Activity Report 2011

Project-Team ARLES

Software architectures and distributed systems

RESEARCH CENTER
Paris - Rocquencourt

THEME
Distributed Systems and Services

Table of contents

Project-Team ARLES

Keywords: Distributed System, Middleware Architectures, Software Engineering, Pervasive Computing, Social Networks, Sensor Networks, Service Oriented Architecture

1. Members

Research Scientists

Valérie Issarny [Team Leader, Inria Senior Researcher, HdR]
Nikolaos Georgantas [Inria Junior Researcher]
Animesh Pathak [Inria Junior Researcher]
Nelly Bencomo [Marie Curie Fellow, since May 2011]

Technical Staff

Emil Andriescu [*Apprenti*, Inria, until September 2011, Research Intern since October 2011]
Mohammad Ashiqur Rahaman [*Ingénieur Expert*, Inria, until January 2011]
Sandrine Beauche [*Ingénieur Expert*, Inria]
Benjamin Billet [Inria Research Intern, since June 2011]
Luca Cavallaro [*Ingénieur Expert*, Inria]
Sneha-Sham Godbole [*Ingénieur Expert*, Inria, until June 2011]
Iraklis Leontiadis [*Ingénieur Jeune Diplomé*, Inria, until September 2011]
Rachid Saadi [*Ingénieur Expert*, Inria]
Daniel Sykes [*Ingénieur Expert*, Inria]
Roberto Speicys Cardoso [*Ingénieur Expert*, Inria, until August 2011]
Bachir Moussa Tari Bako [*Ingénieur Jeune Diplomé*, Inria]

PhD Students

Dionysis Athanasopoulos [Inria, University of Ioannina, Greece]
Nebil Ben Mabrouk [Inria, University Pierre and Marie Curie - Paris 6]
Amel Bennaceur [Inria, University Pierre et Marie Curie - Paris 6]
Sara Hachem [Inria, University of Versailles Saint-Quentin-en-Yvelines]
Pankesh Patel [Inria CORDI-S, University Pierre et Marie Curie - Paris 6]
Amir Seyedi [Inria, University Pierre et Marie Curie - Paris 6]

Post-Doctoral Fellow

Thiago Valladares Sabino Teixeira [Inria, until September 2011]

Visiting Scientists

Oleg Davidyuk [University of Oulu, Finland, December 2011]
Romina Spalazzese [University of L'Aquila, Italy, April 2011 - June 2011]
Apostolos Zarras [University of Ioannina, Greece]

Administrative Assistant

Stéphanie Chaix

2. Overall Objectives

2.1. Overall Objectives

With digital equipment 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 communication 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 distributed infrastructure. It establishes a software layer that homogenizes 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 demanding, the role of middleware has proved central in the software system development practice.

However, the development of middleware itself remains a complex task [1]. In particular, middleware must provide 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, as well as context awareness [5]. Among these new requirements, we are especially interested in applications involving social interactions between the users of modern smart phones, which involve addressing issues of trust and privacy along with efficiency of algorithms to deal with 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 computing [2] and subsequent pervasive computing [3] and ambient intelligence [4] 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

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

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

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

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

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.

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 several new subjects, 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 program of the EC FP7-ICT (§ 6.2, § 7.1.1),
- The use of Models@run.time to extend the applicability of models and abstractions to the runtime environment, arising from our anticipation that Models@run.time will play an integral role in the management of extremely distributed systems. We are exploring the use of Models@run.time to tackle the crucial problem of uncertainty in extremely distributed systems that are aware of their own requirements, as well as to support the runtime synthesis of software that will be part of the executing system (§ 6.2).
- Interaction paradigm abstractions and service oriented middleware for choreographies in the ultra-large scale future Internet. This research is also part of a major European collaborative project within the Software and Service Architectures and Infrastructures programme of the EC FP7-ICT (§ 6.4, § 7.1.2).
- 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 (§ 6.6).

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

- The AMBIENTIC spin-off (http://www.ambientic.com/) was launched 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.

In addition to the above, we co-organized a successful summer school on *Formal Methods for Eternal Networked Software Systems*, in the "SFM: International School on Formal Methods for the Design of Computer, Communication and Software Systems" series at Bertinoro, Italy. It covered topic such as connecting eternal software systems, formal foundations for connectors, dynamic connector synthesis, interaction behavior monitoring and learning, and dependability assurance of connected systems. We also co-organized *FOME: Future of Middleware* event at the 12th ACM/IFIP/USENIX International Middleware Conference in Lisbon, Portugal, which brought together a number of invited leading researchers in the field selected to offer comprehensive coverage of the key issues to be tacked in the near future in the area of Middleware research, such as: right abstractions for the development of future distributed systems; how to achieve interoperability and openness; and how to ensure dependability and security in the face of extremely large scale and heterogeneity in future distributed systems.

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 networked resources in the environment. This leads us to investigate methods and tools supporting the engineering of pervasive software systems, with a special emphasis on associated middleware solutions.

3.2. Engineering Pervasive Software Systems

Since its emergence, middleware has proved successful in assisting 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. In this section, we (i) discuss challenges that middleware brings to software engineering, and (ii) outline a revolutionary approach to middleware-based software engineering aiming at the dynamic runtime synthesis of emergent middleware.

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 transport layer network programming via 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.

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) discipline [5]. The advent of middleware standards have further contributed to the systematic adoption of this paradigm for distributed software development.

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

In spite of the above, mature engineering methodologies to comprehensively assist the development of middleware-based software systems, from requirements analysis to deployment and maintenance, are lagging behind. Indeed, systematic software development accounting for middleware support is rather the exception than the norm [6] [7], 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 [5], raising 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, calling for next-generation middleware platforms and novel software engineering processes integrating middleware features in 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. That said, 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 are currently present in ubiquitous networking environments, and that require 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 currently present in networked environments and further expected to increase dramatically in the future.

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

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 systems [8]. 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; however, 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

[6] R. 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.

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

[8] Valérie 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.

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. Several different approaches to choreography modeling can be found in the literature: *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 hand-held consumer devices, 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 ranges 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 some 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 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 by changing conditions. As dynamic environments call for fulfilling user requests on the fly (i.e., at run-time) 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.

Coordination of heterogeneous distributed systems: Another aspect that we consider important in service composition is enabling integration of services that employ different interaction 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 interaction paradigms, such as message-driven, event-driven and data-driven ones. Different interaction paradigms 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 paradigms is imperative in the extremely 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 interaction paradigm and thus do not or only poorly address cross-paradigm interoperability. Efforts combining diverse interaction paradigms 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.

Evolution 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 of 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 functionalities 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 [9]. Recent efforts in standardizing network-layer protocols for embedded devices 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 network, some middleware solutions propose to manipulate them using semantic techniques, such as in the Triple Space Computing approach, which models 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

[9] L. Mottola and G. P. Picco. Programming Wireless Sensor Networks: Fundamental Concepts and State of the Art. In ACM Computing Surveys. Volume 43, Issue 3. April 2011.

(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 proposing 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 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. Emergent Middleware Enablers

Participant: Valérie Issarny [correspondent].

[10] Lane, 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

As part of our research work on Emergent MIddleware, we have implemented Enablers (or Enabler functionalities) that make part of the overall CONNECT [30] architecture realizing Emergent Middleware in practice.

Discovery Enabler: The CONNECT Discovery Enabler is the component of the overall CONNECT architecture that handles discovery of networked systems (NSs), stores their descriptions (NS models), and performs an initial phase of matchmaking to determine which pairs of systems are likely to be able to interoperate. Such pairs are then passed to the Synthesis Enabler so that mediators can be generated. The Discovery Enabler is written in Java and implements several legacy discovery protocols including DPWS and UPnP. Matchmaking is done on the basis of affordances contained in the system description, that is, ontological concepts describing the system's category. Systems with the same affordance, or affordances standing in a specialization relationship, can be considered for connection. If a system does not provide its affordance, the Discovery Enabler can infer a likely one using text categorization based on the system's interface description. The Discovery Enabler will soon be available for download under an open source license.

