Activity Report 2023

Research-Team ArchWare

Software Architecture

Architecting Software-intensive Systems and Systems-of-Systems

D4 – Language and Software Engineering

Contents

This activity report covers the year of 2023 of the ArchWare Research Team.

1 Team composition

Researchers and faculty

Flavio Oquendo, Full Professor/PR classe exceptionnelle, PEDR, UBS[1] (Head)
Isabelle Borne, Full Professor/PR classe exceptionnelle, UBS
Salah Sadou, Full Professor/PR 1ère classe, PEDR, UBS
Nicolas Belloir, Assistant Professor/MCF hors cl., Ecoles de St-Cyr
Jérémy Buisson, Associate Professor/MCF cl. normale, HDR, Ecoles de St-Cyr[2]
Jamal El Hachem, Assistant Professor/MCF cl. normale, PEDR, UBS
Régis Fleurquin, Associate Professor/MCF cl. exceptionnelle, HDR, UBS
Elena Leroux, Assistant Professor/MCF cl. normale, UBS

Associate members (IRISA/SEGULA Agreement)

Soraya Mesli-Kesraoui, PhD, SEGULA Technologies - R&D Division

Research engineers

Gersan Moguérou, Research Engineer (IGR 1ère classe), Univ. Bretagne Sud

PhD students

Monica Buitrago
Philippe Charton
Ahmed Elmarkez
Mohamed Fendali
Etienne Lemonier
Maykel Mattar
Jesus Ramos
Lucas Tesson
Jeisson Vargas
Tales Viglioni
Sidbewendin Yameogo

External collaborators in particular in cotutelles of PhD theses and joint publications

Thais Batista, Tegawendé F. Bissyandé, Everton Cavalcante, Mohamed Lamine Kerdoudi, Jair Leite, Mohamed Ahmed Nacer, Marcel Oliveira, Wassila Ouerdane, Oscar Pastor, Felipe Restrepo-Calle, Labiba Souici-Meslati, Chouki Tibermacine

[1] Université Bretagne Sud
[2] In leave to Ecole de l'Air et de l'Espace

Administrative assistant

Anne Idier, BIATSS, Université Bretagne Sud

2 Overall objectives

2.1 Overview

The ArchWare Research Team addresses the scientific and technological challenges raised by architecting complex software-intensive systems. Beyond static architectures (i.e. software architectures which are unchanging over time), we focus our research on the dynamic architectures of software-intensive systems (i.e. software architectures which change on the fly during run-time, according to the ways foreseen at design-time). Besides dynamic software-intensive systems, we address an emergent class of evolving software-intensive system that is increasingly shaping the future of our software-reliant world, the so-called System-of-Systems (SoS). SoSs exhibit evolutionary architectures (i.e. software architectures which change dynamically in ways not necessarily foreseen at design-time).

Indeed, since the dawn of computing, the complexity of software and the complexity 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 (often based on static architectures), to be part of networked systems in the present (often based on dynamic architectures), to increasingly become systems-of-systems in the coming future (based on dynamic, evolutionary architectures).

De facto, software-intensive systems have been independently developed, operated, managed, and evolved. Progressively, networks made communication and coordination possible among these autonomous systems, yielding a new kind of complex system, i.e. a system that is itself composed of systems. These systems of systems are evolutionary developed from systems to achieve missions not possible by each constituent system alone.

Different aspects of our lives and livelihoods have become overly dependent on some sort of software-intensive SoS. This is the case of SoSs found in different areas as diverse as aeronautics, automotive, energy, healthcare, manufacturing, and transportation; and applications that addresses societal needs as e.g. in environmental monitoring, distributed energy grids, emergency coordination, global traffic control, and smart cities.

Moreover, emergent platforms such as the Internet of Things and emergent classes of SoSs such as Cyber-Physical SoSs are accelerating the need of constructing rigorous foundations, languages, and tools for supporting the architecture and engineering of trustworthy SoSs.

Complexity is intrinsically associated to SoSs by its very nature that implies emergent behavior: in SoSs, missions are achieved through emergent behavior drawn from the interaction among constituent systems. Hence, complexity poses the need for separation of concerns between architecture and engineering: (i) architecture focuses on reasoning about interactions of parts and their emergent properties; (ii) engineering focuses on designing and constructing such parts and integrating them as architected.

Definitely, the software architecture forms the backbone for taming the complexity of trustworthy software-intensive systems, in particular in the case of evolving systems and systems-of-systems, where architecture descriptions provide the framework for designing, constructing, and dynamically evolving such complex systems, in particular when they operate in unpredictable open-world environments.

Therefore, the endeavor of designing trustworthy systems evolved from architecting complicated systems in the last century, based on static architectures, to architecting trustworthy systems and systems-of-systems in this century, based on dynamic architectures. In particular, trustworthy SoSs, by their very nature, have intrinsic properties that are very hard to address.

Furthermore, the upcoming generation of trustworthy SoSs will operate in environments that are open in the sense of that they are only partially known at design-time. These open-world trustworthy SoSs, in opposite to current closed-world systems, run on pervasive devices and networks providing services that are dynamically selected 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 engineer SoS in a way that enables its operations (acting to fulfill its own mission) without violating the independence of its constituent systems that are autonomous, acting to fulfill 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 engineer an 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 fulfill its own mission.

3. *Distribution of constituent systems:* the participating systems are physically decoupled. Hence, the challenge is to architect and engineer 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, an SoS as a whole may evolve over time to respond to changing characteristics of its environment, constituent systems or even of its own mission. Hence, the challenge is to architect and engineer SoS in a way that it is able to evolve itself to cope with these different kinds of evolution.

