

Project-Team arles

*Software Architectures and Distributed
Systems*

Rocquencourt

THEME COM

Activity
R *report*

2004

Table of contents

1. Team	1
2. Overall Objectives	1
3. Scientific Foundations	3
3.1. Introduction	3
3.2. Software Architectures for Distributed Systems	4
3.2.1. Middleware-based and service-oriented software architectures	4
3.2.2. Architecture-based development of distributed systems	5
3.3. Middleware Architectures for Mobile Systems	6
3.3.1. Ad hoc networking	6
3.3.2. Managing the network's dynamics	7
3.3.3. Enforcing dependability	9
4. Application Domains	10
5. Software	10
5.1. Introduction	10
5.2. WSAMI: A Middleware based on Web Services for Mobile Systems	10
5.3. Scalable Service Discovery Protocol for MANETs	12
6. New Results	12
6.1. Introduction	12
6.2. Software Architectures for Distributed Systems	13
6.2.1. Dependable composition of Web services	13
6.2.2. Semantics-aware services for the mobile computing environment	13
6.3. Middleware Architectures for Mobile Systems	15
6.3.1. Middleware infrastructure for ambient intelligence systems	15
6.3.2. Supporting ad hoc, dynamic composition of mobile services	16
6.3.3. Service discovery protocol interoperability in the mobile environment	17
6.3.4. Multi-protocol service discovery middleware	17
6.3.5. Scalable Service Discovery for MANET	18
7. Other Grants and Activities	19
7.1. European Initiatives	19
7.1.1. IST FP5 OZONE	19
7.1.2. IST FP6 STREP UBISEC	20
7.1.3. IST FP6 IP Amigo	20
7.2. International Research Networks and Work Groups	21
7.2.1. IST WG iTrust	21
7.2.2. ESF Scientific Programme MiNEMA	21
7.2.3. ERCIM WG RISE	22
7.3. Ministry Grant	22
7.3.1. ACI CorSS	22
8. Dissemination	23
8.1. Involvement within Scientific Community	23
8.1.1. Programme Committees	23
8.1.2. Other activities	24
8.2. Teaching	24
8.3. Internships	24
8.4. Invited Conferences	24
9. Bibliography	25

1. Team

Head of project-team

Valérie Issarny [Research director, INRIA]

Administrative assistant

Sylvie Loubressac [INRIA, until August 2004]

Emmanuelle Grousset [INRIA]

INRIA research scientist

Nikolaos Georgantas [Junior research scientist]

Part-time research scientist (partner)

Nicole Levy [Professor UVSQ (University of Versailles-Saint Quentin en Yvelines), until September 2004]

Visiting scientist

Paola Inverardi [Professor, University of L'Aquila, Italy, September-December 2004]

Project technical staff

Rafik Chibout [INRIA-Ozone/UBISEC]

Agnès de La Chapelle [INRIA, *Ingénieur associée*, since December 2004]

Pierre-Guillaume Raverdy [INRIA-UBISEC]

Daniele Sacchetti [INRIA-Ozone/Amigo]

Ph.D. Students

Sonia Ben Mokhtar [INRIA fellowship, University of Paris 6, since November 2004]

Raghav Bhaskar [INRIA fellowship, joint PhD with CODES project-team, Ecole Polytechnique]

Yoram-David Bromberg [INRIA fellowship, UVSQ]

Jinshan Liu [INRIA fellowship, UVSQ]

Françoise Sailhan [INRIA/*region IdF* fellowship, University of Paris 6]

Ferda Tartanoglu [INRIA fellowship, University of Paris 6]

2. 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 due 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;
- 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. Systems' robustness comes from the ability to practically exploit formal methods for modeling

the systems' architectures and hence to reason about the systems' behavior. 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 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 reactivity 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. Regarding specifically the development of software systems for ambient intelligence, a key approach lies in the dynamic composition of software systems according to the environment. The objective of our work is thus to offer comprehensive solutions to the dynamic composition of distributed systems, decomposing into:

- The definition of software architecture styles dedicated to mobile distributed systems that are dynamically composed. This shall serve eliciting key interaction paradigms and core properties of the architectural elements, further leading to devising associated architecture modeling and supporting middleware infrastructure.
- The definition of an architecture-based development environment offering an Architecture Description Language (ADL) and associated methods and tools for assisting the development of systems that are dynamically composed, and may in particular be deployed over resource-constrained mobile devices. Results shall ease the modeling of mobile systems and reasoning about the behavior of the systems regarding both functional and non-functional properties. They shall further serve dynamically composing systems according to the environment (e.g., available resources, connectivity).
- The definition of a middleware infrastructure aimed at mobile distributed systems, which integrates mobile nodes that are possibly resource-constrained and communicate via a variety of wireless networks (i.e., infrastructure-based and ad hoc). We are in particular interested in the exploitation of (multi-hop) ad hoc networks, which do not require any infrastructure for accessing remote resources.

3. Scientific Foundations

3.1. Introduction

Keywords: *Web services, ad hoc networks, ambient intelligence, dependability, distributed systems, middleware, mobile 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 mobile distributed 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, 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 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 aims at system architectures whose components are database applications; message-oriented middleware that targets system architectures whose component interactions rely on publish/subscribe communication schemes; and object-oriented middleware that is originally based on the remote procedure call paradigm and enables the development of system architectures complying with the object paradigm (e.g., inheritance, state encapsulation), and, hence, enforces an object model for the system (i.e., the architectural components are objects). Development of middleware-based systems is now quite mature although middleware heterogeneity is still an open issue. In addition, dealing with middleware heterogeneity in the presence of dynamic composition raises the issue of dynamically integrating and possibly adapting the system's components, which is being investigated in the middleware community.

