



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team ARLES

*Software architectures and distributed
systems*

Paris - Rocquencourt

Theme : Distributed Systems and Services

Activity
R *eport*

2010

Table of contents

1. Team	1
2. Overall Objectives	1
2.1. Overall Objectives	1
2.2. Highlights of the Year	3
3. Scientific Foundations	3
3.1. Introduction	3
3.2. Engineering Pervasive Software Systems	3
3.2.1. Middleware-based Software Engineering	3
3.2.2. Beyond Middleware-based Architectures for Interoperability	4
3.3. Middleware Architectures for Pervasive Computing	5
3.3.1. Service Oriented Middleware	5
3.3.2. Middleware for Wireless Sensor Networks	8
4. Application Domains	9
5. Software	9
5.1. Introduction	9
5.2. Service-oriented Middleware for Pervasive Computing	9
5.3. iBICOOP: Mobile Data Management in Multi-* Networks	10
5.4. Srijan: Data-driven Macroprogramming for Sensor Networks	11
5.5. Yarta: Middleware for supporting Mobile Social Applications	12
6. New Results	12
6.1. Introduction	12
6.2. Data-driven Macroprogramming and Middleware for Sensor Networks	12
6.3. Pervasive Service Oriented Software Engineering	13
6.4. Dynamic Synthesis of Middleware Connectors	16
6.5. Middleware Support for Mobile Social Applications	17
7. Other Grants and Activities	18
7.1. European Contracts and Grants	18
7.1.1. FP7 ICT FET CONNECT	18
7.1.2. FP7 ICT IP CHOReOS	19
7.1.3. FP7 ICT NoE NESSoS	19
7.1.4. FP7 ICT CA EternalS	20
7.2. International Research Networks and Work Groups	20
7.2.1. ForeverSOA Associated Team	20
7.2.2. ERCIM WG SERENE	21
7.2.3. ERCIM WG STM	22
7.3. National Contacts and Grants	22
7.3.1. ANR SemEUSe: SEMantiquE pour bUS de sERVICE	22
7.3.2. ANR ITeMIS: IT and Embedded Integrated Systems	22
7.3.3. INRIA D2T Action de Developpement Technologique Srijan	23
7.3.4. INRIA DTI Action de Transfert iBICOOP	23
8. Dissemination	23
8.1. Involvement within the Scientific Community	23
8.1.1. Programme Committees of Conferences and Workshops	23
8.1.2. Leadership Services in Academic Events and Edited Journals	24
8.1.3. Other Academic Services	24
8.2. Teaching	24
8.3. Internships	24
8.4. Awards	25
9. Bibliography	25

1. Team

Research Scientists

Valérie Issarny [Team Leader, INRIA Senior Researcher, HdR]
Nikolaos Georgantas [INRIA Junior Researcher]
Animesh Pathak [INRIA Junior Researcher]

Technical Staff

Emil Andriescu [*Apprenti*, INRIA, since October 2010]
Mohammad Ashiqur Rahaman [*Ingénieur Expert*, INRIA]
Sandrine Beauche [*Ingénieur Expert*, INRIA]
Sneha-Sham Godbole [*Ingénieur Expert*, INRIA]
Iraklis Leontiadis [*Ingénieur Jeune Diplômée*, INRIA]
Rachid Saadi [*Ingénieur Expert*, INRIA]
Daniel Sykes [*Ingénieur Expert*, INRIA]
Roberto Speicys Cardoso [*Ingénieur Spécialiste*, INRIA]

PhD Students

Hamid Ameziani [INRIA, University Pierre et Marie Curie - Paris 6, until September 2010]
Dionysis Athanasopoulos [INRIA, University of Ioannina, Greece]
Nebil Ben Mabrouk [INRIA CORDI-S, University Pierre and Marie Curie - Paris 6]
Amel Bennaceur [INRIA, University Pierre et Marie Curie Paris - 6]
Pankesh Patel [INRIA CORDI-S, University Pierre et Marie Curie - Paris 6, since October 2010]
Sara Hachem [INRIA, University of Versailles Saint-Quentin-en-Yvelines, since November 2010]
Amir Seyedi [INRIA, University Pierre et Marie Curie - Paris 6]

Post-Doctoral Fellows

Alessandra Toninelli [ERCIM until September 2010, then INRIA]
Thiago Valladares Sabino Teixeira [INRIA, since October 2010]

Visiting Scientist

Oleg Davidyuk [University of Oulu, Finland, April 2010 - July 2010]

Administrative Assistant

Stéphanie Chaix

2. Overall Objectives

2.1. Overall Objectives

With digital equipments becoming increasingly networked, either on wired or wireless networks, for personal and professional use alike, distributed software systems have become a crucial element in information and communications technologies. The study of these systems forms the core of the ARLES project-team's work, which is specifically concerned with defining new system software architectures, based on the use of emerging networking technologies.

Since the 90s, middleware has emerged as a prominent solution to overcome the heterogeneity of the distributed infrastructure. It establishes a software layer that homogenizes the infrastructure diversities by means of a well-defined and structured distributed programming model. Moreover, middleware provides building blocks to be exploited by applications for enforcing non-functional properties, such as dependability and performance. Finally, by providing reusable solutions for the development of distributed systems, which is increasingly complex, the role of middleware has proved central in the software system development practice.

However, the development of distributed systems using middleware remains a complex task¹. In particular, middleware must define the adequate networking and computing abstractions to match the distributed application requirements. Further, while the development of legacy middleware has been significantly driven by requirements of distributed information systems, the ongoing evolution of the networking environment leads to a much broader application of distributed computing — including, among others, the proliferation of disparate mobile computing devices and smart phones, and the inclusion of wireless sensor networks into modern and future pervasive systems. As a result of the above, new requirements arise for middleware, e.g., supporting open and mobile networking, and context awareness [4]. We are especially interested in applications involving social interactions among the users of modern smart phones, which involve addressing issues of trust and privacy along with efficiency of algorithms owing to the lack of resources.

In the above context, ARLES studies the engineering of middleware-based systems, with a special emphasis on enabling interoperability in the ubiquitous computing² and subsequent pervasive computing³ and ambient intelligence⁴ visions, keeping in view the advances in the constitution of these systems, both in terms of the various communication and discovery protocols used, as well as the hardware and software platforms available. Our research is more specifically focused on eliciting middleware for pervasive computing, from foundations and architectural design to prototype implementations. Proposed middleware and programming abstractions shall then effectively leverage networked resources, in particular accounting for advanced wireless networking technologies, while also addressing concerns raised due to the presence of large numbers of extremely low-power devices such as wireless sensors and actuators. This raises a number of complementary research challenges:

- **System architecture:** How to architect and further program pervasive computing systems out of the resources available in the highly dynamic networking environment?
- **System modeling:** How to abstract and further model the networked resources and related networking environment for distributed pervasive computing?
- **Interoperability:** How to actually overcome the heterogeneity of the pervasive computing environment, including middleware heterogeneity?
- **Networking:** How to benefit from the rich wireless networking technologies?
- **Quality of service:** How to effectively master the high dynamics and openness of pervasive computing environments and, in particular, how to ensure dependability and performance in such environments?

All the above mentioned challenges are inter-related, calling for their study in both the software engineering and distributed systems domains. Indeed, proposed middleware abstractions and related programming models shall effectively foster the development of *robust* distributed software systems, which, at the same time, must be implemented in an *efficient* way. Specifically:

- In the **software engineering domain**, ARLES studies resource abstractions and interaction paradigms to be offered by middleware, together with the associated languages, methods and tools for describing and composing the abstracted resources into applications. Our primary goal is to foster the development of *robust* and *interoperable* distributed systems that are highly dynamic to adapt to the ever-changing networking environment, and further meet quality of service requirements.
- In the **distributed systems domain**, ARLES studies innovative middleware architectures and related distributed algorithms and protocols for the *efficient* networking of distributed resources into distributed pervasive systems, in particular taking into account the high mobility and heterogeneity of the constituent nodes.

¹V. Issarny *et al.* Systematic Aid for Developing Middleware Architectures. In Communications of the ACM, Special Issue on Adaptive Middleware, 45(6). 2002.

²M. Weiser. The Computer for the Twenty-First Century. In Scientific American. 1991.

³M. Satyanarayanan. Pervasive Computing: Vision and Challenges. In IEEE Personal Communications. August 2001.

⁴E. Aarts *et al.* Ambient Intelligence. In The Invisible Future: the Seamless Integration of Technology into Everyday Life. McGraw-Hill. 2002.

2.2. Highlights of the Year

During this year, while we have been pursuing our research on advanced service-oriented architectures and related middleware solutions for next generation networking environments, we have made initial progress in research on two new subjects, both called for by the ongoing drastic evolution of the networking environment:

- Dynamic interoperability among networked systems towards making them eternal, by way of on-the-fly generation of connectors based on adequate system models. This research is part of a major European collaborative project within the Future and Emerging Technology programme of the EC FP7-ICT (§ 6.4, § 7.1.1),
- System-level support for mobile social applications, by way of a middleware architecture that involves research in the areas of semantic models for social data, mobile distributed storage, a novel policy framework for access control, and efficient, predictive data-replication on resource-constrained devices, among others (§ 6.5).

Along with the above research, we have initiated transfer of our middleware technology for mobile handheld devices:

- The AMBIENTIC spin-off (<http://www.ambientic.com/>) is being founded for launch in early 2011. AMBIENTIC leverages the ARLES middleware technology that has been developed over the last 10 years for supporting the development of mobile collaborative services. AMBIENTIC specifically develops innovative mobile distributed services on top of the iBICOOP middleware that allows for seamless interaction and content sharing in today's multi-* networks. The AMBIENTIC project is winner of the *Concours national d'aide à la création d'entreprises de technologies innovantes* award (<http://www.enseignementsup-recherche.gouv.fr/pid20162/concours-national-d-aide-a-la-creation-d-entreprises-innovantes.html>) in the *Emergence* category in 2009 and in the *Création* category in 2010.

3. Scientific Foundations

3.1. Introduction

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 methods and tools supporting pervasive software systems engineering, with a special emphasis on associated middleware solutions.

3.2. Engineering Pervasive Software Systems

Since its emergence, middleware has proved successful to assist distributed software development, making development faster and easier, and significantly promoting software reuse while overcoming the heterogeneity of the distributed infrastructure. As a result, middleware-based software engineering is central to the principled development of pervasive computing systems.

