



Activity Report 2013

Project-Team ARLES

Software architectures and distributed systems

RESEARCH CENTER
Paris - Rocquencourt

THEME
Distributed Systems and Services

Table of contents

1. Members	1
2. Overall Objectives	2
2.1. Overall Objectives	2
2.2. Highlights of the Year	3
3. Research Program	3
3.1. Introduction	3
3.2. Engineering Pervasive Software Systems	3
3.2.1. Middleware-based Software Engineering	4
3.2.2. Beyond Middleware-based Architectures for Interoperability	4
3.3. Middleware Architectures for Pervasive Computing	5
3.3.1. Service Oriented Middleware	5
3.3.2. Middleware for Wireless Sensor Networks	7
4. Application Domains	9
5. Software and Platforms	9
5.1. Introduction	9
5.2. iCONNECT – Emergent Middleware Enablers	9
5.3. Service-oriented Middleware for Pervasive Computing	10
5.4. XSB – eXtensible Service Bus for the Future Internet	11
5.5. MobIoT – Service-oriented Middleware for the Mobile IoT	11
5.6. Srijan: Data-driven Macroprogramming for Sensor Networks	12
5.7. Yarta: Middleware for supporting Mobile Social Applications	12
5.8. iBICOOP: Mobile Data Management in Multi-* Networks	13
6. New Results	13
6.1. Introduction	13
6.2. Emergent Middleware	14
6.3. Service-oriented Computing in the Future Internet	16
6.4. Service-oriented Middleware for the Mobile Internet of Things	17
6.5. Composing Applications in the Internet of Things	17
6.6. Lightweight Streaming Middleware for the Internet of Things	18
6.7. Dynamic Decision Networks for Self-Adaptive Systems	19
7. Partnerships and Cooperations	20
7.1. National Initiatives	20
7.1.1. ANR	20
7.1.2. Inria Support	20
7.1.2.1. Inria ADT iConnect	20
7.1.2.2. Inria ADT Yarta	21
7.2. European Initiatives	21
7.2.1. FP7 Projects	21
7.2.1.1. FP7 ICT IP CHOReOS	21
7.2.1.2. FP7 ICT NoE NESSoS	22
7.2.1.3. FP7 ICT CA EternalS	23
7.2.1.4. FP7 PEOPLE Requirements@run.time	23
7.2.2. Collaborations in European Programs, except FP7	23
7.3. International Initiatives	24
7.3.1. Inria International Labs	24
7.3.2. Participation in other International Programs	24
7.4. International Research Visitors	24
8. Dissemination	25
8.1. Scientific Animation	25

8.1.1.	Programme Committees of Conferences and Workshops	25
8.1.2.	Leadership Services in Academic Events and Edited Journals	25
8.1.3.	Other Academic Services	26
8.2.	Teaching - Supervision - Juries	26
8.2.1.	Teaching	26
8.2.2.	Supervision	26
8.2.3.	Juries	27
8.3.	Popularization	27
9.	Bibliography	27

Project-Team ARLES

Keywords: Middleware, Software Engineering, Internet Of Things, Distributed Mobile Systems, Service Oriented Computing, Future Internet, Pervasive Computing

Creation of the Project-Team: 2002 February 01, end of the Project-Team: 2013 December 31.

1. Members

Research Scientists

Valérie Issarny [Team leader, Inria, Senior Researcher, On-leave at UC Berkeley-EECS-PATH & CITRIS since Aug 2013, HdR]
Nelly Bencomo [FP7-Marie Curie Fellow, until May 2013]
Nikolaos Georgantas [Inria, Researcher]
Animesh Pathak [Inria, Researcher]

External Collaborator

Françoise Sailhan [CNAM, Associate Professor, from Apr 2013]

Engineers

Aness Bajia [Inria, ANR MURPHY project, from Aug 2013]
Yesid Jarna Alvis [Inria, FP7 CHOREOS project, until Sep 2013]
Kim Hwa Khoo [Inria, FP7 CHOREOS project, Mar-Dec 2013]
Cong Kinh Nguyen [Inria, FP7 CHOREOS project, until Dec 2013]
Georgios Mathioudakis [Inria, FP7 CHOREOS project & ADT iCONNECT]
Bachir Moussa Tari Bako [Inria, FP7 CHOREOS project, until Jan 2013]
George Rosca [Inria, ADT Yarta]

PhD Students

Emil Andriescu [UPMC, CIFRE Ambientic-Inria]
Amel Bennaceur [UPMC, until Jul 2013]
Benjamin Billet [UVSQ]
Georgios Bouloukakis [UPMC, from Oct 2013]
Sara Hachem [UVSQ]
Pankesh Patel [UPMC, until Nov 2013]

Post-Doctoral Fellows

Oleg Davidyuk [ERCIM, until Jul 2013]
Ajay Kattepur [Inria]

Visiting Scientist

Peter Sawyer [Visiting Professor, until Apr 2013]

Administrative Assistants

Florence Barbara [Inria, until Jul 2013]
Cindy Crossouard [Inria, from Oct 2013]
Emmanuelle Grousset [Inria, *par intérim*, Jul-Oct 2013]

Others

Aness Bajia [Inria, Research intern, Feb-Jul 2013]
Amel Belaggoun [Inria, Research intern, until Aug 2013]
Ankur Gautam [Inria, Research intern, until Feb 2013]
Yijun Liu [Inria, Research intern, Jun-Sep 2013]
Akash Nawani [Inria, Research intern, Jan-Mar 2013]
Dimitrios Soukaras [Inria, Research intern, Feb-Apr 2013]

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 infrastructures. 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 task¹. 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 computing² and subsequent pervasive computing³ and ambient intelligence⁴ visions, keeping in view the advances in the constitution of these systems, both in terms of the various communication and discovery protocols used, as well as the hardware and software platforms available. Our research is more specifically focused on eliciting middleware for pervasive computing, from foundations and architectural design to prototype implementations. Proposed middleware and programming abstractions shall then effectively leverage networked resources, in particular accounting for advanced wireless networking technologies, while also addressing concerns raised due to the presence of large numbers of extremely low-power devices such as wireless sensors and actuators. This raises a number of complementary research challenges:

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

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

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

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

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

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

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

2.2. Highlights of the Year

This year has seen the following acknowledgments of the team's contributions:

PRIZES:

- Valérie Issarny was awarded one of the twelve "Etoiles de l'Europe" for the year 2013. The prize rewards French teams that coordinate European projects as part of the research and innovation framework program, which Valérie received for the FP7 ICT FET CONNECT (Emergent Connectors for Eternal Software-intensive Networked Systems – <https://www.connect-forever.eu/>) project that examined issues facing the Future Internet.
- Animesh Pathak, Sara Hachem, Giorgios Mathioudakis, and George Rosca were awarded the Best Mashup prize of the OpenDataLab organized by RATP, for their "neverBLate" app.

BEST PAPER AWARD :

[12] **Supporting Decision-making for Self-adaptive Systems: From Goal Models to Dynamic Decision Networks in REFSQ 2013 - 9th International Working Conference on Requirements Engineering: Foundation for Software Quality.** N. BENCOMO, A. BELAGGOUN.

3. Research Program

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 the team 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 connectors, a.k.a *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.

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, 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

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

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 systems⁶. 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 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. The approach specifically introduces the *emergent middleware* paradigm, from formal foundations to enabling software tools [2].

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 the middleware paradigms that we leverage in our work: (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 computing⁷. 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. In the following, we discuss the role that *service oriented middleware* is aimed to have within our above sketched vision of the Future Internet idiom [6], 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

⁶Valérie 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.