Evolution of middleware and distributed system technologies has further led to the emergence of service-oriented system architectures to cope with the requirements of Internet-based systems. Software services, in particular in the form of XML Web services, offer a promising paradigm for software integration and interoperation. Simply stated, a service is an instantiated configured system, which may be composed with other services to offer a new system that actually realizes a system of systems. Although the definition of the overall Web services architecture is still incomplete, the base standards have already been released by the W3C¹, 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

¹<http://www.w3.org/2002/ws/arch>

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 ubiquitous 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 mobile 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-team². However, most of the work on the specification of connectors has focused on the characterization of the interaction protocols among components, whilst connectors abstracting middleware embed additional complex functionalities (e.g., support for provisioning fault tolerance, security, transactions). The above concern has led the software architecture community to examine the specification of the non-functional properties offered by connectors. For instance, these may be specified in terms of logic formulae, which further enables synthesizing 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., reflexive middleware) and the rigorous specification of software architectures enabled by ADLs.

²<http://www-rocq.inria.fr/arles/work/aster.html>

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 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 network devices and services. This has led to a variety of protocols for lookup and discovery of network 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 network's identity and its characteristics (e.g., bandwidth, cost, reliability), the users, and the 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 the available services and select the most appropriate one.
- **Protocol interoperability:** Many 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 provider's 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

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 domain, 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, 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 new results are: (i) a middleware infrastructure based on the Web services architecture for mobile distributed computing that is available under LGPL license (§ 5.2), and (ii) a scalable service discovery protocol for Mobile Ad hoc NETWORKS (MANETs) that is to be released as open source software early 2005 (§ 5.3). We are further developing interoperable and context-aware middleware, for which implementation will be released in 2005.

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

Participants: Rafik Chibout, Valérie Issarny, Daniele Sacchetti.

Enabling the ambient intelligence vision means that consumers will be provided with universal and immediate access to available content and services, together with ways of effectively exploiting them. From the standpoint of the software system development aspect, this means that the actual implementation of any ambient intelligence application requested by a user can only be resolved at runtime according to the user's specific situation. Towards that goal, we have introduced the WSAMI middleware [2], which supports the abstract specification of Ambient Intelligence applications in the form of software architectures, together with their dynamic composition according to the environment. The proposed middleware builds on the Web services architecture, whose pervasiveness enables service availability in most environments. In addition, dynamic composition of applications is carried out in a way that enforces quality of service for deployed applications in terms of security and performance through the systematic customization of connectors that dynamically integrates relevant middleware-related services.

We have developed a Java-based prototype of the WSAMI core middleware. We use IEEE 802.11b as the underlying WLAN. The WSAMI core middleware prototype subdivides into: (i) the WSAMI SOAP-based core broker, including the CSOAP SOAP container for wireless, resource-constrained devices, and (ii) the Naming and Discovery (ND) service, including support for connector customization, so as to enforce quality of service through the dynamic integration of middleware-related services over the network's path. Figure 1 depicts the main components of the WSAMI prototype implementation, on top of which Web services

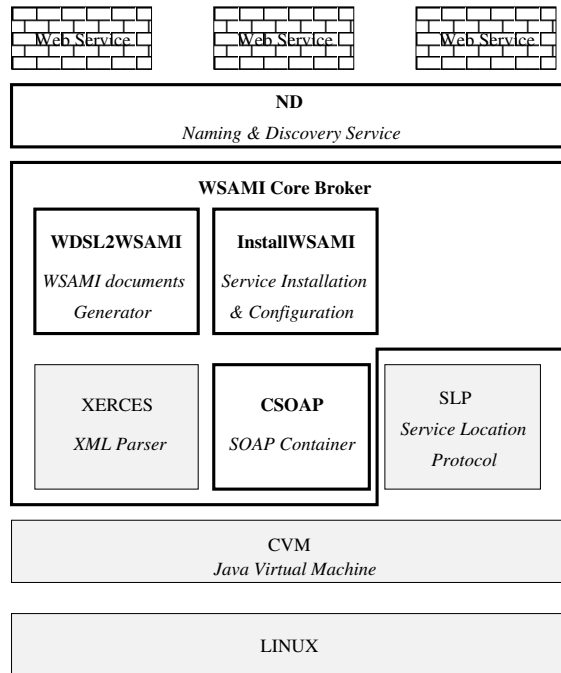


Figure 1. WSAMI core middleware prototype

execute; grayed components denote available implementations that we reuse, while the components that we have developed are highlighted in bold face. Note that the components developed as part of the WSAMI core broker exist in any Web services platform; a new implementation has been elaborated, so as to allow for execution on resource-constrained devices. The CSOAP SOAP container prototype has been developed to cope with the limitations imposed by CVM³ and resource-constrained devices. The CSOAP prototype implementation is mainly based on Sun's JAX-RPC specification⁴, which aims to enable the development of SOAP-based interoperable and portable Web services. Together with the CSOAP container, we have developed the InstallWSAMI tool, which provides deployment and configuration functionalities. The utility for deploying and undeploying Web services is based on a simple XML-based language, specifically defined for the purpose of minimizing the complexity of the system. A related tool for automated stub and skeleton files generation is provided to allow an easy development of Web Services on the server side and of the corresponding client applications. The memory footprint of our CSOAP implementation is 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, including 3MB for the CVM and 815KB for the Xerces XML parser, in addition to the CSOAP implementation. We have further carried out a number of experiments to investigate the performance of our lightweight Web services platform. We have not only assessed it against existing Web services platforms, but also against traditional middleware platforms being considered for mobile environments. Initial results of our assessment are encouraging: (i) the development of ambient intelligence systems using our environment does not add any complexity to the one of Web services, and (ii) performance of the resulting systems is comparable to the one obtained with traditional middleware, and allows for execution on wireless, resource-constrained devices. In addition, the overhead related to the functions handling user mobility (i.e., dynamic

³<http://java.sun.com/products/cdc>

⁴<http://java.sun.com/xml/jaxrpc/>

service composition relying on service discovery and connector customization for enforcing quality of service) compares to the cost of base Web services access.