3.2.1. Middleware-based Software Engineering

Middleware establishes a new software layer that homogenizes the infrastructure's diversities by means of a well-defined and structured distributed programming model, relieving software developers from low-level implementation details, by: (i) at least abstracting socket-level network programming in terms of high-level network abstractions matching the application computational model, and (ii) possibly managing networked resources to offer quality of service guarantees and/or domain specific functionalities, through reusable middleware-level services. More specifically, middleware defines:

- A resource definition language that is used for specifying data types and interfaces of networked software resources;

- A high-level addressing scheme based on the underlying network addressing scheme for locating resources;
- Interaction paradigms and semantics for achieving coordination;
- A transport/session protocol for achieving communication; and
- A naming/discovery protocol with related registry structure and matching relation for publishing and discovering the resources available in the given network.

Over the years, the role of middleware has proven central to address the ever increasing complexity of distributed systems in a reusable way. Further, middleware provides building blocks to be exploited by applications for enforcing non-functional properties, such as dependability and performance. Attractive features of middleware have made it a powerful tool in the software system development practice. Hence, middleware is a key factor that has been and needs to be further taken into account in the Software Engineering (SE) discipline⁵. The advent of middleware standards have further contributed to the systematic adoption of the technology for distributed software development.

However, mature engineering methodologies to comprehensively assist the development of middleware-based software systems, from requirements analysis to deployment and maintenance, are lagging behind⁶. Indeed, software development accounting for middleware support is only emerging⁷, and methods and related tools are dearly required for middleware-based software engineering. This need becomes even more demanding if we consider the diversity and scale of today's networking environments and application domains, which makes middleware and its association with applications highly complex [4].

Even if legacy middleware has well been established and employed, the rapidly evolving computing and networking environments — including new devices and networks, new application domains, and complex associations among them — raise new, challenging requirements for middleware. Among those, access to computational resources should be open across network boundaries and dynamic due to the potential mobility of host- and user-nodes. This urges middleware to support methods and mechanisms for description, dynamic discovery and association, late binding, and loose coordination of resources. In such variable and unpredictable environments, operating not only according to explicit system inputs but also according to the context of system operation becomes of major importance, which should be enabled by the middleware. Additionally, the networking infrastructure is continuing to evolve at a fast pace, and suggesting new development paradigms for distributed systems, leading to new middleware platforms and further calling for novel software engineering methods and tools. Major system requirements posed by today's networking infrastructure relate to openness and mobility, and context-awareness. This results in the need to investigate next generation middleware with support for universal interoperability, open coordination, and context-aware adaptability. Further, with middleware becoming central in software development, software processes shall evolve so as to integrate middleware features at all phases of the software development.

3.2.2. *Beyond Middleware-based Architectures for Interoperability*

As discussed above, middleware stands as the conceptual paradigm to effectively network together heterogeneous systems, specifically providing upper layer interoperability. Still, middleware is yet another technological block, which creates islands of networked systems.

Interoperable middleware has been introduced to overcome middleware heterogeneity. However, solutions remain rather static, requiring either use of a proprietary interface or a priori implementation of protocol translators. In general, interoperability solutions solve protocol mismatch among middleware at syntactic level, which is too restrictive. This is even truer when one considers the many dimensions of heterogeneity, including software, hardware and networks, which now arise in ubiquitous networking environments, and that require

⁵W. Emmerich. Software engineering and middleware: a roadmap. In Proceedings of the Conference on the Future of Software Engineering, Limerick, Ireland, Jun. 2000.

⁶R. E. Schantz and D. C. Schmidt. Research advances in middleware for distributed systems: state of the art". In Proceedings of IFIP World Computer Congress, 2003.

⁷D. Schmidt. Model-driven engineering. IEEE Computer, 39(2), Feb 2006.

fine tuning of the middleware according to the specific capacities embedded within the interacting parties. Thus, interoperable middleware can at best solve protocol mismatches arising among middleware aimed at a specific domain. Indeed, it is not possible to a priori design a universal middleware solution that will enable effective networking of digital systems, while spanning the many dimensions of heterogeneity now arising in networked environments and further called to increase dramatically in the future.

A revolutionary approach to the seamless networking of digital systems is to synthesize on the fly the connectors via which networked systems communicate. The resulting emergent connectors then compose and further adapt the interaction protocols run by the connected systems, which realize application- down to middleware-layer protocols. Hence, thanks to results in this new area, networked digital systems will survive the erosion and further emergence of interaction protocols.

We have specifically undertaken cooperative research on the dynamic synthesis of emergent connectors which shall rely on a formal foundation for connectors that allows learning, reasoning about, and adapting the interaction behavior of networked systems⁸. Further, compared to the state of the art foundations for connectors, it should operate a drastic shift by learning, reasoning about, and synthesizing connector behavior at run-time. Indeed, the use of connector specifications pioneered by the software architecture research field has mainly been considered as a design-time concern, for which automated reasoning is now getting practical even if limitations remain. On the other hand, recent effort in the semantic Web domain brings ontology-based semantic knowledge and reasoning at run-time but networked system solutions based thereupon are currently mainly focused on the functional behavior of networked systems, with few attempts to capture their interaction behavior as well as non-functional properties. In this new approach, the interaction protocols (both application- and middleware layer) behavior will be learnt by observing the interactions of the networked systems, where ontology-based specification and other semantic knowledge will be exploited for generating connectors on the fly.

3.3. Middleware Architectures for Pervasive Computing

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 composed of highly heterogeneous nodes. In this section, we present an overview of: (i) service oriented middleware, a prominent paradigm in large distributed systems today, and (ii) middleware for wireless sensor networks, which have recently emerged as a promising platform.

3.3.1. Service Oriented Middleware

The *Service Oriented Computing*(SOC) paradigm advocates that networked resources should be abstracted as services, thus allowing their open and dynamic discovery, access and composition, and hence reuse. Due to this flexibility, SOC has proven to be a key enabler for pervasive computing. Moreover, SOC enables integrating pervasive environments into broader service oriented settings: the current and especially the *Future Internet* is the ultimate case of such integration. We, more particularly, envision the Future Internet as a ubiquitous setting where services representing resources, people and things can be freely and dynamically composed in a decentralized fashion, which is designated by the notion of service choreography in the SOC idiom. In the following, we discuss the role that *service oriented middleware* is aimed to have within our above sketched vision of the Future Internet, of which pervasive computing forms an integral part.

From service oriented computing to service oriented middleware: In the last few years, there is a growing interest in choreography as a key concept in forming complex service-oriented systems. Choreography is put forward as a generic abstraction of any possible collaboration among multiple services, and integrates previously established views on service composition, among which service orchestration. More specifically, we find in the literature three distinct but overlapping viewpoints (often denoted with varying terms): (1) *Choreography*

⁸Valérie Issarny, Bernhard Steffen, Bengt Jonsson, Gordon S. Blair, Paul Grace, Marta Z. Kwiatkowska, Radu Calinescu, Paola Inverardi, Massimo Tivoli, Antonia Bertolino, Antonino Sabetta: CONNECT Challenges: Towards Emergent Connectors for Eternal Networked Systems. In Proceedings of ICECCS 2009.

captures collaborative processes involving multiple services and especially their interactions seen from a global perspective; (2) *Behavioral interface* captures the behavior of a single service (i.e., dependencies between the interactions that the service supports) which participates in choreography; and (3) *Orchestration* deals with the description of the interactions in which a given service can engage with other services, as well as the internal steps between these interactions. Based on these three viewpoints, choreography modeling finds in the literature different styles: Interaction-oriented models describe choreography as a set of interactions between participants; while process-oriented models describe choreography as a parallel composition of the participants' business processes. Activity-based models focus on the interactions between the parties and their ordering, whereas the state of the interaction is not explicitly modeled or only partly modeled using variables; while state-based models model the states of the choreography as first-class entities, and the interactions as transitions between states.

The above modeling categorizations are applied in the ways in which: service choreographies are specified (e.g., by employing languages such as BPMN, WS-CDL, BPEL); services are discovered, selected and composed into choreographies (e.g., based on their features concerning interfaces, behavior, and non-functional properties such as QoS and context); heterogeneity between choreographed services is resolved via adaptation (e.g., in terms of service features and also underlying communication protocols); choreographies are deployed and enacted (e.g., in terms of deployment styles and execution engines); and choreographies are maintained/adapted given the independent evolution of choreographed services (e.g., in terms of availability and QoS). These are demanding functionalities that service oriented middleware should provide for supporting service choreographies. In providing these functionalities in the context of the Future Internet, service oriented middleware is further challenged by two key Future Internet properties: its ultra large scale as in number of users and services, and the high degree of heterogeneity of services, whose hosting platforms may range from that of resource-rich, fixed hosts to wireless, resource-constrained devices. These two properties call for considerable advances to the state of the art of the SOC paradigm.

Our work in the last years has focused on providing solutions to the above identified challenges, more particularly in the domain of pervasive computing. Given the prevalence of mobile networking environments and powerful user handhelds, we consider resource constrained devices (and things, although we focus on smart, i.e., computation-enabled, things) as first-class entities of the Future Internet. Concerning middleware that enables networking mobile and/or resource constrained devices in pervasive computing environments, several promising solutions have been proposed, such as mobile Gaia, TOTA, AlfredO, or work at UCL, Carnegie Mellon University, and the University of Texas at Arlington. They address issues such as resource discovery, resource access, adaptation, context awareness as in location sensitivity, and pro-activeness in a seamless manner. Other solutions specialize in sensor networks; we, more specifically, discuss middleware for wireless sensor networks in the next section. In this very active domain of service-oriented middleware for pervasive computing environments, we have extensive expertise that covers from lower-level cross-layer networking to higher-level semantics of services, as well as transversal concerns such as context and privacy. We have in particular worked on aspects including semantic discovery and composition of services based on their functional properties, heterogeneity of service discovery protocols, and heterogeneity of network interfaces. Based on our accumulated experience, we are currently focusing on certain of the still unsolved challenges identified above.

