INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team ARLES

Software Architectures and Distributed Systems

Paris - Rocquencourt

THEME COM

Activity Report

2007

Table of contents

1. Team

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

Administrative assistant
Emmanuelle Grousset

Research scientist
Nikolaos Georgantas [INRIA Junior Research Scientist]
Pascal Poizat [Assistant Professor, on leave from University of Evry, France]

Visiting scientist
Marco Autili [PhD Student, University of L'Aquila, Italy, February 2007 – September 2007]
Carlos Flores Cortes [PhD Student, University of Lancaster, United Kingdom, June 2007]
Paola Inverardi [Professor, University of L'Aquila, Italy, December 2007]
Natalia Kokash [PhD Student, University of Trento, Italy, August 2007 – January 2008]
Jorge Parra [PhD Student, Ikerkan, Spain, June 2007 – July 2007]
Aitor Urbieta [PhD Student, University of Mondragón, Spain, March 2007 – July 2007]

Technical staff
Sébastien Bianco [*Ingénieur Associé*, INRIA]
Hassine Moungla [*Ingénieur Expert*, INRIA PLASTIC, since September 2007]
Pierre-Guillaume Raverdy [*Ingénieur Expert*, INRIA PLASTIC]
Letian Rong [*Ingénieur Expert*, INRIA PLASTIC, until November 2007]
Graham Thomson [*Ingénieur Expert*, INRIA Amigo]

Post-doctoral fellows
Mauro Caporuscio [INRIA]
Fabio Mancinelli [INRIA, since December 2007]
Thomas Zahn [INRIA, until August 2007]

PhD students
Sandrine Beauche [DGA fellowship, University Pierre et Marie Curie - Paris 6, since October 2007]
Nebil Ben Mabrouk [INRIA CORDI fellowship, University Pierre and Marie Curie - Paris 6, since October 2007]
Sonia Ben Mokhtar [INRIA fellowship, University Pierre et Marie Curie - Paris 6, until November 2007]
Damien Fournier [INRIA fellowship, University of Versailles Saint-Quentin, until February 2007]
Manel Fredj [INRIA fellowship, University Pierre et Marie Curie - Paris 6]
Roberto Speicys Cardoso [INRIA fellowship, University Pierre et Marie Curie - Paris 6]

Student interns
Sandrine Beauche [MSc SAR, University Pierre et Marie Curie - Paris 6, February 2007 – September 2007]
Nardjesse Miloudi [Institut National d'Informatique (INI), Algiers, Algeria, since November 2007]

2. Overall Objectives

2.1. Introduction

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

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

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

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

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

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

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

bient intelligence/pervasive computing vision. Similarly, multi-radio networking is a major enabler of this vision by potentially improving network connectivity towards ubiquitous networking. In general, numerous challenges remain for middleware architectures to enable the ambient intelligence vision. It is one of the objectives of ARLES to investigate thorough solutions to those issues.

2.2. Highlights of the year

During this year, we have provided complete solutions, including the development of models, algorithms and implementation/middleware for:

- middleware-layer networking in multi-radio multi-network environments (see § 6.2);
- semantic middleware for pervasive services (see § 6.3).

3. Scientific Foundations

3.1. Introduction

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

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

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

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

3.2. Software Architectures for Distributed Systems

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

3.2.1. Middleware-based and service-oriented software architectures

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

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

Although Web services have been primarily designed for realizing complex business processes over the Internet, they are a promising architectural choice for pervasive computing. The pervasiveness of the Web allows anticipating the availability of Web services in most environments, considering further that they may be hosted on mobile devices. Hence, this serves as a sound basis towards dealing with the dynamic composition of services in the pervasive computing environment. However, this further requires specification

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

of the Web services' functional and non-functional behavior that can be exploited for their dynamic selection and integration, which may in particular build upon work on the Semantic Web.

3.2.2. Architecture-based development of distributed systems

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

However, most of the work on the specification of connectors has focused on the characterization of the interaction protocols among components, whilst connectors abstracting middleware embed additional complex functionalities (*e.g.*, support for provisioning fault tolerance, security, transactions). The above concern has led the software architecture community to examine the specification of the non-functional properties offered by connectors. For instance, these may be specified in terms of logic formulae, which further enables synthesizing middleware customized to the application requirements, as supported by the Aster ADL. Another issue that arises when integrating existing components, as provided by middleware platforms, results from assembling components that rely on distinct interaction patterns. This aspect is known as *architectural mismatch* and is one of the criteria substantiating the need for connectors as first-class entities in architecture descriptions. The abstract specification of connector behavior, as, for instance, supported by the Wright ADL, enables reasoning about the correctness of component and connector composition with respect to the interaction protocols that are used. However, from a more pragmatic standpoint, software development is greatly eased when supported by mechanisms for solving architectural mismatches, which further promotes software reuse.

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

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

3.3. Middleware Architectures for Mobile Systems

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

3.3.1. Ad hoc networking

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

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

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

3.3.2. Managing the network's dynamics

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

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

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

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

- **Context and semantic information**: In heterogeneous networks, the simple information used by existing service discovery protocols to define a service is not sufficient. Additional information needs to be collected about the networks' identity and characteristics (*e.g.*, bandwidth, cost, reliability), users, and devices. Semantic information is also necessary since service-rich environments may offer many similar services. Context and semantic information needs to be propagated along with service descriptions so that potential clients can evaluate available services and select the most appropriate one(s).
- **Protocol interoperability**: Many service discovery protocols have been proposed for different environments (Internet, home networks) and several have emerged as the leading protocol in their target environment. A service discovery solution for heterogeneous networks needs to support or interoperate with these service discovery protocols. While discovery information can easily be collected from any protocol, converting service information between different protocols, or injecting information on services from a remote network may not be possible.
- **Network bridging**: The service discovery protocol for heterogeneous networks needs to learn about the different networks available at a location, and about the characteristics of the devices that can act as bridges to access other networks. Similar devices may both be technically able to bridge Wi-Fi and Bluetooth but may provide different QoS due to battery power, user mobility, cost, or installed software packages. As the heterogeneous network topology changes, links to some remote networks may become unavailable, and latency may change drastically as a new route will be used.
- **Information propagation**: The service discovery protocol for heterogeneous networks needs to filter the information that is propagated between networks, as the information usually collected by discovery protocols may not be completely relevant for remote hosts/networks. For example, services on remote networks may not be accessible (*e.g.*, security issues or minimum bandwidth unavailable). The discovery protocol should also evaluate how far discovery information should be propagated, and how it should be cached and managed at the bridges.
- **Remote service access**: Service discovery protocols collect information about available services, and provide this information back to requesting clients. Part of the information is the service providers location (*e.g.*, IP address of host and port number). It is usually assumed that clients can directly contact the service provider and request a service (that may be granted or not). In the case of heterogeneous networks however, it may not be possible to access the service provider due, for example, to IP network accessibility issues. The service bridges, which propagated the service information, may potentially be used to also propagate service request. Further, client and service providers must use semantically and syntactically matching protocols for service access, which cannot be guaranteed in highly open, pervasive networking environments.

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

3.3.3. Enforcing dependability

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

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

4. Application Domains

4.1. Application Domains

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

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

Our application domain is voluntarily broad since we aim at offering generic solutions. However, we examine exploitation of our results for specific applications, as part of the experiments that we undertake to validate our research results through prototype implementation. Applications that we consider in particular include demonstrators developed in the context of the European and Industrial projects to which we contribute (§ 7.1 & 8.1).

