Research-Team ArchWare

Software Architecture

IRISA Site: Vannes

Activity Report

2012

1 Team

Head of the team
Flavio Oquendo, Full Professor, Université de Bretagne-Sud

Administrative assistant
Sylviane Boisadan, BIATSS, Université de Bretagne-Sud

Université de Bretagne-Sud
Isabelle Borne, Full Professor
Jérémy Buisson, Assistant Professor
Régis Fleurquin, Associate Professor, HDR
Elena Leroux, Assistant Professor
Salah Sadou, Associate Professor, HDR
Gersan Moguérou, Research Engineer

PhD students
Tahar Gherbi, Assistant, defence planned on September 2014
Salma Hamza, ARED grant, defence planned on September 2014
Davy Hélard, CIFRE grant with MGDIS, since September 2012
Lucas Oliveira, CsF-CNPq grant, since October 2012
Abderrhamane Seriai, ARED grant with University of Montreal, defence planned on September 2014
Minh Tu Ton That, MESR grant, defence planned on September 2014

Master students
Soraya Mesli, MRI/UBS, February-June 2013

2 Overall Objectives

2.1 Overview

The ArchWare Research Team addresses the scientific and technological challenges raised by architecting complex software-intensive systems. Beyond large-scale distributed systems, it addresses an emergent class of evolving software-intensive systems that is increasingly shaping the future of our software-reliant world, the so-called "Systems-of-Systems" (SoS).

An SoS can be perceived as a composition of systems in which its constituents, i.e. themselves systems, are separately discovered, selected, and composed possibly at run-time to form a more complex system. This composition forms a larger system that performs a mission not performable by one of the constituent systems alone, i.e. it creates an "emergent behaviour". But, if the architect of an SoS has authority on how the different systems work together, it has no direct control over each of the involved systems in the SoS. Component systems fulfil valid purposes in their own right, and continue to operate to fulfil those purposes if disassembled from the encompassing SoS. They are managed, in part, for their own purposes rather than the purposes of the whole. Moreover, a SoS is itself a system and possibly a constituent system of a much more complex SoS in its own right.

Indeed, since the dawn of computing, the complexity of software and the criticality of systems reliant on software have grown at a staggering rate. In particular, software-intensive systems have been rapidly evolved from being stand-alone systems in the past, to being part of networked systems in the present, to becoming systems of systems in the coming future .

In fact, nowadays almost all aspects of our lives and livelihoods depend on some sort of software-intensive system, especially when these systems are critical. This is the case of applications found in different areas as diverse as e.g. aerospace, aeronautics, automotive, energy, healthcare, manufacturing, transportation, civil infrastructures, entertainment, and consumer appliances; and applications that addresses societal needs as e.g. in environmental monitoring, distributed energy grids, emergency coordination, global traffic control, and smart cities. Therefore, the endeavour of constructing critical systems evolved from engineering complicated systems in the last century, to architecting critical SoS in this century. De facto, critical SoS became among the most complex of man-made creations and its very nature has intrinsic properties that are hard to address.

Moreover the upcoming generation of critical SoS will operate in environments that are open in the sense of that they are only partially known at design-time. These open-world critical systems-of-systems, in opposite to current closed-world systems, will run on pervasive devices and networks providing services that are dynamically discovered and used to deliver more complex services, which themselves can be part of yet more complex services and so on. Furthermore, they will often operate in unpredictable environments. Definitely, the unique characteristics of SoS raise a grand research challenge for the future of software-reliant systems in our industry and society due to its simultaneous intrinsic features, which are:

1. *Operational independence:* the participating systems not only can operate independently, they do operate independently. Hence, the challenge is to architect and construct SoS in a way that enables its operations (acting to fulfil its own mission) without violating the independence of its constituent systems that are autonomous, acting to fulfil their own missions.

2. *Managerial independence:* the participating systems are managed independently, and may decide to evolve in ways that were not foreseen when they were originally composed. Hence, the challenge is to architect and construct a SoS in a way that it is able to evolve itself to cope with independent decisions taken by the constituent systems and hence be able to continually fulfil its own mission.

3. *Distribution of constituent systems:* the participating systems are physically decoupled. Hence, the challenge is to architect and construct the SoS in a way that matches the loose-coupled nature of these systems.

4. *Evolutionary development:* as a consequence of the independence of the constituent systems, a SoS as a whole may evolve over time to respond to changing characteristics of its environment, constituent systems or of its own mission. Hence, the challenge is to architect and construct SoS in a way that it is able to evolve itself to cope with these three kinds of evolution.