QoS-aware service composition: With regard to service composition in pervasive environments, taking into account QoS besides functional properties ensures providing a satisfactory experience to the end user. We focus here on the orchestration-driven case, where service composition is performed to fulfill a task requested by the user along with certain QoS constraints. Assuming the availability of multiple resources in service environments, a large number of services can be found for realizing every sub-task part of a complex task. A specific issue emerges in this regard, which is about selecting the best set of services (i.e., in terms of QoS) to participate in the composition, meeting user's global QoS requirements. QoS-aware composition becomes even more challenging when it is considered in the context of dynamic service environments characterized with changing conditions. As dynamic environments call for fulfilling user requests on the fly (i.e., at runtime) and as services' availability cannot be known a priori, service selection and composition must be performed at runtime. Hence, the execution time of service selection algorithms is heavily constrained,

whereas the computational complexity of this problem is NP-hard. Several selection algorithms have been proposed to select service compositions with different composition structures and various QoS constraints. A first class of approaches aims at determining the optimal service composition (i.e., composition with the highest QoS utility) using brute-force-like algorithms (e.g., Global Planning, BBLP, MCSP, WS-IP). Such solutions have high computational cost and they cannot provide a solution in a satisfying amount of time, thus they are inappropriate to be used in the context of dynamic service environments. To cope with this issue, other approaches propose heuristic-based solutions (e.g., WS-HEU, WFlow, genetic algorithms) aiming to find near-optimal compositions, i.e., compositions that respect global QoS constraints and maximize a QoS utility function. More recently, an approach that combines local and global optimization techniques has been introduced. This approach starts from the global level and resolves the selection problem at the local level, i.e., it proceeds by decomposing global QoS constraints (i.e., imposed by the user on the whole composition) into a set of local constraints (i.e., for individual sub-tasks, parts of the whole composition).

Coordination of heterogeneous distributed systems: Another aspect that we consider important in service composition is enabling integration of services that employ different interaction/coordination paradigms. Diversity and ultra large scale of the Future Internet have a direct impact on coordination among interacting entities. Our choice of choreography as global coordination style among services should further be underpinned by support for and interoperability between heterogeneous coordination/interaction models, such as message-driven, event-driven and data-driven models. Different coordination models apply to different needs: for instance, asynchronous, event-based publish/subscribe is more appropriate for highly dynamic environments with frequent disconnections of involved entities. Enabling interoperability between such models is imperative in the very heterogeneous Future Internet integrating services, people and things. Interoperability efforts are traditionally based on, e.g., bridging communication protocols, where the dominant position is held by ESBs, wrapping systems behind standard technology interfaces, and/or providing common API abstractions. However, such efforts mostly concern a single coordination model and thus do not or only poorly address coordination model interoperability. Efforts combining together diverse coordination models include: implementing the LIME tuple space middleware on top of a publish/subscribe substrate; enabling Web services SOAP-based interactions over a tuple space binding; and providing ESB implementations based on the tuple space paradigm.

Maintenance of service oriented applications:: A third issue we are interested in concerns the maintenance of service-oriented applications despite the evolution of employed services. Services are autonomous systems that have been developed independently from each other. Moreover, dynamics of pervasive environments and the Future Internet result in services evolving independently; a service may be deployed, or un-deployed at anytime; its implementation, along with its interface may change without prior notification. In addition, there are many evolving services that offer the same functionality via different interfaces and with varying quality characteristics (e.g., performance, availability, reliability). The overall maintenance process amounts to replacing a service that no longer satisfies the requirements of the employing application with a substitute service that offers the same or a similar functionality. The goal in seamless service substitution is to relate the substitute service with the original service via concrete mappings between their operations, their inputs and outputs. Based on such mappings, it is possible to develop/generate an adapter that allows the employing application to access the substitute service without any modification in its implementation. The service substitution should be dynamic and efficient, supported by a high level of automation. The state of the art in service substitution comprises various approaches. There exist efforts, which assume that the mappings between the original and the substitute service are given, specified by the application or the service providers. The human effort required makes these approaches impractical, especially in the case of pervasive environments. On the other hand, there exist automated solutions, proposing mechanisms for the derivation of mappings. The complexity of these approaches scales up with the cardinality of available services and therefore efficiency is compromised. Again, this is an important disadvantage, especially considering the case of pervasive environments.

3.3.2. Middleware for Wireless Sensor Networks

Wireless sensor networks (WSNs) enable low cost, dense monitoring of the physical environment through collaborative computation and communication in a network of autonomous sensor nodes, and are an area of active research. Owing to the work done on system-level services such as energy-efficient medium access and data-propagation techniques, sensor networks are being deployed in the real world, with an accompanied increase in network sizes, amount of data handled, and the variety of applications. The early networked sensor systems were programmed by the scientists who designed their hardware, much like the early computers. However, the intended developer of sensor network applications is not the computer scientist, but the designer of the system *using* the sensor networks, which might be deployed in a building or a highway. We use the term *domain expert* to mean the class of individuals most likely to use WSNs – people who may have basic programming skills but lack the training required to program distributed systems. Examples of domain experts include architects, civil and environmental engineers, traffic system engineers, medical system designers etc. We believe that the wide acceptance of networked sensing is dependent on the ease-of-use experienced by the domain expert in developing applications on such systems.

The obvious solution to enable this ease-of-use in application development is sensor network middleware, along with related programming abstractions. Recent efforts in standardizing network-layer protocols for embedded devices⁹ provide a sound foundation for research and development of middleware that assist the sensor network developers in various aspects that are of interest to us, including the following.

Data-oriented operations: A large number of WSN applications are concerned with sampling and collection of data, and this has led to a large body of work to provide middleware support to the programmer of WSNs for easy access to the data generated and needed by the constituent nodes. Initial work included Hood, and TeenyLIME, which allowed data-sharing over a limited spatial range. Further work proposed the use of the DART runtime environment, which exposes the sensor network as a distributed data-store, addressable by using logical addresses such as “all nodes with temperature sensors in Room 503”, or “all fire sprinklers in the fifth and sixth floors”, which are more intuitive than say, IP addresses. Taking a different approach toward handling the data in the sensor networks, some middlewares propose to manipulate them using semantic techniques, as done by extending the tsc++ approach to Triple Space Computing, and model the data shared by the nodes in the system as RDF triples (subject-predicate-object groups), a standard method for semantic data representation. They propose to make these triples available to the participating nodes using a tuple space, thus giving it the “triple space” moniker. S-APL or Semantic-Agent Programming Language uses semantic technologies to integrate the semantic descriptions of the domain resources with the semantic prescription of agent behavior.

Integration with non-WSN nodes: Most of the work above focuses on designing applications that exhibit only intra-network interactions, where the interaction with the outside world is only in the form of sensing it, or controlling it by actuation. The act of connecting this data to other systems outside the sensor network is mostly done using an external gateway. This is then supported by middlewares that expose the sensor network as a database (e.g., TinyDB and Cougar), allowing the operator to access the data using a SQL-like syntax, augmented with keywords that can be used to specify the rate of sampling, for example. Another direction of integrating WSNs in general with larger systems such as Web servers has been toward using REST (REpresentational State Transfer) technologies, which are already used for accessing services on the Web as a lightweight alternative to SOAP. There has also been work that proposes a system that will enable heterogeneous sensor and actuators to expose their sensing and actuation capabilities in a plug and play fashion. It proposes a middleware that defines a set of constraints, support services and interaction patterns that follow the REST architectural style principles, using the ATOM Web publishing protocol for service description, and a two-step discovery process. Additionally, there has been work in implementation of a REST-oriented middleware that runs on embedded devices such as Sun SPOT nodes, and the Plogg wireless energy monitors. This involves a two-fold approach — embedding tiny Web servers in devices that can host them, and employing a proxy server in situations where that is not the case. However, it has been noticed that the abstractions provided by REST might be too simplistic to compose complex applications over the

⁹6LoWPAN: <http://tools.ietf.org/wg/6lowpan/>

services provided by WSN nodes. Some of the most recent work in this area also proposes to convert existing (network-layer) gateways into smart gateways, by running application code on them.

In addition to supporting the above interactions, sensor network middleware has also been proposed to address the challenges arising from the fact that a particular sensor or actuator may not be always available. This leads to the need for transparent reconfiguration, where the application developer should not have to care about reliability issues. The PIRATES event-based middleware for resource-rich nodes (hosting sensors/actuators, or just processing data) includes a third-party-remapping facility that can be used to remap a component's endpoints without affecting the business logic. In that sense, it is similar to the RUNES middleware targeted at embedded systems.

Finally, we also note the recent initial WSN middleware research focused on the new nascent classes of systems. Most recently, the field of *participatory sensing*¹⁰ has emerged, where the role of sensing is increasingly being performed by the mobile phones carried by the users of the system, providing data captured using the sound, GPS, accelerometer and other sensors attached to them. This has led to the emergence of middleware such as JigSaw. The core additional challenges in this domain come from the inherent mobility of the nodes, as well as their extremely large scale.

4. Application Domains

4.1. Application Domains

The ARLES project-team is interested in the application of pervasive computing, and as such considers various application domains. Indeed, 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 National projects to which we contribute (§ 7.1 & 7.3).

5. Software

5.1. Introduction

In order to validate our research results, our research activities encompass the development of related prototypes as surveyed below.

5.2. Service-oriented Middleware for Pervasive Computing

Participants: Nikolaos Georgantas [correspondent], Valérie Issarny [correspondent].

In the past years, we have built a strong foundation of service-oriented middleware to support the pervasive computing vision. This specifically takes the form of a family of middlewares, all of which have been released under the open source LGPL license:

- **WSAMI - A Middleware Based on Web Services for Ambient Intelligence:** WSAMI (Web Services for Ambient Intelligence) is based on the Web services architecture and allows for the deployment of services on wireless handheld devices like smartphones and PDAs.

¹⁰Lane, N.D.; Miluzzo, E.; Hong Lu; Peebles, D.; Choudhury, T.; Campbell, A.T.; , "A survey of mobile phone sensing," Communications Magazine, IEEE , vol.48, no.9, pp.140-150, Sept. 2010

URL: <http://www-rocq.inria.fr/arles/download/ozone/index.htm>

- **Ariadne - A Protocol for Scalable Service Discovery in MANETs:** Ariadne enriches WSAMI with the Ariadne service discovery protocol, which has been designed to support decentralized Web service discovery in multi-hop mobile ad hoc networks (MANETs). Ariadne enables small and resource-constrained mobile devices to seek and find complementary, possibly mobile, Web services needed to complete specified tasks in MANETs, while minimizing the traffic generated and tolerating intermittent connectivity.

URL: <http://www-rocq.inria.fr/arles/download/ariadne/index.html>

- **MUSDAC - A Middleware for Service Discovery and Access in Pervasive Networks:** The Multi-protocol Service Discovery and Access (MUSDAC) middleware platform enriches WSAMI so as to enable the discovery and access to services in the pervasive environment, which is viewed as a loose and dynamic composition of independent networks. MUSDAC manages the efficient dissemination of discovery requests between the different networks and relies on specific plug-ins to interact with the various middleware used by the networked services.

URL: <http://www-rocq.inria.fr/arles/download/ubisec/index.html>

- **INMIDIO - An Interoperable Middleware for Ambient Intelligence:** INMIDIO (INteroperable MiddleWare for service Discovery and service InteractiOn) 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.