As illustration of applications investigated within ARLES, the COCOA semantic service middleware together with the INMIDIO interoperable middleware (respectively, § 5.6 & 5.5) support the *networked home* environment. The networked home specifically seeks to combine the home automation, consumer electronics, mobile communications and personal computing domains to provide new user applications that exploit the fluid integration of these traditionally strictly separated domains, and to lay a solid foundation towards realizing the ambient intelligence vision. The applicability of the COCOA semantic service middleware to the networked home environment has in particular been illustrated at the Amigo 2nd Annual Review Meeting and Open Day, at the Philips Research High Tech Campus, Eindhoven, November 2006 (§ 8.1.1). A demonstrator application was presented, which enabled users' personal wireless devices to discover, negotiate permission to use, and control home consumer electronic and personal computing devices.

5. Software

5.1. Introduction

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

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

Participant: Valérie Issarny [correspondant].

WSAMI (Web Services for AMbient Intelligence) is based on the Web services architecture and allows for the deployment of services on wireless handheld devices like smartphones and PDAs [6]. WSAMI further supports the dynamic composition of distributed services over hybrid wireless networks. Moreover, WSAMI takes in charge the customization of the network's path through the dynamic integration of middleware-related services, in order to enforce quality of service with respect to offered dependability and performance properties.

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

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

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

Participant: Valérie Issarny [correspondant].

The Ariadne service discovery protocol for has been designed to support decentralized Web service discovery in multi-hop mobile ad hoc networks (MANETs) [9]. Ariadne enables small and resource-constrained mobile devices to seek and find complementary, possibly mobile, Web services needed to complete specified tasks, while minimizing the traffic generated and tolerating intermittent connectivity. The Ariadne protocol further enables service requesters to differentiate services instances according to non-functional properties. Specifically, the Ariadne protocol is based on the homogeneous and dynamic deployment of cooperating directories within the MANET. Scalability is achieved by limiting the generated traffic related to service discovery, and by using compact directory summaries (i.e., Bloom filters) to efficiently locate the directory that most likely has the description of a given service.

[2] Compact SOAP

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

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

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

Participant: Valérie Issarny [correspondant].

The MUlti-protocol Service Discovery and ACcess (MUSDAC) middleware platform enables the discovery and access to services in the pervasive environment, which is viewed as a loose and dynamic composition of independent networks [7]. MUSDAC manages the efficient dissemination of discovery requests between the different networks and relies on specific plug-ins to interact with the various middleware used by the networked services.

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

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

5.5. INMIDIO: An Interoperable Middleware for Ambient Intelligence

Participant: Nikolaos Georgantas [correspondant].

In the pervasive computing environment, devices from various application domains, *e.g.*, home automation, consumer electronics, mobile and personal computing domains, need to dynamically interoperate irrespectively of the heterogeneity of their underlying hardware and software. Middleware has been introduced in order to overcome this issue by specifying a reference interaction protocol enabling compliant software systems to interoperate. However the emergence of different middleware platforms to address the requirements of specific application domains leads to a new heterogeneity issue among interaction protocols. Thus, at a given time and/or at a specific place, devices hosting the wrong middleware become isolated. In order to overcome this issue, we have developed a system called INMIDIO (INteroperable MIddleware for service Discovery and service InteractiOn) that dynamically resolves middleware mismatch. More particularly, INMIDIO identifies the interaction middleware and also the discovery protocols that execute on the network and translates the incoming/outgoing messages of one protocol into messages of another, target protocol [3]. Specifically, the system parses the incoming/outgoing message and, after having interpreted the semantics of the message, it generates a list of semantic events and uses this list to reconstruct a message for the target protocol, matching the semantics of the original message. The INMIDIO middleware acts in a transparent way with regard to discovery and interaction middleware protocols and with regard to the services running on top of them.

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

5.6. COCOA: A Semantic Service Middleware

Participant: Nikolaos Georgantas [correspondant].

COCOA is a comprehensive approach to semantic service description, discovery, composition, adaptation and execution, which enables the integration of heterogeneous services of the pervasive environment into complex user tasks based on their abstract specification [1], [2]. Using COCOA, abstract user tasks are realized by dynamically composing the capabilities of services that are currently available in the environment. The capabilities that a service provides are presented as a *conversation* – a workflow that specifies data and control dependencies for its capabilities. Similarly, an abstract task is presented as an orchestration of required capabilities. The conversations of the provided service capabilities are integrated to realize the orchestration of the abstract task, while guaranteeing that the dependencies of each of the provided capabilities are preserved. This allows complex user tasks to be created and reliably composed, while offering fine-grained control over the placement of capabilities in the task. This is especially important for the pervasive environment, where user tasks may frequently involve interaction with the user(s). In addition, the service composition can be optimized based on quality of service and context-aware parameters. To accommodate the inherent heterogeneity of services in the pervasive environment, capabilities are described and matched semantically, and adapted when necessary. Furthermore, different service groundings are supported, allowing diverse SOA platforms to be incorporated; interoperability at service grounding, *i.e.*, middleware level may then be ensured by employing INMIDIO (§ 5.5). Once a new service realizing the user task has been created, it is automatically deployed and executed.

The first integrated COCOA prototype was successfully demonstrated at the Amigo 2nd Annual Review Meeting and Open Day, at the Philips Research High Tech Campus, Eindhoven, in November 2006. A prototype version has been released under LGPL as open source software in october 2007 on the Amigo GForge Open Source Software site http://gforge.inria.fr/projects/amigo/.

The COCOA Semantic Service Middleware is a featured component of the Amigo Challenge: http://challenge.amigo-project.org/. The Amigo Challenge is a programming competition to develop novel applications using the Amigo middleware. The Challenge has been promoted through many academic and industrial organisations worldwide, as well as several popular online technical news sites. COCOA is proving to be one the most popular downloads in the Challenge.

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

Participant: Valérie Issarny [correspondant].

The objective of MR_SDP (Multi-Radio Service Discovery Protocol) is to support context awareness based on effective service discovery in multi-radio networks. MR_SDP supports an adaptive service discovery protocol scheme optimized for multi-radio networks [4]. This allows the client device to discover networked services over the embedded network interfaces, by using the relevant functionalities of the multi-radio wireless device while interfacing with the various Service Discovery Protocols (SDPs) that are available at the specific location and time. Further and as importantly, this is realized in a way that both minimizes resource-consumption on the device and offers a response time comparable to related SDPs.

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

6. New Results

6.1. Introduction

The ARLES project-team investigates solutions in the forms of languages, methods, tools and supporting middleware, to assist the development of distributed software 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. We in particular study ambient intelligence system architectures based on the service paradigm, from service modeling to service-oriented middleware for advanced wireless networks. Our research activities over the year 2007 have focused on the following complementary areas:

- Middleware layer B3G networking so as to allow effectively exploiting the rich multi-radio, multi-network environments that are now accessible from most wireless devices (§ 6.2).
- Semantic service-oriented middleware so as to enable truly pervasive services, which may be discovered, accessed and composed anytime, anywhere (§ 6.3).
- Automatic service composition so as to enable distributed software systems to be dynamically composed out of networked resources (§ 6.4).
- Privacy awareness in the design of distributed systems for ambient intelligence, considering more specifically the enhancement of middleware functions from the standpoint of privacy (§ 6.5).

6.2. Middleware-layer B3G Networking

Participants: Marco Autili, Mauro Caporuscio, Manel Frej, Nikolaos Georgantas, Paola Inverardi, Valérie Issarny, Hassine Moungla, Pierre-Guillaume Raverdy, Letian Rong.

