INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team ARLES

Software Architectures and Distributed Systems

Rocquencourt

THEME COM

Activity Report

2005

Table of contents

1. Team

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

Administrative assistant
Emmanuelle Grousset [INRIA]

INRIA research scientist
Nikolaos Georgantas [Junior research scientist]

Visiting scientists
Apostolos Zarras [Lecturer, University of Ioannina, Greece, July-August 2005]
Oriana Riva [PhD, University of Helsinki, Finland, July 2005]

Project technical staff
Rafik Chibout [INRIA-UBISEC & Alcatel]
Damien Charlet [INRIA-Alcatel, since September 2005]
Agnès de La Chapelle [INRIA, *Ingénieur associé*, until September 2005]
Pierre-Guillaume Raverdy [INRIA-UBISEC]
Daniele Sacchetti [INRIA-Amigo]
Ferda Tartanoglu [INRIA-Amigo, since October 2005]

Ph.D. Students
Sonia Ben Mokhtar [INRIA fellowship, University of Paris 6]
Raghav Bhaskar [INRIA fellowship, joint PhD with CODES project-team, Ecole Polytechnique]
Yérom-David Bromberg [INRIA fellowship, UVSQ]
Damien Fournier [INRIA fellowship, UVSQ, since December 2005]
Manel Fredj [INRIA fellowship, University of Paris 6, since November 2005]
Jinshan Liu [INRIA fellowship, UVSQ]
Françoise Sailhan [INRIA/*région IdF* fellowship, University of Paris 6, until March 2005]
Roberto Speicys Cardoso [INRIA fellowship, University of Paris 6, since November 2005]
Ferda Tartanoglu [INRIA fellowship, University of Paris 6, until March 2005]

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 mechanising as far as possible the development process;
- Offering middleware infrastructures for both leveraging the complexity associated with the management of distributed resources and dealing with the efficient integration of new technological development.

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 systems' robustness comes from the ability to practically exploit formal methods for modeling the systems' architectures and hence to reason about the systems' behavior. The systems' performance results from the possibility to specialize the systems' composition 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, 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 infrastructures, 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 shall 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 the key point of 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, the open issues that arose at that time are yet to be addressed. Within the ARLES research team, we aim to contribute to the realization of the ambient intelligence vision by bringing 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 aims 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. Also, dependability of the software systems must 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 enabling the development of dependable, ambient intelligence systems. Middleware provide reusable solutions to the management of the pervasive computing environment, ranging from discovering the networked resources that keep changing (in particular from the standpoint of nomadic users), to accessing them. Hence, middleware leverage the complexity of the pervasive computing environment and further promote 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 as whether service providers or 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. 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: *Web services*, *ambient intelligence*, *dependability*, *distributed systems*, *middleware*, *mobile ad hoc networks*, *mobile computing*, *pervasive computing*, *software architecture*, *software engineering*, *system composition*, *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, and
- Middleware infrastructures for leveraging 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 90s. 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 base ground towards assisting the development of robust distributed systems, which is further eased by middleware infrastructures.

3.2.1. Middleware-based and service-oriented software architectures

Available middleware can be classified into three main categories: transaction-oriented middleware that mainly aim at system architectures whose components are database applications; message-oriented middleware that target system architectures whose component interactions rely on publish/subscribe communication schemes; and object-oriented middleware that are based on the remote procedure call paradigm and enable the development of system architectures complying with the object paradigm (e.g., inheritance, state encapsulation), and, hence, enforce 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 have already been released by the W3C[1], which define a core middleware for Web services, 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 actually 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 pose as 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 base ground 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 infrastructure, 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 is already supported by a number of ADL-based development environments targeting system construction, such as the Aster environment that was 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

[1] W3C

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 middleware customized to the application requirements, as supported by the Aster ADL. Another issue that arises when integrating existing components, as provided by middleware infrastructures, 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 description. 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 provided with means for solving architectural mismatches, which further promotes software reuse.

Connectors that are implemented using middleware infrastructures 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 dealing with the safe dynamic evolution of the middleware architectures according to environmental changes, by exploiting both the support for adaptation offered by novel middleware infrastructures (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 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 infrastructures. 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 infrastructures for leveraging the complexity associated with the management of dynamic networks. In this context, ad hoc networking is amongst the most challenging network infrastructures 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 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: (1) reliance on syntactic matching of resource attributes included in the resource description, and (2) unawareness of the environment where the resources are provided. The development of mobile/handheld devices, and wireless and ad hoc networks (e.g., WiFi, 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 WiFi 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.

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. In this context, we have in particular initiated work on group management over ad hoc networks, which allows to abstractly characterize the mobile network on top of which the application is intended to execute and to manage the network on behalf of the application. 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*, *Web services*, *distributed systems*, *information systems*, *mobile systems*.