Synthesis Enabler: We have implemented (in Java) two approaches to mediator synthesis as part of the CONNECT Synthesis Enabler:

- **Mapping-based mediator synthesis.** This implementation focuses on networked systems that have compatible functionalities but are unable to interact successfully due to mismatching interfaces and/or behaviors. The ontology used in our implementation is encoded so as to make the reasoning more efficient at runtime while considering both subsumption and the union of classes. Based on the interface mapping, a correct-by-construction mediator is generated. In our current implementation, we are leveraging the LTSA (Labeled Transition System Analyser) model checker to generate the parallel composition of the mapping processes and verify that the overall system successfully terminates. In the near future, we will be incorporating our techniques so as to deal with ambiguous mappings, i.e., when an action from one system may semantically be mapped to different actions from the other system.
- **Goal-based abstract mediator synthesis.** This implementation considers the protocols of two networked systems and produces the mediator protocol that allow them to interact so as to satisfy user goals. More specifically, the alphabet of the two protocols are aligned using ontology matching. The aligned protocols as well as the user goal are encoded as a satisfiability problem. The *Zot* model checker solves this problem (if possible) and produces a possible feasible interaction satisfying user goals.

The synthesis enabler will soon be available for download under an open source license.

5.3. 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.
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://gforge.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.

5.4. Supporting Service Orchestrations over Heterogeneous Interaction Paradigms

Participant: Nikolaos Georgantas [correspondent].

Established architectural paradigms enabling open system integration, such as service oriented architecture (SOA) and enterprise service bus (ESB), have provided answers to the essential issue of interoperability in distributed systems. However, realizations of these architectural paradigms fall short when it comes to integrating systems featuring heterogeneous interaction paradigms, such as client/server (CS), publish/subscribe (PS) and tuple space (TS), due to the differing interaction semantics of the latter. Typical solutions constitute in wrapping any system behind RPC-based service interfaces, which results in partial loss of interaction semantics. This can cause suboptimal or even problematic system integration.

Aiming at enabling seamless integration of heterogeneous interaction paradigms, we introduce an interoperability solution based on abstraction and merging of their common high-level semantics, paying special attention to the preservation of semantics. To this end, we propose three abstract connector types for the CS, PS and TS interaction paradigms. We further introduce a higher-level generic application (GA) connector type, which provides an abstract union of the three models, thus preserving their interaction semantics. We express our connector types in terms of application programming interface (API) primitives and related interaction protocol semantics. We then apply our abstractions to rethink a typical SOA- and ESB-based orchestration of heterogeneous distributed systems. Our solution features:

- Extending the BPEL workflow language with GA API primitives in terms of extension activities enabled by the BPEL specification;
- Introducing XSLT transformation between the GA-extended BPEL and the standard BPEL, which consists of encapsulating GA primitives into standard BPEL primitives and enables conveying GA semantics on top of BPEL primitives and subsequently on top of the common bus protocol primitives;
- Providing Java code templates for systematic and highly facilitated building of ESB-embedded binding components;
- Proposing interface description languages in the form of XSDs for systems employing CS-, PS-, TS-, and GA-connectors; and
- Introducing XSLT transformations between native system interface descriptions and GA-based interface descriptions.

We have developed our solution on top of the PEtALS ESB, which provides inherent support for BPEL by embedding the EasyBPEL workflow engine. Our solution considerably facilitates the application developer in designing and executing heterogeneous orchestrations. Furthermore, it is highly extensible, enabling easy integration of support for new middleware platforms. To demonstrate the applicability of our approach, we have implemented an application workflow integrating a JMEDS DPWS Web Service (CS), a JMS system based on Apache ActiveMQ (PS), and a Jini JavaSpaces system (TS). Our software will soon be released under open source license.

5.5. 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. As part of our work in this domain, 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.

The current implementation of *Srijan* targets both the Sun SPOT sensor nodes and larger nodes with J2SE. Most recently, *Srijan* also includes rudimentary support for incorporating Web services in the application being designed. The software is released under open source license, and available as an Eclipse plug-in at http://code.google.com/p/srijan-toolkit/.

5.6. 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/.

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

Participant: Valérie Issarny [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.

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:

- 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 *Security Manager* uses well-established techniques of cryptography and secure communication to provide necessary security.
- The *Partnership Manager* provides device or user information in the form of *profiles*.
- iBICOOP relies on service location protocols for *naming and discovery* of nearby services on currently active network interfaces that support IP multicast.
- Besides normal file managing tasks, the *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.

The iBICOOP middleware has been licensed by AMBIENTIC (http://www.ambientic.com/), a start-up that 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.

URL: https://www-roc.inria.fr/arles/index.php/ongoing-research-projects/74-data-sharing-and-replication-in-pervasive-networks

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 2011 have focused on the following areas:

- Dynamic interoperability among networked systems toward making them eternal, by way of on-the-fly generation of connectors based on adequate system models (§ 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);
- Service oriented middleware for the ultra large scale future Internet of Things (§ 6.4);
- Abstractions for enabling domain experts to easily compose applications on the Internet of Things (§ 6.5); and
- System-level support for application development in the context of mobile social ecosystems, while taking into account privacy, performance, and data interoperability (§ 6.6).

6.2. Emergent Middleware Supporting Interoperability in Extreme Distributed Systems

Participants: Emil Andriescu, Nelly Bencomo, Amel Bennaceur, Luca Cavallaro, Nikolaos Georgantas, Sneha-Sham Godbole, Valérie Issarny, Rachid Saadi, Daniel Sykes.

Interoperability is a fundamental challenge for today's extreme distributed systems. Indeed, the high-level of heterogeneity in both the application layer and the underlying infrastructure, together with the conflicting assumptions that each system makes about its execution environment hinder the successful interoperation of independently developed systems. A wide range of approaches have been proposed to address the interoperability challenge [31]. Solutions that require performing changes to the systems are usually not feasible since the systems to be integrated may be legacy systems, COTS (Commercial Off-The-Shelf) components or built by third parties; neither are the approaches that prune the behavior leading to mismatches since they also restrict the systems' functionality. Therefore, many solutions that aggregate the disparate systems in a non-intrusive way have been proposed. These solutions use intermediary software entities, called *mediators*, to interconnect systems despite disparities in their data and/or interaction models by performing the necessary coordination and translations while keeping them loosely-coupled. However, creating mediators requires a substantial development effort and a thorough knowledge of the application-domain, which is best understood by domain experts. Moreover, the increasing complexity of today's distributed systems, sometimes referred to as Systems of Systems, makes it almost impossible to develop 'correct' mediators manually. Therefore, formal approaches are used to synthesize mediators automatically.