Beyond 3rd Generation (B3G) networks combine multiple wireless networking technologies in order to benefit from their respective advantages and specificity. Further, the increase in computing and communication capacities of portable devices, as well as their mass marketing, make possible the widespread deployment of such multi-networks pervasive environments, where B3G-capable devices hold several radio interfaces, and the possibility to switch from one radio interface to another. B3G-enabled users shall then benefit from such pervasive networks by increasing the perimeter of reachable service providers. However, this should not be realized at the expense of a greater complexity. In fact, dealing with B3G networks opens new challenges and issues in the development and deployment of distributed systems. In this setting, a B3G-oriented middleware shall cope with the complexity induced by the heterogeneity of the wireless technologies, by hiding it to the user and, to be effective, to the application developer as well. In light of the above discussion, the B3G-oriented middleware shall be able to: (i) capture the various networks and observe their status (e.g., connectivity and quality of service), and (ii) abstract their properties (e.g., accessibility and offered capabilities), in order to let users fully exploit the underlying network diversity without relying on any pre- established knowledge or infrastructure.

In order to address the above issues, we have been developing a B3G-oriented communication middleware which implements two inter-working layers: *Multi-radio networking* and underlying *Multi-radio device management*. Specifically, the Multi-radio networking layer is in charge of:

1. Selecting the best available network (with respect to both application requirements and networks characteristics) [25].
2. Managing message exchange between devices by means of device addressing and packet delivering. Specifically, this layer defines the addressing scheme that uniquely identifies a device (and consequently the services hosted by it, as we consider that a software service instance remains deployed on the very same host over its lifetime) as an enhanced set of IP addresses, and provides a one-hop communication. In particular, based on the routing scheme of the current upper-layer protocol, it is in charge of delivering given packets to the next hop by properly choosing one of the available networks.

The Multi-radio device management layer manages the low-level characteristics of the perceived networks in terms of functionality and QoS properties. That is, it is in charge of:

1. Sensing the available networks and retrieving their characteristics (attributes and offered capabilities).
2. Monitoring the networks' status.
3. Controlling the network interfaces to exploit the respective offered capabilities.

In order to enable service oriented computing over multi-radio networks, the above-mentioned B3G networks, we have further worked on enabling SOAP-based interaction in these networks. Providing B3G SOAP communication relates to comprehensively exploiting the rich, heterogeneous B3G networking environment for the handling of SOAP messaging. In particular, B3G SOAP enables:

- Efficient SOAP interactions in multi-paths configurations (i.e., when multiple network paths exist between the client and the service). SOAP messages are exchanged over the most effective network link among those currently eligible, considering in particular energy efficiency and enforcement of QoS;
- Seamless mobility so that active sessions with services are maintained transparently to the application layer despite the mobility of nodes, as long as a network path exists;
- Multi-network routing so that access to services in distant networks is enabled as long as there exists a path bridging the heterogeneous networks between the client and target service;
- Point-to-point and group communications using the same abstractions (e.g., URL, service endpoint) are supported.

To validate the above, we have implemented a prototype of the B3G SOAP protocol which decomposes into:

- A Bridge component in charge of routing B3G SOAP messages between independent networks. This component is deployed on multi-radio devices and interacts with the PLASTIC multi-radio networking layer to get connectivity information, send and receive packets, and monitor PLASTIC addresses;
- Implementations of SOAP transports (much like HTTP or SMTP transports) for both point-to-point and group B3G communication. These transports are in charge of resolving PLASTIC addresses used as end point references (EPRs) for services, and encapsulating applications' SOAP messages into a B3GSOAP message in case of multi-network routing;
- A B3GSOAP handler activated on clients, services, and bridges which is in charge of updating and processing the multi-network routing information and the context information inserted in the SOAP message header based on local information.

As explained above, B3GSOAP strives to provide a solution for multi-network communication that can be deployed and used as easily and transparently as possible for application developers (both client and service applications), is compatible with existing standards, and is implemented on top of major SOAP implementations. Indeed, overlay networks usually provide a specific API for communications, which requires all clients and services to use this specific API and thus limits the set of services that can be accessed. Such proprietary approach also prevents the use of other tools based on standard communication API (e.g., QoS monitors). B3GSOAP enables SOAP communication in B3G environments, is compatible with existing SOAP standards and implementations, and can be easily deployed and used, even for existing service implementations. A further innovation of the proposed B3G service oriented communication is the support for SOAP-based group (multicast) communication.

A complementary issue we have focused on in B3G networks is dealing with mobility. In service-oriented environments, B3G networks enable mobile users to roam freely between heterogeneous networks on an all-IP platform, consuming dynamically the networked services. Specifically, we view B3G networks as an infrastructure-less aggregation of heterogeneous infrastructure-based and ad hoc networks. However, in such dynamic networks, connectivity with the networked services is not guaranteed due to users' and service hosts' mobility. Tolerating connectivity loss at runtime in SOA-based B3G environments is mandatory to take fully advantage of their openness. In our recent researches, we focus on handling mobility in B3G networks and ensuring service continuity despite connectivity loss. In [34], we propose a hybrid approach that uses either vertical handoff solutions to switch from a radio network interface to another, maintaining thereby connectivity with the same service instance, or service reconfiguration solutions to substitute a disconnected service with a semantically equivalent one, while maintaining the consistency at both the client and the service sides. The hybrid solution is enabled by acting on two layers of the middleware: the lowest or network layer and the highest layer. At (lowest) network layer, the mobility can be handled seamlessly using network handoffs, which relies upon additional network infrastructure and radio interface switching. Alternatively to handoff, the mobility awareness can be achieved – at higher layer of the middleware – through service reconfiguration. Service reconfiguration can be preferable in cases where shifting to another network (e.g, from a WIFI spot to a 3G network) is not possible or may introduce a loss of quality or high billing cost. More specifically, we are investigating the adaptation of fault tolerance means to the specifics of SOA-based B3G systems.

6.3. Semantic Middleware for Service-Oriented Pervasive Computing

Participants: Sébastien Bianco, Sonia Ben Mokhtar, Carlos Flores Cortes, Nikolaos Georgantas, Valérie Issarny, Graham Thomson, Aitor Urbieta.

Realizing the vision of pervasive computing requires dealing with a number of issues, mainly due to the environment's heterogeneity, dynamics and user-centrism. Among the various investigated middleware paradigms, service-oriented middleware (SOM) appears to be most appropriate to tackle the requirements of pervasive environments by abstracting the software and hardware resources of such environments as services and by supporting service discovery, access and composition. Nevertheless, the affluence of SOM technologies and platforms that have been put forward to address the heterogeneity and dynamics of pervasive environments has engendered a new kind of heterogeneity, i.e., middleware heterogeneity. This heterogeneity concerns the protocols associated to base middleware functionalities, which are service discovery and service access, as well as the multitude of networks in which service providers and requesters may reside. Thus, a SOM for pervasive computing should provide multi-protocol and multi-network interoperability mechanisms (see the previous section). Still, even after interoperability has been established at the networking and middleware levels, the dynamic discovery and composition of networked services by applications further require service providers and requesters to agree on the semantics of services, so that these can integrate and interact in a way that guarantees dependable service provisioning and consumption. Such an agreement cannot be carried out at the syntactic level, i.e., assuming that service providers and requesters use a common syntax for denoting service semantics, as this is a too strong assumption for pervasive environments. Then, a promising approach towards addressing syntactic heterogeneity relies on semantic modeling of service features by employing relevant semantic technologies. Nevertheless, assessing the conformance between service semantics as announced by service providers and requested by service requesters induces costly semantic reasoning (in terms of time and computation), which makes existing solutions inappropriate for the highly interactive and resource constrained pervasive environment. Finally, besides dealing with the functional features of services, user-centrism of pervasive environments calls for the awareness of service non-functional features, i.e., Quality of Service (QoS).