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 projects to which we contribute (§ 7.1). We have in particular contributed to the development of a demonstrator focused on the away situation in the context of the extended home environment, in collaboration with the INRIA IMARA, LED, MAIA, PAROLE and SARDES project-teams. The demonstrator allows for seamless access to, possibly composite and/or mobile, Web services, both in the local and the wide area, from various mobile terminals (e.g., wireless PDAs, terminals in the car) through a multimodal interface combining speech and gesture.

5. Software

5.1. Introduction

For the sake of validation of our research results, our research activities encompass development of related prototypes. Available prototypes related to our recent results are: (i) a middleware infrastructure based on the Web services architecture for mobile distributed computing that is released as open source software (§ 5.2), and (ii) a scalable service discovery protocol for Mobile Ad hoc NETworks (MANETs) that was released as open source software early 2005 (§ 5.3). We are further developing interoperable and context-aware middleware whose prototype implementation will be released in 2006 (§ 5.4 and 5.5).

5.2. WSAMI: A Middleware based on Web Services for Mobile Distributed Computing

Participants: Rafik Chibout, Valérie Issarny, Daniele Sacchetti.

WSAMI (Web Services for AMbient Intelligence) is based on the Web services architecture and supports the dynamic composition of distributed services over hybrid wireless networks [9]. WSAMI in particular takes in charge the customization of the networks 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 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 of 90KB, as opposed to the 1100KB of the Sun's reference implementation. The overall memory footprint of our Web services platform is of 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 at LGPL). Our prototype is being used for the implementation of demonstrator applications in the field of ambient intelligence [22]. It will further be extended with a number of value-added middleware services for mobility management that integrate our research results in the area.

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

Participants: Rafik Chibout, Valérie Issarny, Françoise Sailhan.

The Ariadne service discovery protocol for MANET has been designed to support decentralized Web service discovery in an hybrid network composed of multi-hop mobile ad hoc and/or infrastructure-based and/or wired networks [23]. 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 services 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 Ariadne solution to service discovery in MANETs is further presented in Section 6.3.5 and detailed in [23]. 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 services-oriented middleware such as WSAMI.

[2]Compact SOAP

The Ariadne prototype is an open source software freely distributed under the terms of the GNU Lesser Public License at (LGPL). Our prototype is being used for the implementation of demonstrator applications exploiting MANETs, in the field of ambient intelligence.

5.4. MSDA: A Middleware for Service Discovery and Access in Pervasive Computing

Participants: Rafik Chibout, Agnès de La Chapelle, Valérie Issarny, Pierre-Guillaume Raverdy.

With the MSDA (Multi-protocol Service Discovery and Access) middleware, the pervasive computing environment is viewed as a dynamic composition of independent networks in which software applications use different protocols for discovering and accessing networked services. MSDA relies on specific plug-ins to interact with existing middleware, manages the efficient dissemination of the service information between the different networks, and enables clients to access all the networked services in them [19]. We have implemented a first prototype of MSDA in Java (J2SE 1.4.2 and 1.5) that 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. As demonstrated by the first MSDA prototype, diverse service discovery protocols can easily be integrated and early performance results are encouraging.

We are currently finalizing the prototype implementation packaging so as to release MSDA as open source software early 2006.

5.5. Interoperable Middleware for Ambient Intelligence

Participants: Sonia Ben Mokhtar, Yérom-David Bromberg, Nikolaos Georgantas, Valérie Issarny, Daniele Sacchetti.

In order to support the development of applications enabling the ambient intelligence vision in open networked environment, we are developing an interoperable middleware supporting the dynamic integration and composition of service providers and consumers, independently of the middleware technology upon which services rely to interact with peer networked services. Our solution decouples into: (i) semantic-based modeling of services to enable semantic matching between provided and required services, and (ii) event-based interoperability that realizes middleware interoperability transparently to applications. First prototype of the event-based interoperability system is already available and has been demonstrated at the Middleware conference (December 2005) [21]; the next prototype version will be released as open source software in 2006.

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, Nikolaos Georgantas, Valérie Issarny, Ferda Tartanoglu, Damien Fournier, Manel Fredj, Apostolos Zarras.

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. In that context, our research activities over year 2005 have focused on the following complementary areas: (i) supporting the dependable composition of systems with a special focus on composite Web services (§ 6.2.1), (ii) semantics-based interoperability for pervasive computing systems (§ 6.2.2), and (iii) semantics-aware services for the mobile computing environment (§ 6.2.3).

6.2.1. Dependable composition of Web services

Web services offer a number of valuable features towards supporting the development of open distributed systems built out of the composition of autonomous services. Nonetheless, the resulting systems must offer a number of non-functional properties and in particular dependability-related ones. However, dependability of composite services can only be achieved according to the recovery property of composed Web services. This calls for the rigorous specification of the standard and exceptional behavior of Web services.