In light of the above, we have introduced the notion of *emergent middleware* for realizing mediators. Our research on enabling emergent mediators is done in collaboration with our partners of the CONNECT project (§ 7.1.1). Our work during the year has more specifically focused on:

- **Supporting architecture.** We have been working together with our partners in the CONNECT project on the refinement of an overall architecture supporting emergent middleware, from the discovery of networked systems to the learning of their respective behavior, and synthesis of emergent middleware enabling them to interoperate [30].
- **Affordance inference.** We have proposed an ontology-based formal model of networked systems based on their affordances, interfaces, behavior, and non-functional properties, each of which describes a different facet of the system [2]. However, legacy systems do not necessarily specify all of the aforementioned facets. Therefore, we are currently exploring techniques to infer the affordance by using textual descriptions of the interface of networked systems. More specifically, we rely on machine learning techniques to automate the inference of the affordance from the interface description by classifying the natural-language text according to a predefined ontology of affordances [17].
- **Mediator synthesis for emergent connectors.** We focus on systems that have compatible functionality, i.e., semantically matching affordances, but are unable to interact successfully due to mismatching interfaces or behaviors. We propose two approaches to enable communication between such systems:

  1. A *mapping based* approach, whose goal is to automatically synthesize a mediator model that ensures their *safe* interaction, i.e., deadlock-freedom and the absence of unspecified receptions. Our approach combines semantic reasoning and constraint programming to identify the semantic correspondence between networked systems' interfaces, i.e., *interface mapping*. Unlike existing approaches that only tackle the one-to-one correspondence between actions, this approach handles the more general cases of one-to-many and many-to-many mappings.
  2. A *goal based* approach, which enables the communication of two networked systems, so that the communication satisfies a given user goal. It aligns their actions using ontology matching. The aligned processes as well as the user goal are encoded as a satisfiability problem. It relies on model checking to determine if a feasible communication trace exists that satisfies the user goal. The model checking process is reiterated so as to discover all the feasible satisfying traces, which are finally concatenated to build the mediator.

  The feasibility of both of our approaches has been demonstrated through prototype tools and real-world scenarios involving heterogeneous systems.
- **Mediator synthesis for streaming connectors.** In the context of dynamic mediator synthesis, we have targeted the domain of mobile multimedia streaming, resulting in a first step that statically solves the hard problem of streaming interoperability across heterogeneous smartphone multimedia platforms. With the recent evolution of mobile phones, multimedia streaming is now commonly used in smartphones for purposes such as video broadcast, video conferencing and place shifting, which in turn highlights the importance of multimedia enabled applications. However, peer-to-peer solutions are difficult to implement because of increased node heterogeneity and their low processing power. Furthermore, existing mobile platforms such as Android, iOS, Blackberry and Windows Phone 7 support multimedia streaming (as resource consumers) either through platform specific APIs or system services. However, they use heterogeneous protocols and data formats, thus compromising interoperability.

  Given the challenges above, we designed AmbiStream [11], a lightweight middleware for heterogeneous mobile devices, capable of “on the fly" adaptation. AmbiStream relies on the highly-optimized multimedia software stacks provided by smartphone platforms and adds the necessary layers to solve interoperability. More specifically, the middleware targets: a) Streaming of prerecorded or live audio/video using an intermediary real-time protocol; b) Managing streaming protocol translation and multimedia container format adaptation to the ones supported natively by each device; and c) Extensibility in order to support new multimedia streaming protocols and multimedia container formats given its plug-in based architecture. We have used a model-driven approach to generate multi-platform plug-ins from higher level descriptions in the form of a Domain Specific Language (DSL). The defined DSL takes into account multimedia specific operations such as timing, fragmenting, multiplexing, congestion control and buffering.

- **Models@run.time.** We have recently integrated the notion of *Models@run.time* [11] in our research towards emergent middleware. We use Models@run.time to extend the applicability of models and abstractions to the runtime environment. As is the case for software development models, a run-time model is often created to support reasoning. However, in contrast to development models, run-time models are used to reason about the operating environment and runtime behavior, and thus these models must capture abstractions of runtime phenomena. Different dimensions need to be balanced, including resource-efficiency (time, memory, energy), context-dependency (time, location, platform), as well as personalization (quality-of-service specifications, profiles). The hypothesis is that because Models@run.time provide meta-information for these dimensions during execution, run-time decisions can be facilitated and better automated. Thus, we anticipate that Models@run.time will play an integral role in the management of extremely distributed systems. Our work on the use of Models@run.time has two aspects:
  - We have used Models@run.time to tackle the crucial problem of uncertainty in extremely distributed systems that are aware of their own requirements. Requirements awareness helps optimize requirements satisfaction when factors that were uncertain at design time are resolved at runtime. Using our approach, we are able to maintain goal-based models in memory while the system is running. The executing system, therefore, is able to introspect and consult it goals during runtime. Crucially, at runtime we use the notion of claims to represent assumptions that cannot be verified with confidence at design time. Such claims are attached to the goal-based runtime models. By monitoring claims at runtime, their veracity can be tested. If falsified, the effect of claim negation can be propagated to the system's goal model and an alternative means of goal realization can be selected automatically, allowing the dynamic adaptation of the system to the prevailing environmental context [14], [15], [16].
  - In a complementary way to the mediator synthesis approaches discussed above, we further promote the use of Models@run.time to support the runtime synthesis of software that will be part of the executing system. Specifically, we focus on the use of runtime models to support the realization of emergent middleware, i.e., the synthesis of mediators that define a sequences of actions to translate semantic actions of one system developed using a particular middleware protocol to the semantic actions of another system developed using an alternate middleware built with no prior knowledge on the former. Discovery and learning enablers capture the required knowledge of the context and environment during runtime. Supported by that knowledge, a runtime model of the mediator-to-be is reified. Reification means that the knowledge is explicitly formulated and made available for computational manipulation. The form of the runtime models is based on labeled transition systems (LTSs) which offer the behavioral semantics needed to model the interaction protocols. Ontologies complement the LTSs providing semantic reasoning about the mapping between protocols. Specifically the LTS of each protocol is annotated using ontologies to support the subsequent mapping between the protocols. From the LTS-based runtime models, mediators are synthesized.

6.3. Revisiting the Abstractions of Service Oriented Computing for the Future Internet

[11] Models@run.time Dagstuhl Seminar, http://www.dagstuhl.de/en/program/calendar/semhp/?semnr=11481

Participants: Mohammad Ashiqur Rahaman, Dionysis Athanasopoulos, Sandrine Beauche, Nebil Ben Mabrouk, Nikolaos Georgantas, Valérie Issarny.

A software architecture style characterizes, via a set of abstractions, the types of: components (i.e., units of computation or data stores), connectors (i.e., interaction protocols) and possibly configurations (i.e., system structures) that serve to build a given class of systems. As such, the definition of a software architectural style is central toward eliciting appropriate design, development and runtime support for any family of systems. The service oriented architecture style may then be briefly defined as follows: (1) components map to services, which may be refined into consumer, producer or prosumer services; (2) connectors map to traditional client-service interaction protocols; and (3) configurations map to compositions of services through (service-oriented) connectors, e.g., choreography and orchestration structures. While the service-oriented architecture style is well suited to support the development of Internet-based distributed systems, it is largely challenged by the Future Internet that poses new demands in terms of sustaining *ities* such as scalability, heterogeneity, mobility, awareness & adaptability that come in extreme degrees compared to the current Internet. Therefore, we have been working on eliciting software architectural abstractions for the Future Internet by building upon the service-oriented architecture style, as well as on applying them to system design, development and execution.

Complex distributed applications in the Future Internet will be to a large extent based on the open 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 Cloud infrastructures). These heterogeneous system domains differ significantly in terms of interaction paradigms, communication protocols, and data representation models, provided by supporting middleware platforms. Specifically considering interaction paradigms, the client/server (CS), publish/subscribe (PS), and tuple space (TS) paradigms are among the most widely employed ones today, with numerous related middleware platforms. In light of the above, we have aimed at eliciting abstractions that (i) leverage the diversity of interaction paradigms associated with today's and future complex distributed systems, as well as (ii) enable cross-paradigm interaction to sustain interoperability in the highly heterogeneous Future Internet [19].