To address the above challenges, we have introduced an efficient, semantic, QoS-aware service-oriented middleware for pervasive computing. After a survey of existing semantic service description languages, we have identified the requirement for a new semantic service model to support interoperability between these languages, which is at the heart of interoperability enabled by our middleware [13]. The proposed model supports the specification of both semantic and syntactic service descriptions. For semantic-based service descriptions, our model further enhances the specification of semantic annotations, where an additional source of heterogeneity has been identified. This enables service providers and requesters to provide more accurate semantic specifications, which allows our middleware to perform more accurate semantic service matching. Furthermore, our model supports formal specification of service conversations as finite state automata.

This enables automated reasoning about service behavior independently from the underlying conversation specification language. Hence, pervasive service conversations described with different service conversation languages can be integrated towards the realization of a user task. Finally, our model supports the specification of service non-functional properties based on existing QoS models to meet the specific requirements of each pervasive application.

Building on our semantic service model, we have further proposed an efficient semantic service registry for pervasive computing environments [18]. The proposed registry supports a set of conformance relations for matching both syntactic and rich semantic service descriptions, as well as their heterogeneous non-functional properties. As finding a service that exactly matches a client request is rather the exception than the rule in pervasive environments, our registry identifies the semantic distance between semantic service descriptions, and rates services with respect to their suitability for a specific client request, so that selection can be made among them. The evaluated semantic distance takes into account both functional and non-functional characteristics of services. Additionally, our registry supports efficient reasoning on semantic service descriptions, which makes it applicable to highly interactive pervasive environments. Service descriptions in our registry are semantically organized to enable both efficient service publication and location. Thanks to the proposed optimizations, our registry performs better than existing semantic service registries that opt for overloading the service publication phase to achieve efficiency at service location.

Another functionality supported by our semantic middleware is service composition, as discussed in the next section. We have further developed a prototype implementation of our semantic service-oriented middleware complemented with the multi-network and multi-protocol interoperability methods coming from the MUSDAC middleware (see § 5.6 & 5.4) [24]. The overall prototype, which constitutes an innovative, efficient and comprehensive solution towards the realization of the pervasive computing vision, is evaluated in terms of the execution overhead of each of its constituent middleware functionalities presented above. Furthermore, our semantic service-oriented middleware has been integrated as a key element in the ambient intelligence framework developed by the IST Amigo project (see § 8.1.1) [50], [51].

6.4. Automatic Service Composition

Participants: Sandrine Beauche, Nebil Ben Mabrouk, Fabio Mancinelli, Sonia Ben Mokhtar, Nikolaos Georgantas, Valérie Issarny, Pascal Poizat.

Automatic semantic service composition is a key functionality supported by our semantic middleware [17]. This functionality supports flexible QoS-aware service composition towards the realization of user-centric tasks abstractly described on the user's handheld. Flexibility is enabled by a set of composition algorithms that may be run alternatively according to the current resource constraints of the user's device. These algorithms further support the fulfillment of the QoS requirements of user tasks by aggregating the QoS provided by the composed networked services. Unlike existing research efforts on service composition that assume complex behavior for either services or tasks but not for both, our proposed composition algorithms support the integration of services that have a complex behavior to realize a user task also specified with a complex behavior. This allows taking full advantage of the diverse pervasive functionalities in the vicinity of a user at the specific time and place. Furthermore, our service composition is performed efficiently as it relies on our efficient semantic service registry to discover services and on efficient formal verification algorithms to build the user task realizations.

Incompatibilities between the behavioral interfaces (protocols) of interacting components or services may prevent their composition or lead to incorrect composition behaviors (e.g., deadlocks or QoS loss). The objective of software adaptation is to compensate such incompatibilities building as automatically as possible corrective connectors or components called adaptors. In previous work we have developed a model-based approach to deadlock-freedom component adaptation and a dedicated tool for the synthesis of adaptor protocols from user-defined adaptation contracts, Adaptor. This proposal grounds on a close system hypothesis and is not suited to contexts where the architecture evolves, e.g., service-oriented pervasive computing where services may enter or leave the system at any moment. Moreover, the complexity of computing centralized adaptors (i.e., global ones, as opposed to distributed, local adaptors) is an issue. This has been tackled in two different and complementary ways:

- in [32], [33], we support the adaptation of open-systems and an incremental adaptation process where adaptors are distributed over the architecture and where only parts of the adaptors have to be recomputed when this architecture evolves. This avoids the global adaptor computation, yet cannot ensure the composition full-correctness unless the final system is closed;
- in [31], we keep the centralized (and correct by construction) adaptation approach. Yet we have defined a first on-the-fly adaptation technique through the encoding of the adaptation constraints in the LOTOS process algebra and the use of on-the-fly model-checking and behavioral reduction techniques.

In both cases, we have developed prototype tools to support the adaptor computation. Independently, the size of composition and adaptation protocols may prevent their implementation. Therefore, we have defined in [29] a behavioral reduction technique supporting the composite structure of compositions, and introducing a separation between the synchronizability and observability concepts, which are usually mixed in reduction techniques, yielding compositors or adaptors that, if smaller, cannot be correctly implemented. The enhancement of the reduction complexity into an on-the-fly reduction technique that could be applied while computing compositors or adaptors is a perspective. Adaptor implementation issues have been addressed in [27] with a first approach for the adaptation of WF/.NET components.

All the above-mentioned works apply to component-based architectures. More recently, we have addressed the inclusion of adaptation features in composition processes for service architectures. We have in particular investigated a task-oriented service composition process based on graph planning techniques (graphplan) with two specific features:

- vertical adaptation: vertical mismatch is supported using task decomposition guidelines;
- horizontal adaptation: horizontal mismatch is supported using ontologies.

This technique generates service compositions (orchestrations) under the form of plans which are then translated into the WS-BPEL service orchestration language.

On-going research includes the study of the underlying planning structure used for semantic service composition to support service disconnection. As this structure supports a concise representation of several service composition plans from a end-user task, it may be interesting to compute it off-line to define possible compositions and then to use it on-line to perform service(s) replacement when a service to be used later on in the run-time composition gets unavailable. This later perspective takes place in a wider study on the development of safe systems of systems based on the composition and adaptation of services.

6.5. Privacy Aware Solutions

Participants: Valérie Issarny, Natalia Kokash, Roberto Speicys Cardoso, Pierre-Guillaume Raverdy.

Privacy is a major concern in today's information society. As computers are used to mediate a growing number of human activities, a greater amount of personal data is digitalized and can be easily transmitted, stored and analyzed. If, in the past, private information such as shopping habits, activities, preferences and whereabouts vanished with time, today this data can be either directly collected or deduced from other available data, and correlated to infer a number of individual's personal details without one's awareness. Privacy protection becomes even more significant in pervasive computing systems, where the surrounding space incorporates computing devices that the user manipulates unconscious of their existence to execute daily activities. Pervasive information systems, thus, can collect a larger number of personal data and amplify the possibilities for privacy invasions.