⁷Valérie Issarny, Daniele Sacchetti, Ferda Tartanoglu, Françoise Sailhan, Rafik Chibout, Nicole Lévy, Angel Talamona: Developing Ambient Intelligence Systems: A Solution based on Web Services. Autom. Softw. Eng. 12(1): 101-137 (2005)

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 [1], heterogeneity of service discovery protocols, and heterogeneity of network interfaces [3]. 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 abstractions⁸. 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 (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 sensors 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*⁹ 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

⁸L. 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.

⁹Lane, 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

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].

4. Application Domains

4.1. Pervasive Software Applications

The ARLES project-team is interested in the application of pervasive computing, and as such considers various application domains, especially considering the increasing pervasiveness of the digital world. 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).

5. Software and Platforms

5.1. Introduction

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

5.2. iCONNECT – Emergent Middleware Enablers

Participant: Valérie Issarny [correspondent].

As part of our research work on Emergent Middleware, we have implemented Enablers (or Enabler functionalities) that make part of the overall CONNECT architecture realizing Emergent Middleware in practice [2]. The focus of ARLES work is on the: *Discovery enabler* that builds on our extensive background in the area of interoperable pervasive service discovery; and *Synthesis enabler* that synthesizes mediators that allow Networked Systems (NSs) that have compatible functionalities to interact despite mismatching interfaces and/or behaviors.

The 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.

The Synthesis Enabler assumes semantically-annotated system descriptions *à la* OWL-S, which are made available by the Discovery Enabler, together with a domain ontology, and produces the mediators that enable functionally compatible networked systems to interoperate. The semantically-annotated interfaces of the NSs that need to communicate are processed to compute the semantic mapping between their respective operations using a constraint solver. The resulting mapping serves generating a mediator that coordinates the behaviors of the NSs and guarantees their successful interaction. Only when the mediator includes all the details about the communication of NSs, can interoperability be achieved, which calls for the adequate concretization of synthesized mediators.

The *concretization of mediators* bridges the gap between the application level, which provides the abstraction necessary to reason about interoperability and synthesize mediators, and the middleware-level, which provides the techniques necessary to implement these mediators. Concretization entails the instantiation of the data structures expected by each NS and their delivery according to the interaction pattern defined by the middleware, based on which the NS is implemented. Therefore, we have been developing a mediation engine that, besides executing the data translations specified by the mediator, generates composed parsers and composers, which can process complex messages, by relying on existing libraries associated with standard protocols and state-of-the-art middleware solutions.

The Discovery and Synthesis Enablers have been integrated and experimented with by the CONNECT consortium to effectively enable Emergent Middleware. Part of them are available for download under an open source license at the CONNECT Web site at <https://www.connect-forever.eu/software.html>.

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.
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. XSB – eXtensible Service Bus for the Future Internet

Participant: Nikolaos Georgantas [correspondent].

The eXtensible Service Bus (XSB) is a development and runtime environment dedicated to complex distributed applications of the Future Internet. Such applications will be based, to a large extent, 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). Such heterogeneous systems are supported by enabling middleware platforms, particularly for their interaction. With regard to middleware-supported interaction, the client-service (CS), publish-subscribe (PS), and tuple space (TS) paradigms are among the most widely employed ones, with numerous related middleware platforms, such as: Web Services, Java RMI for CS; JMS, SIENA for PS; and JavaSpaces, Lime for TS. XSB then provides support for the seamless integration of heterogeneous interaction paradigms (CS, PS and TS).

In a nutshell, our systematic interoperability approach implemented by the proposed XSB is carried out in two stages. First, a middleware platform is abstracted under a corresponding interaction paradigm among the three base ones, i.e., CS, PS and TS. To this aim, we have elicited a connector model for each paradigm, which comprehensively covers its essential semantics. Then, these three models are abstracted further into a single generic application (GA) connector model, which encompasses their common interaction semantics. Based on GA, we build abstract connector converters that enable interconnecting the base interaction paradigms.

Following the above, XSB is an abstract service bus that prescribes only the high-level semantics of the common bus protocol, which is the GA semantics. Furthermore, we provide an implementation of the XSB, building upon existing SOA and ESB realizations. XSB features richer interaction semantics than common ESBs 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. Services relying on different interaction paradigms can be plugged into XSB by employing binding components (BCs) that adapt between their native middleware and the common bus protocol. This adaptation is based on the abstractions, and in particular on the conversion between the native middleware, the corresponding CS/PS/TS abstraction, and the GA abstraction.

Furthermore, we provide a companion implementation, named Light Service Bus (LSB), targeting the Internet of Things(IoT) domain. LSB forms a concrete access solution for IoT systems as it is able to cope with the diversity of the involved interaction protocols and take care of the IoTS specifics, such as resource constraints, dynamic environments, data orientation, etc. It is implemented to be lightweight in nature and uses REST as the common protocol/bus in place of an ESB solution. In LSB, we confirm the wide use of the aforementioned interaction paradigms (CS/PS/TS) but also underline the existence of an additional paradigm focused on continuous interaction known as Streaming (STR).

Both the XSB and LSB solutions are available for download under open source licenses at <http://xsb.inria.fr> and <http://websvn.ow2.org/listing.php?reparent=choreos&path=%2Ftrunk%2Fextensible-service-access%2Flsb%2FLsbBindingComponents%2F> respectively.

5.5. MobIoT – Service-oriented Middleware for the Mobile IoT

Participant: Valérie Issarny [correspondent].

MobIoT is a service-oriented middleware aimed at the mobile Internet of Things (IoT), which in particular deals with the ultra-large scale, heterogeneity and dynamics of the target networking environment. MobIoT offers novel probabilistic service discovery and composition approaches, and wraps legacy access protocols to be seamlessly executed by the middleware. The middleware exposes two levels of service abstractions: Thing as a service (on the service provider side); and Things measurements/actions as a service (on the service consumer side).

Key features of MobIoT lie in: (i) the exploitation of ontologies to overcome the heterogeneity of the Things network, (ii) the introduction of probabilistic approaches for both registering and retrieving networked things so as to filter out the ones that are redundant with already known alternatives, and finally, (iii) the exploitation of Thing services composition for responding to user queries asking information about the physical world so as to ease interaction with such a complex and dynamic networking environment.

MobIoT is implemented using Java and the Android platform, and consists of two complementary components: The MobIoT Mobile middleware and the MobIoT Web Service. The MobIoT Mobile middleware is deployed on mobile devices (e.g., smartphones, tablets, sensor devices). It wraps: (i) the Query component that enables the querying of the physical world, (ii) the Registration component that deals with the probabilistic registration of local sensors and actuators, (iii) the domain ontology that allows reasoning about the features of Things, and (iv) the Sensor Access component that enables the sensor data retrieval and exposure. The MobIoT Web Service wraps: (i) the Registry component that keeps tracks of the registered services, (ii) the probabilistic Lookup component that allow retrieving relevant services in a scalable way, and (iii) the Composition & Estimation component to answer queries over the physical world using available Thing services, and finally domain and devices ontologies.

The MobIoT middleware is available for download under an open source license at http://choreos.eu/bin/Documentation/IoTS_Middleware.

5.6. 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/>.

5.7. 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/>.

5.8. 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.

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.

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 2013 have in particular accounted for the increasingly connected networking environment, as envisioned by the Future Internet, and further focused on one of its major components that is the Internet of Things, which allows connecting the physical with the digital world. In more detail, our research has 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);
- Revisiting service-oriented computing toward the Future Internet, in particular dealing with the composition of highly heterogeneous services while ensuring quality of service (§ 6.3);
- Service oriented middleware for the ultra large scale future mobile Internet of Things (§ 6.4);
- Abstractions for enabling domain experts to easily compose applications on the Internet of Things (§ 6.5);
- Lightweight streaming middleware for the Internet of Things (§ 6.6); and
- Dynamic decision networks for decision-making in self-adaptive systems (§ 6.7).