Existing cross-domain interoperability efforts are based on bridging communication protocols, wrapping systems behind standard technology interfaces, and/or providing common API abstractions. In particular, such techniques have been applied by the two widely established system integration paradigms, that is, service oriented architecture (SOA) and enterprise service bus (ESB). However, state of the art interoperability efforts do not or only poorly address interaction paradigm interoperability. Indeed, systems integrated via SOA and ESB solutions have their interaction semantics transformed to the CS paradigm. Then, potential loss of interaction semantics can result in suboptimal or even problematic system integration. To overcome the limitation of today's ESB-based connectors for cross-domain interoperability in the Future Internet, we introduce a new connector type, called GA connector, which stands for "Generic Application connector". The proposed connector type is based on the service bus paradigm in that it achieves bridging across heterogeneous connector types. However, the behavior of the GA connector type differs from that of classical ESB connectors by bridging protocols across heterogeneous paradigms, which is further realized by paying special attention to the preservation of the semantics of the composed protocols. Indeed, the GA connector type is based on the abstraction and semantic-preserving merging of the common high-level semantics of base interaction paradigms.

Eliciting Interaction Paradigm Abstractions: We introduce a systematic abstraction of interaction paradigms with the following features:

- First, we introduce base CS, PS and TS connector types, which formally characterize today's core interaction paradigms. The proposed types comprehensively cover the essential semantics of the considered paradigms, based on a thorough survey of the related literature and representative middleware instances.
- Then, we further abstract these connector types into a single higher-level one, the GA connector type.

GA is a comprehensive connector type based on the abstract union of CS, PS, and TS, where precise identification of the commonalities or similarities between the latter has enabled the optimization of the former. Further, GA preserves by construction the semantics of CS, PS, and TS.

- In more detail, connector types are formally specified in terms of: (i) their API (Application Programming Interface), and (ii) their roles, i.e., the semantics of interaction of the connected component(s) with the environment via the connector. Regarding the latter, the behavioral specification of roles from a middleware perspective relates to specifying the production and consumption of information in the network, while the semantics of the information are abstracted and dealt with at the application layer. The behaviors of the connector roles are then specified using Labeled Transition Systems (LTS). We precisely define the mapping of the roles implemented by the base connector types to/from the corresponding roles of the GA connector type.
- For both the above abstraction transformations, we provide counterpart concretizations, which enable transforming GA connector primitives to CS, PS, or TS connector primitives and then to concrete middleware platforms primitives.
- Furthermore, based on the GA abstraction, we introduce mapping transformations between any pair from the set {CS, PS, TS} via GA. The fine knowledge of CS, PS, and TS semantics, as embedded in GA, enables these mappings to be precise: differing semantics are mapped to each other in such a way that loss of semantics is limited to the minimum. These mappings relate to the definition of the glue process implemented by the GA connector, which defines how a pair of producer and consumer roles coordinates in the environment. The GA glue reconciles consumer and producer roles that may differ with respect to time and space coupling as well as scoping. Hence, GA connectors support interactions among highly heterogeneous services of the Future Internet, and especially across domains.

eXtensible Service Bus: We apply the above connector abstractions to introduce an enhanced bus paradigm, the *eXtensible Service Bus (XSB)*. XSB features richer interaction semantics than common ESB implementations to deal effectively with the increased Future Internet heterogeneity. Moreover, from its very conception, XSB incorporates special consideration for the cross-integration of heterogeneous interaction paradigms. When mapping between such paradigms, special attention is paid to the preservation of interaction semantics. XSB has the following features:

- XSB is an abstract bus that prescribes only the high-level semantics of the common bus protocol. The XSB common bus protocol features GA semantics.
- Heterogeneous systems can be plugged into the XSB by employing binding components that adapt between the native middleware of the deployed system and the common bus protocol. This adaptation is based on the systematic abstractions and mappings discussed above
- XSB, being an abstract bus, can have different implementations. This means that it needs to be complemented with a substrate which at least supports: (1) deployment (i.e., plugging) of various systems on the bus, and (2) a common bus protocol implementing GA semantics. With respect to the latter, we envision that a GA protocol realization may either be designed and built from scratch (still supposing at least an IP-based transport substrate) or be implemented by conveying GA semantics on top of an existing higher-level protocol used as transport carrier. The latter solution can be attractive, as it facilitates GA protocol realizations in different contexts and domains.

We have carried out an early realization of XSB on the PEtALS ESB. In particular, we addressed the workflow-based orchestration of heterogeneous systems, which is a preliminary step before dealing with peer-wise system integration. This work already provides a successful feasibility study of the XSB concept. This work comprises: (i) extending the BPEL workflow language with GA API primitives; (ii) introducing transformation between the GA-extended BPEL and the standard BPEL, which consists in encapsulating GA primitives into standard BPEL primitives and enables conveying GA semantics on top of BPEL primitives and subsequently on top of the common bus protocol primitives; (iii) providing templates for systematic and highly facilitated building of binding components; and (iv) introducing transformations between native system interface descriptions and GA-based interface descriptions.

6.4. Service Oriented Middleware facing the Challenges of the Internet of Things

Participants: Benjamin Billet, Nikolaos Georgantas, Sara Hachem, Valérie Issarny, Roberto Speicys Cardoso, Thiago Valladares Sabino Teixeira.

Over the years, the Internet has become the most important networking infrastructure, providing an integrated entity enabling sharing, contributing, creating, using, collaborating and integrating information and knowledge by all. As a result, the Internet is changing at fast pace and is expected to evolve into the *Future Internet*, i.e., service-aware and self-aware federated networks that provide built-in and integrated capabilities such as: contextualization, reliability, robustness, mobility, security, service support, and self-management of communication resources and services. In our vision, The Future Internet can be defined as the union and cooperation of the Internet of Content, Internet of Services, Internet of Things, and 3D interactive Internet, supported by an expanding network infrastructure foundation. In ARLES, we chose to pay special attention to the Internet of Things (IoT). IoT is characterized by the integration of large numbers of real-world objects (or “things") onto the Internet, with the aim of turning high-level interactions with the physical world into a matter as simple as is interacting with the virtual world today. As such, two devices that will play a key role in the IoT are *sensors* and *actuators*. In fact, such devices are already seeing widespread adoption in the highly localized systems within our cars, mobile phones, laptops, home appliances, etc. In their current incarnation, however, sensors and actuators are used for little more than low-level inferences and basic services. This is partly due to their highly specialized domains (signal processing, estimation theory, robotics, etc.), which demand application programmers to also be domain experts, and partly due to a glaring lack of interconnectivity between all the different devices. Our work within that domain was focused on three related directions:

- **Challenges related to IoT**: To prepare the ground for our research on middleware for the Internet of Things, we identified the set of challenges in the IoT, namely [10]: the large scale of the Internet of Things, heterogeneity of things, unknown and dynamic network topology, unknown data-point availability, incomplete or inaccurate metadata, and conflict resolution. The *scale* issue arises with the millions of devices, millions of users, large amounts of data to share and services to request. The *heterogeneity* of the IoT is due to the fact that the network will be composed of different types of devices from different vendors with varying sensing/actuating characteristics. The *unknown dynamic network topology* results from the fact that devices will be mostly mobile and their availability is unknown. A related challenge is the *unknown data-point availability* as things, which provide the desired measurements, may leave the network or malfunction at any time. A data point is measurement of an entity of interest at a specific time. As for *metadata inaccuracy*, this issue is a direct result of humans, who are prone to making errors, being the main source of metadata specification. Last but not least, *conflict resolution* is due to the multiple stakeholders involved in the Internet of Things
- **Middleware Requirements for the Internet of Things**: The middleware we plan on implementing should abstract things (IoT devices) as services and support dynamic service composition. To handle the IoT challenges, the middleware should also support a probabilistic discovery approach where only a subset, instead of a whole set, of devices is selected in a way that provides a good enough answer that satisfies an application’s request [10]. However, and prior to designing the middleware architecture, we extensively surveyed the literature in order to identify research challenges for service-oriented middleware design, therefore investigating service description, discovery, access and composition in the Future Internet of services [7].
- **Ontologies for the Internet of Things**: As part of our middleware architecture, we specified a set of ontologies [20] that model real-world entities as physical concepts, along with things that measure those entities. Further, to support a smarter service composition, we also modeled mathematical formulas and physics relations as services to substitute missing thing-based services. Those services will instead compute the value of a desired measurement of an entity of interest. Finally, we also specified an ontology that describes estimation models that can be used to estimate the value of a

measurement in case of a missing data point or a missing data source. Estimation models can further be used to define probabilistic discovery functions that will be executed by the middleware.