To effectively protect user's privacy, the infrastructure, middleware and application layers must cooperate, supported by a privacy legislation that enables legal measures against identified privacy invasions. In this work, we focus on the pervasive middleware, which must not only provide mechanisms to encourage the development of privacy-aware pervasive applications but also must be designed taking privacy into account. Particularly, we consider privacy issues that arise on service-oriented middleware both when applications interact with middleware services as well as on the middleware design and implementation.

In [37], we investigate issues that appear when performing multi-network service discovery. Pervasive computing environments are usually composed of heterogeneous networks in terms of technology (WiFi, Bluetooth, Ethernet, GPRS, etc.), discovery protocols (SDP, UPnP, WS-Discovery, etc.), organization (ad hoc, structured) and administration (security policies, access control, etc.). Users must be able to discover and compose services regardless of network heterogeneity, which is typically obtained by forwarding service requests to adjacent networks - as opposed to forwarding service advertisements, which is inefficient on pervasive computing. We proposed a series of mechanisms to (1) allow clients to control *where* service requests are forwarded, (2) allow clients to control *how* service requests are forwarded and (3) enhance middleware privacy-awareness when handling multi-network service discovery.

The focus of [38] is on how to support highly distributed syntactic and semantic service discovery without risking the privacy of clients. Service discovery directories are critical for the operation of a service discovery protocol, and many approaches propose directory distribution as a means to improve reliability. However, existing protocols consider the service directory as a trusted component of the architecture which creates additional barriers for directory distribution. Moreover, the use of semantics to enrich service descriptions increase the expressiveness of service requests and advertisements but also allows attackers to infer further client private information from service discovery data. We proposed a privacy-enhanced service discovery protocol that supports both syntactic and semantic service descriptions and reduces trust requirements on service directories, simplifying directory distribution. Service descriptions are modified in such a way that a privacy-enhanced description can represent multiple original service descriptions. Service directories are unable to relate a given privacy-enhanced service request or advertisement to the original request or advertisement, but are still able to find matches between privacy-enhanced requests and advertisements. A privacy-enhanced match, however, can contain false positives that the client must filter. As a result, service directories cannot recognize which services are announced or requested, which reduces the privacy risks posed by untrusted directories. The protocol was demonstrated in [24].

Our ongoing work concerns two other dimensions of service-oriented middleware, namely service access and service composition. More precisely, we are investigating how multiple network connections can be explored to improve privacy-protection during service access and also how privacy preferences can be expressed on the specification of a service composition and enforced during service assembly to enable clients to compose services that respect privacy principles such as separation of duty.

7. Contracts and Grants with Industry

7.1. Contracts and Grants with Industry

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

Participants: Sonia Ben Mokhtar, Sébastien Bianco, Nikolaos Georgantas, Valérie Issarny.

- **Name:** Carroll Dyonisos – Phase 1 – *Dynamic Organization and Instantiation of Systems-of-Systems*
- **Related activities:** § 6.3
- **Period:** [April 2006- February 2007]
- **Partners:** CEA, THALES.

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

8. Other Grants and Activities

8.1. European Contracts and Grants

8.1.1. IST FP6 IP Amigo

Participants: Sonia Ben Mokhtar, Sébastien Bianco, Nikolaos Georgantas, Valérie Issarny, Graham Thomson.

- **Name:** IST Amigo – *Ambient Intelligence for the networked home environment*
- **URL:** http://www.extra.research.philips.com/euprojects/amigo/index.htm
- **Related activities:** § 6.3
- **Period:** [September 2004 - February 2008]
- **Partners:** Philips Research Eindhoven (The Netherlands) – project coordinator, Philips Design - Philips Consumer Electronics (The Netherlands), Fagor (Spain), France Télécom (France), Fraunhofer IMS (Germany), Fraunhofer IPSI (Germany), Ikerlan (Spain), INRIA (URs Rocquencourt, Futurs, Loraine, Rhône Alpes), Italdesign Giugiaro (Italy), Knowledge (Greece), Microsoft (Germany), Telin (The Netherlands), ICCS (Greece), Telefónica I+D (Spain), University of Paderborn (Germany), VTT (Finland).

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

8.1.2. IST FP6 STREP PLASTIC

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

- **Name:** IST PLASTIC – *Providing Lightweight and Adaptable Service Technology for Pervasive Information and Communication*
- **URL:** http://www.ist-plastic.org/
- **Related activities:** § 6.2

- **Period:** [February 2006 - July 2008]
- **Partners:** INRIA (UR Rocquencourt) – project coordinator, 4D Soft (Hungary), CNR (Italy), IBM (Belgium), Siemens Business Services (Germany), Telefónica I+D (Spain), UCL (United-Kingdom), Universitá di L'Aquila (Italy), Universitá della Svizzera Italiana (Switzerland), Virtual Trip (Greece), Pragmatica Technologies (Argentina).

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

8.2. International Research Networks and Work Groups

8.2.1. ASA Associated Team

- **Name:** ASA – *Adaptive SoftwAre*
- **Period:** [created 2007]
- **Participants:** Universita dell'Aquila, Dipartimento di Informatica, Italy

The objective of the team is assisting the development of dynamic distributed software systems for next generation ubiquitous communication and computing infrastructures. Software in the near ubiquitous future (Softure) will need to cope with variability, as software systems get deployed on an increasingly large diversity of computing platforms and should further deliver applications ubiquitously. Heterogeneity of the underlying communication and computing infrastructures, mobility and continuously evolving requirements demand new software paradigms that span the entire life-cycle, from development to deployment and execution. Softure must be developed in a way that facilitates both its deployment over heterogeneous networks of heterogeneous nodes, and its interaction with end users, their environment and/or other existing systems, depending on the application domain. Moreover, Softure should be reliable and meet the user's performance requirements and needs. Last but not least, Softure should be dynamic so that the applications they implement can be provisioned ubiquitously, despite the high dynamics of the pervasive networking and computing environment. Looking at the software life cycle, one key issue in this domain appears to be the disappearance of a clear distinction between static and dynamic aspects. Indeed, the adaptability requirement imposed by ubiquity makes software become evolving in nature, therefore introducing a strong interaction between the development environment and the middleware one. The goal of the ASA associated team is to research design and programming techniques and innovative middleware models that can be profitably integrated to support this new generation of software systems.

8.2.2. 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 Program aiming to bring together European groups from different communities working on middleware for mobile environment. The program intends to foster the definition and implementation of widely recognized middleware abstractions for new and emerging mobile applications. The program includes the following planned activities:

- Short term visit exchanges among the program participants (PhD students).
- Organization of a "closed" workshop for program 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 program.

8.2.3. 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: Software/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.

8.2.4. ERCIM WG SESAMI

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

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

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

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

8.2.5. ERCIM WG STM

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

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

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

8.3. National Contacts and Grants

8.3.1. EXOTICUS: Etude et eXpérimentation des Outils & Technologies IMS Compatibles avec les USages

Participants: Valérie Issarny, Pierre-Guillaume Raverdy.

- **Name:** EXOTICUS – *Etude et eXpérimentation des Outils & Technologies IMS Compatibles avec les USages*
- **Related activities:** § 6.2
- **Period:** [November 2007 – October 2009]
- **Partners:** Alcatel Lucent - coordinator, Legos, PragmaDev, Archos, Citypassenger, Deveryware, Transatel, ENST, GET/INT, INRIA (CRI Paris-Rocquencourt)

Introduced by the 3GPP standardization forum in relation with 3G, IMS (IP Multimedia Subsystem) is a key element for telecom operators trying to increase the status of their service provider activities, in a highly concurrential marketplace. The incoming IMS should enable the interconnexion of fix and mobile access networks around a same IP core, and making an open service platform available. The objective of the EXOTICUS project is to tackle the technological locks related to this new architecture, by contributing to:

- bridge the weakness of the norm about processes for the service composition and integration,
- foster the service creation dynamics thanks to new innovative service primitives,
- accelerate the development cycles, experiment new services and assess their use.