5. *Emergent behaviours:* from the collaboration of the participating systems may emerge new behaviours. Furthermore, these behaviours may be ephemeral because the systems composing the SoS evolve independently, which may impact the availability of these behaviours. Hence, the challenge is to architect and construct a SoS in a way that emergent behaviours and their subsequent evolution can be discovered and controlled. In the case of an open-world environment, one can add the following characteristics and challenges:

6. *Unpredictable environment:* the environment in which the open-world SoS operates is only partially known at design-time, i.e. it is too unpredictable to be summarised within a fixed set of specifications, and thereby there will inevitably be novel situations to deal with at run-time. Hence, the challenge is to architect and construct such a system in a way that it can dynamically accommodate to new situations while acting to fulfil its own mission.

7. *Unpredictable constituents:* the participating systems are only partially known at design-time. Hence, the challenge is to architect and construct an open-world SoS in a way that constituent systems are dynamically discovered, composed, operated, and evolved in a continuous way at run-time, in particular for achieving its own mission.

8. *Long-lasting:* as an open-world SoS is by nature a long-lasting system, re-architecting must be carried out dynamically. Hence, the challenge is to evolutionarily re-architects and evolves its construction without interrupting it.

In essence, the long-term research challenge raised by SoSs calls for a paradigm shift in Software Architecture: SoSs call for a novel paradigm, complementing the traditional use of architectures at design-time traditionally applied to closed-world systems by novel trustful approaches blurring the boundary between design-time and run-time supporting continuous correctness of open-world systems.

Actually, in the literature, SoSs are classified according to four categories:

1. *Directed SoS:* a SoS that is centrally managed and which constituent systems have been especially developed or acquired to fit specific purposes in the SoS - they operate under tight subordination;

2. *Acknowledged SoS:* a SoS that is centrally managed and that operates under loose subordination - the constituent systems retain their operational independence;

3. *Collaborative SoS:* a SoS in which there is no central management and constituent systems voluntarily agree to fulfil central purposes;

4. *Virtual SoS:* a SoS in which there is no central authority or centrally agreed purpose.

Regarding the state-of-the-art, our last systematic literature review searching all key bibliographic databases relevant to computer science, software and systems engineering, looking for publications from academia and experience reports from industry shows that, today, proposed approaches only support the architecture and construction of SoS matching the core characteristics, basically directed and acknowledged SoSs limited to non-critical systems. Therefore, achieving the targeted breakthrough will be a major advance in the state-of-the-art by delivering a conceptual, theoretical, and technological foundation for architecting and constructing the new generation of critical SoS, i.e. collaborative and virtual SoS.

For addressing the scientific challenge raised for architecting SoS, the targeted breakthrough for the ArchWare Research Team is to conceive sound foundations and a novel holistic approach for architecting open-world critical software-intensive systems-of-systems, encompassing:

1. suitable architectural abstractions for formulating the architecture and re-architecture of SoS;

2. an appropriate formalism and its underlying computational model to rigorously specify the architecture and re-architecture of SoS;

3. the supporting mechanisms to construct and manage SoSs driven by architecture descriptions, while resiliently enforcing their correctness, effectiveness, and efficiency;

4. the supporting mechanisms to evolve SoSs driven by architecture descriptions, while resiliently enforcing their correctness, effectiveness, and efficiency throughout the evolution.

The research approach we adopt in the ArchWare Research Team for developing the expected breakthrough is based on well-principled design decisions:

1. to conceive architecture description, analysis, transformation, and evolution languages based on suitable SoS architectural abstractions;

2. to formally ground these SoS-specific architecture languages on well-established concurrent constraint process calculi and associated logics;

3. to conceptually and technologically ground the construction and management of SoSs on architecture descriptions defined by models-at-runtime in the service-oriented computing style;

4. to architecturally and technologically ground the dynamic evolution of SoS architectures around the concept of "anytime architecture".

2.2 Key Issues

As stated, an SoS can be perceived as a composition of systems in which its constituents, i.e. themselves systems, are separately discovered, selected, and composed possibly at run-time to form a more complex system, the SoS. Hence, four points are at the core of our approach:

- SoS architectures as compositions of systems: an SoS depends on composed systems and their interactions to undertake capabilities. Composition is thereby more challenging in SoS as systems being combined are active. This challenge beyond the state-of-the-art will be overcome in our approach by the development of SoS-specific composition mechanisms that are explicit, formally defined, and operate on active architectural models at run-time.