The WSAMI middleware prototype is an open-source software freely distributed under the terms of the GNU Lesser Public License (LGPL) at <http://www-rocq.inria.fr/arles/download/ozone/index.html>. Our prototype is being used for the implementation of demonstrator applications in the field of ambient intelligence. 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. Scalable Service Discovery Protocol for MANETs

Participants: Rafik Chibout, Valérie Issarny, Françoise Sailhan.

Our 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. It 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. Our protocol further enables services requesters to differentiate services instances according to non-functional properties. Specifically, our 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 (minimizing the broadcast duplicate retransmissions, aggregation of messages), 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. Our solution to service discovery in MANETs is further presented in Section 6.3.5 and detailed in [14]. The prototype of our service discovery protocol is 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.

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:

- Software architectures for distributed systems, where we investigate architectural styles dedicated to mobile distributed systems from which to derive languages for system modeling and related methods and tools for supporting system development (§ 6.2).
- Middleware architectures for mobile systems, building upon architectural styles elicited for mobile distributed software 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 Distributed Systems

Participants: Sonia Ben Mokhtar, Nikolaos Georgantas, Paola Inverardi, ValÃ©rie Issarny, Ferda Tartanoglu.

Building upon our past work on modeling software architectures of closed distributed systems for supporting the systems' analysis and synthesis, we are investigating architectural styles of dependable, mobile, distributed systems that can be dynamically composed out of resources available in the network. Our work in this area over year 2004 has focused on two complementary issues: (i) supporting the dependable composition of systems with a special focus on composite Web services (§ 6.2.1), and (ii) modeling services towards enabling dynamic composition in the mobile environment (§ 6.2.2).

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, for effective exploitation in the e-business domain. 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.

We have defined the WS-RESC (Web Service REcovery Support Conversations) conversation language that addresses the above issue [15]. WS-RESC allows the thorough specification of both the standard and exceptional behavior of autonomous, composable Web services, further assisting the development of dependable composite services. In a way similar to existing conversation languages, WS-RESC includes constructs for defining ordering and choices. Moreover, WS-RESC defines constructs for specifying concurrency since it is an inherent feature of distributed systems, and constructs for specifying exceptional behaviors, timing constraints and recovery properties of conversation since these are key behavioral properties in the context of dependability. The language in particular enables the definition of equivalence relationships over conversations with respect to their recovery behavior, which may be exploited for the design of fault-tolerant composite actions. WS-RESC is an XML-based language to be directly used by Web service developers to describe recovery-related properties of Web services. In addition, we provide a formal specification of the language through translation into the pi-calculus that makes available a large number of tools for reasoning about Web service properties. In particular, it allows the automated analysis of the correct composition of Web services with respect to the services' behavior.

6.2.2. Semantics-aware services for the mobile computing environment

Mobile distributed systems are characterized by a number of features, such as the highly dynamic character of the computing and networking environment due to the intense use of the wireless medium and the mobility of devices; the resource constraints of mobile devices; and the high heterogeneity of integrated technologies in terms of networks, devices and software infrastructures. To deal with high dynamics, mobile distributed systems tend to be dynamically composed according to the networking of mobile services. Nevertheless, such a composition shall be addressed in a way that enforces correctness of the composite systems with respect to both functional and non-functional properties and deals with the interoperability issue resulting from the high heterogeneity of integrated components. A number of paradigms may be employed to address these objectives, such as the Service-Oriented Architecture (SOA) and the Semantic Web.

SOA allows developing software as services delivered and consumed on demand. The benefit of this approach lies in the looser coupling of the software components making up an application, hence the increased ability to make systems evolve according to the dynamics of the environment. The SOA approach appears to be a convenient architectural style enabling dynamic integration of application components deployed on the diverse devices of today's wireless networks. The most popular existing software technology complying with the SOA architectural style is the Web Services Architecture. However, the SOA paradigm alone cannot meet the interoperability requirements for mobile distributed systems. Drawbacks include: (i) support of a single specific core middleware platform to ensure integration at the communication level; and (ii) interaction between services based on syntactic description, for which common understanding is hardly achievable in

an open environment. A promising approach towards addressing the interoperability issue relies on semantic modeling of information and functionality, that is, enriching them with machine-interpretable semantics. This concept originally emerged as the vehicle towards the Semantic Web. Semantic modeling is based on the use of ontologies and ontology languages that support formal description and reasoning on ontologies; the Ontology Web Language (OWL) is a recent proposition by W3C. A natural evolution to this has been the combination of the Semantic Web and Web Services into Semantic Web Services. This effort aims at the semantic specification of Web services towards automating Web services discovery, invocation, composition and execution monitoring. The most complete approach towards Semantic Web Services has been elaborated by the OWL-S community. OWL-S is an OWL-based ontology for semantically describing Web services' properties and capabilities. The Semantic Web and Semantic Web Services paradigms have emerged as a decisive factor towards interoperability, which up to then was being pursued based on agreements on common syntactic standards; such agreements cannot scale in the open, highly diverse mobile environment. Related efforts elaborating semantic approaches are addressing application-level interoperability in terms of information and functionality. However, interoperability requirements of mobile distributed systems are wider, concerning functional and non-functional interoperability that spans both middleware and application level.