This research activities will bring fuel, all along the project, to the experiment platform that is deployed in southern Ile de France, and which is available for both professional (especially from the telecom and automotive sectors) and public population.

8.3.2. PERSO: PERvasive Service cOmposition

Participants: Manel Fredj, Nikolaos Georgantas, Pascal Poizat.

- **Name:** PERSO – *PERvasive Service cOmposition*
- **Related activities:** § 6.4
- **Period:** [November 2007 – October 2010]
- **Partners:** INRIA (CRI Paris-Rocquencourt) - coordinator, University Paris Dauphine - LAMSADE, University of Evry - IBISC.

The objectives of the project comprise the study, analysis and elaboration of a comprehensive approach to service composition for pervasive environments, supporting three description levels for service interfaces that enrich the standard description level for service interfaces (signatures): behavioral level, non-functional level and semantic level. We specifically aim at supporting these three levels and their integration towards a thorough approach to service composition. Two axes are further identified. The first axis concerns the foundations of service composition in terms of underlying formal models and related algorithms for service interface specification, service discovery, and service composition, including adaptation. The second axis concerns the application of the elaborated first axis outcomes to the runtime pervasive environment, tackling issues such as: efficiency and performance of algorithms for the interactive pervasive environment; resource consumption on resource-constrained portable devices; monitoring mechanisms for detecting change of conditions and for triggering composite service reconfiguration; runtime mechanisms for ensuring QoS in the face of change.

8.3.3. SemEUsE: SEMantiquE pour bUS de sErvice

Participants: Nebil Ben Mabrouk, Nikolaos Georgantas, Valérie Issarny, Fabio Mancinelli.

- **Name:** SEMEUSE – *SEMantiquE pour bUS de sErvice*
- **Related activities:** § 6.4
- **Period:** [December 2007 – May 2009]
- **Partners:** Thales - coordinator, France Télécom R&D, EBM WebSourcing, Université Pierre et Marie Curie - LIP6/MoVe, INSA Lyon, INRIA (CRI Paris-Rocquencourt), ObjectWeb, GET/INT.

The aim of the SemEUsE project is to study a semantic based service infrastructure that will provide the foundational services required for service-oriented applications to exchange information in a ubiquitous, reliable environment. The combination of an emerging semantic service infrastructure and associated engineering techniques will make it possible to produce flexible, mission-critical, software-based service applications that are dependable and manageable, and to provide high levels, business-focused, guaranteed end-to-end quality-of-services for all users.

9. Dissemination

9.1. Involvement within the Scientific Community

9.1.1. Program Committees

- Nikolaos Georgantas is co-organizer and PC co-chair of RSPSI'07: 2nd International Workshop on Requirements and Solutions for Pervasive Software Infrastructures, at the 9th International Conference on Ubiquitous Computing (Ubicomp'07);
- Nikolaos Georgantas is PC member of the ICSOFT'07 and PerSys'07 international conferences;
- Nikolaos Georgantas is PC member of the CODS'07, IMAI'07, MoSO'07, and SIPE'07 international workshops;
- Valérie Issarny is PC chair of ESEC/FSE'09: 7th joint meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering;
- Valérie Issarny is PC co-chair of Middleware'08: 9th International Middleware Conference;
- Valérie Issarny is PC member of the CFSE'08, Coordination'07, FASE'08, ICSE'09, IFIPTM'07, Middleware'07, Mobiquitous'07, and WICSA'07 international conferences;
- Valérie Issarny is PC member of the Adamus'08, ESSPE'07, FOCLASA'07, MP2P'07, MW4SOC'07, and WOSP'07 international workshops;
- Pascal Poizat is co-organizer and PC co-chair of FOCLASA'07: 6th International Workshop on Foundations of Coordination Languages and Software Architectures, at the 18th International Conference on Concurrency Theory (CONCUR'07);
- Pascal Poizat is co-organizer and PC co-chair of WCAT'07: 4th International Workshop on Coordination and Adaptation Techniques for Software Entities, at the 21st European Conference on Object-Oriented Programming (ECOOP'07);
- Pascal Poizat is PC member of the PROVECS'07 international workshop;
- Pierre-Guillaume Raverdy is PC chair of CAPS'07: 3rd Workshop on Context Awareness for Proactive Systems;
- Pierre-Guillaume Raverdy is PC member of the DANMS'07 international workshop.

9.1.2. Other Activities

- The COCOA Semantic Service Middleware is a featured component of the Amigo Challenge: http://challenge.amigo-project.org/. The Amigo Challenge is a programming competition to develop novel applications using the Amigo middleware. The Challenge has been promoted through many academic and industrial organizations worldwide, as well as several popular online technical news sites. COCOA is proving to be one the most popular downloads in the Challenge.
- Nikolaos Georgantas has been reviewer for the ACM Computing Surveys, ACM Transactions on Software Engineering and Methodology (TOSEM), IEEE Transactions on Mobile Computing (TMC), and International Journal of Semantic Web and Information Systems (IJSWIS) journals;
- Valérie Issarny is associate editor of the ACM Computing Surveys journal;
- Valérie Issarny is member of the evaluation committee for ANR TechLog (French National Agency for Research, Software Technologies Program), call 2007;
- Valérie Issarny is member of the Execution Board of RNTL (French National Network for Software Technologies);
- V. Issarny is coordinator of the OFTA (http://www.ofta.net) working group on Pervasive Computing;

- Pascal Poizat is member of the review committee of Science of Computer Programming special issue on FOCLASA'06 and Fundamentae Informatica special issue on FSEN'07;
- Pascal Poizat is co-editor and member of the review committee of the special issue on the *3rd International Workshop on Coordination and Adaptation Techniques for Software Entities* of the journal Electronic Notes in Theoretical Science;
- Pascal Poizat was expert for the evaluation of a project of "Contrat de Plan Etat-Region Lorraine" (Research grant for region Lorraine, France);
- Pascal Poizat is member of the recruiting commission for Computer Science Assistant Professors in University of Evry, France;
- Pascal Poizat is member of the council of laboratory IBISC FRE 2873 CNRS, Evry, France.

9.2. Teaching

- Roberto Speicys Cardoso gives a course on Distributed Objects Architectures (laboratory). Final year of the five-year computer engineering degree at the Institut des Sciences et Techniques des Yvelines of the University of Versailles Saint-Quentin en Yvelines;
- Manel Fredj gives a course on Middleware Architectures (laboratory). Final year of the five-year computer engineering degree at the Ecole Supérieure d'Ingénierie Léonard de Vinci of the Pôle Universitaire Léonard de Vinci;
- Nikolaos Georgantas gives a course on Middleware Architectures (lectures). Final year of the five-year computer engineering degree at the Ecole Supérieure d'Ingénierie Léonard de Vinci of the Pôle Universitaire Léonard de Vinci;
- Nikolaos Georgantas gives a course on Distributed Objects Architectures (lectures). Final year of the five year computer engineering degree at the Institut des Sciences et Techniques des Yvelines of the University of Versailles Saint-Quentin en Yvelines;
- Valérie Issarny gives a course on Software Architectures for Distributed Systems (lectures), as part of the SAL course of the Master 2 COSY of the University of Versailles Saint-Quentin en Yvelines.