6.2. Emergent Middleware

Participants: Emil Andriescu, Amel Bennaceur, Valérie Issarny.

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. At the application layer, components may exhibit disparate data types and operations, and may have distinct business logics. At the middleware layer, they may rely on different communication standards, which define disparate data representation formats and induce different architectural constraints. Finally, at the network layer, data may be encapsulated differently according to the network technology in place.

A wide range of approaches have thus been proposed to address the interoperability challenge, as surveyed in [26]. However, solutions that require performing changes to the systems are usually not feasible since the systems to be integrated may be built by third parties (e.g., COTS—Commercial Off-The-Shelf—components or legacy systems); no more appropriate are 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 investigated. 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; correct mediators guarantee that the components interact without errors (e.g., deadlocks) and reach their termination successfully. Therefore, formal approaches are used to synthesize mediators automatically.

We posit that interoperability should neither be achieved by defining yet another middleware nor yet another ontology but rather by exploiting existing middleware together with knowledge encoded in existing domain ontologies to synthesize and implement mediators automatically. In [2], we have introduced the notion of *emergent middleware* for realizing mediators, which was initiated as part of the FP7 FET IP CONNECT project. Our work during the year 2013 has more specifically focused on the further elaboration of a comprehensive approach to mediator synthesis, including dealing with interoperability across protocol layers.

Mediator synthesis for emergent middleware: We focus on functionally-compatible components, i.e., components that at some high level of abstraction require and provide compatible functionalities, but are unable to interact successfully due to mismatching interfaces and behaviors. To address these differences without changing the components, mediators that systematically enforce interoperability between functionally-compatible components by mapping their interfaces and coordinating their behaviors are required. Our approach for the automated synthesis of mediators is performed in several steps.

The first step is interface matching, which identifies the semantic correspondence between the actions required by one component and those provided by the other. We incorporate the use of ontology reasoning within constraint solvers, by defining an encoding of the ontology relations using arithmetic operators supported by widespread solvers, and use it to perform interface matching efficiently. For each identified correspondence, we generate an associated matching process that performs the necessary translations between the actions of the two components' interfaces. The second step is the synthesis of correct-by-construction mediators. To do so, we analyze the behaviors of components so as to generate the mediator that combines the matching processes in a way that guarantees that the two components progress and reach their final states without errors. The synthesized mediator is the most general component that ensures freedom of both communication mismatches and deadlock in the composition of the components [15]. The last step consists in making the synthesized mediator concrete by incorporating all the details about the interaction of components. To do so, we compute the translation functions necessary to reconcile the differences in the syntax of the input/output data used by each component and coordinate the different interaction patterns that can be used by middleware solutions.

We refer the interested reader to [7] for a complete description of the approach. Our contribution primarily lies in handling interoperability from the application to the middleware layer in an integrated way. The mediators we synthesize act as: (i) translators by ensuring the meaningful exchange of information between components, (ii) controllers by coordinating the behaviors of the components to ensure the absence of errors in their interaction, and (iii) middleware by enabling the interaction of components across the network so that each component receives the data it expects at the right moment and in the right format.

Automated mediation for cross-layer protocol interoperability: Existing approaches to interoperability are restricted to solving either application heterogeneity when the underlying middlewares are compatible, or solving middleware heterogeneity at each protocol layer separately. In real world scenarios, this does not suffice: application and middleware boundaries are ill-defined and solutions to interoperability must consider them in conjunction. We have been studying the case of cross-layer interoperability where protocol mediation is performed between protocol stacks, rather than between protocol layers separately. Such interoperability approaches are appropriate for systems that rely on complex protocol stacks, where application and middleware layers are tightly coupled.

Systems relying on tightly coupled protocol stacks exchange complex messages that consist of a composition of heterogeneous data formats. To enable interoperation, complex messages from one system must be translated into a different complex format that another system accepts such that the two can interact. While Off-The-Shelf and third party message parsers are widely available for simple message formats (i.e., message formats corresponding to a single protocol layer), complex message formats are typically unique since they are the result of a protocol binding. Protocol binding represents the connection between one protocol and another to create a new communication flow. Some middleware protocols recommend or restrict to certain types of default binding (e.g., HTTP provides an extensive set of rules for binding, such as Content-Encoding and Content-Type). However, real systems are often designed following a custom binding mechanism, restricting the application of automated mediation solutions. This problem occurs primarily because complex message formats cannot be easily interpreted.

Many solutions address this composition issue by introducing Domain Specific Languages that can be used by experts to specify parsers for complex message formats. Yet, whenever messages have a more complicated syntax, providing their DSL descriptions becomes difficult as well. Further, such approaches are not future proof as more protocols are expected to emerge, which will not be accounted for by DSLs that are defined according to known message formats. An alternative is to generate parsers based on the composition of third-party parsers that are usually included with protocol implementations. However, third-party parsers cannot be used unless the protocol binding rules are identified by an expert, further allowing to implement the bridge between one parser's output data and the other parser's input data. To this end, we designed an approach for generating composed parsers that can process complex messages, accompanied by a formal mechanism for defining complex message formats based on existing data formats. Our approach relies on user-provided parser composition rules, which reflect the binding requirements of complex message formats.

We posit that our method is more efficient than implementing complex parsers, defining them using DSLs, or directly implementing the binding of protocols. Furthermore, with this solution, we support the automated synthesis of mediators at the application layer using the mapping-based approach discussed above, by automatically generating an abstract representation of the application data exchanged by the interoperating components.

6.3. Service-oriented Computing in the Future Internet

Participants: Georgios Bouloukakis, Nikolaos Georgantas, Valérie Issarny, Ajay Kattapur.

With an increasing number of services and devices interacting in a decentralized manner, *choreographies* represent a scalable framework for the Future Internet. The service oriented architecture inherent to choreographies allows abstracting multiple devices as components, that interact through middleware connectors via standard protocols. However, the heterogeneous nature of devices leads to choreographies that not only include conventional services, but also sensor-actuator networks, databases and service feeds. We reason about their behavior through abstract middleware interaction paradigms, such as client-service (CS), publish-subscribe (PS) and tuple space (TS), made interoperable through the *eXtensible Service Bus* (XSB) connector.

Extensible Service Bus for the Future Internet: XSB is an abstract service bus that deals effectively with the cross-integration of heterogeneous interaction paradigms [17]. Inside the XSB, the CS, PS and TS paradigms are modeled as abstract base connectors. Their *space coupling* semantics are represented with programming interfaces used by applications (APIs) and corresponding application interface description languages (IDLs). Their behavioral semantics are formally specified in terms of LTS (Labeled Transition Systems). We formally verify the correctness of these behavioral specifications with respect to *time coupling* and *concurrency* properties expressed in LTL temporal logic. This allows stating the correctness of the connector models with respect to the semantics that they must have. This further enables identifying the behavioral semantics of the XSB connector derived from the interconnection of base connectors. More specifically, in order to identify the time coupling and concurrency semantics of XSB and construct a converter among the base connectors, we build upon the formal method of *protocol conversion via projections*¹⁰. According to this method, conversion between two different protocols is possible if both protocols can be projected (where projection is an abstraction defined as a set of transformations on the protocol LTS) to a *functionally sufficient* common *image protocol*. Then, the end-to-end protocol of the interconnection of the two protocols is this image protocol.