A new Web service conversation language called WS-RESC (Web Service REcovery Support for Conversations) is defined in [24] for the specification of the individual behavior and capabilities of Web services that are relevant to providing dependability. We introduced the WS-RESC language in order to support reasoning about which recovery strategies can be implemented at the composition level. Using WS-RESC, the interface of each Web service can be complemented with additional recovery-related information such as the correct ordering of interactions that is assumed by the service, the transactional behavior, the support for concurrency, etc.

The WS-RESC language, used complementarily with our previously defined service composition language WSCAL (Web Service Composition Action Language), enables the integration of composed Web services with different dependability properties into a dependable composite Web service, called WSCA. The integration of both languages together with a supporting middleware platform for deploying and executing WSCA composite Web services is presented in [4]. In particular, implemented fault tolerance mechanisms and concurrency control in WSCA are customized according to the capabilities of the composed Web services defined in WS-RESC.

6.2.2. Semantics-based interoperability for pervasive computing systems

Among the features characterizing pervasive computing systems are the highly dynamic character of the computing and networking environment and the high heterogeneity of 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 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 from system components abstracted as services is *Service-Oriented Architectures (SOA)*. 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 open pervasive computing environments. Then, 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, interoperability requirements of pervasive computing systems are wider, concerning functional and non-functional interoperability that spans both the middleware and application levels. Thus, interoperability in pervasive computing systems is a major and still open issue. We have been studying in depth the architectural and behavioral features of pervasive computing systems that need to be interoperable and have elaborated a solution towards ad hoc interoperability in open pervasive computing environments, building upon research background from the SOA and Semantic Web Services paradigms [7].

More specifically, we have introduced an *abstract reference service-oriented architecture*, which can represent and integrate existing individual service infrastructures by abstracting their fundamental features [13]. The reference architecture further incorporates support for context-awareness and quality of service (QoS) -awareness, which are key requirements for pervasive computing systems. To enable interoperability between heterogeneous service infrastructures represented by the reference architecture, we have introduced an *enriched service description* which covers the fundamental features included in the reference architecture and establishes a common architectural and behavioral description at a higher, technology-independent level, based on semantic concepts. Based on the enriched service description, we have proposed the deployment of *interoperability mechanisms* within the reference architecture at both application and middleware levels, which aim at establishing semantics-based service interoperability. The proposed interoperability mechanisms comprise *conformance relations* and *interoperability methods*. Conformance relations aim at checking conformance (matching) between services for assessing their capacity to interoperate. Interoperability methods aim at enabling integration of partially conforming services. Then, two services coming from heterogeneous service infrastructures will be interoperable if: (i) they conform semantically (at least partially) in terms of functional and non-functional properties; and (ii) they implement complementary interoperability methods.

Building on the introduced reference architecture and the associated principles, we have worked on two principal research axes: the first one concerns the enriched service description and the second one the interoperability mechanisms. The enriched service description calls for a specialized *language for semantic service specification*. We have introduced a declarative service specification language, which: (i) embodies combined architectural and semantic modeling; (ii) is service & middleware infrastructure-independent; (iii) models both application-level and middleware-level service features; (iv) provides adequate specification abstractions enabling interoperability; and (v) supports both functional and non-functional properties. We have included a set of elements in the specification language, using as starting point Semantic Web Services-oriented languages, and, more specifically, OWL and OWL-S. Regarding the interoperability mechanisms, we have outlined a comprehensive conformance relation that is based on the introduced service specification, covering both application-level and middleware-level conformance and addressing both functional and non-functional service properties. Finally, we are currently working on the runtime execution of interoperability mechanisms (both conformance relations and interoperability methods), which involves associated semantic reasoning. We are targeting lightweight, efficient interoperability mechanisms that can execute on-line on the resource-constrained pervasive devices with acceptable runtime overhead.

6.2.3. Semantics-aware services for the mobile computing environment

Due to the large success of wireless networks and handheld devices, the pervasive computing paradigm is becoming a reality. One of the most challenging objectives to be achieved in pervasive computing environments is to assist nomadic users in realizing tasks anytime, anywhere, on the fly, through the ad hoc composition of functionalities provided by the networked services of the pervasive computing environment. Additionally, context awareness and management allow to adapt the behavior of services according to users and to the networked environment.

In order to enable the ad hoc composition of networked services in pervasive computing environments, we are developing COCOA, a COnversation-based service COmposition middlewAre, in which both networked services of the pervasive computing environment and user tasks are described as workflows of high level capabilities [17]. COCOA integrates two key algorithms: (i) COCOA-SD (COCOA-Service Discovery), which deals with the discovery of networked services of the pervasive computing environment and the selection of services to be integrated for realizing the user task, and (ii) COCOA-CI (COCOA Conversation Integration), which deals with the integration of the conversations of the selected services. The distinctive feature of COCOA is the ability of integrating the conversations of networked services to realize the conversation of the user task, which enforces valid service consumption and conformance to the user's requirements. We have conducted experiments to evaluate the performance of COCOA. Results show that in most realistic cases, COCOA overhead is negligible compared to XML parsing time of services and task descriptions.