9.3. Internships

During year 2007, members of the ARLES project-team supervised the work of the following student interns:

- Sandrine Beauche, *Adaptation de services en informatique diffuse*, Master of Science, SAR speciality, University Pierre et Marie Curie, Paris, France.

9.4. Invited Talks

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

- Nikolaos Georgantas, *Interoperable Service-Oriented Architecture for Ambient Intelligence in the Networked Home*, Tutorial, SmartUniversity, Ubiquitous Computing: State of the Art and Challenges for the Software Infrastructure, September 2007, Sophia Antipolis, France.
- Nikolaos Georgantas, *The Amigo project*, EC workshop on Software and Service Architectures and Infrastructures, January 2007, Brussels, Belgium.
- Pascal Poizat, *Model-based Software Adaptation. Introduction and Contributions*, invited talk, PGL team, LRI, University Paris Sud, October 2007, France.
- Pascal Poizat, *Model-based Software Adaptation. Open Systems & Semantics*, invited talk, University of Málaga, March 2007, Málaga, Spain.

- Valérie Issarny, *Dependability of SoS*, tutorial, School on Systems of Systems, March 2007, Paris, France.
- Valérie Issarny, *Service Oriented Architectures*, tutorial, School on Systems of Systems, March 2007, Paris, France.

- Roberto Speicys Cardoso was awarded the Best Student Paper at IFIPTM 2007 - Joint iTrust and PST Conferences on Privacy, Trust Management and Security.

10. Bibliography

Major publications by the team in recent years

[1] S. BEN MOKHTAR, N. GEORGANTAS, V. ISSARNY. *COCOA: COnversation-based Service Composition in PervAsive Computing Environments with QoS Support*, in "Journal of Systems and Software, Special Issue on ICPS'06", vol. 80, n° 12, 2007, p. 1941–1955.

[2] S. BEN MOKHTAR, D. PREUVENEERS, N. GEORGANTAS, V. ISSARNY, Y. BERBERS. *EASY: Efficient SemAntic Service DiscoverY in Pervasive Computing Environments with QoS and Context Support*, in "Journal of Systems and Software, Special Issue on Web Services Modelling and Testing", to appear, 2007.

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

[4] D. CHARLET, V. ISSARNY, R. CHIBOUT. *Energy-efficient Middleware-layer Multi-radio Networking: An Assessment in the Area of Service Discovery*, in "Computer Networks", to appear, 2007.

[5] V. ISSARNY, M. CAPORUSCIO, N. GEORGANTAS. *A Perspective on the Future of Middleware-based Software Engineering*, in "Future of Software Engineering 2007", IEEE-CS Press, 2007.

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

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

[8] F. SAILHAN, V. ISSARNY. *Energy-aware Web Caching over Hybrid Networks*, in "Handbook of Mobile Computing", CRC Press, 2004.

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

Year Publications

Books and Monographs

[10] S. BECKER, C. CANAL, N. DIAKOV, J. M. MURILLO, P. POIZAT, M. TIVOLI (editors). *Special Issue on Proceedings of the Third International Workshop on Coordination and Adaption Techniques for Software Entities (WCAT 06)*, Electronic Notes in Theoretical Computer Science, vol. 189, 2007.

[11] C. CANAL, J. M. MURILLO, P. POIZAT (editors). *Practical Approaches for Software Adaptation. Proceedings of the Fourth International Workshop on Coordination and Adaptation Techniques for Software Entities (WCAT'07)*, Held in conjunction with ECOOP'2007, 2007.

[12] V. ISSARNY (editor). *ARAGO 31: Informatique Diffuse*, Rapport de l'OFTA, ISBN 2-906028-17-7, OFTA, 2007.

Doctoral dissertations and Habilitation theses

[13] S. BEN MOKHTAR. *Intergiciel Sémantique pour les Services de l'Informatique Diffuse*, Ph. D. Thesis, Université Pierre et Marie Curie – Paris 6, December 2007.

Articles in refereed journals and book chapters

[14] D. ATHANASOPOULOS, A. ZARRAS, V. ISSARNY, E. PITOURA, P. VASSILIADIS. *CoWSAMI: Interface-Aware Context Gathering in Ambient Intelligence Environments*, in "Pervasive and Mobile Computing Journal", to appear, 2007.

[15] C. ATTIOGBÉ, P. POIZAT, G. SALAÜN. *A Formal and Tool-Equipped Approach for the Integration of State Diagrams and Formal Datatypes*, in "IEEE Transactions on Software Engineering", vol. 33, n^o^ 3, 2007, p. 157–170.

[16] D. AUGOT, R. BHASKAR, V. ISSARNY, D. SACCHETTI. *A three round authenticated group key agreement protocol for ad hoc networks*, in "Pervasive and Mobile Computing Journal", vol. 3, n^o^ 1, 2007, p. 36–52.

[17] S. BEN MOKHTAR, N. GEORGANTAS, V. ISSARNY. *COCOA: COnversation-based Service Composition in PervAsive Computing Environments with QoS Support*, in "Journal of Systems and Software, Special Issue on ICPS'06", vol. 80, n^o^ 12, 2007, p. 1941–1955.

[18] S. BEN MOKHTAR, D. PREUVENEERS, N. GEORGANTAS, V. ISSARNY, Y. BERBERS. *EASY: Efficient SemAntic Service DiscoverY in Pervasive Computing Environments with QoS and Context Support*, in "Journal of Systems and Software, Special Issue on Web Services Modelling and Testing", to appear, 2007.

[19] Y.-D. BROMBERG, C. CONSEL, W. JOUVE, S. BEN MOKHTAR, N. GEORGANTAS, V. ISSARNY, P.-G. RAVERDY. *Intergiciels orientés services pour l'informatique diffuse*, in "ARAGO 31: Informatique Diffuse", Rapport de l'OFTA, ISBN 2-906028-17-7, OFTA, 2007.

[20] D. CHARLET, V. ISSARNY, R. CHIBOUT. *Energy-efficient Middleware-layer Multi-radio Networking: An Assessment in the Area of Service Discovery*, in "Computer Networks", to appear, 2007.

[21] J. LIU, V. ISSARNY. *An incentive compatible reputation mechanism for ubiquitous computing environments*, in "International Journal of Information Security", vol. 6, n^o^ 5, 2007, p. 297–311.

[22] T. ZAHN, J. SCHILLER. *Designing structured peer-to-peer overlays as a platform for distributed network applications in mobile ad hoc networks*, in "Journal of Computer Communications, Special Issue on Foundation of Peer-to-Peer Computing", to appear, 2007.

Publications in Conferences and Workshops

[23] S. BECKER, C. CANAL, N. DIAKOV, J. M. MURILLO, P. POIZAT, M. TIVOLI. *Coordination and Adaptation Techniques: Bridging the Gap Between Design and Implementation*, in "ECOOP 2006 Workshop Reader", Lecture Notes in Computer Science, vol. 4379, Springer, 2007, p. 72–86.

[24] S. BEN MOKHTAR, P.-G. RAVERDY, R. SPEICYS CARDOSO, A. URBIETA, N. GEORGANTAS. *Discovering Social Services in Pervasive Environments with Privacy (Demonstration Paper)*, in "Proceedings of the ACM/IFIP/USENIX 8th International Middleware Conference", 2007.