We have implemented our XSB solution into an extensible development and execution platform for application and middleware designers. Using this platform, they can easily develop composite applications: they only need to build descriptions for the constituent services and directives for data mapping among them. Our platform then deals with reconciling among the heterogeneous interaction paradigms and protocols of the services by employing *binding components* (BCs) that adapt between the native middleware of the services and the XSB bus protocol. The XSB itself is implemented on top of an existing ESB substrate. Support for new middleware platforms, new ESB substrates, or even new interaction paradigms can be incorporated in a facilitated way thanks to the provided XSB architectural framework.

QoS composition and analysis of heterogeneous choreographies: Leveraging on the functional interoperability across interaction paradigms offered by the XSB, we study the Quality of Service (QoS) performance of choreographies [21]. QoS dependency plays an important role in the service oriented system lifecycle, including discovery, runtime selection, replacement and contractual guarantees. Consequently, QoS composition among choreographed devices should tackle multi-dimensional probabilistic metrics combined with message passing constraints imposed at design-time. We make use of an algebraic QoS composition model that is applied at the interaction paradigm level to study the composition of QoS metrics, and the subsequent tradeoffs. While traditional QoS composition analysis has been done purely at the application level, analyzing the effect of middleware interactions allows us to study CS, PS and TS based device compositions. This produces interesting insights such as selection of a particular system and its middleware during design-time, or end-to-end

¹⁰Lam, S.S.: Protocol Conversion. IEEE Trans. Softw. Eng. 14(3) (1988) 353–362.

QoS expectation/guarantees during runtime. Our formulation also allows for runtime reconfiguration, in order to optimally produce design time QoS expectations. Such flexible reconfiguration policies are crucial in the case of large scale choreographies with high variability in runtime performance of participating devices.

Further, we study the effect of time/space coupling on the latency of successful transactions across the XSB connector [20]. XSB models the message passing among peers through generic `post` and `get` operations, that represent peer behavior with both tight (CS) and loose (PS/TS) time/space coupling. The heterogeneous *lease* and *timeout* behaviors of these operations severely affect latency and success rates of messages passed either synchronously or through callbacks. By precisely studying the timing thresholds using timed automata models, we verify conditions for accurate message transactions with XSB connectors. This offers choreography designers the ability to set these timing thresholds (bottom-up) or select a particular interaction paradigm (top-down) for runtime enactment.

6.4. Service-oriented Middleware for the Mobile Internet of Things

Participants: Sara Hachem, Valérie Issarny, Georgios Mathioudakis, Animesh Pathak.

The Internet of Things (IoT) is characterized by an increasing number of Things embedding sensing, actuating, processing, and communication capacities. A considerable portion of those Things will be *mobile* Things, which come with several advantages yet lead to unprecedented challenges. The most critical challenges, that are directly inherited from, yet amplify, today's Internet issues, lie in handling i) the large scale of users and mobile Things, ii) providing interoperability across the heterogeneous Things, and iii) overcoming the unknown dynamic nature of the environment, due to the mobility of an ultra-large number of Things.

Service-Oriented Architecture (SOA) provides solid basis to address the above challenges as it allows the functionalities of sensors/actuators embedded in Things to be provided as services, while ensuring loose-coupling between those services and their hosts, thus abstracting their heterogeneous nature. In spite of its benefits, SOA has not been designed to address the ultra-large scale of the mobile IoT. Consequently, an alternative is provided within a novel Thing-based Service-Oriented Architecture, that revisits SOA interactions and functionalities, service discovery and composition in particular. The novel architecture is concretized within MobIoT, a middleware solution that is specifically designed to manage and control the ultra-large number of mobile Things in partaking in IoT-related tasks.

In accordance with SOA, MobIoT comprises *Discovery*, *Composition & Estimation*, and *Access* components, yet modifies their internal functionalities. In more detail, the Discovery component enables Thing-based service registration (for Things to advertise hosted services) and look-up (for Things to retrieve remote services of interest). In order to handle the ultra large number of mobile Things and their services in the IoT, the component revisits the Service-Oriented discovery and introduces *probabilistic discovery* to provide, not *all*, but only a sufficient *subset of services that can best approximate* the result that is being sought after [18], [11]. Furthermore, the Composition & Estimation component (C&E) provides automatic composition of Thing-based services. This capacity is of interest in the case where no service can perform a required measurement/action task directly (based on its atomic functionalities). Thing-based service composition executes in three phases: i) *expansion*, where composition specifications are automatically identified; ii) *mapping*, where actual service instances (running services) are selected based on their functionalities and the physical attributes of their hosts; and iii) *execution*, where the services are accessed and the composition specifications are executed. Thing-based service composition revisits Service-Oriented composition by executing seamlessly with no involvement from developers or end users. Last but not least, the Access component provides an easy to use interface for developers to sample sensors/actuators while abstracting sensor/actuator hardware specifications. Additionally, it revisits Service-Oriented access by executing access to services transparently and wrapping access functionalities internally. Thus, it alleviates that burden from users, initially in charge of this task. The Access component supports access to remote services and to locally hosted services.

6.5. Composing Applications in the Internet of Things

Participants: Aness Bajia, Pankesh Patel, Animesh Pathak, Françoise Sailhan.

As introduced above, the Internet of Things 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. 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 goal of our research is then to propose a suitable application development framework, where our work this year covered the two following related areas.

Multi-stage model-driven approach for IoT application development: We have proposed a multi-stage model-driven approach for IoT application development based on a precise definition of the role to be played by each stakeholder involved in the process: domain expert, application designer, application developer, device developer, and network manager [22]. The metamodels/abstractions available to each stakeholder are further customized using the inputs provided in the earlier stages by other stakeholders. We have also implemented code-generation and task-mapping techniques to support our approach. Our evaluation based on two realistic scenarios shows that the use of our techniques/framework succeeds in improving productivity in the IoT application development process. More details of our approach can be found in [8].

Integrating support for non-functional requirements while programming IoT applications: Given that devices and networks constituting the IoT are prone to failure and consequent loss of performance, it is natural that IoT applications are expected to encounter and tolerate several classes of faults - something that still largely remains within the purview of low-level-protocol designers. As part of our work on the MURPHY project (§ 7.1.1.1), we are addressing this issue by proposing: i) a set of abstractions that can be used during macroprogramming to express fault tolerance requirements, and ii) a runtime system that employs adaptive fault tolerance (AFT) to provide fault tolerance to the sensing application. Complementary to this, we have proposed task mapping algorithms to satisfy those requirements through a constraint programming approach [19]. Through evaluations on realistic application task graphs, we show that our constraint programming model can effectively capture the end-to-end requirements and efficiently solves the combinatorial problem introduced.

We have continually incorporating our research results in the above areas into *Srijan* (§ 5.6), 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. Lightweight Streaming Middleware for the Internet of Things

Participants: Benjamin Billet, Valérie Issarny.

The Internet of Things (IoT) is a promising concept toward pervasive computing as it may radically change the way people interact with the physical world. One of the challenges raised by the IoT is the in-network continuous processing of data streams presented by Things, which must be investigated urgently because it affects the future data models of the IoT. This cross-cutting concern has been previously studied in the context of Wireless Sensor and Actuator Networks (WSAN) given the focus on the acquisition and in-network processing of sensed data. However, proposed solutions feature heterogeneous technologies that are difficult to integrate and complex to use, which represents a hurdle to their wide deployment. In addition, new types of smart sensors are emerging due to technological advances (e.g., Oracle SunSpot), enabling the implementation of complex processing tasks directly into the network, without using proxies or sending every data to the cloud. There is thus a need for a distributed middleware solution for data stream management that leverages existing WSAN work, while integrating it with today's Web technologies in order to improve the flexibility and the interoperability of the future IoT. Toward that goal, we have been developing Diopbase, a Data Stream Management System for the IoT, which aims to integrate the Things and their streams into today's Web by presenting sensors and actuators as services. The middleware specifically provides a way to describe complex

fully-distributed stream-based mashups and to deploy them dynamically, at any time, as task graphs, over available Things of the network, including resource-constrained ones. To this end, Diopbase enables task graphs to be composed of Thing-specific tasks (directly implemented on the Thing) and dynamic tasks that communicate using data streams. Dynamic tasks are then described in a lightweight DSL, which is directly interpreted by the middleware and provides specific primitives to manipulate data streams.