As concluded from the above, the Service-Oriented Architecture with Web Services as its main representative, further semantically enhanced by Semantic Web principles into Semantic Web Services, only partially address the composability and interoperability requirements of mobile distributed systems. On the other hand, mobile services may be conveniently modeled using concepts from the software architecture field: architectural components abstract mobile services corresponding to application level and connectors abstract interaction protocols above the wireless network corresponding to middleware level. Based on these concepts, we have elaborated in our work on software architecture for mobile computing, base modeling of mobile software components, which integrates key features of the mobile environment and allows for reasoning on the correctness of dynamically composed systems with respect to both functional and non-functional properties [8]. Building on this work, we have introduced semantic modeling of mobile services so as to offer enhanced support to the interoperability requirements of mobile distributed systems. We have focused on the functional behavior of services; semantic modeling of the non-functional behavior of services is part of our future work. Specifically, we have introduced OWL-based ontologies to model functional properties of mobile components and associated wireless connectors. We have further elaborated conformance relations over component and connector models so as to be able to reason on the correctness of the composition of peer mobile services with respect to offered functional properties. Our conformance relations enable identifying partial conformance between components and between connectors, thus reasoning on interoperability. Based on these conformance relations, we have further specified appropriate interoperability methods to realize composition and interoperation of heterogeneous mobile services. More specifically, to compose two non-absolutely conforming connectors, we employ a connector customizer that serves as an intermediate for the message exchange between the two connectors. The customizer takes all appropriate action to remedy the incompatibilities between the two connectors. Further, our high-level conformance relation for components states that two components may be composed if they require and provide in a complementary way semantically conforming capabilities. To compose these two components, we intervene in the execution properties of the component requiring the specific capability. First, the component providing the specific capability is a normal component, the executable of which integrates the hard-coded implementation of the conversation associated to the capability. Regarding the component requiring the specific capability, its executable is built around this capability, which may be represented as a high-level local function call. This component integrates further an execution engine able to execute on the fly the specific conversation associated to this capability and supported by its peer component. Both conformance relations and interoperability methods are based on the reasoning capacity of OWL. In our modeling, we have adopted some existing results from the OWL-S community. Nevertheless, our approach is wider and treats in a comprehensive way the interoperability requirements of mobile distributed systems. As mentioned above, our modeling needs to be complemented with specification of the non-functional behavior of services and definition of related ontologies. We plan to do this building on our work on software architecture for mobile computing, which has identified key non-functional features of the mobile environment.

6.3. Middleware Architectures for Mobile Systems

Participants: Raghav Bhaskar, Yoram-David Bromberg, Rafik Chibout, Agnès de La Chapelle, Valérie Issarny, Jinshan Liu, Pierre-Guillaume Raverdy, Daniele Sacchetti, Françoise Sailhan.

In order to ease the development of mobile systems enabling the ambient intelligence vision, we are investigating supporting core middleware infrastructure and associated services. Our work in this area over year 2004 has concentrated on the following aspects: (i) development of a core middleware infrastructure based on the Web services architecture that enables the situation-sensitive composition of (possibly mobile) networked services (§ 6.3.1), (ii) middleware functionalities for the ad hoc, dynamic composition of mobile services (§ 6.3.2), (iii) middleware interoperability and more specifically service discovery protocol interoperability, investigating both decentralized (§ 6.3.3) and centralized (§ 6.3.4) solutions, and (iv) design of a scalable service location protocol for mobile ad hoc networks (§ 6.3.5).

6.3.1. Middleware infrastructure for ambient intelligence systems

Enabling the ambient intelligence vision means that consumers will be provided with universal and immediate access to available content and services, together with ways of effectively exploiting them. Concentrating on the software system development aspect, this means that the actual implementation of any ambient intelligence application requested by a user can only be resolved at runtime according to the user's specific situation. To support such a feature, we have introduced a base declarative language and associated core middleware, which respectively allow for the abstract specification of ambient intelligence applications, and for the dynamic composition of applications according to the environment. Our solution primarily builds on results of component-based software engineering and architecture-based development of software systems, which have been proven successful for the development of distributed software systems: ambient intelligence applications are developed through the composition of services that are defined by their abstract interfaces. The ambient intelligence requirement of enabling anytime, anywhere access to applications from any terminal further leads to binding with related services instances at runtime, according to the environment (including network connectivity) in which services are requested. Such a facility then requires a software technology that is pervasive enough for being able to rely on both consistent specification and availability of services in most environments, so as to actually support anytime, anywhere discovery of service instances from abstract interfaces. This has led us to base our solution on the Web, and more specifically on the Web services architecture. Our main design objective was then to offer a solution that could be deployed in any environment, and effectively supported by mobile, resource-constrained devices.

We have introduced the XML-based WSAMI declarative language for the specification of Web services taking part in the realization of ambient intelligence applications, together with associated core SOAP-based WSAMI middleware [2]. The language allows for dynamically retrieving instances of services matching a requested application by comparing only URIs of XML documents. Hence, the cost associated with the dynamic composition of applications, in terms of resource consumption (and in particular energy), is kept very low, which is mandatory for wireless devices. The language further allows for composing applications that guarantee security and performance properties, which we consider as foremost requirements for the actual acceptance of ambient intelligence applications by consumers. Actual composition of applications at runtime relies on the core middleware, which amounts to supporting SOAP and to providing a naming and discovery service for the dynamic retrieval of service instances, both in the local area over WLAN and in the wide area, according to the network connectivity and associated cost. We have implemented a first prototype of the WSAMI middleware (see § 5.2) so as to practically validate our solution with respect to offered development support and performance.