We have further proposed: (i) COCOA-QoS, which extends COCOA with QoS-awareness [18], and (ii) COCOA-CTX, which enriches COCOA with context-awareness [16]. COCOA-QoS accounts for the QoS requested by the user task and the one offered by the networked services, and finds composition schemes that provide an estimated QoS that fulfills the one required by the target user task. Experimental results show that the runtime overhead of COCOA-QoS is reasonable, and further, that QoS-awareness improves the overall performance of the composition process. In addition, COCOA-CTX allows providing users with complex, context-sensitive applications, which effectively enables the realization of the pervasive computing vision. Context-awareness is achieved by integrating and disseminating information provided by various sources (e.g., sensors, databases, applications) about the overall context (e.g., users, devices, physical environment).

6.3. Middleware Architectures for Ambient Intelligence Systems

Participants: Raghav Bhaskar, Yérom-David Bromberg, Rafik Chibout, Agnès de La Chapelle, Valérie Issarny, Jinshan Liu, Pierre-Guillaume Raverdy, Oriana Riva, Daniele Sacchetti, Françoise Sailhan, Roberto Speicys Cardoso.

In order to ease the development of distributed systems enabling the ambient intelligence vision, we are investigating supporting core middleware infrastructure and associated services. Our work in this area over year 2005 has concentrated on the following aspects: (i) offering an interoperability system for ambient intelligence systems, which will allow networked software services that are based on heterogeneous middleware technologies (e.g., Jini, UPnP, Web services,) to seamlessly interoperate (§ 6.3.1), (ii) enabling service discovery and access in the heterogeneous networking environment in a way that overcomes both the limited interconnectivity between the different networks available at a location, and the lack of interoperability of the existing discovery and access protocols (§ 6.3.2), (iii) supporting cost-effective service discovery in multi-radio networks (§ 6.3.3), and (iv) enabling efficient service discovery and access in MANETs (§ 6.3.4 and 6.3.5).

6.3.1. Interoperability system for ambient intelligence

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 the inter-networking of heterogeneous middleware. This interoperability system must at least overcome both the heterogeneity of discovery protocols and the one of communication protocols among services. As part of our research work, we have been eliciting such an interoperability system. Specifically, our interoperability 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 [11]. Our interoperability system is further designed so that it can be accommodated by wireless, resource-constrained devices, whether service providers or consumers.

An implementation of our interoperability system is being developed to validate our approach in terms of performance. We provide an interoperability layer that hides middleware heterogeneity to the applications. Thus, we provide efficient interoperability between networked devices, including resource-constrained ones, without requiring any change to applications/services and related middleware.

6.3.2. Multi-protocol service discovery and access

In pervasive environments, the use of various wireless technologies (e.g., cellular networks, WiFi, or Bluetooth) and network management models (e.g., ad hoc- or infrastructure-based) result 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). Moreover, 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.

To overcome these limitations, we have designed the Multi-Protocol Service Discovery and Access (MSDA) middleware that enables mobile users to benefit from all the services in their networked environment [20]. The specifics of our approach is that: (i) MSDA integrates existing discovery and access protocols instead of providing an interoperability layer between them, which allows to benefit from the semantics diversity of protocols at application layer; (ii) MSDA is itself provided as a service through existing protocols; (iii) MSDA manages the dynamic association between independent networks at a location and the dissemination of discovery and access requests; and (iv) MSDA allows for context-aware service discovery.

The MSDA middleware is instantiated in each independent network and is composed of: (i) The MSDA Manager that provides service discovery and access to clients within the network; (ii) Service Discovery and Access (SDA) plug-ins that interact with specific discovery domains (i.e., service discovery protocol in the target wireless network) to collect service information and perform service access; and (iii) MSDA Bridges that assist MSDA Managers in providing service discovery and service access across networks. We have further built and integrated in MSDA, support for context-awareness in order to accomplish more efficient service discovery. Indeed, context information is used not only to select the most appropriate networked service instances, but also to improve the dissemination of service requests across heterogeneous pervasive environments, thus minimizing resource consumption.

We have implemented a first prototype of MSDA in Java (J2SE 1.4.2 and 1.5) that currently includes support for 5 different service discovery protocols (allowing to experiment with both repository-based and mulicast-based protocols), and remote access for SOAP-based services (see § 5.4). We now plan to investigate caching policies for service descriptions in MSDA, first to improve the MSDA response time to clients and also to recover quickly following changes in the network topology. We also plan to investigate additional access protocols in order to support more clients and services, and to tune our implementation for low-power, mobile devices.