As part of the design of Diopbase, we have been investigating dedicated task mapping. Task mapping, which basically consists of mapping a set of tasks onto a set of nodes, is a well-known problem in distributed computing research. However, as a particular case of distributed systems, the Internet of Things (IoT) poses a set of renewed challenges, because of its scale, heterogeneity and properties traditionally associated with WSN, shared sensing, continuous processing of data streams and real time computing. To handle IoT features, we present a formalization of the task mapping problem that captures the varying consumption of resources and various constraints (location, capabilities, QoS) in order to compute a mapping that guarantees the lifetime of the concurrent tasks inside the network and the fair allocation of tasks among the nodes (load balancing). It results in a binary programming problem for which we provide an efficient heuristic that allows its resolution in polynomial time. Our experiments show that our heuristic: (i) gives solutions that are close to optimal and (ii) can be implemented on reasonably powerful Things and performed directly within the network, without requiring any centralized infrastructure.

6.7. Dynamic Decision Networks for Self-Adaptive Systems

Participants: Amel Belaggoun, Nelly Bencomo, Valérie Issarny, Peter Sawyer.

Different modeling techniques have been used to model requirements and decision-making of self-adaptive systems [25]. Important successful techniques based on goal models have been prolific in supporting decision-making according to partial and total fulfillment of functional (goals) and non-functional requirements (softgoals). The final decision about what strategy to use is based on a utility function that takes into account the weighted sum of the different effects of the non-functional requirements. Such solutions have been used both at design and run time including our own solutions using runtime goal models. Different modeling techniques have been used to model requirements and decision-making of self-adaptive systems [25]. Important successful techniques based on goal models have been prolific in supporting decision-making according to partial and total fulfillment of functional (goals) and non-functional requirements (softgoals). The final decision about what strategy to use is based on a utility function that takes into account the weighted sum of the different effects of the non-functional requirements. Such solutions have been used both at design- and run-time including our own solutions using runtime goal models.

We have enriched the decision-making supported by goal models with the use of Bayesian Dynamic Decision Networks (DDNs) [12]. Our novel approach supports reasoning about partial satisfaction of soft-goals using probabilities and uses machine learning. When using DDNs, we introduce new ways to tackle uncertainty based on probabilities that can be updated based on runtime evidence. We have reported the results of the application of the approach on two different cases, one of them being the case of dynamic reconfiguration of a remote data mirroring network that must spread data among servers while minimizing costs and loss of data. Our early results suggest the decision-making process of self-adaptive systems can be improved by using DDNs.

This work has been developed under the umbrella of the Marie Curie Project Requirements@run.time (§ 7.2.1.4). The main results achieved during the year 2013 are:

- A Bayesian-based technique to support the decision making of self-adaptive systems [14]. DDN-based approaches adopt probabilistic methods (i.e., Bayesian methods) and decision theory to assess the consequences of uncertainty. Using the approach, suitable choices to satisfy functional requirements of the system are identified from a range of alternative decisions and their expected utilities. Satisfaction of NFRs is modeled using conditional probabilities given the design decisions. Preferences over decisions are modeled using weights associated with pairs of design alternatives and NFRs, and used when computing the expected utilities of the architectural design alternatives. The decision taken by the DDN is that with the highest expected utility. The approach offers the benefits

of machine learning.

- A formal Bayesian definition of surprise as the basis for quantitative analysis to measure degrees of uncertainty and deviation of self-adaptive systems from normal behavior [13]. Specifically, a Bayesian surprise quantifies how new evidence affects assumptions of the world (properties in the models). A “surprising” event may provoke a large divergence between the beliefs distributions prior and posterior to that event. As such and depending on how big or small this divergence is, the running system may decide to either: (i) dynamically adapt accordingly, or (ii) temporarily avoid any action of adaptation and flag up the fact that a potential abnormal situation has been found. While doing (ii) we are offering a specific implementation of the RELAX language previously developed by Bencomo and her co-authors.

7. Partnerships and Cooperations

7.1. National Initiatives

7.1.1. ANR

7.1.1.1. ANR MURPHY

Participant: Animesh Pathak [correspondent].

- **Name:** MURPHY – *Dependability-focused Evaluation of Sensor Networks*
- **URL:** <http://cedric.cnam.fr/~sailhanf/murphy/>
- **Related activities:** § 6.5
- **Period:** [January 2011 – September 2014]
- **Partners:** CNAM (**Coordinator**), 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 that 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 enable the dynamic and distributed insertion of faults and hide from the end user the complexity related to this task; and
- 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.

7.1.2. Inria Support

7.1.2.1. Inria ADT iConnect

Participant: Valérie Issarny [correspondent].

- **Name:** iConnect – *Emergent Middleware Enablers*
- **Related activities:** § 6.2 and 6.3
- **Period:** [October 2013 – September 2015]
- **Partners:** Inria ARLES.

The pervasive computing vision is hampered by the extreme level of heterogeneity in the underlying infrastructure, which impacts on the ability to seamlessly interoperate. Further, the fast pace at which technology evolves at all abstraction layers increasingly challenges the lifetime of networked systems in the digital environment.

Overcoming the interoperability challenge in pervasive computing systems has been at the heart of the FP7 FET IP CONNECT project (<http://www.connect-forever.eu>), which ran from 2009 to 2012, and was coordinated by Inria ARLES. Specifically, CONNECT has been investigating the paradigm of *Emergent middleware*, where protocol mediators are dynamically synthesized so as to allow networked systems that provide complementary functionalities to successfully coordinate. The CONNECT project has in particular delivered prototype implementation of key enablers for emergent middleware, spanning discovery, protocol learning, and mediator synthesis and deployment. Further, while CONNECT focused on learning and reconciling interaction protocols at the application layer, the FP7 project CHOReOS (<http://www.choreos.eu>) to which ARLES contributed as well, investigated a complementary enabler that supports interoperability across systems implementing heterogeneous interaction paradigms (i.e., client-service, event-based and shared memory). The proposed enabler introduces the concept of XSB - eXtensible Service Bus, which revisits the notion of Enterprise Service Bus and features an end-to-end interaction protocol that preserves the interaction paradigms of the individual components, while still allowing interoperability.

The objective of the Inria iConnect ADT is to leverage and integrate the above complementary results, packaging and further enhancing enabler prototypes, for take-up of the results by the relevant open source community. The work will involve development effort focused on the following core enablers:

- Universal discovery of resources composing legacy discovery protocols,
- Dynamic synthesis and deployment of mediators specified as enhanced labelled transition systems,
- XSB as underlying run-time support for mediators so as to support interoperability across systems based on heterogeneous interaction paradigms,
- Experiment in the area of federated social networking.

We intend to release the software prototypes through the newly created OW2 open source initiative FISSi (Future Internet Software and Services initiative – http://www.ow2.org/view/Future_Internet/) as our solutions are of direct relevance to sustaining interoperability in the future Internet.

7.1.2.2. Inria ADT Yarta

Participant: Animesh Pathak [correspondent].