6.5. Composing Applications in the Internet of Things

Participants: Iraklis Leontiadis, Pankesh Patel, Animesh Pathak.

As introduced above, the Internet of Things(IoT) integrates the physical world with the existing Internet, and is rapidly gaining popularity, thanks to the increased adoption of smart phones and sensing devices. Several IoT applications have been reported in recent research, and we expect to see increased adoption of IoT concepts in the fields of personal health, inventory management, and domestic energy usage monitoring, among others.

An important challenge to be addressed in the domain of IoT is to enable domain experts (health-care professionals, architects, city planners, etc.) to develop applications in their fields rapidly, with minimal support from skilled computer science professionals. Similar challenges have already been addressed in the closely related fields of Wireless Sensor and Actuator Networks (WSANs) and Pervasive/Ubiquitous computing. While the main challenge in the former is the extremely *large scale* of the systems (hundreds to thousands of largely similar nodes, sensing and acting on the environment), the primary concern in the latter has been the *heterogeneity* of nodes and the major role that the user's own interaction with these nodes plays in these systems (cf. the classic "smart home" scenario where the user interacts with a smart display which works together with his refrigerator and toaster). The upcoming field of IoT includes both WSANs as well as smart appliances, in addition to the elements of the "traditional" Internet such as Web and database servers, exposing their functionalities as Web services etc. Consequently, an ideal application development abstraction of the IoT will allow (domain expert) developers to intuitively specify the rich interactions between the extremely large number of disparate devices in the future Internet of Things.

The larger goal of our research is to propose a suitable application development framework which addresses the challenges introduced above. This will most likely be achieved by a domain specific language (DSL) that exposes specific functionalities to the domain experts. The first logical step was to construct a domain model. Towards that end, we took advantage of the CRC — Classes, Responsibility, Collaboration — technique, defining the main abstract concepts, their responsibilities, and associations that represent their relationship with each other in the IoT. Specifically, we used this technique to propose a domain model [22] that addresses the following challenges:

- **Creation of common understanding.** The different terms used by different people in the IoT domain can lead to confusion, which can be alleviated by the usage of a common lexicon, as provided by a domain model. This lexicon can then be used by researchers, system programmers, as well as domain experts.
- **Modeling invariant properties.** The domain model represents the invariant properties of the domain — concepts and relationships which do not change from one application to the other. An instance of this in the IoT domain can be the notion of a *sensor* attached to a *device*. Depending on the specific applications, the type of sensors and devices can change (e.g, a light sensor attached to a smart phone), but the inherent relationship between the types of entities they represent does not.
- **Enabling modular design.** Application needs often tend to arrive in terms of behavior, which needs to be broken down and divided among the entities in the system. A good domain model aids in this process, since the capabilities of each type of entity are clearly identified. E.g., the application requirement of "the system senses the temperature of a room and keeps it steady" can be easily broken down into an application consisting of temperature sensors, computational components, and HVAC actuators, each performing its well-known role in this sense-compute-actuate loop.

As part of a related effort with a narrower focus on the domain of sensor network *macroprogramming* — a technique that 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 — we have explored techniques to bring *Web services* in the gamut of sensor network macroprogramming. Our research addresses the challenges faced by developers of systems where sensors (e.g., RFID badge sensors in an office) interact with pre-existing larger software components exposed as Web services (e.g., the office personnel access control database). As part of our work, we have proposed extensions to the data-driven ATaG macroprogramming language using which developers can easily incorporate existing Web services in their applications.

We have incorporated our continued research in the above areas into *Srijan* (§ 5.5), which provides an easy-to-use graphical front-end to the various steps involved in developing an application using the ATaG macroprogramming framework.

6.6. Addressing Middleware Challenges in Large Scale Mobile Social Networks of the Future

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

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. 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 on addressing the following challenges:

- **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 [21] 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.
- **Inferring trust from proximity.** In mobile social network, highly sensitive private data is at risk of being shared with unwanted peers, since users may not have any knowledge about the users they socially connect with. Trust management then appears as a promising decision support for

mobile users in establishing social links. However, while the literature is rich in trust models, most approaches lack appropriate *trust bootstrapping*, i.e., the initialization of trust values. In [24], we address this challenge by introducing proximity-based trust initialization based on the users' behavioral data available from their mobile devices or other types of social interactions. The proposed approach is further assessed in the context of mobile social networking using users behavioral data collected by the MIT reality mining project. Results show that the inferred trust values correlate with the self-reported survey of users relationships.

We have incorporated our research in the above areas into Yarta [25], a middleware for mobile social applications. Our prototype middleware, as discussed in § 5.6, 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. Partnerships and Cooperations

7.1. European Contracts and Grants

7.1.1. FP7 ICT FET IP CONNECT

Participants: Emil Andriescu, Amel Bennaceur, Luca Cavallaro, Nikolaos Georgantas, Sneha-Sham Godbole, Valérie Issarny, Rachid Saadi, Daniel Sykes.

- **Name:** CONNECT – *Emergent Connectors for Eternal Software Intensive Networked Systems*
- **URL:** http://www.connect-forever.eu/
- **Related activities:** § 6.2
- **Period:** [February 2009 - July 2012]
- **Partners:** Inria (CRI Paris-Rocquencourt) [**project coordinator**], CNR (Italy), DoCoMo (Germany), Lancaster University (UK), Thales Communications SA (France), Universita degli Studi L'Aquila (Italy), Technische Universitaet 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 investigates 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 is 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, Nebil Ben Mabrouk, Benjamin Billet, Nikolaos Georgantas, Sara Hachem, Valérie Issarny, Animesh Pathak, Roberto Speicys Cardoso.

- **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) [**scientific leader**], 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 Telecommunicazioni 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 PEOPLE Requirements@run.time

Participants: Nelly Bencomo, Valérie Issarny.

- **Name:** Requirements@run.time: *Requirements-aware systems*
- **URL:** https://www-roc.inria.fr/arles/index.php/members/220-marie-curie-project-requirements-aware-systems-requirementsruntime
- **Related activities:** § 6.2
- **Period:** [May 2011 - May 2013]
- **Partners:** Inria (CRI Paris-Rocquencourt).

This project uses the novel notion of requirements reflection, that is, the ability of a system to dynamically observe and reason about its requirements. It aims to address the need of having systems requirements-aware by reifying requirements as run-time objects (i.e. requirements@run.time). These systems provide a runtime model of their requirements that allow them to reason, evaluate and report on their conformance to their requirements during execution.This project contributes towards development of conceptual foundations, engineering techniques, and computing infrastructure for the systematic development of dynamically-adaptive systems based on the principle of requirements reflection The researchers build upon their extensive expertise in the area of reflective middleware and reflective architectures and research projects like CONNECT.

7.1.4. FP7 ICT NoE NESSoS

Participants: Valérie Issarny, Animesh Pathak, Rachid Saadi, Amir Seyedi.

- **Name:** NESSoS – *Network of Excellence on Engineering Secure Future Internet Software Services and Systems*
- **URL:** http://www.nessos-project.eu
- **Related activities:** § 6
- **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 exposed by the Future Internet, new results are 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 integrates the research labs involved; NESSoS re-addresses, integrates, harmonizes and fosters the research activities in the necessary areas, and increases and spreads the research excellence. NESSoS also impacts training and education activities in Europe to grow a new generation of skilled researchers and practitioners in the area. NESSoS collaborates with industrial stakeholders to improve the industry best practices and support a rapid growth of software-based service systems in the Future Internet.