Supporting the development of ambient intelligence or pervasive computing systems has given rise to extensive research over the last couple of years, which has led to a number of complex middleware-related services that place high demand on the underlying platform and hence limit deployment in most environments. Our contribution lies in the definition of a minimal middleware infrastructure for the actual dynamic composition of services, i.e., a naming and discovery service in addition to SOAP, which allows for

its wide deployment but also incurs minimal overhead in terms of resource consumption and response time. Additional middleware-related services may be exploited for increased quality of service, depending on the specific target application, but they do not have to be supported in all environments. Our solution resembles work in the area of service discovery, given the base support offered by the platform. However, results in the area mainly target local area networks, while our solution targets composition of services that may be retrieved both in the local and the wide area. In addition, by building upon the Web services architecture, availability of services is promoted. We are currently working on complementing our core solution with support for off-line analysis of Web services so as to enforce robustness of the composed applications, regarding in particular behavioral matching with respect to both functional and non-functional properties. We are also investigating the exploitation of user and service profiles for the naming and discovery service, which will in particular enable an enhanced service selection process with respect to matching user and service requirements, and the integration of advanced prefetching techniques for enhancing response time.

6.3.2. Supporting ad hoc, dynamic composition of mobile services

The advent of light-weight terminals (e.g., PDAs) with integrated communication capabilities facilitates service accessing and hosting anytime, anywhere. Effective and universal service access requires deployment of services on possibly thin, wireless devices for increased availability. A mobile ad hoc network comprises a set of devices connected by wireless links in a temporary manner. The ease of deployment makes MANETs an attractive choice in a variety of applications, e.g., disaster recovery. However, service deployment and composition on such a distributed system face challenges including: (1) the limited computation power and communication capacity of thin devices, (2) the lack of infrastructure, and (3) the nodes' mobility and transient nature of the wireless connection. In this context, we are devising base middleware functionalities towards supporting the dynamic composition of mobile services over MANETs.

First, to realize a mobile distributed system that provides access to information and services spread among autonomous devices, cooperation among those devices is a necessity. Unfortunately, cooperative behavior implies resource consumption (e.g., battery), which is not in the interest of the autonomous devices. Thus how to stimulate service provision in ad hoc networks is vital to the operation of MANETs. Not only do we need to stimulate service provision in ad hoc networks, we also need to allocate services in an efficient manner, that is the allocation result should be a social, system-wide choice instead of an individual one. We utilize the Vickrey auction for service allocation in MANETs in an efficient, distributed and autonomy-preserving manner, thanks to its incentive compatibility and resource saving in computation and communication. Specifically, we have analyzed the challenges of applying the Vickrey auction in MANETs and proposed a viable service allocation model, as detailed in [12].

Second, interactions between entities unknown to each other are inevitable in MANETs that feature openness and mobility. Trust management through a reputation mechanism to facilitate such interactions is recognized as a vital part of mobile ad hoc networks. However, the design of a reputation mechanism is faced by challenges of how to enforce reputation information sharing and honest recommendation elicitation. We have designed a reputation model, which incorporates two essential dimensions, time and context, along with mechanisms supporting reputation formation, evolution and propagation [9]. By introducing the notion of recommendation reputation, our reputation mechanism shows effectiveness in distinguishing truth-telling and lying agents, obtaining true reputation of an agent, and ensuring reliability against attacks of defame and collusion.

Third, the network topology dynamics due to nodes' mobility is among the challenges raised by MANETs. Group management appears as a promising paradigm to ease the development of distributed applications over dynamic, mobile networks [5]. Different from existing approaches that focus on communication services and do not generally account for the network's highly dynamic topology, we have designed a middleware group service based on various attributes of group membership, such as geographical location, network scale, trustworthiness and QoS awareness. Our group service can divide into three parts: (1) group member discovery for discovering nodes that are interested in joining the group and satisfy local group constraints; (2)

group initialization for group formation, leader election and enforcing group-wide constraints; and (3) group dynamics management for handling the leaving and joining of nodes (including the group leader) [13].

6.3.3. Service discovery protocol interoperability in the mobile environment

In the mobile computing domain, communication relationships amongst application components involve the use of specific middleware protocols, making applications tightly coupled to middleware. Additionally, to overcome wireless networks constraints, e.g., limited bandwidth, poor network quality of service and either voluntary or forced frequent disconnection, several communication models have arisen. Thus, as there specifically 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. Service discovery protocols enable mobile nodes to find and use networked services without any previous knowledge of their specific location. Several Service Discovery Protocols (SDPs) are now available. And, with the advent of both mobility and wireless networking, SDPs are taking on a major role, and are the source of a major heterogeneity issue across middleware. Summarizing, interoperability among entities of a spontaneous ad hoc network, which is formed by the random arrival of mobile devices for short periods of time, is becoming a real issue to overcome. A portable computer must be aware of its dynamic environment that evolves over time, and further adapt its communication paradigms according to the environment. Thus, mobile distributed systems must provide efficient mechanisms to detect and interpret protocols currently used, which are not known in advance. We provide base mechanisms for achieving interoperability among heterogeneous SDPs, taking into account the above mobility requirements. The design of our solution is based on a software architecture enriched with event-based parsing techniques to drastically improve SDP interoperability, enabling mobile applications to be efficiently aware of their environment [6].

A Java implementation of our service discovery protocol interoperability architecture is being developed to validate our approach in terms of performance. We provide an interoperability layer that hides middleware heterogeneity to the applications. Thus, SDP interoperability is guaranteed to any existing applications, even those tied to a specific SDP.

6.3.4. Multi-protocol service discovery middleware

As part of our research on achieving middleware interoperability, we have further designed the Multi-Protocol Service Discovery (MSD) middleware to both facilitate interoperability among existing SD protocols in pervasive environments and enable context-aware service discovery. Within a homogeneous network (e.g., a WiFi hotspot or a home network), an MSD Manager is elected to centralize the service descriptions from the different SD protocols used within the homogeneous network. This manager interacts with basic service discovery protocols through specific SD plugins that collect service announcements, and extend them with context and semantic information. All services are described using a common format, the MSD service description. Extended service descriptions as well as context-aware discovery requests issued by client applications, are propagated to nearby networks using appropriate bridging devices selected by the manager. We argue that such federation of centralized managers is well suited to provide accurate, responsive, and stable service discovery while limiting processing and communication overhead. While converting protocol-specific service descriptions into a common format to enable protocol interoperability is not a new approach, we extend it with the following novel features:

- Service descriptions are not only converted into a common format, but are also extended with context and semantic information. This additional information is used to filter the available services to be presented to the user, a key issue in service-rich pervasive environments.
- Service descriptions are not only available to applications using different SD protocols within the same network, but are propagated to other networks by cooperating gateway devices.