- **Name:** Yarta – *Middleware for mobile social ecosystems*
- **Period:** [October 2012 – September 2014]
- **Partners:** Inria ARLES.

Yarta is a middleware for managing mobile social ecosystems, which builds upon existing research in context-awareness in the pervasive computing domain. The work involves development effort in the multi-layer middleware architecture of Yarta, providing the needed functionalities, including: (i) Storage of social data in an interoperable format, using semantic technologies such as RDF; (ii) Extraction of social ties from context (both physical and virtual); (iii) Enforcement of access control to protect social data from arbitrary access; and (iv) A rich set of mobile social ecosystem (MSE) management functionalities, using which mobile social applications can be developed. Specifically, the ADT supports the public open source release and evolution of the Yarta middleware, which is currently a research prototype.

7.2. European Initiatives

7.2.1. FP7 Projects

7.2.1.1. FP7 ICT IP CHOReOS

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

Name: CHOReOS – *Large Scale Choreographies for the Future Internet*

URL: <http://www.choreos.eu/>

Type: COOPERATION (ICT)

Defi: Internet of Services, Software & Virtualisation

Instrument: Integrated Project (IP)

Related activities: § 6.3

Period: [October 2010 - September 2013]

Partners: NoMagic Europe (Lithuania), CEFRIEL (Italy), CNR (Italy), Linagora (France), Inria ARLES [**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 Telecomunicazioni S.p.A (Italy).

CHOReOS aims at assisting the engineering of software service composition 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 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.

7.2.1.2. FP7 ICT NoE NESSoS

Participants: Valérie Issarny [correspondent], Animesh Pathak [correspondent].

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

URL: <http://www.nessos-project.eu>

Type: COOPERATION (ICT)

Defi: Trustworthy ICT

Instrument: Network of Excellence (NoE)

Related activities: § 6

Period: [October 2010 - March 2014]

Partners: Atos Origin (Spain), CNR (Italy) [**coordinator**], ETH Zürich (Switzerland), IMDEA Software (Spain), Inria (EPIs 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 NESSoS on "Engineering Secure Future Internet Software Services and Systems" 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.

7.2.1.3. FP7 ICT CA EternalS

Participant: Valérie Issarny [correspondent].

Name: EternalS – *Trustworthy Eternal Systems via Evolving Software, Data and Knowledge*

URL: <http://www.eternals.eu>

Type: CAPACITIES (ICT)

Defi: FET - Proactive

Instrument: Coordination and Support Action (CSA)

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) [**coordinator**], 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 (CA) 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 are used to conduct a successful CA.

7.2.1.4. FP7 PEOPLE Requirements@run.time

Participant: Nelly Bencomo [correspondent].

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>

Type: PEOPLE

Instrument: Marie Curie Intra-European Fellowships for Career Development (IEF)

Related activities: § 6.7

Period: [May 2011 - May 2013]

Partners: Inria ARLES.

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.

7.2.2. Collaborations in European Programs, except FP7

7.2.2.1. EIT ICT Labs TravelDashboard

Participant: Animesh Pathak [correspondent].

Name: TravelDashboard – *Personalized Mobility to Urban Travelers*

URL: <https://www.rocq.inria.fr/arles/traveldashboard/>

Period: [January 2013 - December 2013]

Partners: Alcatel/Lucent (Ir and Be), Ambientic (F), Inria (CRI Paris-Rocquencourt), Systématique (F), Thales [**coordinator**], Transport for London (UK), UC London (UK).

With over 70% of the world's entire population expected to be living in cities by 2050, supporting citizens' mobility within the urban environment is a priority for municipalities worldwide. Although public multi-modal transit systems, coupled with integrated fare management and road user charging, are necessary to better manage mobility, they are not sufficient. Citizens must be offered personalized travel information, where and when such information is needed to take decisions that will make their journeys more efficient and enjoyable. Notably, such information is not purely qualitative (e.g., bus timetable, live bus tracking), but crucially subjective (e.g., crowdedness of trains, heat of tube platforms, sociability of the coaches). The perception and value attached to this information varies substantially, not only across people (e.g., different tolerance to delays, different perception of crowdedness, different taste in the social environment), but also for the same person in different contexts (e.g., work commute, leisure trip with the family). Thanks to the increased abundance of smart phones (equipped with various types of physical sensors, as well as enabling the users to easily report phenomena), the field of mobile participatory sensing has emerged recently, and can be leveraged towards providing a more fine-grain and up-to-date view of the city's transportation system. In that direction, the TravelDashboard project works towards an open source middleware platform, enriched with personalized mobility services for urban travelers, evaluated via real-life demonstrators assessment, and accompanied by novel business models.

7.3. International Initiatives

7.3.1. Inria International Labs

Valérie Issarny acts as scientific manager of the Inria@Silicon Valley program (<https://project.inria.fr/inria-siliconvalley/>) for the academic year 2013-14, and is on leave at UC Berkeley since August 2013.

7.3.2. Participation in other International Programs

7.3.2.1. International scientific cooperation program Inria/Brazil – Project M@TURE

Participant: Nikolaos Georgantas [correspondent].

Name: M@TURE – *Models @ runtime for self-adaptive pervasive systems*

Instrument: Inria-Brazil cooperation programme

Period: [October 2012 - September 2014]

Partners: Institute of Informatics of Federal University of Goias (Brazil), Inria ARLES.

The overall goal of the M@TURE project is to design, implement and evaluate a novel approach and architecture - comprising conceptual foundations, engineering techniques, and supporting middleware infrastructure - for self-adaptive pervasive systems by building on the notion of Models@run.time. Models@run.time extends the applicability of models and abstractions to the runtime environment. In contrast to design-time models, runtime models are used to reason about the running system taking into account its operating environment, and thus these models enable automating runtime decisions and actions regarding the creation, configuration, and evolution of the system. We in particular focus on the following dimensions and related models: (i) Requirements models making a system requirements-aware at runtime; (ii) Application- and middleware-level interoperability models exposing to an external observer the technological and business features of a system; and (iii) End-user and system engineer models modeling the internal elements of a system at two different abstraction levels. These models are considered both independently and, more importantly, in synergy in order to introduce a comprehensive conceptual and architectural solution for self-adaptive pervasive systems.

7.4. International Research Visitors

7.4.1. Visits of International Scientists

Prof. Peter Sawyer from Lancaster University (UK), visited the ARLES team during Q1 2013, where he investigated how to leverage requirements engineering in the context of distributed software systems, with a special emphasis on the exploitation of requirements@runtime.

7.4.1.1. Internships

Aness Bajia (from Feb. 2013 until Jul. 2013)

Subject: *Fault Tolerance in Sensor Network Macroprogramming*

Institution: Faculté des sciences de Tunis (Tunisia)

Amel Belaggoun (from Jan. 2013 until Aug. 2013)

Subject: *Runtime and Representation of Requirements in Self-Adaptive Systems*

Institution: Université de Versailles Saint-Quentin-en-Yvelines (France)

Ankur Gautam (from Jan. 2013 until Feb. 2013)

Subject: *Semantic Composition of Services in the Internet of Things*

Institution: Indian Institute of Technology, Banaras Hindu University, Varanasi (India)

Yijun Liu (from Jun. 2013 until Sep. 2013)

Subject: *Smartphone-supported Indoor Location System*

Institution: Stanford University (USA)

Akash Nawani (from Jan. 2013 until Mar. 2013)

Subject: *Middleware Support for Federated Social Networking*

Institution: Indian Institute of Technology, Banaras Hindu University, Varanasi (India)

Dimitrios Soukaras (from Feb. 2013 until Apr. 2013)