- Analysis of SoS architectures: SoS architecture descriptions, by their formal nature, will support automated analysis with a view in particular to evaluating and predicting non-functional qualities of the modelled SoS. In our approach, we will make a significant progress beyond the state-of-the-art of SoS analysis by developing techniques and tools for the architecture-centric model-based analysis of SoS. It will support verification of different sorts of properties and interleaving of these properties, including structural (e.g. connectivity, topology), behavioural (e.g. safety, liveness, fairness), and quality properties (e.g. agility, performance). We will develop different complementary analysis techniques combining simulation, model checking, and testing. The aim is to enable analysis and validation of a SoS architecture anytime along the whole SoS life-cycle by means of automated verification. Automated verification will include infinite-state model checking and abstract interpretation techniques for the critical parts of SoS, and testing for non-critical parts to ensure the correct functioning of SoS.

- Architecture-driven construction of SoSs: SoS architecture will drive the initial construction and subsequent evolutions of a SoS. Initial construction will be developed through refinement, where abstract SoS architectures are refined to "meet in the middle" with concrete selection and integration of discovered systems, consequently defining concrete SoS architectures. In our approach, significant progress beyond the state-of-the-art on SoS construction will be achieved by conceiving abstractions and mechanisms for expressing architecture transformations, where the application of these transformations will support refinement from abstract to concrete SoS architectures, taking into account discovered SoSs. Each transformation can be seen as an architecture rewrite in a formal rewriting system. These transformations are enforced to be refinements if preconditions are satisfied and proof obligations discarded. Concrete architectures are deployed to directly implement running SoSs in terms of middleware-based glue code.

- Evolution of SoS architectures and underlying SoSs: an evolving system must be continuously aware of its own behaviour and surrounding environment to execute efficiently and to react to new situations. In the ArchWare Research Project, we will develop SoS-specific concepts and mechanisms of software instrumentation, based on probes and gauges, to implement and support such a continuous feedback. We will utilise a set of probes to observe and quantify significant events as defined by SoS requirements. The

probes will have the ability to be placed into running SoSs. Each SoS will use the input from the probes according to the interpretation of SoS-specific gauges, together with the models of the mission and the architecture to decide when, how, and where evolution is appropriate. For supporting feedback, due to the specific nature of a SoS, monitoring will be required at two distinct levels: the architecture of the SoS and its assigned mission. On the one hand, monitoring the architecture of the SoS implies maintaining a suitable model of all dependencies among the systems involved in the SoS. Those dependencies being of various types, we will formally describe them in such a way that any change in the architecture will be automatically detected. On the other hand, monitoring the mission implies collecting data from the systems comprising the SoS, indicating that the mission is in progress, or detecting any event that must be addressed to decide if one has to re-architect the SoS and/or change the mission itself. Due to the nature of the SoS, a challenge is that the mission monitoring system must be itself evolvable as systems involved in the SoS may evolve over time.

2.3 Key Activities

During the reporting year, we have deeply analyzed the domain of software-intensive systems-of-systems from the Software Engineering viewpoint. On the one hand we have organized several meetings for raising the needs of industrials working on critical systems-of-systems and on the other we have surveyed the litterature with studies of the state-of-the-art and in particular systematic mappings and reviews.

From the results of these studies, we have analyzed in greater depth the state-of-the-art from the Software Architecture viewpoint issuing the set of key research challenges that needs to be addressed in the next 10 years. The scientific project of the ArchWare team has then been constructed, written down and presented to the IRISA Scientific Direction for launching the process of officially accreditating the ArchWare team in its new composition and thematics.

In parallel, we have worked internally in the foundations of a novel approach for architecting software-intensive systems-of-systems to pave the way for publications in the coming year.

Keywords: Software Architecture, Architecture Description, Architecture Analysis, Software-intensive Systems-of-Systems.

3 Contracts and Grants with Industry

3.1 Grants Involving Industry

Collaboration on systems for processing big-data (CIFRE)

Participants: MGDIS.

Collaboration on description of systems-of-systems

Participants: DCNS.

3.2 National Cooperative Projects

PEPS API (Projet Exploratoire Pluridisciplinaire CNRS sur l'Automatique pour l'informatique autonomique)

- Funding: CNRS
- Period: July 2011 - July 2013
- Partners: LIG, INRIA Rhône-Alpes, GIPSA-Lab, Lab-STICC
- Objective: To explore how control theory, software adaptability and reconfigurable architectures contribute to each other in order to build autonomic systems.