We are implementing an instance of the MSD Middleware architecture using J2SE v1.4.2. A UPnP plugin has been developed and a Web Services plugin based on WSAMI is under development. These SD protocols enable us to experiment with both service announcements and discovery requests. We are also integrating the MANET-oriented service discovery protocol that is introduced below.

6.3.5. Scalable Service Discovery for MANET

The vision of pervasive computing and/or ambient intelligence requires enabling ubiquitous computing and networking so that mobile users can seamlessly get access to digital services anywhere, anytime. MANETs are one enabler of such a vision, providing networking capabilities to mobile devices without requiring any infrastructure. However, the specifics of MANETs such as potentially highly dynamic topology and networking of heterogeneous wireless nodes whose energy needs to be saved for enhanced autonomy, require special care in the handling of distributed service provisioning. In particular, the discovery of services must allow accessing services of the overall MANET for increased availability, while limiting resource consumption in the network. Existing discovery protocols including the most recent ones are either centralized, i.e., they are based on a central directory that stores information about services available in the network, or decentralized, i.e., clients and services discover each other directly, and rely typically on a multicast or broadcast protocol. Centralized resource discovery is much suited to wireless infrastructure-based networks. However, this makes the service discovery process dependent upon the availability of the central directory, which further constitutes a bottleneck and a single point of failure. Decentralized resource discovery protocols appear better suited to ad hoc networks. However, they rely either on a pull-based discovery (when a client wants to access a service, it broadcasts or multicasts a service query that is handled by available service providers) or on a push-based discovery (service providers broadcast or multicast service advertisements, which are possibly cached by nodes for later service invocation). Thus, the major drawback of the above decentralized solutions is that broadcasting floods the network. It may be concluded from the above that existing service discovery protocols for the wireless environment are mostly aimed at infrastructure-based network or at small-scale MANETs or also at 1-hop MANETs. Following the above survey, our design is based on a centralized discovery architecture, as it induces less traffic. Directories are distributed and deployed dynamically for the sake of scalability [14]. In order to deploy dynamically and homogeneously directories in the MANET, directories are elected on the fly by mobile nodes. The election process accounts for directories that are already deployed so as to keep to a minimum duplicate directories' coverage. Our discovery architecture is then structured as a virtual network composed of a subset of nodes of the MANET acting as directories. These directories represent a backbone of nodes responsible for handling service discovery. The scalability of our protocol comes from the minimization of the generated traffic, and the use of compact directory summaries that enable efficiently locating the directory that most likely caches the description of a given service.

Moreover, while service discovery protocols were initially introduced to deal with services provided by devices like printers or cameras, the increasing popularity of handheld devices has made possible a rich variety of mobile services as illustrated by scenarios in the area of pervasive computing, ambient intelligence or mobile commerce. This requires service-level interoperability among mobile devices, despite the heterogeneity of the supported software platforms. The pervasiveness of the Web allows assuming that a significant level of interoperability can already be achieved using the Web services architecture. We thus consider a service-oriented architecture based on Web services, with wireless nodes hosting Web services, as enabled by, e.g., the WSAMI open source middleware. We then build on the standardization effort of the Web services activities that has in particular introduced XML languages for describing Web services' interfaces. Specifically, we use WSDL for service description. We further enrich WSDL to allow the specification of Quality of Service (QoS) parameters, as QoS-awareness is a key requirement in the mobile environment [8]. Thus, service providers proactively provide QoS parameters together with the service's functional interface, to service requesters. Based on our work on QoS-aware service discovery [11], QoS specification includes both service-related and resource-related QoS parameters.

Another critical issue that is addressed by our service discovery protocol lies in enabling bridging MANETs with infrastructure-based networks since Intranet and/or Internet connectivity remains the primary source of

service provisioning. Such a feature allows devices that are located in the MANET to transparently access services offered in the interconnected networks (Internet or Intranet), and devices located outside the network to discover services provided in the interconnected MANETs. With our approach, services provided within the ad hoc network are visible from mobile terminals in the interconnected and external networks forming the hybrid network. This is especially useful in an enterprise network where some ad hoc networks can be created and interconnected to the Intranet network.

Finally, we have evaluated the performance of our protocol through simulation with the ns simulator, which shows the efficiency of our protocol in terms of generated traffic and delay. In addition, we have implemented our protocol as a middleware service of the WSAMI middleware.

7. Other Grants and Activities

7.1. European Initiatives

7.1.1. IST FP5 OZONE

Participants: Rafik Chibout, Nikolaos Georgantas, ValÃ©rie Issarny, Daniele Sacchetti.

- **Name:** IST OZONE – *New technologies and services for emerging nomadic societies*
- **URL:** <http://www.extra.research.philips.com/euprojects/ozone/>
- **Related activities:** §6.3.1, §5.2
- **Period:** [November 2001 - August 2004]
- **Partners:** Philips Research Eindhoven (The Netherlands) – project coordinator, Epictoid (The Netherlands), IMEC (Belgium), INRIA (URs Lorraine, Rennes, Rhône-Alpes, Rocquencourt), Technical University of Eindhoven (The Netherlands), THOMSON (France).