Subject: *Enabling High-level Application Development in the Internet of Things*

Institution: University of Peloponnese (Greece)

8. Dissemination

8.1. Scientific Animation

8.1.1. Programme Committees of Conferences and Workshops

- Nelly Bencomo is PC chair of SEAMS'13;
- Nelly Bencomo is PC member of the following international conferences: MODELS'13, CibSE'13, Poster track at RE'13, SANES session at PECCS'13, SBES'13;
- Nelly Bencomo is PC member of the VAMOS'13 international workshop;
- Nikolaos Georgantas is PC member of the following international conferences: CloudCom'13, AmI'13, DATA'13, ICSOFT-EA'13, ICSOFT-PT'13, SCC'13, ANT'13, SOSE'13;
- Nikolaos Georgantas is PC member of the following international workshops: MW4NG'13, ARM'13;
- Valérie Issarny is PC member of the following international conferences: CBSE'13&14, Coordination'14, ECSA'13, ESSOS'14, FASE'13&15, FSE'14, ICDCS'13, ICSE'13, ICSE-SEIP'14, IFIPTM'13, ISARCS'13, Middleware'13&14, MobiCASE'13, SEAMS'13&14;
- Valérie Issarny is PC member of the following international workshops: SESO@ICSE'13, SERENE'13;
- Animesh Pathak is PC member of the following international conferences: S-Cube'13, EWSN'14, and Sensornets'14; and
- Animesh Pathak is PC member of the GeoHCI'13 international workshop.

8.1.2. Leadership Services in Academic Events and Edited Journals

- Nikolaos Georgantas is associate editor of the International Journal of Ambient Computing and Intelligence (IJACI);
- Nikolaos Georgantas is associate editor of the International Journal of Advances in Human-Computer Interaction;
- Valérie Issarny is member of the Steering Committee of the ESEC/FSE conference;
- Valérie Issarny is associate editor of the Springer JISA Journal of Internet Services and Applications;
- Valérie Issarny is the general chair of the ESSoS 2013, the International Symposium on Engineering Secure Software and Systems; and
- Animesh Pathak is the Production Chair of HiPC 2013, and Publicity Chair of DCOSS 2014.

8.1.3. Other Academic Services

- Nikolaos Georgantas is member of the PhD monitoring committee at Inria Paris-Rocquencourt;
- Valérie Issarny is scientific leader of the EC FP7 ICT IP CHOReOS;
- Valérie Issarny is member of the IFSTTAR Scientific Council & “Commission d’évaluation des chercheurs”;
- Valérie Issary is member of the GDR GPL scientific council;
- Valérie Issarny is member of the evaluation committee of ANR Blanc JCJC’13, and of the Swedish Research Council subcommittee for Computer Science’2013.

8.2. Teaching - Supervision - Juries

8.2.1. Teaching

Nikolaos Georgantas gives a 12 hours lecture on “Pervasive Service Oriented Computing” as part of the SWAS lecture of the M2 COSY of the University of Versailles Saint-Quentin en Yvelines;

Animesh Pathak co-taught a M2 COSY level course on “Web sémantique, contenus et usages” at University of Versailles Saint-Quentin-en-Yvelines in Spring 2013; and

Animesh Pathak gave two three-hour guest lectures at CNAM, Paris as part of a M2 level course on embedded systems in Spring 2013.

8.2.2. Supervision

In 2013, the following students successfully defended their PhD:

- Amel Bennaceur, *Dynamic Synthesis of Connectors in Pervasive Environments*, UPMC-EDITE, defended in July 2013, advised by V. Issarny
- Pankesh Patel, *Enabling High-Level Application Development for the Internet of Things*, UPMC-EDITE, defended in November 2013, advised by A. Pathak and V. Issarny

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

- Emil Andriescu, *Synthèse et déploiement dynamiques de protocoles de médiation pour l’interopérabilité dans les environnements collaboratifs nomades*, started October 2012, UPMC-EDITE, CIFRE Inria-Ambientic, advised by Valérie Issarny and Roberto Speicys-Cardoso (Ambientic).
- Benjamin Billet, *Self-adaptive Streaming Middleware for the Internet of Things*, started October 2012, UVSQ-EDSTV, advised by Valérie Issarny
- Sara Hachem, *Middleware pour l’Internet des objets intelligents*, started October 2012, UVSQ-EDSTV, advised by Valérie Issarny and Animesh Pathak.
- Georgios Bouloukakis, *Runtime adaptation of middleware connectors for emergent mobile systems*, started October 2013, UPMC-EDITE, advised by Nikolaos Georgantas and Valérie Issarny.

Also, Valérie Issarny is co-advising with Ansgar Radermacher from CEA-LISE, the PhD thesis of Amel Belaggoun on *Adaptabilité et reconfiguration des systèmes temps-réel embarquées*; this is a PhD from UPMC-EDITE with the research being undertaken at CEA-LISE.

8.2.3. Juries

- Valérie Issarny served on the HDR committee of Marc-Olivier Killijian (Université Paul Sabatier, Toulouse).
- Animesh Pathak served on the PhD committee of Miruna Stoicescu (CNRS-LAAS & INP, Toulouse).

8.3. Popularization

- Benjamin Billet was invited to speak at a one-day Android tutorial event organized by the ARAMIS business network at Lyon.
- Animesh Pathak attended the "India-France Technology Summit" in New Delhi, India as part of the Inria delegation, where he was part of a panel on urban rail, and also presented a technology showcase of "TravelDashboard: Smart Mobility for Urban Travelers".
- Animesh Pathak was invited to be part of a panel on the "Chronique marché des applications mobiles" television programme on the French TV channel TVFIL78.
- Emil Andriescu, Sara Hachem, Kim Hwa Khoo, Georgos Mathioudakis Animesh Pathak, and George Rosca represented Inria at the "Futur en Seine" technology expo in Paris.

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, pp. 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", 2012, vol. 99 [DOI : 10.1109/TSC.2010.60], <http://hal.inria.fr/inria-00519577>
- [4] S. HACHEM, A. PATHAK, V. ISSARNY. *Service-Oriented Middleware for Large-Scale Mobile Participatory Sensing*, in "Pervasive and Mobile Computing", 2014, <http://hal.inria.fr/hal-00872407>
- [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, pp. 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, pp. 23-45 [DOI : 10.1007/s13174-011-0021-3], <http://hal.inria.fr/inria-00588753/en>

Publications of the year

Doctoral Dissertations and Habilitation Theses

- [7] A. BENNACEUR. , *Synthèse dynamique de médiateurs dans les environnements ubiquitaires*, Université Pierre et Marie Curie - Paris VI, July 2013, <http://hal.inria.fr/tel-00849402>
- [8] P. PATEL. , *Environnement de développement d'applications pour l'Internet des objets*, Université Pierre et Marie Curie - Paris VI, November 2013, <http://hal.inria.fr/tel-00927150>

Articles in International Peer-Reviewed Journals

- [9] M. AUTILI, M. CAPORUSCIO, V. ISSARNY, L. BERARDINELLI. *Model-driven engineering of middleware-based ubiquitous services*, in "Software & Systems Modeling", April 2013, pp. 1-31 [DOI : 10.1007/s10270-013-0344-6], <http://hal.inria.fr/hal-00866590>
- [10] A. BENVENISTE, C. JARD, A. KATTEPUR, S. ROSARIO, J. THYWISSEN. *QoS-Aware Management of Monotonic Service Orchestrations*, in "Formal Methods in System Design", February 2014, vol. 44, n^o 1, pp. 1-43 [DOI : 10.1007/s10703-013-0191-7], <http://hal.inria.fr/hal-00840362>
- [11] S. HACHEM, A. PATHAK, V. ISSARNY. *Service-Oriented Middleware for Large-Scale Mobile Participatory Sensing*, in "Pervasive and Mobile Computing", 2014, <http://hal.inria.fr/hal-00872407>