UBISOA (Self-evolving Service-Oriented Architectures: On-the-fly Service Compositions based on Formal Foundations)

- Funding: Brittany Region and Morbihan Department
- Period: December 2009 - December 2012
- Partners: DCNS Engineering France
- Objective: To support the description of self-evolving service-oriented architectures from the structural and behavioural viewpoints, enabling to rigorously reason about and verify their quality attributes, in particular related to conformance and correctness.

3.3 International Cooperative Projects

FP7 NoE S-CUBE (European Network of Excellence in Software Services and Systems)

- Funding: European Commission under the ICT Programme
- Period: 2010-2012
- Partners: Associated Partner of the S-CUBE Consortium with Univ. of Duisburg-Essen / Paluno - Ruhr Institute for Software Technology (DE), Tilburg Univ. (NL), City Univ. London (UK), CNR (IT), FBK (IT), INRIA (FR), Lero (IE), PJIIT (PL), Politecnico di Milano (IT), MTA SZTAKI (HU), Vienna Univ. of Technology (AT), Univ. Claude Bernard Lyon (FR), Univ. of Crete (GR), Univ. Politécnica de Madrid (ES), Univ. of Stuttgart (DE)
- Objective: In the Network of Excellence, our aim was to foster novel formal languages for describing service-oriented architectures.

AD-AUTO (Adaptive Software Architectures for Autonomic Systems)

- Funding: CNPq (Brazilian National Research Agency)
- Period: 2011 - 2013
- Partners: UFRN - Federal University of Rio Grande do Norte, UFPE - Federal University of Pernambuco (Brazil)
- Objective: To develop mechanisms for dynamic reconfiguration based on reflective architectural models and middleware supporting the architecture and engineering of autonomic systems.

ProSA-RAES (Framework to Support Reference Architectures for Embedded Systems)

- Funding: FAPESP (Sao Paulo State Research Agency)
- Period: 2012-2014
- Partners: Fraunhofer-Institut für Experimentelles Software Engineering IESE (Germany), ICMC Research Institute at USP (Brazil)
- Objective: To develop a formal language and framework for defining reference architectures of embedded systems based on ArchWare languages.

4 Other Grants and Activities

4.1 International Collaborations

- Flavio Oquendo:
  - Federal University of Rio Grande do Norte (UFRN), Brazil (Thais Batista, Jair Leite): architecture-based dynamic reconfiguration
  - University of Sao Paulo (USP), Brazil (José Carlos Maldonado, Elisa Nakagawa): reference architectures of embedded systems
- Salah Sadou:
  - Université de Montréal (Houari Sahraoui): Restructuring Object-oriented Systems into Component-Based Systems (PhD in co-tutelle)

4.2 National Collaborations

- Jérémy Buisson is an external collaborator of Télécom Bretagne (PASS team), contributing to the supervision of PhD student Sébastien Martinez of that team
- Salah Sadou has a collaboration on Reuse of Architectural Constraints with the Université de Montpellier 2 (Chouki Tibermancine and Christophe Dony)
- Flavio Oquendo participates in a collective book on software architecture organized by the LINA - Université de Nantes (Mourad Oussalah).

4.3 National CNRS GDR Networks

- GT COSMAL of GDR CNRS GPL on Software Architecture
- GT ADAPT of GDR CNRS ASR on Software Adaptation
- GT RIMEL of GDR CNRS GPL on Software Evolution
- Action IDM of GDR CNRS ASR/GPL on Software Modelling

5 Dissemination

5.1 Editorial Boards and Guest Editions

- Flavio Oquendo:
  - Springer Journal of Software Engineering Research and Development (Member of the Editorial Board)
  - Special Issue on Software Architecture Modelling of the Journal on Software and System Modelling (Guest Editor)
  - Special Issue on Software Components, Architectures and Reuse of the International Journal of Universal Computer Science (Guest Editor)

5.2 General Chairs, Steering Committees

- Flavio Oquendo:
  - IEEE/IFIP Working International Conference on Software Architecture - WICSA (Steering Committee Member)
  - European Conference on Software Architecture - ECSA (Steering Committee Chair)
  - Conférence francophone sur les architectures logicielles - CAL (Steering Committee Member)
  - International Conference on Communication Languages and Signal Processing in Reference to 4G Technologies - ICCLSP (International Advisory Board Member)

5.3 Program Committees

- Jérémy Buisson:
  - ICCS: International Conference on Computational Science, 2012
- Flavio Oquendo:
  - ICECCS: IEEE International Conference on Engineering of Complex Computer Systems, 2012
  - ECSA: European Conference on Software Architecture, 2012