7.1.5. FP7 ICT CA EternalS

Participants: Amel Bennaceur, Valérie Issarny, Animesh Pathak, Daniel Sykes.

- **Name:** EternalS – *Trustworthy Eternal Systems via Evolving Software, Data and Knowledge*
- **URL:** http://www.eternals.eu
- **Related activities:** § 6.2
- **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.

7.1.6. PHC Ulysses: Middleware for Mobile Social Applications in Smart Urban Environments

Participant: Animesh Pathak.

- **Name: Middleware for Mobile Social Applications in Smart Urban Environments**
- **Related activities:** § 6.6
- **Period:** [Jan 2011 - December 2011]
- **Partners:** Inria (CRI Paris-Rocquencourt), Trinity College, Dublin, Ireland.

This project aims at investigating how the exploitation of novel information and communication technologies (ICT) in the field of mobile social networking can improve the quality of life of citizens. In particular, it investigates how novel shared urban infrastructures, such as bike sharing schemes, can become neighborhood hubs and offer community services to users. For example as users collect a bike, the application that they have installed on their smart phone synchronizes with the infrastructure installed on the bike sharing station, automatically retrieving information relevant to their interests and publishing any prepared postings. Users can read information or prepare postings at their leisure. The main research questions that need to be addressed to fulfill this vision include: (i) the design of appropriate data representation, management and exchange models, to support different types of data (local vs. global, short-lived vs. long-lived), to deal with distributed/inconsistent knowledge, as well as with data provenance and authentication; (ii) the seamless integration of different computing platforms and architectures (e.g., user devices, city infrastructure); (iii) the need for adequate privacy and security support to protect personal social data; and (iv) the need to design applications that are able to deal with the scale of urban environments. The project relies on the Yarta middleware (§ 5.6), which includes a flexible and expressive representational framework for social data, tools to develop application-specific data models, and a set of middleware components to manage social information in mobile environments.

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:** [2009 - 2011]
- **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).

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), 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), NTNU (Norway), Oak Ridge National Laboratory (USA), Polit. di Milano (Italy), Poznan University of Technology (Poland), 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), University of Southern Denmark in Odense (Denmark), 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:** AARIT Research (Austria), ATOS Research(Spain), British Telecom, CNR (Italy), CETIC (Belgium), CWI (The Netherlands), DTU (Denmark), FORTH-ICS (Greece), FNR (Luxembourg), FNRS (Belgium), Fraunhofer SIT (Germany), HP Labs (UK), IBM Research, Ie Business School (Spain), 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), TUBITAK UEKAE (Turkey), VTT (Finland), University of East London (UK), University of Luxembourg (Luxembourg), University of Milan (Italy), University of Portsmouth (UK), University of Roma Tor Vergata (Italy), University of Trento (Italy), University of Twente (The Netherlands), VCPC (Austria), VTT (Finland), 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 ITEmIS: IT and Embedded Integrated Systems

Participants: Mohammad Ashiqur Rahaman, Sandrine Beauche, Amir Seyedi, 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).

7.3.2. ANR MURPHY: Dependability-focused Evaluation of Sensor Networks

Participant: Animesh Pathak.

- **Name:** MURPHY – *Dependability-focused Evaluation of Sensor Networks*
- **Related activities:** § 6.5
- **Period:** [January 2011 – December 2013]
- **Partners:** CNAM, Inria ARLES, LAAS - CNRS, SmartGrains, Univ. Valenciennes.

Murphy aims at easing the development of dependable and pervasive applications built on top of robust wireless sensor networks, thus providing a mean for early detection of possible failures, by estimating dependability metrics. This endeavor is undertaken by providing:

- Fault detection based on in-network event processing,
- Fault injection which attempts to accelerate the occurrence of faults so as to judge the quality of the error handling and hence, facilitate the evaluation of dependability,
- Advanced code dissemination across sensor networks, which is intended to (i) enable the dynamic and distributed insertion of faults and (ii) hide from the end user the complexity related to this task,
- Suitable abstractions to reason on faults, wireless sensor networks, data-centric and event-driven applications.

The aforementioned components enable to detect faults, diagnose possible causes and select appropriate corrective actions, and therefore to consolidate the dependability of sensor applications.

7.3.3. Inria D2T Action de Developpement Technologique Srijan

Participants: Animesh Pathak, Iraklis Leontiadis.

- **Name:** *Srijan – Data-driven Macroprogramming for Heterogeneous Sensor*
- **Related activities:** § 6.5, § 5.5
- **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 D2T Action de Developpement Technologique MobiTools

Participants: Valérie Issarny, Bachir Moussa Tari Bako.

- **Name:** *MobiTools – Environnement de développement logiciel pour plateforme mobiles*
- **Related activities:** § 5
- **Period:** [January 2011 – December 2012]
- **Partners:** Inria (CRI Paris-Rocquencourt, EPI ARLES)

As part of the development of our software prototypes, MobiTools focuses on setting a supporting continuous integration platform (compilation, test, profiling, quality).

7.3.5. Inria DTI Action de Transfert iBICOOP

Participants: Valérie Issarny, Roberto Speicys Cardoso.

- **Name:** iBICOOP – *Middleware for mobile collaborative applications*
- **Related activities:** § 5.7
- **Period:** [November 2009 - April 2011]
- **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

- Nelly Bencomo is PC member of the following international conferences: SEAMS '11 & '12, SBES '11, MODELS '12, CIbSE '12, Poster track at RE '11 & '12;
- Nelly Bencomo is PC member of the following international workshops: MAPLE '10 at SPLC '10, WCSI '10, SOAR '11, MoDRE '11 at MODELS'11, VAST '11 UsARE '12 at ICSE '12;
- Nikolaos Georgantas is PC member of the ICSOFT '11, AmI '11, NAS '11 international conferences, and the CFSE '11 national conference;
- Nikolaos Georgantas is PC member of the M-MPAC '11, MW4SOC '11, QASBA '11, MAASC '11 international workshops, and the ICSOC '11 PhD Symposium;
- Valérie Issarny is PC member of the following international conferences: COMSWARE '11, ESEC/FSE '11, ESSOS'12, FASE '11 & '13, FMOODS '11, ICDCS '11, ICSE '13, ICSOC '11, IFIPTM '11, ISARCS '11 & 12, Middleware '11, SEAMS '11 ServiceWave '11, Mobile track at ESWC '11, Demonstration track at ICSE '11, NIER Track at ICSE '12;

- Valérie Issarny is PC member of the following international workshops: ESWC '11, EternalS '11, MW4SOC '11, Serene '11, SESENA '11;
- Animesh Pathak is PC member of ISSNIP '11, S-Cube '11 & '12, and SECON '11 international conferences; and
- Animesh Pathak is PC member of SESENA '12 and NPC '12 international workshops.

8.1.2. Leadership Services in Academic Events and Edited Journals

- Nelly Bencomo is Student Volunteers Co-Chair at ICSE 2012;
- 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 was associate editor the of ACM Computing Surveys till September 2011;
- Valérie Issarny is associate editor the Springer JISA Journal of Internet Services and Applications;
- Valérie Issarny is co-chair of the *Future of Middleware* event, co-located with Middleware '11, 13 December 2011 in Lisbon, Portugal;
- Valérie Issarny is co-chair of SFM-CONNECT: The 11th International School on Formal Methods for the Design of Computer, Communication and Software Systems: Connectors for Eternal Networked Software Systems, on 13-18 June, 2011 in Bertinoro; and
- Animesh Pathak is the Production Chair of HiPC 2011 & 2012.

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 (IFSTTAR since 2011) & "Commission d'évaluation des chercheurs";
- Valérie Issary is member of the GDR GPL scientific council; and
- Valérie Issarny is member of the evaluation committee of the ANR "Programme Blanc" and "Programme Jeunes Chercheuses et Jeunes Chercheurs" for the 2011 & 2012 call.