6.3.3. Service discovery in multi-radio networks

As in particular addressed by Beyond 3G networks, mobile distributed systems shall be deployed on heterogeneous (or multi-radio) networks, combining infrastructure-based and ad hoc networks, so as to benefit from the respective advantages of the various networks. Development of distributed systems over heterogeneous wireless networks remains an open challenge, which requires dedicated middleware solutions for in particular managing the network's dynamics and resources.

In the above context, we study middleware services for the cost-effective networking of mobile, wireless resources, in particular optimizing service access over heterogeneous networks, with respect to both computer-centric (e.g., resource usage) and user-centric (e.g., perceived quality of service) criteria. We more specifically investigate middleware services for effective service discovery in multi-radio networks. Those services are further being investigated in the context of the WSAMI/CSOAP middleware supporting mobile Web Services.

6.3.4. Efficient service access in mobile ad hoc networks

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 will 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. The access will need to take care of special challenges posed by MANET: (1) the lack of infrastructure; (2)

the limited resources on the terminals; and (3) the terminal mobility. And for broader application of such technologies in civil applications, the service access will also need to take into account (4) the autonomy and resulting selfishness of the interconnected hosts of services and resources.

To address the above challenges, our research includes studying the specification of services, which includes both their functional and non-functional properties. Due to 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 rely on the signal strength to discover reliable services [14] 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 an input regarding how the actual provided QoS is compared to the advertised values, for modeling the trustworthiness of a terminal. And to facilitate the access of services on stranger terminals, we introduce a reputation mechanism to enforce reputation information sharing and honest recommendation elicitation.

6.3.5. *Resource discovery in mobile ad hoc networks*

Ad hoc networks are well suited to support ambient intelligence applications, that is, to provide an immediate access to resources (i.e., content or services) anywhere, anytime, at low cost. In this context, automatic discovery of resources within the network plays an essential role. However, conventional resource discovery solutions are not well suited for ad hoc networks, as they, in particular, use broadcasting to discover service providers. This results in the unavailability of the ad hoc network, which is induced by broadcast storms.

Our work focuses on designing a resource discovery system for MANET (Mobile Ad hoc NETworks) [3]. Our solution is designed so as to limit the induced traffic load, particularly when the number of users increases dramatically [23]. Our solution is based on a subset of self-organized devices (called directories), which are periodically elected to store information about networked resources for the surrounding devices. Then, devices can access information from directories without flooding the network to discover resources. In addition, the system is designed to cope with moving resources, and, thus, related mobility-induced failures, by accounting for the existence of resource replicas within the network together with the quality of service offered by eligible resources. We have implemented a prototype of the proposed resource discovery protocol in the context of mobile Web services, as presented in Section 5.3. The resulting Ariadne protocol allows for the discovery of Web services deployed in the MANET.

Another critical issue in ad hoc networks is to enable users to easily access information from both the local and the wide area (e.g., the Internet). However, we have not yet reached the point where anywhere, anytime network access is actually offered. Infrastructure-based wireless networks use fixed network access points with which mobile terminals interact for communicating. Unfortunately, the unavailability of a base station results in network failure. Ad hoc networking may then be exploited for accessing resources available in the local area, which comes at no cost for users, and possibly accessing a WLAN base station to reach resources available in the wide area. The issue that we are addressing is on setting up an ad hoc network of mobile terminals that cooperate to access resources in the local network, and also to offer utilities intended to discover resources in the global network (i.e., the Internet), when needed. This requires interaction with the base station to gain access to the rich set of available Internet resources, when sought resources are not available in the local area. In this context, we have first concentrated on how to improve the Web latency using a WLAN, exploiting both the ad hoc and infrastructure-based capabilities of the network. Our main design objective was to minimize the energy cost of peer-to-peer communication among mobile terminals, so as to allow for inexpensive access when a fixed access point is not available in the communication range of the mobile terminal.

7. Other Grants and Activities

7.1. European Initiatives

7.1.1. IST FP6 STREP UBISEC

Participants: Rafik Chibout, Agnès de La Chapelle, 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, §6.3.5, §5.3, §5.4
- **Period:** [January 2004 - February 2006]
- **Partners:** Siemens Business Services (Germany) – project coordinator, Orga Systems (Germany), France Télécom (France), INRIA (UR Rocquencourt), Universidad Carlos 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 infrastructure 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 will be 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 will demonstrate the feasibility of the UBISEC approach.

7.1.2. IST FP6 IP Amigo

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

- **Name:** IST Amigo – *Ambient Intelligence for the networked home environment*
- **URL:** http://www.extra.research.philips.com/euprojects/amigo/index.htm
- **Related activities:** §6.2.2, §6.2.3, §6.3.1, §5.5
- **Period:** [September 2004 - February 2008]
- **Partners:** Philips Research Eidhoven (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 and attractiveness. 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 infrastructure) and basic user services available as open source software together with architectural rules for everyone to use.