URL: <http://www-rocq.inria.fr/arles/download/inmidio/index.html>

- **COCOA - A Semantic Service Middleware:** COCOA is a comprehensive approach to semantic service description, discovery, composition, adaptation and execution, which enables the integration of heterogeneous services of the pervasive environment into complex user tasks based on their abstract specification. Using COCOA, abstract user tasks are realized by dynamically composing the capabilities of services that are currently available in the environment.

URL: <http://forge.inria.fr/projects/amigo/>

- **ubiSOAP - A Service Oriented Middleware for Seamless Networking:** ubiSOAP brings multi-radio, multi-network connectivity to services through a comprehensive layered architecture: (i) the multi-radio device management and networking layers together abstract multi-radio connectivity, selecting the optimal communication link to/from nodes, according to quality parameters; (ii) the communication layer allows for SOAP-based point-to-point and group-based interactions in the pervasive network; and (iii) the middleware services layer brings advanced distributed resource management functionalities customized for the pervasive networking environment.

URL: <http://www.ist-plastic.org>.

5.3. iBICOOP: Mobile Data Management in Multi-* Networks

Participants: Valérie Issarny [correspondent], Roberto Speicys Cardoso [correspondent].

Building on the lessons learned with the development of pervasive service oriented middleware and of applications using them, we have been developing the custom iBICOOP middleware. iBICOOP specifically aims at assisting the development of advanced mobile, collaborative application services by supporting interactions between mobile users. Target application services in particular include the U-EVENT suite of services for professional events. iBICOOP is now being transferred to the AMBIENTIC spin-off (<http://www.ambientic.com/>) that is being founded.

Briefly, the iBICOOP middleware addresses the challenges of easily accessing content stored on mobile devices, and consistent data access across multiple mobile devices by targeting both fixed and mobile devices, leveraging their characteristics (e.g., always on and unlimited storage for home/enterprise servers, ad hoc communication link between mobile devices), and by leveraging the capabilities of all available networks (e.g., ad hoc networks, Internet, Telecoms infrastructure networks). It also relies on Web and Telecoms standards to promote interoperability.

The base architecture of the iBICOOP middleware consists of core modules on top of which we can develop applications that may arise in the up-coming multi-device, multi-user world.

- *Communication Manager:* The iBICOOP Communication Manager is aimed to overcome the constraints of different network characteristics that different devices may have. The Communication Manager provides mechanisms to communicate over different available network interfaces of a device — Bluetooth, WiFi, Cellular — and also using different technologies e.g., Web services, HTTP/TCP sockets, ad hoc mode. The communication between two devices is always secured with SSL.
- *Security Manager:* The iBICOOP Security Manager uses well-established techniques of cryptography and secure communication to provide necessary security. Presently, we are providing RSA, AES, DES, and Diffie-Hellman for generating keys that can be used for encryption/decryption of data for storing on device or sending over the net. The Security Manager is also responsible for access control on shared devices.
- *Partnership Manager:* The iBICOOP Partnership Manager provides device or user information in the form of *profiles*. Profiles are stored as XML files. This allows us to easily integrate our profiles with the IMS architecture using XDMS (XML Data Management Server). We have defined an XML schema for profiles. In the iBICOOP middleware, profiles are used by core modules to make a service available to the user. To make our solution privacy-aware, the Partnership Manager module provides filters to control the information exchanged in a profile.
- *Naming & Discovery:* iBICOOP's naming scheme is based on hierarchical names in the pattern of URI scheme¹¹; We combine names of protocol, devices, services, email etc. to give a human-readable name to a resource. An example for a service is: "ibic://john@work.com:ipaqpda/axis2/services/exchange". For iBICOOP, we rely on service location protocols to find nearby services on currently active network interfaces that support IP multicast. Setting up partnerships between users (bootstrapping a relationship) may also use the discovery service, and may happen after a multicast-based local discovery (e.g., over WiFi) for any Partnership Manager service, or by searching for such services on a public repository.
- *Local File Manager:* Besides normal file managing tasks, the iBICOOP Local File Manager gives the user clear cues to the files that have been replicated across multiple devices or shared among different users by using different icons. Later it will provide extra functionalities for content-sharing and data management in emerging pervasive computing environments

5.4. Srijan: Data-driven Macroprogramming for Sensor Networks

Participant: Animesh Pathak [correspondent].

Macroprogramming is an application development technique for wireless sensor networks (WSNs) where the developer specifies the behavior of the system, as opposed to that of the constituent nodes. In this research, we are working on *Srijan*, a toolkit that enables application development for WSNs in a graphical manner using data-driven macroprogramming.

It can be used in various stages of application development, *viz.*,

1. Specification of application as a task graph,
2. Customization of the auto-generated source files with domain-specific imperative code,
3. Specification of the target system structure,
4. Compilation of the macroprogram into individual customized runtimes for each constituent node of the target system, and finally
5. Deployment of the auto generated node-level code in an over-the-air manner to the nodes in the target system.

¹¹<http://tools.ietf.org/html/rfc3986>

The current implementation of *Srijan* targets both the Sun SPOT sensor nodes and larger nodes with J2SE. The software is released under open source license at <https://gforge.inria.fr/projects/srijan/>.

5.5. Yarta: Middleware for supporting Mobile Social Applications

Participant: Animesh Pathak [correspondent].

With the increased prevalence of advanced mobile devices (the so-called “smart” phones), interest has grown in *Mobile Social Ecosystems* (MSEs), where users not only access traditional Web-based social networks using their mobile devices, but are also able to use the context information provided by these devices to further enrich their interactions. We are developing a middleware framework for managing mobile social ecosystems, having a multi-layer middleware architecture consisting of modules, which will provide the needed functionalities, including:

- Extraction of social ties from context (both physical and virtual),
- Enforcement of access control to protect social data from arbitrary access,
- A rich set of MSE management functionalities, using which mobile social applications can be developed.

Our middleware adopts a graph-based model for representing social data, where nodes and arcs describe socially relevant entities and their connections. In particular, we exploit the Resource Description Framework (RDF), a basic Semantic Web standard language that allows representing and reasoning about social vocabulary, and creating an interconnected graph of socially relevant information from different sources.

The current implementation of the Yarta middleware targets both desktop/laptop nodes running Java 2 SE, as well as Android smart phones. The software is released under open source license at <https://gforge.inria.fr/projects/yarta/>.

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/pervasive computing vision. Our research activities in 2010 have focused on the following areas:

- High-level application development in sensor networks, focusing specifically on data-driven macro-programming (§ 6.2);
- Pervasive service-oriented software engineering, focusing on supporting service composition in an increasingly heterogeneous and dynamic networking environment, while enforcing quality of service (§ 6.3);
- Dynamic interoperability among networked systems toward making them eternal, by way of on-the-fly generation of connectors based on adequate system models (§ 6.4); and
- System-level support for application development in the context of mobile social ecosystems, while taking into account privacy, performance, and data interoperability (§ 6.5).

6.2. Data-driven Macroprogramming and Middleware for Sensor Networks

Participants: Iraklis Leontiadis, Pankesh Patel, Animesh Pathak.

Since their early days, WSNs have been viewed as a special class of distributed systems, and have been approached as such from an application development perspective as well. Consequently, application developers have thus far specified their applications at the level of the individual node where they use a language such as nesC, galsC, C++ or Java to write the program. This *node-level* program written by the developer can then read the values from local sensing interfaces, maintain application level state in the local memory, send messages to other nodes addressed by node ID or location, and process incoming messages from other nodes.

Owing to the large size and heterogeneity of the systems involved, as well as the limited distributed programming expertise of the domain experts, the above paradigm of node-level programming is not easy to use for sensor networks. We believe that this is a large obstacle holding back the wide-acceptance of WSNs. For example, to develop an environment management application in nesC, a commonly used language in WSNs, the developer has to specify the functions at each node in terms of the respective components - one each for sensing the environment, communicating with other nodes, as well as controlling the actuators attached to the node. In addition to the above, the application developer is also responsible for ensuring that the distributed application that results from these communicating node-level programs performs the necessary functions as desired, and is also efficient in terms of the energy spent during its operations.

Sensor network *macroprogramming* aims to aid the wide adoption of networked sensing by providing the domain expert the ability to specify their applications at a high level of abstraction. In macroprogramming, abstractions are provided to specify the high-level collaborative behavior at the system level, while intentionally hiding most of the low-level details concerning state maintenance or message passing from the programmer. As a result of this, macroprogramming is emerging as a viable technique for developing complex embedded applications, as demonstrated by the several efforts currently underway in this field.

In the past year, we have worked on the following research topics in the area of macroprogramming abstractions for sensor networks and middleware to support them:

Modeling and supporting new interaction patterns: We expect future networked sensing systems to not only consist of isolated islands; instead the nodes in-charge of sensing will actively interact with their (larger) peers such as Web and database servers, and also provide control and access directly to the users via the use of mobile devices such as smart phones. This rich variety of nodes leads to the need to include adequate models of new interaction patterns such as request-response in macroprogramming frameworks. We have extended the ATaG macroprogramming language to adequately support these new behaviors by introducing new types of primitives in the language.

System-level support for heterogeneous WSN nodes: The new interaction patterns discussed above are closely linked with the increased heterogeneity among the nodes used in these systems. We have been working on updating the DART runtime system, which now successfully provides a uniform layer above various types of platforms, including mobile systems such as Java ME, desktop-class systems, as well as mobile operating systems such as Google's Android. Another area of research in this domain has been toward exploiting the higher computation power and better network links of these nodes to enable a two-level routing protocol for the system.

We have incorporated our continued research in the above areas into *Srijan* (§ 5.4), which provides an easy-to-use graphical front-end to the various steps involved in developing an application using the ATaG macroprogramming framework. The *Srijan* toolkit is supported by a modular compilation framework [20] and task-mapping algorithms [7] designed especially for data-driven macroprogramming.

6.3. Pervasive Service Oriented Software Engineering

Participants: Mohammad Ashiqur Rahaman, Hamid Ameziani, Dionysis Athanasopoulos, Sandrine Beauche, Nebil Ben Mabrouk, Nikolaos Georgantas, Sneha-Sham Godbole, Valérie Issarny.

