environments*, in "Proceedings of PST'06: International Conference on Privacy, Security & Trust", October 2006.

[19] P. POIZAT, G. SALAÜN, M. TIVOLI. *An Adaptation-based Approach to Incrementally Build Component Systems*, in "International Workshop on Formal Aspects of Component Software (FACS'2006)", September 2006.

[20] P.-G. RAVERDY, S. ARMAND, V. ISSARNY. *Scalability Study of the MUSDAC Platform for Service Discovery in B3G Networks*, in "Proceedings of Wireless World Research Forum Meeting (WWRF-17)", November 2006.

[21] P.-G. RAVERDY, V. ISSARNY, R. CHIBOUT, A. DE LA CHAPELLE. *A Multi-Protocol Approach to Service Discovery and Access in Pervasive Environments*, in "Proceedings of MOBIQUITOUS - The 3rd Annual International Conference on Mobile and Ubiquitous Systems: Networks and Services", July 2006.

[22] P.-G. RAVERDY, O. RIVA, A. DE LA CHAPELLE, R. CHIBOUT, V. ISSARNY. *Efficient Context-aware Service Discovery in Multi-Protocol Pervasive Environments*, in "Proceedings of the 7th International Conference on Mobile Data Management (MDM'06)", May 2006.

[23] R. SPEICYS CARDOSO, V. ISSARNY. *Architecting Pervasive Computing Systems for Privacy : A Survey*, in "Proceedings of 6th Working IEEE/IFIP Conference on Software Architecture (WICSA)", to appear, January 2007.

[24] T. ZAHN, J. SCHILLER. *Performance Evaluation of a DHT-based Approach to Resource Discovery in Mobile Ad Hoc Networks*, in "Fourth Annual Conference on Wireless On demand Network Systems and Services (IEEE/IFIP WONS 2007)", to appear, January 2007.

[25] T. ZAHN, G. WITTENBURG, J. SCHILLER. *Towards Efficient Range Queries in Mobile Ad Hoc Networks using DHTs*, in "Proceedings of the 1st International Workshop on Decentralized Resource Sharing in Mobile Computing and Networking (ACM MobiShare 2006)", September 2006.

Internal Reports

[26] *D3.2: Amigo Middleware Core: Prototype Implementation & Documentation*, Technical report, Amigo Project Deliverable, March 2006.

[27] *D3.3: Amigo Middleware Core Enhanced: Prototype Implementation & Documentation*, Technical report, Amigo Project Deliverable, September 2006.

Miscellaneous

[28] Y.-D. BROMBERG, V. ISSARNY, D. SACCHETTI. *Amigo Interoperable Middleware for the Networked Home Environment*, February 2006, Demonstration at Linux Solution + Solution Open Source for Business.

[29] V. ISSARNY, N. GEORGANTAS, S. BEN MOKHTAR. *Networking Semantic Services for Pervasive Computing*, October 2006, ERCIM News - Issue on Embedded Intelligence.

[30] F. SAILHAN, V. ISSARNY. *Ariadne, un intergiciel pour les réseaux ad hoc*, August 2006, Techniques de l'ingénieur.