7.2. International Research Networks and Work Groups

7.2.1. IST WG iTrust

- **Name:** IST WG iTrust – *Working group on trust management in dynamic open systems*
- **URL:** http://www.itrust.uoc.gr/
- **Period:** [September 2002 - August 2005]
- **Partners:** University of Crete (Greece) – project coordinator, CCLRC (UK), CNR-ISTC (Italy), HP (UK), Imperial College (UK), INRIA (UR Rocquencourt), Intracom SA (Greece), King's College (UK), Nine by Nine Co (UK), Plefsis Information Systems SA (Greece), Queen Mary University College (UK), Sintef Telecom and Informatics (Norway), Trinity College Dublin (Ireland), Autonomous University of Barcelona (Spain), University of Dortmund (Germany), University of Oslo (Norway), University of Strathclyde (UK), Virtual Trip Ltd (Greece).

The aim of iTrust is to provide a forum for cross-disciplinary investigation of the application of trust as a means of establishing security and confidence in the global computing infrastructure, recognizing trust as a crucial enabler for meaningful and mutually beneficial interactions. The proposed forum brings together researchers with a keen interest in complementary aspects of trust, from both technology-oriented disciplines and the field of law, social sciences and philosophy. Hence, it aims to provide the consortium participants (and the research communities associated with them) with the common background necessary for advancing toward an in-depth understanding of the fundamental issues and challenges in the area of trust management in open systems.

7.2.2. 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.

7.2.3. 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.

7.2.4. 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 2005]
- **Partners:** INRIA (UR Rocquencourt), University of Ioannina-DSLab.

A common research project of INRIA-ARLES team and the University of Ioannina-DSLab, aiming at QoS-aware mobile e-business systems. We contribute to 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*
- **Period:** [July 2005 - June 2006]
- **Partner:** Alcatel

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

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 activity:** §6.2.1
- **Period:** [September 2003 - August 2006]
- **Partners:** SVF FERIA (Toulouse) – project coordinator, ARLES at INRIA-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 features 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.

8. Dissemination

8.1. Involvement within Scientific Community

8.1.1. Programme Committees

- S. Ben Mokhtar is PC member of the workshop on Mobile Services and Ontologies (MoSO 2006), 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.
- V. Issarny is PC member of MP2P'05 and '06: International Workshop on Mobile Peer-to-Peer Computing.
- V. Issarny is PC member of iTrust'05 and '06: International Conference on Trust Management.
- V. Issarny is PC member of EWSA'05: European Workshop on Software Architecture.
- V. Issarny is PC member of WICSA'05: IEEE/IFIP Working Conference on Software Architecture.
- V. Issarny is PC member of ACM SAC TRECK: Track on Trust, Recommendations, Evidence and other Collaboration Know-how at SAC'2005, 20th Annual ACM Symposium on Applied Computing.
- V. Issarny is PC member of ICDE'05: IEEE 21st International Conference on Data Engineering.
- V. Issarny is PC member of ICDCS'05 and '06: IEEE International Conference on Distributed Computing Systems.
- V. Issarny is PC member of TSPUC'2005: First International Workshop on Trust, Security and Privacy for Ubiquitous Computing. Affiliated with IEEE WOWMOM'2005.

- V. Issarny is PC member of HotDep'2005: First Workshop on Hot Topics in System Dependability. In conjunction with DSN'2005.
- V. Issarny is Topic vice-chair of Euro-Par'05 and '06 – Topic on Mobile and Ubiquitous Computing.
- V. Issarny is PC member of DSN-DCCS'2005: IEEE Dependable Computing and Communication Symposium at International Conference on Dependable Systems and Networks.
- V. Issarny is PC member of Middleware'2005: ACM/IFIP/USENIX 6th International Middleware Conference.
- V. Issarny is PC member of ICEBE'2005: IEEE Conference on e-Business Engineering.
- V. Issarny is PC member of MPAC'2005: 3rd International Workshop on Middleware for Pervasive and Ad-Hoc Computing. A Workshop of Middleware'2005.
- 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. In conjunction with MDM'2006.
- V. Issarny is PC member of MDM'2006: IEEE 7th 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 WOSP'2007: ACM 6th Workshop on Software and Performance.

8.1.2. Other activities

- Y-D Bromberg is reporter of the OFTA working group on Pervasive Computing.
- N. Georgantas is Demonstration Chair of iTrust'2005: 3rd International Conference on Trust Management. May 2005, INRIA, Rocquencourt, France.
- V. Issarny is General Chair of iTrust'2005: 3rd International Conference on Trust Management. May 2005, INRIA, Rocquencourt, France.
- 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 coordinating the OFTA (http://www.ofta.net) working group on Pervasive Computing.
- J. Liu is Publicity Chair of iTrust'2005: 3rd International Conference on Trust Management. May 2005, INRIA, Rocquencourt, France.
- J. Liu is Publicity Chair of STM'2005: 1st International Workshop on Security and Trust Management.