Over the last years, service oriented architecture has been identified as a key enabler for pervasive computing. Moreover, the world is getting closer and closer to the realization of the ubiquitous vision of computing with the boundaries between networks getting blurred and the Future Internet aiming to encompass networked resources, people and things; service orientation then proves to be a unifying element. Additionally, enabling service composition in its more generic form, as in service choreographies, seems to be a promising and challenging approach. Building on our long expertise on middleware solutions for service-based pervasive environments, we focused in the year 2010 on three aspects of pervasive middleware that make part of and provide preliminary solutions to the above vision:

- QoS-aware service composition, for providing a satisfactory experience to the end user, where we cover the specific case of service orchestration,
- Composition of services that employ different interaction/coordination paradigms, where we extend the viewpoint of only synchronous interaction based services, again in the context of service orchestration, and
- Maintenance of service-oriented applications despite the evolution of employed services, where we propose a solution to individual service substitution, which can be later extended in the context of complex service compositions.

QoS-aware service composition: We investigate a service-oriented middleware-based solution addressing QoS concerns in pervasive computing. The proposed solution focuses on QoS-aware service composition as an approach allowing to fulfill users' required tasks in pervasive environments while meeting QoS requisites. It includes: (i) a semantic end-to-end QoS model enabling the specification of QoS-aware composition, (ii) a QoS-aware service selection algorithm enabling the construction of such compositions, and (iii) a reconfiguration approach allowing to cope with QoS dynamics at runtime.

Addressing QoS-aware service composition requires defining a QoS model which provides the appropriate ground for QoS specification and provision in pervasive environments. Toward this purpose, we proposed in 2009 a semantic QoS model that addresses QoS on an end-to-end basis by covering QoS properties of the main elements acting in pervasive environments; in particular, it deals with network and hardware resources, application services and end-users. Our model is designed according to a layered approach integrating four QoS ontologies: (1) QoS Core ontology, (2) Infrastructure QoS ontology, (3) Service QoS ontology, and (4) User QoS ontology. Based on our design, we provided in 2010 a complete implementation of the QoS model in OWL-DL 2.0.

We exploit our QoS model to trigger QoS-aware service selection, which is a key requirement in pervasive environments, since it enables selecting services able to fulfill complex user tasks while meeting their QoS requirements. A challenging issue toward this purpose is to determine the best set of services to compose, while meeting global QoS constraints imposed by the user, which is traditionally known to be NP-hard. Addressing this problem in the context of pervasive environments is a challenging task. Two specific issues arise to this regard. First, pervasive environments advocates fulfilling user required functionalities on the fly, thus the time available for services' selection and composition is limited compared to the computational complexity of the problem. Second, existing solutions to QoS-aware service composition employ centralized infrastructures with rich computational resources for performing service selection. However, in pervasive environments it is not always possible to assume such infrastructures. Thus, selection algorithms should be conceived in a distributed fashion enabling them to execute on resource-constrained devices. To cope with the above issues, we present an efficient service selection algorithm for pervasive environments. The proposed algorithm is the fruitful result of our extensive research investigating clustering techniques (i.e., namely the K-means algorithm) for service selection. It is conceived as a heuristic which uses K-means to classify services with respect to their QoS and determine the best services able to meet user QoS requirements. We further improved our approach by guiding in a second step, based on this classification, a global selection algorithm that reinforces the accuracy of the composition results. Furthermore, we derived a distributed version of the algorithm, which is able to execute on infrastructure-less pervasive environments.

Nevertheless, addressing QoS awareness during the construction of service compositions is not enough to steadily ensure a satisfying level of QoS, since QoS can change dynamically at runtime. To this regard, QoS-aware compositions have to be adapted according to QoS fluctuations. In the case of a service failure, we apply an adaptation approach which consists in reselecting services for substituting the failed one, and partially applying the global selection on the still unexecuted part of the task.

We implemented the aforementioned research results and integrated our prototype components in a complete service-oriented architecture, in the context of the SemEUsE project, which includes support for semantic service discovery, dynamic QoS-aware composition and deployment of service orchestrations, runtime QoS monitoring based on Service Level Agreements, and QoS-aware runtime reconfiguration of orchestrations.

Coordination of heterogeneous distributed systems: In the near future, complex distributed applications will be to a large extent based on the integration of extremely heterogeneous systems, such as lightweight embedded systems (e.g., sensors, actuators and networks of them), mobile systems (e.g., smartphone applications), and resource-rich IT systems (e.g., systems hosted on enterprise servers and Grid infrastructure). These heterogeneous system domains develop individually, with significant advancements in terms of coordination styles (e.g., message-, event-, data-driven), communication protocols, and data representation models, as well as related middleware paradigms (e.g., client/server - CS, publish/subscribe - PS, tuple space - TS middleware). These three middleware paradigms and homonymous coordination models are the most widely employed ones. To enable complex distributed applications, the heterogeneity between the involved system domains needs to be tackled. Cross-domain interoperability efforts are then based on, e.g., bridging communication protocols, wrapping systems behind standard technology interfaces, and/or providing common API abstractions. However, such efforts commonly cover part of the heterogeneity issues (regarding coordination, communication, data) and are applicable to specific cases. In particular, existing solutions do not or only poorly address coordination model interoperability.

We have been working on a system integration approach that is sufficiently general to deal with very diverse existing systems, while focusing in particular on integrating heterogeneous coordination models. Firstly, our integration solution features workflow-based orchestration of the constituent systems. A number of models, languages and associated support platforms have been proposed for designing and executing orchestrations, however, focusing almost exclusively on service-oriented systems, i.e., assuming synchronous coordination. Envisaging heterogeneous orchestrations, we extend the workflow-based orchestration model to cover diverse coordination models. However, this results into additional complexity and effort for the application workflow designer, who will be required to deal in the workflow with the union of all the heterogeneous coordination primitives of the constituent systems. Thus, secondly, our integration solution introduces a single generic application (GA) coordination model to abstract from the underlying heterogeneous coordination models of the constituent systems, while paying special attention to preserving their coordination and data semantics. Our systematic abstraction approach is carried out in two stages. First, a middleware technology is abstracted under a corresponding middleware coordination model among the three principal ones, i.e., CS, PS and TS. To this end, we have elicited generic coordination primitives for each model comprehensively covering their essential semantics, after having thoroughly surveyed the related literature. Then, these three models are abstracted into the GA coordination model. We have introduced a set of high-level generic coordination primitives for the GA model and mappings between them and the primitives of each one of CS, PS and TS. This results into a decoupling between application- and middleware-level coordination in addition to the preservation of semantics, thus enabling orchestrations that are agnostic to the underlying middleware technologies of the heterogeneous constituent systems.

The above concepts are incorporated into an abstract layered coordination middleware architecture, which we implement into a prototype extensible coordination middleware by building upon BPEL and its extensibility support mechanism. We further explicitly identify the roles of the application and middleware designers and provide them with tool-assisted design-time support: enabling the former to easily design application workflows integrating heterogeneous constituent systems; and the latter to rapidly add support for new middleware technologies into the prototype coordination middleware. We have evaluated our approach by

applying design time metrics for measuring this support. Our solution considerably reduces efforts of both designers in dealing with multiple heterogeneous middleware technologies.

Maintenance of service oriented applications: Service substitution is a key issue toward dealing with the independent evolution of services along with their variation in quality (e.g., performance, availability, reliability). Research efforts that focus on service substitution can be divided in two categories. The first category consists of *abstraction-based* approaches, which propose development methodologies and frameworks that allow the development from scratch of client applications that use service abstractions that are mapped into alternative concrete services. The second category comprises *adapter-based approaches*, which deal with existing client applications that use concrete services. The basic concept in this case is to derive a *mapping* between the target service that should be substituted and a *substitute service* that offers similar functionality through a different interface. Based on such a mapping, an adapter is generated, which allows accessing the functionality of the substitute service through the original target interface, without modifying the client application code.

While considering adapter-based approaches the following issue is raised: the effort and time required by the service substitution process scale up with the number of available services that should be examined as potential candidate substitutes of the target service. This problem is a serious drawback toward a practical service substitution approach, if we consider that the service substitution process involves human intervention to validate the mapping between target and substitute services. Our work shares the objective of adapter-based approaches. However, our specific goal is to reduce the effort and time required to achieve this objective.

To this end, we investigated the issue of systematically recovering service abstraction definitions, along with mappings between the abstractions' unified interfaces and the interfaces of the represented alternative services. We further proposed an approach for the recovery of service abstractions [23] out of sets of available services that play the role of alternative design-decisions, which can be used in a service-oriented application. A service abstraction provides a uniform interface that hides differences in the interfaces of alternative services and consequently allows reducing the coupling between the application and the services. To this end, we *formally defined the notion of service abstraction* and proposed a hierarchical clustering algorithm that incrementally recovers a hierarchy of service abstractions out of a given set of alternative design-decisions/services. Finally, we evaluated the proposed algorithm with real-world sets of services.

6.4. Dynamic Synthesis of Middleware Connectors

Participants: Amel Bennaceur, Nikolaos Georgantas, Valérie Issarny, Rachid Saadi, Daniel Sykes.

As networked systems are becoming increasingly pervasive, they need to compose dynamically with their ever evolving environment according to functionalities they provide and/or request. However, such dynamic composition is greatly challenged by the heterogeneity and autonomy of today's digital systems, which are not designed in concert, but are instead independently developed and deployed within pervasive networking environments. As a result, although networked systems may possibly match from the standpoint of provided and required functionalities, actual behavioral matching is unlikely due to inherent design diversity. Therefore, what is needed for enabling the composition of pervasive networked systems is *emergent connectors*, which are synthesized on the fly and embed a mediation process so as to adapt the systems' respective interaction behaviors for the sake of coordination.

The notion of *mediator* underlying emergent connectors is not new. It has indeed been investigated since the need for interoperability in distributed systems was identified. However, this was initially a design-time concern whereas today's dynamic distributed systems require on-the-fly adaptation. On-the-fly protocol adaptation has in particular been studied quite extensively in the context of Web services to deal with either dynamic service composition or substitution. Still, existing work on the runtime synthesis of mediators concentrates on application-layer protocols; while the heterogeneity of open networked systems may concern both the application and middleware layers. Indeed, a one-size-fits-all middleware cannot be assumed. Middleware solutions will keep emerging according to specific requirements coming from application domains and/or networking environments. In addition, most of the existing solutions to middleware interoperability rely upon the design-time choice to develop applications using the proposed interoperability platform; and thus do not allow for on-the-fly interoperability between networked applications embedding different

legacy middleware. Middleware interoperability further needs to cope with the many middleware interaction paradigms that now need to coexist. This includes accessing the same functionality through distinct paradigms (e.g., context-awareness through access to a data-centric sensor network or a RPC-based context server).