[25] M. CAPORUSCIO, D. CHARLET, V. ISSARNY, A. NAVARRA. *Energetic Performance of Service-oriented Multi-radio Networks: Issues and Perspectives*, in "Proceedings of the 6th International Workshop on Software and Performance (WOSP 07), Buenos Aires, Argentina", February 2007.

[26] J. CUBO, G. SALAÜN, C. CANAL, E. PIMENTEL, P. POIZAT. *Relating Model-Based Adaptation and Implementation Platforms: A Case Study with WF/.NET 3.0*, in "Proceedings of the 12th International Workshop on Component-Oriented Programming (WCOP 07)", to appear, 2007.

[27] J. CUBO, G. SALAÜN, C. CANAL, E. PIMENTEL, P. POIZAT. *Verification and Adaptation of WF/.NET Components*, in "Proceedings of the Workshop on Formal Aspects of Component Software (FACS 07)", to appear, 2007.

[28] N. GEORGANTAS. *Dynamic services for pervasive environments*, in "Proceedings of Dagstuhl Seminar 07462, Assisted Living Systems & Models, Architectures and Engineering Approaches", November 2007.

[29] S. HADDAD, P. POIZAT. *Transactional Reduction of Component Compositions*, in "Proceedings of the IFIP International Conference on Formal Methods for Networked and Distributed Systems (FORTE 07)", Lecture Notes in Computer Science, vol. 4574, Springer, 2007, p. 341–357.

[30] V. ISSARNY, M. CAPORUSCIO, N. GEORGANTAS. *A Perspective on the Future of Middleware-based Software Engineering*, in "Future of Software Engineering 2007", IEEE-CS Press, 2007.

[31] R. MATEESCU, P. POIZAT, G. SALAÜN. *Behavioral Adaptation of Component Compositions based on Process Algebra Encodings (Short Paper)*, in "Proceedings of the 22nd IEEE/ACM International Conference on Automated Software Engineering (ASE 07)", 2007.

[32] P. POIZAT, G. SALAÜN. *Adaptation of Open Component-based Systems*, in "Proceedings of the IFIP International Conference on Formal Methods for Open Object-Based Distributed Systems (FMOODS 07)", Lecture Notes in Computer Science, vol. 4468, Springer, 2007, p. 141–156.

[33] P. POIZAT, G. SALAÜN, M. TIVOLI. *An Adaptation-based Approach to Incrementally Build Component Systems*, in "Proceedings of the International Workshop on Formal Aspects of Component Software (FACS 06)", Electronic Notes in Theoretical Computer Science, vol. 182, Elsevier, 2007, p. 155–170.

[34] L. RONG, M. FREDJ, V. ISSARNY, N. GEORGANTAS. *Mobility Management in B3G Networks: a Middleware based Approach*, in "Proceedings of the International Workshop on the Engineering of Software Services for Pervasive Environments (ESSPE 07), as part the ACM SIGSOFT Symposium on the Foundations of Software Engineering", 2007, p. 41–45.

[35] L. RONG, T. WALLET, M. FREDJ, N. GEORGANTAS. *Mobile Medical Diagnosis: an m-Health Initiative through Service Continuity in B3G (Demonstration Paper)*, in "Proceedings of the ACM/IFIP/USENIX 8th International Middleware Conference", 2007.

[36] R. SPEICYS CARDOSO, V. ISSARNY. *Architecting Pervasive Computing Systems for Privacy : A Survey*, in "Proceedings of the 6th Working IEEE/IFIP Conference on Software Architecture (WICSA 07)", 2007.

[37] R. SPEICYS CARDOSO, P.-G. RAVERDY, V. ISSARNY. *A Privacy-Aware Service Discovery Middleware for Pervasive Environments*, in "Proceedings of the IFIPTM 2007 Joint iTrust and PST Conferences on Privacy, Trust Management and Security", 2007.

[38] R. SPEICYS CARDOSO, S. BEN MOKHTAR, A. URBIETA, V. ISSARNY. *Privacy-Aware Service Discovery for Pervasive Computing (Work In Progress)*, in "Proceedings of the ACM/IFIP/USENIX 8th International Middleware Conference", 2007.

[39] G. THOMSON, S. BIANCO, S. BEN MOKHTAR, N. GEORGANTAS, V. ISSARNY. *Amigo aware services*, in "Proceedings of the Amigo EU Project Workshop, European Conference on Ambient Intelligence", 2007.

[40] G. THOMSON, D. SACCHETTI, Y.-D. BROMBERG, J. PARRA, N. GEORGANTAS, V. ISSARNY. *Amigo Interoperability Framework: Dynamically Integrating Heterogeneous Devices and Services*, in "Proceedings of the Amigo EU Project Workshop, European Conference on Ambient Intelligence", 2007.

[41] P. VINK, N. GEORGANTAS. *Amigo Architecture: Service-Oriented Architecture for Intelligent Future In-Home Networks*, in "Proceedings of the Amigo EU Project Workshop, European Conference on Ambient Intelligence", 2007.

[42] G. XIE, Z. LI, J. CHEN, V. ISSARNY, A. CONTE. *Efficient Soft QoS Guarantee in Mobile Ad Hoc Networks*, in "Proceedings of the ISCA 20th International Conference on Parallel and Distributed Computing Systems (ISCA PDCS 2007)", 2007.

[43] G. XIE, Z. LI, J. CHEN, Y. WEI, V. ISSARNY, A. CONTE. *DTCS: A Dynamic Tree-based Consistency Scheme of Cooperative Caching in Mobile Ad Hoc Networks*, in "Proceedings of the 3rd IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMOB'2007)", 2007.

[44] G. XIE, J. YANG, V. ISSARNY, A. CONTE. *An Accurate and Efficient 3-Phase Measurement Method for IP Traffic Flow on High Speed Link*, in "Proceedings of the IEEE International Conference on Networking (ICN'07)", 2007.

[45] G. XIE, G. ZHANG, J. YANG, Y. MIN, V. ISSARNY, A. CONTE. *The Survey on Traffic of Metro Area Network with Measurement On-line*, in "Proceedings of the 20th International Teletraffic Congress", 2007.

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

Internal Reports

[47] E. BACCELLI, T. ZAHN, J. SCHILLER. *DHT-OLSR*, Research Report, n^o RR-6194, INRIA, 2007, http://hal.inria.fr/inria-00148347/fr/.

[48] R. MATEESCU, P. POIZAT, G. SALAÜN. *Behavioral Adaptation of Component Compositions based on Process Algebra Encodings*, Technical report, n^o 6362, INRIA, 2007, http://hal.inria.fr/inria-00189246/fr/.

Miscellaneous

[49] M. CAPORUSCIO, V. ISSARNY (editors). *PLASTIC D3.1: Middleware Specification and Architecture*, March 2007, Project IST FP6 STREP PLASTIC deliverable.

[50] G. THOMSON, N. GEORGANTAS (editors). *Amigo D3.4 Amigo overall middleware: First Prototype implementation & documentation*, May 2007, Project Amigo deliverable.

[51] G. THOMSON, N. GEORGANTAS (editors). *Amigo D3.5 Amigo overall middleware: Final Prototype implementation & documentation*, December 2007, Project Amigo deliverable.

[52] S. BEAUCHE. *Composition adaptative de services en informatique diffuse*, Technical report, Université Pierre et Marie Curie – Paris 6, 2007.

[53] V. ISSARNY, M. CAPORUSCIO, P.-G. RAVERDY. *A Service-Oriented Middleware for B3G Networking*, October 2007, ERCIM News 70 - Issue on Service-Oriented Computing.