The objective of the OZONE project is to specify and implement a generic architecture/framework that will support the effective acceptance and use of ambient intelligence in the consumer domain. The OZONE project aims at the development of novel concepts, techniques and tools to provide invisible computing for the domestic and nomadic personal use of information technology. The developed concepts aim at improving the acceptability and usability for the average customer. One of the important concepts is the application of advanced technologies to support the user-centric retrieval and consumption of information compared to the current-practice, computer-centric approach. This requires special emphasis on natural interfaces that put the user in the foreground and the system in the background. Security and privacy are prerequisites for consumer acceptance of these systems and are covered by the project's software environment. A final objective deals with the provision of a strong technology base enabling powerful, but energy-efficient, computing.

7.1.2. IST FP6 STREP UBISEC

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

- **Name:** IST UBISEC – *Ubiquitous Networks with Secure Provision of Services, Access and Content Delivery*
- **URL:** <http://www.c-lab.de/ubisec/>
- **Related activities:** §6.3.4, §6.3.5, §5.3
- **Period:** [January 2004 - December 2005]
- **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), Universitat Politècnica 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 personalised 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.

7.1.3. IST FP6 IP Amigo

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

- **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, §6.3.3
- **Period:** [September 2004 - February 2008]
- **Partners:** Philips Research Eindhoven (The Netherlands) – project coordinator, Philips Design - Philips Consumer Electronics (the Netherlands), Fagor (Spain), France Telecom (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 people's lives is still dogged by complex installation procedures, the lack of interoperability between different manufacturer's 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 Nive 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.

7.3. Ministry Grant

7.3.1. 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

- V. Issarny is PC member of MDM'04: 5th IEEE International Conference on Mobile Data Management. January 2004, Berkeley, California, USA.
- V. Issarny is PC member of PIM'2004: International Workshop on Pervasive Information Management. In conjunction with EDBT'2004, March 2004, Heraklion, Crete, Greece.
- V. Issarny is PC member of MP'04: International Workshop on Middleware Performance. April 2004, Phoenix, Arizona, USA.
- V. Issarny is PC member of ICSE'04: 26th International Conference on Software Engineering. May 2004, Edinburgh, UK.
- V. Issarny is PC member of WADS'04: ICSE & DSN'04 Workshop on Architecting Dependable Systems. May 2004, Edinburgh, UK and July 2004, Florence, Italy.
- V. Issarny is PC member of FAST'2004: Workshop on Formal Aspects in Security & Trust. Co-located with 18th IFIP World Computer Congress (WCC'2004), August 2004, Toulouse, France.
- V. Issarny is PC member of SEM'2004: Workshop on Software Engineering and Middleware. Co-located with ASE'2004, September 2004, Linz, Austria.
- V. Issarny is PC member of Workshop on Software Architecture Description & UML. Co-located with UML'2004, October 2004, Lisbon, Portugal.
- V. Issarny is PC member of RISE'04: ERCIM International Workshop on Rapid Integration of Software Engineering Techniques. November 2004, Luxembourg.
- V. Issarny is PC member of IISW'04: International Infrastructure Survivability Workshop. In conjunction with RTSS'04, December 2004, Lisbon, Portugal.
- V. Issarny is PC member of MP2P'04 and '05: International Workshop on Mobile Peer-to-Peer Computing.
- V. Issarny is PC member of iTrust'04 and 05: International Conference on Trust Management.
- V. Issarny is PC member of EWSA'04 and '05: European Workshop on Software Architecture.
- V. Issarny is PC member of WICSA'04 and '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. March 2005, Santa Fe, USA.
- V. Issarny is PC member of ICDE'05: 21st International Conference on Data Engineering. April 2005, Tokyo, Japan.
- V. Issarny is PC member of ICDCS'2005: 25th International Conference on Distributed Computing Systems. June 2005, Columbus, Ohio, USA.
- V. Issarny is PC member of TSPUC'2005, First International Workshop on Trust, Security and Privacy for Ubiquitous Computing. Affiliated with IEEE WOWMOM'2005, June 2005, University of Texas at Arlington, USA.
- V. Issarny is PC member of HotDep'2005: First Workshop on Hot Topics in System Dependability. In conjunction with DSN'2005, June 2005, Yokohama, Japan.
- V. Issarny is Topic vice-chair of Euro-Par'2005 – Topic on Mobile and Ubiquitous Computing. September 2005, Lisboa, Portugal.
- V. Issarny is PC member of DSN-DCCS'2005: Dependable Computing and Communication Symposium at International Conference on Dependable Systems and Networks. June 2005, Yokohama, Japan.
- V. Issarny is PC member of Middleware'2005: ACM/IFIP/USENIX 6th International Middleware Conference. November 2005, Grenoble, France.

8.1.2. Other activities

- V. Issarny is chair of the executive committee of the AIR&D consortium on Ambient Intelligence Research and Development (<http://www.air-d.org>).
- V. Issarny is coordinating the OFTA (<http://www.ofta.net>) working group on Pervasive Computing.
- Y-D Bromberg is reporter of the OFTA working group on Pervasive Computing.
- V. Issarny is General Chair of iTrust'2005: 3rd International Conference on Trust Management. May 2005, INRIA, Rocquencourt, France.
- N. Georgantas is Demonstration Chair of iTrust'2005.
- J. Liu is Publicity Chair of iTrust'2005.

8.2. Teaching

- 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.
- N. Georgantas gives a course on Middleware Architectures (both lectures and laboratory), as part of the final year of the five-year computer engineering degree at the Ecole Supérieure d'Ingénierie Léonard de Vinci of the Université Universitaire Léonard de Vinci.
- F. Tartanoglu gives a course on "Web services" at the *Licence professionnelle ISDRN (Intégrateur de Systèmes Distribués et Réseaux Numériques)*, IUT Velizy, University of Versailles Saint-Quentin-en-Yvelines.