Our research on enabling emergent connectors is done in collaboration with our partners of the CONNECT project (§ 7.1.1). Our work during the year was more specifically focused on:

- *A theory of mediators for the ubiquitous networking environment* [13]. The proposed formal model: (i) precisely characterizes the problem of interoperability between networked systems, and (ii) paves the way for automated reasoning about protocol matching (interoperability) and related mediator synthesis. Using our theory, we achieve automated and on-the-fly mediation between behaviorally mismatching, yet functionally matching application-layer interaction protocols.
- *A model-based approach enabling emergent connectors*, which builds upon the above theory of mediators. Emergent connectors specifically adapt the protocols run by networked systems implementing a matching functionality, but possibly mismatching from the standpoint of associated application protocol and even middleware technology used for interactions. The enabling of emergent connectors then lies in the appropriate modeling of the networked systems' high-level functionalities and related protocols, for which we exploit ontology so as to enable unambiguous specification. This then allows us to introduce a rigorous definition of protocol matching and adaptation, together with related algorithms, supporting on-the-fly synthesis of emergent connectors for the sake of interoperability. Compared to related work that deals with either automated protocol conversion/mediation or middleware interoperability, our contributions lies in comprehensively dealing with both the application and middleware layers. Furthermore, through the alignment of middleware concepts, we are able to deal with interoperability between networked systems relying on heterogeneous middleware paradigms.
- *Dependability of emergent connectors*. We have investigated dependability with respect to the threats deriving from on-the-fly synthesis of emergent connectors [11], leading to the study of appropriate means for assessing/guaranteeing that the networked systems connected via emergent connectors yield acceptable levels for different non-functional properties. This has in particular led us to study the integration of our automated technique for the synthesis of mediators with a monitoring mechanism [12]. The mediators provide functional interoperability and the monitors make it possible to assess the non-functional characteristics of the connected systems at runtime that cannot be assessed statically at synthesis time.
- *Supporting architecture*. Finally, we have been working on the definition of an overall architecture supporting emergent connectors, from the discovery of networked systems to the learning of their respective behavior, and synthesis of emergent connectors enabling them to interoperate [10].

6.5. Middleware Support for Mobile Social Applications

Participants: Sara Hachem, Valérie Issarny, Animesh Pathak, Amir Seyedi, Alessandra Toninelli.

With the increased prevalence of advanced mobile devices (the so-called “smart” phones), interest has grown in *Mobile Social Ecosystems* (MSE), where users not only access traditional on line Web-based social networks using their mobile devices, but are also able to use the context information provided by these devices to further enrich their interactions. In complex mobile social ecosystems of the future, the heterogeneity of software platforms on constituent nodes, combined with their intrinsic distributed nature and heterogeneity in representation of data and context, as well as user’s privacy and trust concerns, raises the need for middleware support for the development of mobile social applications [15]. We believe that the development of mobile social applications can be greatly simplified by the presence of middleware support. To that end, we have been working in the directions below.

Semantic models for mobile social ecosystems: In order to enable re-use of data between different social applications run by the same user, we have proposed an expressive and extensible model using semantic techniques to represent MSE and the interactions possible in them. This supports semantic interoperability between separately developed applications and minimizes resource-consuming operations such as data mapping and replication.

Efficient decentralized storage of social data: Instead of storing the social knowledge of the whole world with a single provider — a practice performed today by common social networks such as Facebook — which can lead to privacy issues, our research endeavors to propose a middleware using which users can store their personal knowledge in a distributed manner on the devices owned by them (e.g., smart phone, home desktop, laptop). This also allows users to provide selective access to other users based on semantically defined access control policies.

Socially aware policies for access control: Since social data is private and sensitive in nature, we have proposed a policy framework where the user can specify both the data to be protected as well as the relevant set of peers with access to that data in a socially-aware manner (e.g., “only let my colleagues know my location during weekdays from 9 – 5”). This policy framework can be used as a guard around the user’s knowledge base, allowing access only to authorized peers. We are also working on providing end-users an easy to use editor so as to be able to specify these socially-aware policies easily.

Social data extraction from existing sources: Our research includes work in enabling users to populate their social knowledge base by extracting data from their existing repositories. We have identified two types of sources of such data. The first already contain social links such as “friendship” in addition to general information, while the second do not contain social links, but may contain information which can be correlated to infer social links (e.g., call and SMS logs). We are working on a framework where adapters can be written for the former using their API to import their data; while for the latter, inference algorithms can be used to correlate data and guess/recommend social links.

We have incorporated our research in the above areas into Yarta, a middleware for mobile social applications. Our prototype middleware, as discussed in § 5.5, currently supports application development for laptops as well as Android-powered smart phones, providing distributed storage of semantically-modeled social knowledge guarded by a rich policy framework.

7. Other Grants and Activities

7.1. European Contracts and Grants

7.1.1. FP7 ICT FET CONNECT

Participants: Amel Bennaceur, Nikolaos Georgantas, Valérie Issarny, Rachid Saadi, Daniel Sykes.

- **Name:** CONNECT – *Emergent Connectors for Eternal Software Intensive Networked Systems*
- **URL:** <http://www.connect-forever.eu/>
- **Related activities:** § 6.4
- **Period:** [February 2009 - July 2012]
- **Partners:** INRIA (CRI Paris-Rocquencourt) [**project coordinator**], CNR (Italy), DoCoMo (Germany), Lancaster University (UK), Thales Communications SA (France), Università degli Studi L’Aquila (Italy), Technische Universität Dortmund (Germany), University of Oxford (UK), Uppsala Universitet (Sweden), Peking University (China).

The CONNECT Integrated Project aims at enabling continuous composition of networked systems to respond to the evolution of functionalities provided to and required from the networked environment. At present the efficacy of integrating and composing networked systems depends on the level of interoperability of the systems's underlying technologies. However, interoperable middleware cannot cover the ever growing heterogeneity dimensions of the networked environment. CONNECT aims at dropping the interoperability barrier by adopting a revolutionary approach to the seamless networking of digital systems, that is, synthesizing on the fly the connectors via which networked systems communicate. The resulting emergent connectors are effectively synthesized according to the behavioral semantics of application- down to middleware-layer protocols run by the interacting parties. The synthesis process is based on a formal foundation for connectors, which allows learning, reasoning about and adapting the interaction behavior of networked systems at run-time. Synthesized connectors are concrete emergent system entities that are dependable, unobtrusive, and evolvable, while not compromising the quality of software applications. To reach these objectives the CONNECT project undertakes interdisciplinary research in the areas of behavior learning, formal methods, semantic services, software engineering, dependability, and middleware. Specifically, CONNECT will investigate the following issues and related challenges: (i) Modeling and reasoning about peer system functionalities, (ii) Modeling and reasoning about connector behaviors, (iii) Runtime synthesis of connectors, (iv) Learning connector behaviors, (v) Dependability assurance, and (vi) System architecture. The effectiveness of CONNECT research will be assessed by experimenting in the field of wide area, highly heterogeneous systems where today's solutions to interoperability already fall short (e.g., systems of systems).

7.1.2. FP7 ICT IP CHOReOS

Participants: Sandrine Beauche, Nikolaos Georgantas, Sara Hachem, Valérie Issarny, Animesh Pathak.

- **Name:** CHOReOS – *Large Scale Choreographies for the Future Internet*
- **URL:** <http://www.choreos.eu/>
- **Related activities:** § 6.3
- **Period:** [February October 2010 - September 2013]
- **Partners:** BPI (Lithuania), CEFRIEL (Italy), CNR (Italy), eBM WebSourcing S.A.S (France), INRIA (CRI Paris-Rocquencourt), MLS Multimedia A.E. (Greece), OW2 Consortium, Thales Communications S.A. (France) [**coordinator**], The City University, London (UK), Università degli Studi dell'Aquila (Italy), Universidade de São Paulo (Brazil), University of Ioannina (Greece), SSII VIA (Latvia), Virtual Trip Ltd. (Greece), Wind Telecomunicazioni S.p.A (Italy).

CHOReOS aims at assisting the engineering of software service compositions in the revolutionary networking environment created by the Future Internet. Indeed, sustaining service composition and moving it closer to the end users in the Future Internet is a prime requirement to ensure that the wealth of networked services will get appropriately leveraged and reused. This again stresses the required move from static to dynamic development, effectively calling for adequate support for service reuse; much like software reuse has been a central concern in software engineering over the last two decades. This is why CHOReOS adopts the Service Oriented Computing (SOC) paradigm, where networked resources are abstracted as services so as to ease their discovery, access and composition, and thus reuse. However, although latest advances in the SOC domain enable facing (at least partly) the requirements of today's Internet and related networking capabilities, engineering service compositions in the light of the Future Internet challenges — in particular the ultra large scale (ULS) on all imaginable dimensions as well as the evolution of the development process from a mostly static process to a dynamic user-centric one — is far from adequately addressed. Therefore, the CHOReOS goal is to address these challenges by devising a dynamic development process, and associated methods, tools and middleware, to sustain the composition of services in the Future Internet.

7.1.3. FP7 ICT NoE NESSoS

Participant: Valérie Issarny.

- **Name:** NESSoS – *Network of Excellence on Engineering Secure Future Internet Software Services and Systems*

- **URL:** <http://www.nessos-project.eu>
- **Related activities:** § 6.4, § 6.3
- **Period:** [October 2010 - March 2013]
- **Partners:** Atos Origin (Spain), CNR (Italy) [**coordinators**], ETH Zürich (Switzerland), IMDEA Software (Spain), INRIA (EPI ARLES, CASSIS, and TRISKELL), KU Leuven (Belgium), LMU München (Germany), Siemens AG (Germany), SINTEF (Norway), University Duisburg-Essen (Germany), Universidad de Malaga (Spain), Università degli studi di Trento (Italy).

The Network of Excellence on Engineering Secure Future Internet Software Services and Systems (NESSoS) aims at constituting and integrating a long lasting research community on engineering secure software-based services and systems. The NESSoS engineering of secure software services is based on the principle of addressing security concerns from the very beginning in system analysis and design, thus contributing to reduce the amount of system and service vulnerabilities and enabling the systematic treatment of security needs through the engineering process. In light of the unique security requirements the Future Internet will expose, new results will be achieved by means of an integrated research, as to improve the necessary assurance level and to address risk and cost during the software development cycle in order to prioritize and manage investments. NESSoS will integrate the research labs involved; NESSoS will re-address, integrate, harmonize and foster the research activities in the necessary areas, and will increase and spread the research excellence. NESSoS will also impact training and education activities in Europe to grow a new generation of skilled researchers and practitioners in the area. NESSoS will collaborate with industrial stakeholders to improve the industry best practices and support a rapid growth of software-based service systems in the Future Internet.

7.1.4. FP7 ICT CA *EternalS*

Participants: Amel Bennaceur, Valérie Issarny, Animesh Pathak.