8.2. Teaching

- S. Ben Mokhtar gives a course on Introduction to Java programming as part of the first year of engineering degree at Ecole Polytechnique.
- 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.
- N. Georgantas gives a course on Middleware Architectures (both lectures and 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 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 lecture on Software Architectures for Distributed Systems, as part of the AOD course of the Master 2 COSY of the University of Versailles Saint-Quentin en Yvelines.

8.3. Internships

During year 2005, 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, KTH, Stockholm, Sweden.
- Damien Fournier, "Context Management for distributed mobile systems" *Master de Recherche en Informatique: des Concepts aux Systèmes*, University of Versailles Saint-Quentin-en-Yvelines.
- Manel Fredj, "Dynamic reconfiguration of mobile distributed services", *Master de recherche Systèmes et Applications Répartis (SAR)*, University of Paris 6.
- 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.
- Sonia Nimour, "Group-based picture sharing on pervasive environments", *Stage IUT*, University of Versailles Saint-Quentin-en-Yvelines.

8.4. Invited Conferences

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. “*Composition Dynamiques de Services Web*. Project CoRRS, INRIA Lorraine (LORIA). September 2005, Nancy, France.
- S. Ben Mokhtar. “QoS- and Context-aware Ad Hoc Composition of User Tasks in Pervasive Computing Environments”. Mindswap Lab, University of Maryland. November 2005, College Park, USA.
- S. Ben Mokhtar. “QoS- and Context-aware Ad Hoc Composition of User Tasks in Pervasive Computing Environments”. LSDIS Lab, University of Georgia. November 2005, Athens, USA.
- V. Issarny. “Interoperable middleware for Ambient Intelligence Systems”. EUNICE’2005: "Networked Applications" – 11th Open European Summer School, IFIP WG6.6, WG6.4, WG6.9 Workshop. July 2005, Colmenarejo, Spain.
- V. Issarny. “Interoperable Middleware for Pervasive Computing and its Application to the Networked Home”. ING’2005: 9eme Ecole d’été Internet Nouvelle Génération. June 2005, Côte d’Opal, France.
- V. Issarny. "Ubiquitous Computing Systems for Ambient Intelligence". 2005 Science Week of the Basque Country focused on Ambient Intelligence. November 2005, Bilbao, Spain.
- V. Issarny. "Développement des systèmes intelligents ambiants : prospectives et concepts du futur". Atelier CGTec - Réseaux domestiques et applications, AFNOR. November 2005, Saint-Denis, France.
- J. Liu. “A Distributed Reputation Mechanism for Ubiquitous Computing Environments”. Journées sécurité de l’INRIA. December 2005, Grenoble, France.

9. Bibliography

Books and Monographs

[1] C. CREMERS, V. ISSARNY, S. MAUW (editors). *Proceedings of the 1st workshop on Security & Trust Management*, To appear, ENTCS, 2005.

[2] P. HERMANN, V. ISSARNY, S. SHIU (editors). *Trust Management: Third International Conference, iTrust 2005*, Springer-Verlag GmbH, May 2005.

Doctoral dissertations and Habilitation theses

[3] F. SAILHAN. *Localisation de ressources dans les reseaux ad hoc*, Ph. D. Thesis, University of Paris 6, France, July 2005.

[4] F. TARTANOGLU. *Composition sûre de fonctionnement de services Web (Dependable composition of Web services)*, Ph. D. Thesis, University of Paris 6, France, December 2005.

Articles in refereed journals and book chapters

[5] N. GEORGANTAS, P. INVERARDI, V. ISSARNY. *Software Platforms*, in "True Visions: Tales on the Realization of Ambient Intelligence", LNCS, To appear, Springer Verlag, 2005.

[6] N. GEORGANTAS, V. ISSARNY, C. CERISARA. *Dynamic Synthesis of Natural Human-Machine Interfaces in Ambient Intelligence Environments*, in "Towards Ambient Intelligence for Human-Centric Mobile and Wireless World", To appear, 2005.

[7] N. GEORGANTAS, S. B. MOKHTAR, F. TARTANOGLU, V. ISSARNY. *Semantics-Aware Services for the Mobile Computing Environment*, in "Architecting Dependable Systems III", LNCS, vol. 3549, 2005.

[8] T. HIGUERA, V. ISSARNY. *Improving the Memory Management Performance of RTSJ*, in "Concurrency and Computation: Practice and Experience", April-May 2005.