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;
- Valérie Issarny gave a lecture on "Middleware-Layer Connector Synthesis" at SFM-CONNECT: The 11th International School on Formal Methods for the Design of Computer, Communication and Software Systems: Connectors for Eternal Networked Software Systems, on 13-18 June, 2011 in Bertinoro;
- Valérie gave a lecture "Revisiting the Middleware Paradigm to Meet the Challenge of Interoperability in Pervasive Networks" at The workshop on Ubiquitous Computing Middleware 2011 (UCM 2011), 27 - 28 of June 2011 at the Department of Informatics of the University of Fribourg, Switzerland;

- Valérie Issarny gave a lecture on "Model-based Emergent Middleware to Meet the Challenges of Interoperability in Pervasive Networks" at ESEC/FSE'11 Tech Briefings, 6 September 2011, in Szeged, Hungary;
- Valérie Issarny gave a lecture on "Towards Future Proof Interoperability" at the SERENE'2011 Autumn School, 27-28 September 2011 in Geneva, Switzerland;
- Animesh Pathak co-taught a Master's level course on "Web sémantique, contenus et usages" at University of Versailles Saint-Quentin-en-Yvelines in Spring 2011; and
- Animesh Pathak gave two three-hour guest lectures at CNAM, Paris as part of a Master-level course.

8.3. PhD

The following PhD theses are currently in progress at the ARLES project-team:

- Dionysis Athanasopoulos, *"Evolution and Maintenance in Service-oriented Software"*, started October 2008, advised by Apostolos Zarras and Valérie Issarny
- Nebil Ben Mabrouk, *"QoS Service Oriented Middleware for Pervasive environments"*, started October 2007, advised by Nikolaos Georgantas and Valérie Issarny
- Amel Bennaceur, *"Synthèse dynamique de connecteurs pour les réseaux ubiquitaires"*, started October 2009, advised by Valérie Issarny
- Sara Hachem, *"Middleware pour l'Internet des objets intelligents"*, started October 2012, advised by Valérie Issarny
- Pankesh Patel, *"Enabling High-Level Application Development in Sensing-and-Actuation-augmented Pervasive Systems*, started October 2010, advised by Animesh Pathak and Valérie Issarny
- Amir Seyedi, *"Impact of Social Networks on Middleware for Mobile Environment"*, started December 2009, advised by Animesh Pathak and Valérie Issarny.

8.4. Internships

During the year 2011, members of the ARLES project-team supervised the work of the following student interns:

- Benjamin BILLET, *En quoi les architectures logicielles et integicielles peuvent-elles simplifier le processus de développement et d'évolution des applications multi-plateformes?*, EPSI, Montpellier, France.
- Ajay CHHATWAL, *Supporting application development for Future Internet of smart things and services*, Integrated M.Tech Dual-degree program, Institute of Technology, Banaras Hindu University, India.
- Fatma GHACHEM, *Développer et intégrer de nouveaux services d'infrastructure pour faciliter les interactions entre les terminaux mobiles*, M2, ESPRIT Ecole Supérieure Privée D'Ingenerie et de Technologie, Tunisia.
- Mahesh GONDI, *Middleware support for managing social data in smart urban environments*, Bachelor of Technology, Institute of Technology, Banaras Hindu University, India.
- Shashank KHARE, *Realization of a generic trust simulator for distributed systems*, Integrated M.Tech Dual-degree program, 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.
- Valentin VILLANOVA, *Ergonomie des services coopératifs mobiles*, IUT Cergy-Pontoise, France.

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] G. Blair, A. Bennaceur, N. Georgantas, P. Grace, V. Issarny, V. Nundloll, M. Paolucci. *The Role of Ontologies in Emergent Middleware: Supporting Interoperability in Complex Distributed Systems*, in "Big Ideas track of ACM/IFIP/USENIX 12th International Middleware Conference", Lisbon, Portugal, 2011, http://hal.inria.fr/inria-00629059/en.

[3] M. Caporuscio, P.-G. Raverdy, V. Issarny. *ubiSOAP: A Service Oriented Middleware for Ubiquitous Networking*, in "IEEE Transactions on Services Computing", 2010, vol. 99, n^{o} PrePrints [*DOI :* 10.1109/TSC.2010.60], http://hal.inria.fr/inria-00519577.

[4] D. Charlet, V. Issarny, R. Chibout. *Energy-efficient middleware-layer multi-radio networking: An assessment in the area of service discovery*, in "Comput. Netw.", January 2008, vol. 52, p. 4–24 [*DOI :* 10.1016/j.comnet.2007.09.018], http://dl.acm.org/citation.cfm?id=1321781.1321914.

[5] 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.

[6] V. Issarny, N. Georgantas, S. Hachem, A. Zarras, P. Vassiliadis, M. Autili, M. A. Gerosa, A. Ben Hamida. *Service-Oriented Middleware for the Future Internet: State of the Art and Research Directions*, in "Journal of Internet Services and Applications", May 2011, vol. 2, n^{o} 1, p. 23-45 [*DOI :* 10.1007/s13174-011-0021-3], http://hal.inria.fr/inria-00588753/en.

Publications of the year

Articles in International Peer-Reviewed Journal

[7] V. Issarny, N. Georgantas, S. Hachem, A. Zarras, P. Vassiliadis, M. Autili, M. A. Gerosa, A. Ben Hamida. *Service-Oriented Middleware for the Future Internet: State of the Art and Research Directions*, in "Journal of Internet Services and Applications", May 2011, vol. 2, n^{o} 1, p. 23-45 [*DOI :* 10.1007/s13174-011-0021-3], http://hal.inria.fr/inria-00588753/en.

[8] C. Pham, A. Makhoul, R. Saadi. *Risk-based adaptive scheduling in randomly deployed video sensor networks for critical surveillance applications*, in "J. Netw. Comput. Appl.", March 2011, vol. 34, p. 783–795, http://dx.doi.org/10.1016/j.jnca.2010.10.002.

[9] I. Schaefer, R. Rabiser, D. Clarke, L. Bettini, D. Benavides, G. Botterweck, A. Pathak, S. Trujillo, K. Villela. *Software Diversity – State of the Art and Perspectives*, in "International Journal on Software Tools for Technology Transfer", 2012, to appear.

Invited Conferences

[10] T. TEIXEIRA, S. HACHEM, V. ISSARNY, N. GEORGANTAS. *Service Oriented Middleware for the Internet of Things: A Perspective*, in "ServiceWave 2011", Poznan, Poland, Springer-Verlag, 2011, p. 220-229, http://hal.inria.fr/inria-00632794/en.

International Conferences with Proceedings

[11] E. ANDRIESCU, R. SPEICYS CARDOSO, V. ISSARNY. *AmbiStream: A Middleware for Multimedia Streaming on Heterogeneous Mobile Devices*, in "ACM/IFIP/USENIX 12th International Middleware Conference", Lisbon, Portugal, A.-M. KERMARREC, F. KON (editors), Lecture Notes in Computer Science, Springer, December 2011, vol. 7049, http://hal.inria.fr/hal-00639633/en.

[12] D. ATHANASOPOULOS, A. ZARRAS, V. ISSARNY. *ForeverSOA: Towards the Maintenance of Service Oriented Software*, in "SQM 2011 - Fifth CSMR International Workshop on Software Quality and Maintainability", Oldenburg, Germany, 2011, http://hal.inria.fr/inria-00628706/en.

[13] D. ATHANASOPOULOS, A. ZARRAS, P. VASSILIADIS, V. ISSARNY. *Mining Service Abstractions*, in "NIER Track of 33rd International Conference on Software Engineering (ICSE)", Hawaii, United States, May 2011, 4, http://hal.inria.fr/inria-00583780/en.