- **Name:** *EternalS – Trustworthy Eternal Systems via Evolving Software, Data and Knowledge*
- **URL:** <http://www.eternals.eu>
- **Related activities:** § 6.4
- **Period:** [March 2010 - February 2013]
- **Partners:** INRIA (CRI Paris-Rocquencourt), KU Leuven (Belgium), Queen Mary University (UK), University of Chalmers (Sweden), University of Trento (Italy), Waterford Institute of Technology (Ireland).

Latest research work within ICT has allowed to pinpoint the most important and urgently required features that future systems should possess to meet users' needs. Accordingly, methods making systems capable of adapting to changes in user requirements and application domains have been pointed out as key research areas. Adaptation and evolution depend on several dimensions, e.g., time, location, and security conditions, expressing the diversity of the context in which systems operate. A design based on an effective management of these dimensions constitutes a remarkable step toward the realization of Trustworthy Eternal Systems. The EternalS Coordination Action specifically aims at coordinating research in that area based on a researcher Task Force together with community building activities, where the organization of large workshops and conferences is just one of the tools that will be used to conduct a successful CA.

7.2. International Research Networks and Work Groups

7.2.1. *ForeverSOA Associated Team*

- **Name:** *ForeverSOA – A rigorous approach to the evolution of service-oriented software*
- **URL:** <http://dmod.cs.uoi.gr/ForeverSOA/index.htm>
- **Related activities:** § 6.3
- **Period:** [Created 2009]

- **Participants:** Joint team with University of Ioannina (UoI), Department of Computer Science, Greece.

This objective of the team is to study a principled approach for the dynamic maintenance of service-oriented software (i.e., software that is built by composing available services) on the basis of fundamental design principles and middleware that supports their adoption. The need for maintaining service-oriented software may be triggered by changes in the quality requirements of the end-users of service-oriented software (e.g., performance, availability, reliability), or by the the independent evolution of constituent services (e.g., services may be deployed or undeployed at anytime). Specifically, the proposed approach shall consist of:

- A systematic approach that allows to reverse engineer service abstractions out of a given set of available services. A service abstraction shall group concrete services that provide similar functionalities through different interfaces and with different quality characteristics.
- A service discovery middleware that embeds the abovementioned approach and further enables service providers to register available services and service users to discover services exposed as reverse-engineered service abstractions.
- Middleware poly-proxies that serve as local representatives of reverse engineered service abstractions on the side of the service-oriented software that uses them. Poly-proxies shall be dynamically customizable middleware elements that map the changing quality requirements (e.g., performance, reliability, availability) of the end-users of the service-oriented software to the concrete services that can fulfill these requirements. The end-users' quality requirements shall serve as input rules to the dynamically customizable poly-proxies, which will rely on online quality monitoring toward performing the selection of the appropriate concrete services. A second critical property of the poly-proxy middleware elements shall be the ability to dynamically upgrade them with respect to the evolving availability of the concrete services that are registered in / unregistered from the service-discovery mechanism.

7.2.2. ERCIM WG SERENE

- **Name:** ERCIM Working Group – *Software EngineeRing for rEsilieNt systEms*
- **URL:** <http://serene.uni.lu/tiki/tiki-index.php>
- **Period:** [Created 2004]
- **Participants:** Aabo Akademi (Finland), Birkbeck College (UK), BUTE (Hungary), CNR (Italy), CWI (The Netherlands), FNR (Luxembourg), FORTH (Greece), Fraunhofer FOKUS & IPSI (Germany), INRIA (CRI Paris-Rocquencourt), LAAS-CNRS (France), National Aerospace University (Ukraine), Nokia Research (Finland), Polit. di Milano (Italy), Poznan University of Technology (Poland), NTNU (Norway), SARIT (Switzerland), SpaRCIM (Spain), SZTAKI (Hungary), University of L'Aquila (Italy), University Mc Gill (Canada), University Mc Master (Canada), University of Florence (Italy), University of Ioannina (Greece), University of Groningen (The Netherlands), University of Newcastle (UK), University Roma Tor Vergata (Italy), VTT (Finland).

SERENE considers resilient systems as open and distributed systems that can dynamically adapt in a predictable way to unexpected events. Engineering such systems is a challenging issue still not solved. Achieving this objective is a very complex task since it implies reasoning explicitly and in a combined way, on system's functional and non-functional characteristics.

SERENE advocates that resilience should be explicitly included into traditional software engineering theories and practices and should become an integral part of all steps of software development. As current software engineering practices tend to capture only normal behavior, assuming that all abnormal situations can be removed during development, new software engineering methods and tools need to be developed to support explicit handling of abnormal situations. Moreover, every phase in the software development process needs to be enriched with phase specific resilience means.

7.2.3. 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 (CRI Paris-Rocquencourt & Sophia Antipolis), IUC (Ireland), L3S (Germany), Marasyk University (Czech Republic), Microsoft EMIC (Germany), NTNU (Norway), Politecnico Torino (Italy), SAP (Germany), SARIT (Switzerland), SICS (Sweden), Siemens Corporate Technology, SparCIM (Spain), SZTAKI (Hungary), VTT (Finland), Eindhoven University of Technology (The Netherlands), University of Milan (Italy), University of Roma Tor Vergata (Italy), University of Trento (Italy), University of Twente (The Netherlands), VCPC, W3C.

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

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

7.3. National Contacts and Grants

7.3.1. ANR SemEUsE: SEMantiquE pour bUS de sErvice

Participants: Sandrine Beauche, Nebil Ben Mabrouk, Nikolaos Georgantas.

- **Name:** SEMEUSE – *SEMantiquE pour bUS de sErvice*
- **Related activities:** § 6.3
- **Period:** [December 2007 – May 2010]
- **Partners:** Thales [**coordinator**], France Télécom R&D, EBM WebSourcing, Université Pierre et Marie Curie - LIP6/MoVe, INSA Lyon, INRIA, Télécom Sud Paris.

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.

7.3.2. ANR ITeMIS: IT and Embedded Integrated Systems

Participants: Hamid Ameziani, Mohammad Ashiqur Rahaman, Nikolaos Georgantas.

- **Name:** ITeMIS – *IT and Embedded Integrated Systems*
- **Related activities:** § 6.3
- **Period:** [January 2009 – December 2011]
- **Partners:** Thales Communications S.A, EBM Websourcing, INRIA ARLES, INRIA ADAM, LAAS - CNRS, ScalAgent, IRIT.

Service-Oriented Architecture (SOA), as a key architectural pattern for prompt and rapid integration, is today a cornerstone of the agile Information Technology (IT) wave. Indeed, most of today's greatest successes, in terms of bringing agility to the whole enterprise through its IT backbone, have been provided by SOA and its major technological counterparts that are the Web Services and the Enterprise Service Bus (ESB). At the same time, large control and command systems are envisaged, which may roughly be described as net-centric assemblies of heterogeneous lightweight sensors and actuators along with several large control systems. To accomplish such systems, there is currently a strong need of techniques at the cutting edge of technology that could bring seamless integration and deployment of lightweight embedded applications and IT services in a global agile system of services. In this context, ITeMIS aims at easing the evolution from today's world of separate lightweight embedded applications and IT services to the future world of seamlessly integrated services, thus qualifying and defining a new generation SOA enabling IT and Embedded Integrated Systems (ITeMIS systems).

7.3.3. INRIA D2T Action de Développement Technologique Srijan

Participants: Animesh Pathak, Iraklis Leontiadis.

- **Name:** *Srijan – Data-driven Macroprogramming for Heterogeneous Sensor*
- **Related activities:** § 6.2, § 5.4
- **Period:** [October 2009 – September 2011]
- **Partners:** INRIA (CRI Paris-Rocquencourt, EPI ARLES)

Macroprogramming is an application development technique for wireless sensor networks (WSNs) where the developer specifies the behavior of the system, as opposed to that of the constituent nodes. In this research, we are working on *Srijan*, a toolkit that enables application development for WSNs in a graphical manner using data-driven macroprogramming, including specification of application as a task graph, customization of the auto-generated source files with domain-specific imperative code, specification of the target system structure, and compilation and deployment of the macroprogram into individual customized runtimes for each constituent node of the target system.

7.3.4. INRIA DTI Action de Transfert iBICOOP

Participants: Valérie Issarny, Roberto Speicys Cardoso.

- **Name:** *iBICOOP – Middleware for mobile collaborative applications*
- **Related activities:** § 5.3
- **Period:** [2010]
- **Partners:** INRIA (CRI Paris-Rocquencourt, EPI ARLES)

The *Action de transfert* iBICOOP supports the development of the iBICOOP middleware towards its transfer to industry and more specifically its exploitation by the AMBIENTIC spin-off for the development of innovative, mobile collaborative services.

8. Dissemination

8.1. Involvement within the Scientific Community

8.1.1. Programme Committees of Conferences and Workshops

- Nikolaos Georgantas is PC member of the DAIS '10, ICSOFT '10 and AmI '10 international conferences;
- Nikolaos Georgantas is PC member of the RoSOC-M '10, M-MPAC '10 and MW4SOC '10 international workshops;

- Valérie Issarny is PC member of the COMSWARE '11, ESEC/FSE '11, FASE '11, FMOODS '10 & '11, ICDCS '11, ICSOC '10, IFIPTM '10 & '11, ISARCS '10 & '11, Middleware '10, ServiceWave '10, Mobile track at ESWC '11, Demonstration track at ICSE '11 international conferences;
- Valérie Issarny is PC member of the CAMPUS '10, EPEW '10, & MW4SOC '10 international workshops;
- Animesh Pathak is PC member of HiPC '10, ISSNIP '10, ICPADS '10, S-Cube '10, and SECON '10 & '11 international conferences;
- Alessandra Toninelli is PC chair of the Mobile Web track at ESWC '11 conference; and
- Alessandra Toninelli is PC member of MeSSa '10, SISS '11, and SocialComNet '11 workshops.

8.1.2. Leadership Services in Academic Events and Edited Journals

- Nikolaos Georgantas is Workshops Chair of the AmI '10 international conference;
- Nikolaos Georgantas is associate editor of the International Journal of Ambient Computing and Intelligence (IJACI);
- Valérie Issarny is member of the Steering Committee of the Middleware and ESEC/FSE conferences;
- Valérie Issarny is associate editor the of ACM Computing Surveys; and
- Animesh Pathak is the Production Chair of HiPC 2010 & 2011, Web Publicity Chair of DCOSS 2010, and Web Publicity chair of COMSWARE 2011.