International Conferences with Proceedings

- [12] *Best Paper*
N. BENCOMO, A. BELAGGOUN. *Supporting Decision-making for Self-adaptive Systems: From Goal Models to Dynamic Decision Networks*, in "REFSQ 2013 - 9th International Working Conference on Requirements Engineering: Foundation for Software Quality", Essen, Germany, J. DOERR, A. L. OPDAHL (editors), Lecture Notes in Computer Science, Springer, April 2013, vol. 7830, pp. 221-236 [DOI : 10.1007/978-3-642-37422-7_16], <http://hal.inria.fr/hal-00927156>
- [13] N. BENCOMO, A. BELAGGOUN, V. ISSARNY. *Bayesian artificial intelligence for tackling uncertainty in self-adaptive systems: The case of dynamic decision networks*, in "RAISE 2013 - 2nd International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering", San Francisco, United States, IEEE, May 2013, pp. 7-13 [DOI : 10.1109/RAISE.2013.6615198], <http://hal.inria.fr/hal-00927162>
- [14] N. BENCOMO, A. BELAGGOUN, V. ISSARNY. *Dynamic Decision Networks for Decision-Making in Self-Adaptive Systems: A Case Study*, in "SEAMS '13 - 8th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), 2013", San Francisco, United States, IEEE, May 2013, pp. 113-122, <http://hal.inria.fr/hal-00927159>

- [15] A. BENNACEUR, C. CHILTON, M. ISBERNER, B. JONSSON. *Automated Mediator Synthesis: Combining Behavioural and Ontological Reasoning*, in "SEFM - 11th International Conference on Software Engineering and Formal Methods", Madrid, Spain, 2013, <http://hal.inria.fr/hal-00838788>
- [16] G. BOULOUKAKIS, I. BASDEKIS, C. STEPHANIDIS. *myWebAccess: A platform for repairing, enhancing and re-distributing Web Services accessible to people with disability*, in "IADIS - International Conference on Interfaces and Human Computer Interaction - 2013", Prague, Czech Republic, International Association for Development of the Information Society (hereinafter IADIS), July 2013, <http://hal.inria.fr/hal-00843464>
- [17] N. GEORGANTAS, G. BOULOUKAKIS, S. BEAUCHE, V. ISSARNY. *Service-oriented Distributed Applications in the Future Internet: The Case for Interaction Paradigm Interoperability*, in "ESOC 2013 - European Conference on Service-Oriented and Cloud Computing", Malaga, Spain, K.-K. LAU, W. LAMERSDORF, E. PIMENTEL (editors), Lecture Notes in Computer Science, Springer, July 2013, vol. 8135, pp. 134-148 [DOI : 10.1007/978-3-642-40651-5_11], <http://hal.inria.fr/hal-00841332>
- [18] S. HACHEM, A. PATHAK, V. ISSARNY. *Probabilistic Registration for Large-Scale Mobile Participatory Sensing*, in "PerCom 2013 - IEEE International Conference on Pervasive Computing", Californie, United States, Elsevier, March 2013, <http://hal.inria.fr/hal-00769087>
- [19] F. HASSANI BIJARBOONEH, A. PATHAK, J. PEARSON, V. ISSARNY, B. JONSSON. *A constraint programming approach for managing end-to-end requirements in sensor network macroprogramming*, in "SENSOR-NETS 2014 - 3rd International Conference on Sensor Networks", Lisbon, Portugal, 2014, <http://hal.inria.fr/hal-00927148>
- [20] A. KATTEPUR, N. GEORGANTAS, V. ISSARNY. *QoS Analysis in Heterogeneous Choreography Interactions*, in "11th International Conference on Service Oriented Computing (ICSOC)", Berlin, Germany, December 2013, <http://hal.inria.fr/hal-00866190>
- [21] A. KATTEPUR, N. GEORGANTAS, V. ISSARNY. *QoS Composition and Analysis in Reconfigurable Web Services Choreographies*, in "International Conference on Web Services", Santa Clara, California, United States, IEEE, June 2013, <http://hal.inria.fr/hal-00841485>
- [22] P. PATEL, A. PATHAK, D. CASSOU, V. ISSARNY. *Enabling High-Level Application Development in the Internet of Things*, in "4th International Conference on Sensor Systems and Software", Lucca, Italy, April 2013, <http://hal.inria.fr/hal-00809438>
- [23] M. STOICESCU, J.-C. FABRE, M. ROY, A. PATHAK. *From Resilient Computing Architectural Concepts to Wireless Sensor Network-based Applications*, in "EDCC 2014 - Tenth European Dependable Computing Conference", Newcastle upon Tyne, United Kingdom, May 2014, <http://hal.inria.fr/hal-00938389>

Scientific Books (or Scientific Book chapters)

- [24] M. CAPORUSCIO, M. FUNARO, C. GHEZZI, V. ISSARNY. *ubiREST: A RESTful Service-oriented Middleware for Ubiquitous Networking*, in "Advanced Web Services", A. BOUGUETTAYA, Q. SHENG, F. DANIEL (editors), Springer, 2014, vol. III, pp. 475-500 [DOI : 10.1007/978-1-4614-7535-4_20], <http://hal.inria.fr/hal-00927085>
- [25] R. DE LEMOS, H. GIESE, H. A. MULLER, M. SHAW, J. ANDERSSON, L. BARESI, B. BECKER, N. BENCOMO, Y. BRUN, B. CUKIC, R. DESMARAIS, S. DUSTDAR, G. ENGELS, K. GEIHS, K. GOESCHKA,

A. GORLA, V. GRASSI, P. INVERARDI, G. KARSAI, J. KRAMER, M. LITOIU, A. LOPES, J. MAGEE, S. MALEK, S. MANKOVSKII, R. MIRANDOLA, J. MYLOPOULOS, O. NIERSTRASZ, M. PEZZE, C. PREHOFE, W. SCHÄFER, R. SCHLICHTING, B. SCHMERL, D. SMITH, J. SOUSA, G. TAMURA, L. TAHVILDARI, N. VILLEGAS, T. VOGEL, D. WEYNS, K. WONG, J. WUTKE. *Software Engineering for Self-Adaptive Systems: A Second Research Roadmap*, in "Software Engineering for Self-Adaptive Systems", R. DE LEMOS, H. GIESE, H. MÜLLER, M. SHAW (editors), Dagstuhl Seminar Proceedings, Springer, 2013, vol. 7475, pp. 1-26, <http://hal.inria.fr/inria-00638157>

- [26] V. ISSARNY, A. BENNACEUR. *Composing Distributed Systems: Overcoming the Interoperability Challenge*, in "FMCO 2012", F. DE BOER, M. BONSANGUE, E. GIACHINO, R. HÄHNLE (editors), Lecture Notes in Computer Science, Springer, 2013, pp. 168-196 [DOI : 10.1007/978-3-642-40615-7_6], <http://hal.inria.fr/hal-00828801>

Scientific Popularization

- [27] L. CAPRA, P. CHÂTEL, A. PATHAK, R. SPEICYS CARDOSO. *TravelDashboard - a Framework for the Delivery of Personalized Mobility Services to Urban Travellers*, in "ERCIM News", April 2013, vol. 2013, n^o 93, <http://hal.inria.fr/hal-00939031>

Other Publications

- [28] V. ISSARNY. , *CHOReOS Deliverables*, September 2013, <http://www.choreos.eu/>, <http://hal.inria.fr/hal-00939027>