[9] V. ISSARNY, D. SACCHETTI, F. TARTANOGLU, F. SAILHAN, R. CHIBOUT, N. LEVY, A. TALAMONA. *Developing Ambient Intelligence Systems: A Solution based on Web Services*, in "Journal of Automated Software Engineering", vol. 12, 2005.

Publications in Conferences and Workshops

[10] D. AUGOT, R. BHASKAR, V. ISSARNY, D. SACCHETTI. *An Efficient Group Key Agreement Protocol for Ad hoc Networks*, in "Proceedings of the 1st International Workshop on Trust, Security and Privacy for Ubiquitous Computing (TSPUC'2005)", June 2005.

[11] Y.-D. BROMBERG, V. ISSARNY. *INDISS: Interoperable Discovery System for Networked Services*, in "Proceedings of ACM/IFIP/USENIX 6th International Middleware Conference (Middleware'2005)", November 2005.

[12] M. FILALI, V. ISSARNY, P. MAURAN, G. PADIOU, P. QUEINNEC. *Maximal group membership in ad hoc networks*, in "Proceedings of the 6th International Conference on Parallel Processing and Applied Mathematics (PPAM)", September 2005.

[13] N. GEORGANTAS, S. B. MOKHTAR, Y.-D. BROMBERG, V. ISSARNY, J. KALAOJA, J. KANTAROVITCH, A. GÉRODOLLE, R. MEVISSEN. *The Amigo Service Architecture for the Open Networked Home Environment*, in "Proceedings of 5th Working IEEE/IFIP Conference on Software Architecture (WICSA)", 2005.

[14] J. LIU, V. ISSARNY. *Signal Strength based Service Discovery (S3D) in Mobile Ad Hoc Networks*, in "Proceedings of the 16th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)", September 2005.

[15] J. LIU, F. SAILHAN, D. SACCHETTI, V. ISSARNY. *Group Management for Mobile Ad hoc Networks: Design, Implementation and Experiment*, in "Proceedings of the 6th IEEE International Conference on Mobile Data Management (MDM'2005)", May 2005.

[16] S. B. MOKHTAR, D. FOURNIER, N. GEORGANTAS, V. ISSARNY. *Context-aware Service Composition in Pervasive Computing Environments*, in "Proceedings of the 2nd International Workshop on Rapid Integration of Software Engineering Techniques (RISE'2005)", September 2005.

[17] S. B. MOKHTAR, N. GEORGANTAS, V. ISSARNY. *Ad hoc Composition of User Tasks in Pervasive Computing Environments*, in "Proceedings of the 4th International Workshop on Software Composition", April 2005.

[18] S. B. MOKHTAR, J. LIU, N. GEORGANTAS, V. ISSARNY. *QoS-aware Dynamic Service Composition in Ambient Intelligence Environments*, in "Proceedings of the 2005 IEEE/ACM International Conference on Automated Software Engineering (ASE 2005)", 2005.

[19] P.-G. RAVERDY, R. CHIBOUT, A. DE LA CHAPELLE, V. ISSARNY. *The MSDA Multi-Protocol Approach to Service Discovery and Access in Pervasive Environments (Demonstration)*, in "Middleware'2005 CD", November 2005.

[20] P.-G. RAVERDY, V. ISSARNY. *Context-aware Service Discovery in Heterogeneous Networks*, in "Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'2005)", June 2005.

[21] D. SACCHETTI, Y.-D. BROMBERG, N. GEORGANTAS, V. ISSARNY, J. PARRA, R. POORTINGA. *The Amigo Interoperable Middleware for the Networked Home Environment (Demonstration)*, in "Middleware'2005 CD", November 2005.

[22] D. SACCHETTI, A. TALAMONA, C. CERISARA, R. CHIBOUT, S. B. ATALLAH, W. V. RAEMDONCK. *Seamless Access to Mobile Services for the Mobile User (Demonstration)*, in "Video at The 3rd International Conference on Pervasive Computing (Pervasive 2005)", May 2005.

[23] F. SAILHAN, V. ISSARNY. *Scalable Service Discovery for MANET*, in "Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom'2005)", March 2005.

[24] F. TARTANOGLU, V. ISSARNY. *Specifying Web Services Recovery Support with Conversations*, in "Proceedings of the 38th Hawaii International Conference on System Sciences (HICSS'2005)", January 2005.

Internal Reports

[25] *D2.1: Specification of the Amigo Abstract Middleware Architecture*, Technical report, Amigo Project Deliverable, April 2005.

[26] *D2.2: State of the Art Analysis Including Assessment of System Architectures for Ambient Intelligence*, Technical report, Amigo Project Deliverable, April 2005.

[27] *D3.1b: Detailed Design of the Amigo Middleware Core - Service Specification, Interoperable Middleware Core*, Technical report, Amigo Project Deliverable, September 2005.