8.1.3. Other Academic Services

- Nikolaos Georgantas is member of the PhD monitoring committee at INRIA Paris-Rocquencourt;
- Nikolaos Georgantas is member of the organizing committee of the monthly colloquium at INRIA Paris-Rocquencourt;
- Valérie Issarny is coordinator of the EC FP7 FET IP CONNECT;
- Valérie Issarny is scientific leader of the EC FP7 IP CHOReOS;
- Valérie Issarny is expert for the EC FP7 ICT work programme, and ARTEMIS Joint Undertaking;
- Valérie Issarny is member of the INRETS scientific council & "Commission d'évaluation des chercheurs";
- Valérie Issary is member of the GDR GPL scientific council;
- Valérie Issarny is member of the evaluation committee of the ANR "Programme Blanc" and "Programme Jeunes Chercheuses et Jeunes Chercheurs" for the 2010 & 2011 call; and
- Valérie Issarny was invited to deliver keynote talks at Vidas'10 workshop & SC'10 international conference.

8.2. Teaching

- Valérie Issarny gives a 12 hours lecture on "Pervasive Service Oriented Computing" as part of the SWAS lecture of the Master 2 COSY of the University of Versailles Saint-Quentin en Yvelines.

8.3. Internships

During the year 2010, members of the ARLES project-team supervised the work of the following student interns:

- Ajay CHHATWAL, *Designing an application development environment for sensing-and-actuation augmented pervasive systems*, Integrated M.Tech Dual-degree program, Institute of Technology, Banaras Hindu University, India.
- Mohamed Larbi ELHANI, *Design and realization of a generic trust simulator for distributed systems*, M2R, University of Versailles St. Quentin-en-Yvelines.
- Fatma GHACHEM, *Développer et intégrer de nouveaux services d'infrastructure pour faciliter les interactions entre les terminaux mobiles*, M2, ESPRIT Ecole Supérieure Privée D'Ingénierie et de Technologie, Tunisia.
- Sara HACHEM, *Développement d'une architecture middleware pour le contrôle d'accès aux ressources distribuées dans un réseau mobile pair-à-pair*, M2R, University of Versailles St. Quentin-en-Yvelines.
- Nishant KUMAR, *Middleware for managing mobile social ecosystems*, Bachelor of Technology, Institute of Technology, Banaras Hindu University, India.
- Shashank TYAGI, *Ontology based interoperability of instant messaging applications*, Integrated M.Tech Dual-degree program, Institute of Technology, Banaras Hindu University, India.

8.4. Awards

- The AMBIENTIC spin-off project, led by Pierre-Guillaume Raverdy, formerly ARLES team member, is winner of the *Concours national d'aide à la création d'entreprises de technologies innovantes* award in the *Création* category (see <http://www.inria.fr/innovation/actualites/cinq-projets-de-creations-d-entreprises-de-l-inria-primex> to know more).

9. Bibliography

Major publications by the team in recent years

- [1] 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", 2008, vol. 81, n^o 5, p. 785-808.
- [2] M. CAPORUSCIO, P.-G. RAVERDY, V. ISSARNY. *ubiSOAP: A Service Oriented Middleware for Ubiquitous Networking*, in "IEEE Transactions on Services Computing", 2010, accepted, <http://hal.inria.fr/inria-00519577>.
- [3] D. CHARLET, V. ISSARNY, R. CHIBOUT. *Energy-efficient middleware-layer multi-radio networking: An assessment in the area of service discovery*, in "Comput. Netw.", 2008, vol. 52, n^o 1, p. 4–24, <http://dx.doi.org/10.1016/j.comnet.2007.09.018>.
- [4] V. ISSARNY, M. CAPORUSCIO, N. GEORGANTAS. *A Perspective on the Future of Middleware-based Software Engineering*, in "FOSE '07: 2007 Future of Software Engineering", Washington, DC, USA, IEEE Computer Society, 2007, p. 244–258, <http://dx.doi.org/10.1109/FOSE.2007.2>.

Publications of the year

Articles in International Peer-Reviewed Journal

- [5] M. CAPORUSCIO, P.-G. RAVERDY, V. ISSARNY. *ubiSOAP: A Service Oriented Middleware for Ubiquitous Networking*, in "IEEE Transactions on Services Computing", 2010, accepted, <http://hal.inria.fr/inria-00519577>.

- [6] G. COSTA, A. LAZOUSKI, F. MARTINELLI, I. MATTEUCCI, V. ISSARNY, R. SAADI, N. DRAGONI, F. MASSACCI. *Security-by-Contract-with-Trust for Mobile Devices*, in "Journal of Wireless Mobile Networks, Ubiquitous Computing and Dependable Applications (JoWUA)", 2010, vol. 4, n^o 1, p. 75-91.
- [7] A. PATHAK, V. K. PRASANNA. *Energy-Efficient Task Mapping for Data-driven Sensor Network Macroprogramming*, in "IEEE Transactions on Computers", 2010.
- [8] C.-D. PHAM, M. ABDALLAH, R. SAADI. *Risk-based adaptive scheduling in randomly deployed video sensor networks for critical surveillance applications*, in "Journal of Network and Computer Applications", Nov 2010, <http://hal.inria.fr/inria-00530355>.

International Peer-Reviewed Conference/Proceedings

- [9] M. ABDALLAH, R. SAADI, C.-D. PHAM. *Risk Management in Intrusion Detection Applications with Wireless Video Sensor Networks*, in "IEEE WCNC international conference", Australie Sydney, 2010, 0, <http://hal.inria.fr/hal-00530724>.
- [10] A. BENNACEUR, G. BLAIR, F. CHAUVEL, N. GEORGANTAS, P. GRACE, F. HOWAR, P. INVERARDI, V. ISSARNY, M. PAOLUCCI, A. PATHAK, R. SPALAZZESE, B. STEFFEN, B. SOUVILLE. *Towards an architecture for runtime interoperability*, in "In Proceedings of ISoLA 2010 - 4th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation", Grèce Crete, 2010, <http://hal.inria.fr/inria-00512446>.
- [11] A. BERTOLINO, F. DI GIANDOMENICO, A. DI MARCO, V. ISSARNY, F. MARTINELLI, P. MASCI, I. MATTEUCCI, R. SAADI, A. SABETTA. *Dependability in dynamic, evolving and heterogeneous systems: the CONNECT approach*, in "2nd International Workshop on Software Engineering for Resilient Systems (SERENE 2010)", Royaume-Uni London, 2010, <http://hal.inria.fr/inria-00512426>.
- [12] A. BERTOLINO, P. INVERARDI, V. ISSARNY, A. SABETTA, R. SPALAZZESE. *On-the-fly Interoperability through Automated Mediator Synthesis and Monitoring*, in "In Proceedings of ISoLA 2010 - 4th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation", Grèce Crete, 2010, <http://hal.inria.fr/inria-00512449>.
- [13] P. INVERARDI, V. ISSARNY, R. SPALAZZESE. *A Theory of Mediators for Eternal Connectors*, in "In Proceedings of ISoLA 2010 - 4th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation", Grèce Crete, 2010, <http://hal.inria.fr/inria-00512443>.
- [14] R. SAADI, J.-M. PIERSON, L. BRUNIE. *T2D: A Peer to Peer trust management system based on Disposition to Trust*, in "ACM SAC: 25th Symposium On Applied Computing", Suisse Sierre, Mar 2010, <http://hal.inria.fr/inria-00469649>.
- [15] A. TONINELLI, A. PATHAK, A. SEYEDI, R. SPEICYS CARDOSO, V. ISSARNY. *Middleware Support for Mobile Social Ecosystems*, in "Second IEEE International Workshop on Middleware Engineering (COMPSAC Workshops)", Corée, République De Seoul, May 2010, <http://hal.inria.fr/inria-00483736>.

Workshops without Proceedings

- [16] I. ALSHABANI, J. E. HADDAD, N. GEORGANTAS, T. MELLITI, L. MOKDAD, P. POIZAT. *Projet ANR PERSO: Pervasive Service Composition*, in "Grand Colloque STIC, Paris", January 2010, Poster.

- [17] I. ALSHABANI, J. E. HADDAD, N. GEORGANTAS, T. MELLITI, L. MOKDAD, P. POIZAT. *Projet ANR PERSO: Pervasive Service Composition*, in "Journées nationales du GDR Génie de la Programmation et du Logiciel (GPL)", Pau, March 2010, Poster.

Scientific Books (or Scientific Book chapters)

- [18] O. DAVIDYUK, N. GEORGANTAS, V. ISSARNY, J. RIEKKI. *MEDUSA: Middleware for End-User Composition of Ubiquitous Applications*, in "Handbook of Research on Ambient Intelligence and Smart Environments: Trends and Perspectives", F. MASTROGIOVANNI, N. CHONG (editors), IGI Global, 2010, <http://hal.inria.fr/inria-00432675>.
- [19] N. GEORGANTAS, V. ISSARNY, S. B. MOKHTAR, Y.-D. BROMBERG, S. BIANCO, G. THOMSON, P.-G. RAVERDY, A. URBIETA, R. S. CARDOSO. *Middleware Architecture for Ambient Intelligence in the Networked Home*, in "Handbook of Ambient Intelligence and Smart Environments", H. NAKASHIMA, H. AGHAJAN, J. C. AUGUSTO (editors), Springer US, 2010, p. 1139-1169, http://dx.doi.org/10.1007/978-0-387-93808-0_42.
- [20] A. PATHAK, V. K. PRASANNA. *High-Level Application Development for Sensor Networks - Data-Driven Approach*, in "Theoretical Aspects of Distributed Computing in Sensor Networks", S. NIKOLETSEAS, J. ROLIM (editors), Springer, October 2010.

Books or Proceedings Editing

- [21] B. DE RUYTER, R. WICHERT, D. V. KEYSON, P. MARKOPOULOS, N. STREITZ, M. DIVITINI, N. GEORGANTAS, A. M. GOMEZ (editors). *Proceedings of First International Joint Conference on Ambient Intelligence, Aml 2010*, November 2010.
- [22] N. GEORGANTAS, V. ISSARNY (editors). *Special Issue on Ad Hoc Ambient Computing. International Journal of Ambient Computing and Intelligence*, IGI Global, 2010, vol. 2.

Research Reports

- [23] D. ATHANASOPOULOS, A. ZARRAS, V. ISSARNY, P. VASSILIADIS. *Hiding Design-Decisions in Service-Oriented Software via Service Abstraction Recovery*, INRIA, Jun 2010, <http://hal.inria.fr/inria-00491349>.
- [24] CONNECT PROJECT TEAM MEMBERS. *CONNECT project technical reports*, INRIA, 2010, <http://connect-forever.eu/publication.html>.