- WICSA: IEEE/IFIP Working International Conference on Software Architecture, 2012
- ISARCS: International ACM Symposium on Architecting Critical Systems, 2012
- ICSEA: International Conference on Software Engineering Advances, 2012
- ICSOFT: International Conference on Software and Data Technologies, 2012
- WETICE: IEEE International Conference on Collaboration Technologies and Infrastructures, 2012
- AROSA: IEEE WETICE Conference Track on Adaptive and Reconfigurable Service-oriented and component-based Applications and Architectures, 2012
- PESOS: International Workshop on Principles of Engineering Service-Oriented Systems at ACM/IEEE International Conference on Software Engineering (ICSE), 2012
- ICAART: International Conference on Agents and Artificial Intelligence, 2012
- SBES: National Symposium on Software Engineering, 2012
- SBCARS: National Symposium on Software Components, Architectures and Reuse, 2012
- I-ESA: International Conference on Interoperability for Enterprise Systems and Applications, 2012
- COMPUTATION: International Conference on Computational Logics, Algebras, Programming, Tools, and Benchmarking, 2012
- ADAPTIVE: International Conference on Adaptive and Self-Adaptive Systems and Applications, 2012
- AOSE: International Workshop on Agent-Oriented Software Engineering, 2012
- QSIC: International Conference On Quality Software, 2012

5.4 Doctoral Examination Boards

Flavio Oquendo:

- Sébastien Dourlens, University of Versailles, France (Rapporteur)
- Ivano Malavolta, University of L?Aquila, Italy (Rapporteur)

Isabelle Borne:

- Amine Raji, ENSTA Bretagne, 2012 (President)

5.5 Industrial Collaborations

- MGDIS: CIFRE (Flavio Oquendo)
- DCNS: Collaboration sur les Systèmes-de-Systèmes (Flavio Oquendo)

5.6 Research Excellence Awards (PES)

- Flavio Oquendo: PES A (2011-2015)
- Salah Sadou: PES A (2010-2014)

5.7 Teaching Responsabilities

- Régis Fleurquin: Head of Computing Department of IUT (UBS, Vannes), 2012
- Flavio Oquendo: Head of Computing Degree of ENSIBS School of Engineering, 2012
- Flavio Oquendo: Member of the Steering Board of ENSIBS School of Engineering, 2012

5.8 Expertises

- Flavio Oquendo: Expert acting as reviewer and evaluator of R&D Projects for the European Commission (Framework Program 7) for:
  - Software-intensive Systems Engineering,
  - Systems-of-Systems, and
  - Trustworthy ICT, in particular, acting as one of the 6 experts composing the Evaluation Panel for Large-Scale Research Projects (IPs) of the European Commission?s Framework Programs in Trustworthy ICT.

6 Bibliography

Major publications by the team in the current year

Publications in Conferences and Workshops

[1] M. Guessi, E. Yumi Nakagawa, F. Oquendo, J. Carlos Maldonado, "Architectural Description of Embedded Systems: a Systematic Review", *in : International ACM Symposium on Architecting Critical Systems (ISARCS)*, p. 31–40, Italy, June 2012, `https://hal.archives-ouvertes.fr/hal-00760747`.

[2] J.-E. Méhus, T. Batista, J. Buisson, "ACME vs PDDL: support for dynamic reconfiguration of software architectures", *in : 6ème édition de la Conférence Francophone sur les Architectures Logicielles (CAL 2012)*, p. 48–57, Montpellier, France, May 2012, `https://hal.archives-ouvertes.fr/hal-00703176`.

[3] L. Minora, J. Buisson, F. Oquendo, T. Batista, "Issues of Architectural Description Languages for Handling Dynamic Reconfiguration", *in : 6ème Conférence francophone sur les architectures logicielles (CAL'2012)*, D. Beslimane, A.-D. Seriai (editors), p. 69–80, Montpellier, France, May 2012, `https://hal.archives-ouvertes.fr/hal-00699895`.

[4] T. M. Ton That, S. Sadou, F. Oquendo, "Using Architectural Patterns to Define Architectural Decisions", *in : Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture*, p. 196–200, Helsinki, Finland, August 2012, `https://hal.archives-ouvertes.fr/hal-00758792`.

[5] E. Yumi Nakagawa, F. Oquendo, M. Becker, "RAModel: A Reference Model for Reference Architectures", *in : Joint 2012 Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture*, p. 297–301, Helsinki, Finland, August 2012, `https://hal.archives-ouvertes.fr/hal-00760751`.