8.3. Internships

During year 2003, members of the ARLES project-team supervised the work of the following student interns:

- Sonia Ben Mokhtar, "*Synthèse ad hoc dans les systèmes de l'informatique diffuse.*", Graduate Student intern, *DEA Informatique Distribuée*, University of Paris-6.
- Agnès de La Chapelle, "*Caching et préchargement dans les architectures de services*", Graduate Student intern, *DEA MISI*, 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:

- F. Sailhan. "*Algorithmes de découverte de services et de contenus dans un réseau ad hoc*". Journées thématiques : Algorithmique distribuée et applications. September 2004, Porsquerolles, France.
- F. Sailhan. "Scalable service discovery for MANETs". Mimema Workshop. November 2004, Lancaster, UK.

9. Bibliography

Articles in referred journals and book chapters

- [1] T. HIGUERA, V. ISSARNY, M. BANÂTRE, G. CABILLIC, F. PARAIN, J. LESOT. *Memory Management for Real-time Java: an Efficient Solution using Hardware Support*, in "Real-Time Systems journal (Kluwer Pub.)", January 2004.
- [2] 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", January 2005.
- [3] F. SAILHAN, V. ISSARNY. *Handbook of Mobile Computing*, chap. Energy-aware Web Caching over Hybrid Networks, CRC Press, December 2004.

Publications in Conferences and Workshops

- [4] M. BOULKENAFED, V. ISSARNY, J. LIU. *Group Management for In-home Ad Hoc Networks*, in "Proc. of the ECRTS International Workshop on Real-Time for Multimedia - Special Focus on Real-time Middleware for Consumer Electronics (RTMM)", June 2003.
- [5] M. BOULKENAFED, D. SACCHETTI, V. ISSARNY. *Using Group Management to Tame Mobile Ad Hoc Networks*, in "Proc. of the IFIP TC8 Working Conference on Mobile Information Systems (MOBIS)", September 2004.
- [6] D. BROMBERG, V. ISSARNY. *Service Discovery Protocols Interoperability in the Mobile Environment*, in "Proc. of the International Workshop Software Engineering and Middleware (SEM)", September 2004.
- [7] N. GEORGANTAS, V. ISSARNY. *User Activity Synthesis in Ambient Intelligence Environments*, in "Adjunct Proc. of the 2nd European Symposium on Ambient Intelligence (EUSAI)", September 2004.
- [8] V. ISSARNY, F. TARTANOGLU, J. LIU, F. SAILHAN. *Software Architecture for Mobile Distributed Computing*, in "Proc. of the 4th IEEE/IFIP Conference on Software Architecture (WICSA)", June 2004.
- [9] J. LIU, V. ISSARNY. *Enhanced Reputation Mechanism for Mobile Ad Hoc Networks*, in "Proc. of the 2nd International Conference on Trust Management (iTrust'2004)", March 2004.
- [10] J. LIU, V. ISSARNY. *On Service Allocation in Selfish Mobile Ad Hoc Networks*, in "Proc. of EDBT International Workshop on Pervasive Information Management (PIM'2004)", March 2004.
- [11] J. LIU, V. ISSARNY. *QoS-aware Service Location in Mobile Ad Hoc Networks*, in "Proc. of the 5th IEEE International Conference on Mobile Data Management (MDM'04)", January 2004.
- [12] J. LIU, V. ISSARNY. *Service Allocation in Selfish Mobile Ad Hoc Networks using Vickrey Auction*, in "Current Trends in Database Technology - EDBT Workshops revised papers", LNCS 3268, Springer, 2004.

- [13] J. LIU, F. SAILHAN, D. SACCHETTI, V. ISSARNY. *Group Management for Mobile Ad hoc Networks: Design, Implementation and Experiments*, in "Proc. of the 6th IEEE International Conference on Mobile Data Management (MDM'05)", 2005.
- [14] F. SAILHAN, V. ISSARNY. *Scalable service discovery for MANETs*, in "Proc. of the 3rd IEEE International Conference on Pervasive Computing and Communications (PERCOM'2005)", March 2005.
- [15] F. TARTANOGLU, V. ISSARNY. *Specifying Web Services Recovery Support with Conversations*, in "Proc. of the 38th Hawaii International Conference on System Sciences (HICSS'2005)", January 2005.
- [16] A. ZARRAS, P. VASSILIADIS, V. ISSARNY. *Model-driven Dependability Analysis of Web Services*, in "Proc. of the International Conference on Distributed Objects and Applications (DOA)", October 2004.

Miscellaneous

- [17] *D10: Interim report on the integration process (Revised version)*, March 2004, OZONE Project Deliverable.
- [18] *D12a: Implementation of the Functionality Required to Support the Distributed Middleware Infrastructure, Parts 1, 2, 4*, May 2004, OZONE Project Deliverable.
- [19] *D12: Report on the Implementation of the Ozone Software Environment Layer*, May 2004, OZONE Project Deliverable.
- [20] *D14: Report on integration process and assessment of technical achievements*, June 2004, OZONE Project Deliverable.
- [21] *D15d: User experience report of the three demonstrators and the external application of the OZONE framework*, September 2004, OZONE Project Deliverable.
- [22] *D15: Report on the OZONE demonstrators and their assessment*, September 2004, OZONE Project Deliverable.
- [23] *D1.1: State of the art and requirements analysis for service discovery in heterogeneous networks*, June 2004, UBISEC Project Deliverable.
- [24] *D1.2: Specification and proof of concept implementation of a base discovery service for heterogeneous networks*, December 2004, UBISEC Project Deliverable.
- [25] *D4.1: Specification of UBISEC applications and scenarios*, December 2004, UBISEC Project Deliverable.