[14] N. BENCOMO. *Run-time Resolution of Uncertainty*, in "International Requirements Engineering Conference, RE 2011", Trento, Italy, August 2011, http://hal.inria.fr/inria-00623785/en.

[15] N. BENCOMO. *Tracing Requirements for Adaptive Systems using Claims*, in "6th International Workshop on Traceability in Emerging Forms of Software Engineering", Honolulu, United States, May 2011 [*DOI :* 10.1145/1987856.1987865], http://hal.inria.fr/inria-00623812/en.

[16] N. BENCOMO, K. WELSH, P. SAWYER. *Towards Requirements Aware Systems: Run-time Resolution of Design-time Assumptions*, in "ASE'11 - 26th IEEE/ACM International Conference On Automated Software Engineering", Lawrence, Kansas, United States, November 2011, http://hal.inria.fr/inria-00623780/en.

[17] A. BENNACEUR, J. RICHARD, M. ALESSANDRO, S. ROMINA, D. SYKES, R. SAADI, V. ISSARNY. *Inferring Affordances Using Learning Techniques*, in "International Workshop on Eternal Systems (EternalS'11)", Budapest, Hungary, 2011, http://hal.inria.fr/inria-00591264/en.

[18] G. BLAIR, A. BENNACEUR, N. GEORGANTAS, P. GRACE, V. ISSARNY, V. NUNDLOLL, M. PAOLUCCI. *The Role of Ontologies in Emergent Middleware: Supporting Interoperability in Complex Distributed Systems*, in "Big Ideas track of ACM/IFIP/USENIX 12th International Middleware Conference", Lisbon, Portugal, 2011, http://hal.inria.fr/inria-00629059/en.

[19] N. GEORGANTAS, M. A. RAHAMAN, H. AMEZIANI, A. PATHAK, V. ISSARNY. *A Coordination Middleware for Orchestrating Heterogeneous Distributed Systems*, in "International Conference on Grid and Pervasive Computing, GPC 2011", 2011, p. 221-232.

[20] S. HACHEM, T. TEIXEIRA, V. ISSARNY. *Ontologies for the Internet of Things*, in "Doctoral Symposium at ACM/IFIP/USENIX 12th International Middleware Conference", Lisbon, Portugal, Springer, 2011, http://hal.inria.fr/hal-00642193/en.

[21] S. HACHEM, A. TONINELLI, A. PATHAK, V. ISSARNY. *Policy-based Access Control in Mobile Social Ecosystems*, in "IEEE International Symposium on Policies for Distributed Systems and Networks", Pisa, Italy, IEEE computer society, 2011, http://hal.inria.fr/inria-00608201/en.

[22] P. PATEL, A. PATHAK, T. TEIXEIRA, V. ISSARNY. *Towards Application Development for the Internet of Things*, in "Doctoral Symposium at ACM/IFIP/USENIX 12th International Middleware Conference", Lisbon, Portugal, Springer, December 2011.

[23] R. SAADI, M. A. RAHAMAN, V. ISSARNY, A. TONINELLI. *Composing Trust Models towards Interoperable Trust Management*, in "5th IFIP WG 11.11 International Conference on Trust Management", Copenhagen, Denmark, I. WAKEMAN, E. GUDES, C. JENSEN, J. CRAMPTON (editors), Springer Boston, June 2011, vol. 358, p. 51-66, http://hal.inria.fr/inria-00617629/en.

[24] A. SEYEDI, R. SAADI, V. ISSARNY. *Proximity-Based Trust Inference for Mobile Social Networking*, in "IFIPTM 2011 - 5th IFIP WG 11.11 International Conference on Trust Management", Copenhagen, Denmark, I. WAKEMAN, E. GUDES, C. JENSEN, J. CRAMPTON (editors), IFIP Advances in Information and Communication Technology, Springer Boston, June 2011, vol. 358, p. 253-264 [*DOI :* 10.1007/978-3-642-22200-9_20], http://hal.inria.fr/inria-00617630/en.

[25] A. TONINELLI, A. PATHAK, V. ISSARNY. *Yarta: A Middleware for Managing Mobile Social Ecosystems*, in "International Conference on Grid and Pervasive Computing, GPC 2011", 2011, p. 209-220.

Scientific Books (or Scientific Book chapters)

[26] G. BLAIR, M. PAOLUCCI, P. GRACE, N. GEORGANTAS. *Interoperability in Complex Distributed Systems*, in "11th International School on Formal Methods for the Design of Computer, Communication and Software Systems: Connectors for Eternal Networked Software Systems", M. BERNARDO, V. ISSARNY (editors), Springer, 2011, http://hal.inria.fr/inria-00629057/en.

[27] G. COSTA, F. MARTINELLI, V. ISSARNY, R. SAADI, I. MATTEUCCI. *Security and Trust*, in "11th International School on Formal Methods for the Design of Computer, Communication and Software Systems: Connectors for Eternal Networked Software Systems", Bertinoro, Italy, M. BERNARDO, V. ISSARNY (editors), Lecture notes in computer science, Springer, Bertinoro, Italy, 2011, vol. 6659, p. 393-416, http://hal.inria.fr/hal-00646860/en.

[28] 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, 2011, vol. 11, p. 197-219 [*DOI :* 10.4018/978-1-61692-857-5.CH011], http://hal.inria.fr/inria-00432675/en.

[29] M. FREDJ, A. ZARRAS, N. GEORGANTAS, V. ISSARNY. *Dynamic Maintenance of Service Orchestrations*, in "Service Intelligence and Service Science: Evolutionary Technologies and Challenges", IGI Global, January 2011 [*DOI :* 10.4018/978-1-61520-819-7], http://hal.inria.fr/inria-00628561/en.

[30] P. GRACE, N. GEORGANTAS, A. BENNACEUR, G. BLAIR, F. CHAUVEL, V. ISSARNY, M. PAOLUCCI, R. SAADI, B. SOUVILLE, D. SYKES. *The CONNECT Architecture*, in "11th International School on Formal Methods for the Design of Computer, Communication and Software Systems: Connectors for Eternal Networked Software Systems", M. BERNARDO, V. ISSARNY (editors), Lecture notes in computer science, Springer, 2011, vol. 6659, p. 27-52 [*DOI :* 10.1007/978-3-642-21455-4], http://hal.inria.fr/inria-00586638/en.

[31] V. Issarny, A. Bennaceur, Y.-D. Bromberg. *Middleware-layer Connector Synthesis: Beyond State of the Art in Middleware Interoperability*, in "11th International School on Formal Methods for the Design of Computer, Communication and Software Systems: Connectors for Eternal Networked Software Systems", M. Bernardo, V. Issarny (editors), Lecture notes in computer science, Springer, 2011, vol. 6659, p. 217-255 [*DOI :* 10.1007/978-3-642-21455-4], http://hal.inria.fr/inria-00586630/en.

Books or Proceedings Editing

[32] M. Bernardo, V. Issarny (editors). *Formal Methods for Eternal Networked Software Systems - 11th International School on Formal Methods for the Design of Computer, Communication and Software Systems, SFM 2011, Bertinoro, Italy, June 13-18, 2011. Advanced Lectures*, Lecture Notes in Computer Science, Springer, 2011, vol. 6659.

[33] G. Damianos, B. Paolo, C. Jiannong, V. Issarny (editors). *Journal of Systems and Software, Special Issue on Mobile Applications: Status and trends*, Springer, 2011, vol. 84, n^{o} 11, http://hal.inria.fr/inria-00616306/en.

Other Publications

[34] *CHOReOS consortium. CHOReOS Deliverables. IP CHOReOS EU project, FP7 grant agreement number 257178*, 2011, http://www.choreos.eu/.

[35] Connect *consortium.* Connect *Deliverables. FET IP* Connect *EU project, FP7 grant agreement number 231167*, 2011, http://www.connect-forever.eu/.