5. *Emergent behaviors:* from the collaboration of the participating systems may emerge new macroscale behaviors. Furthermore, these macroscale behaviors may be ephemeral because the systems composing the SoS evolve independently, which may impact the availability of these behaviors. Hence, the challenge is to architect and engineer an SoS in a way that emergent behaviors and their subsequent evolution can be discovered and controlled.

In the case of an open-world environment, one can add the following characteristics:

1. *Unpredictable environment:* the environment in which the open-world SoS operates is only partially known at design-time, and thereby there will inevitably be novel situations to deal with at run-time. Hence, the challenge is to architect and engineer such a system in a way that it can dynamically accommodate to unprecedented situations while acting to fulfill its own mission.

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

3. *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.

The importance of developing novel theories and technologies for architecting and engineering SoSs is highlighted in several roadmaps.

In France, SoS architecture and engineering is explicitly targeted in the report prepared by the French Ministry of Economy as one of the key technologies for the period 2015-2025 (étude prospective sur les technologies clés 2015-2025, Direction Générale de la Compétitivité, de l'Industrie et des Services du Ministére de l'Economie).

In Europe, SoSs are explicitly targeted in the studies developed by the initiative of the European Commission, i.e. Directions in Systems-of-Systems Engineering, and different Networks of Excellence (NoE), in particular HiPEAC (NoE on high-performance and embedded computing systems) and HYCON2 (NoE on highly-complex and networked control systems) and different European Technological Platforms (ETP) and Industrial Associations (IA), specifically ARTEMIS-IA (industrial association for actors in embedded and cyber-physical systems) and NESSI-ETP (European platform on software and services) point out the relevance and timeliness of addressing the SoS challenge.

In Europe also, two roadmaps for SoSs were produced under the support of the European Commission, issued from the CSAs ROAD2SoS (Development of strategic research and engineering roadmaps in Systems-of-Systems) and T-Area-SoS (Transatlantic research and education agenda in Systems-of-Systems) as well as a research agenda for developing cyber-physical SoSs by the CSA CPSoS.

All these roadmaps and the just released INCOSE Systems Engineering Vision for 2035 emphasize the importance of progressing from the current situation, where SoSs are basically developed in ad-hoc way in specific application sectors, to a scientific approach providing rigorous theories, technologies, and methodologies for mastering the complexity of SoSs in general.

Overall, the long-term research challenge raised by SoSs has called for a novel paradigm and novel trustful approaches for architecting, analyzing, constructing, and assuring the continuous correctness of systems-of-systems, often deployed in unpredictable environments, taking into account all together their intrinsic characteristics.

The targeted breakthrough for the ArchWare Research Team is to conceive sound foundations and a novel holistic approach for architecting open-world trustworthy software-intensive SoSs, encompassing:

- Concepts and abstractions for formulating the architecture and re-architecture of SoS;
- Formalism and underlying computational model to rigorously specify the architecture and re-architecture of SoS;
- Mechanisms to construct, manage, and evolve SoSs driven by architecture descriptions, while resiliently enforcing their correctness, effectiveness, and efficiency;
- Concepts, formalisms and mechanisms for specifying and operating SoS missions, deriving abstract architectures, as well as generating concrete SoS architectures;
- Concepts, formalisms and mechanisms for specifying and enforcing safety/liveness properties as well as cybersecurity to achieve trustworthiness in SoS architectures.

Keywords: Software Architecture, Architecture Description, Architecture Analysis, Safety Architecture, Cybersecurity Architecture, Mission Specification, Software-intensive Systems, Software-intensive Systems-of-Systems, Architecture-based Evolutionary Development.

2.2 Scientific foundations

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. Architectural abstractions for formulating the architecture and re-architecture of SoS;
2. Formalism and underlying computational model to rigorously specify the architecture and re-architecture of SoS;
3. Mechanisms to construct, manage, and evolve SoSs driven by architecture descriptions, while resiliently enforcing their correctness, effectiveness, and efficiency;
4. Formalism and mechanisms for ensuring safety and cybersecurity at the architectural level and their transformations towards implementation.
5. Concepts and formalisms for specifying and operating SoS missions and generating abstract and concrete SoS architectures.

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, 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 executable models;
4. To derive/generate abstract/concrete architectural descriptions from well-defined mission specifications.

2.3 Application domains

The ArchWare Research Team develops formalisms, languages and software technologies which are transverse to application domains while providing mechanisms for customization to different architectural styles and application areas. In these different application areas, extra-functional properties, in particular safety and cybersecurity, are addressed.

During 2023, addressed applications areas include:

1. Internet-of-Things (IoT), Industrial Internet-of-Things (IIoT);
2. Intelligent Transportation Systems;
3. Cybersecurity of major events reliant on digital systems.

3 Scientific achievements

3.1 Formal approaches for systems-of-systems architectures: the SoS Architecture Description Language (SosADL)

Keywords: Architecture Description Language (ADL), Architecture Synthesis, Uncertainty, Software-intensive Systems-of-Systems (SoS).

Participants: Flavio Oquendo, Jérémy Buisson, Elena Leroux, Gersan Moguérou.

The architecture provides the right abstraction level to address the complexity of software-intensive Systems-of-Systems (SoSs). The research challenges raised by SoSs are fundamentally architectural: they are about how to organize the interactions among the constituent systems to enable the emergence of SoS-wide behaviors and properties derived from local behaviors and properties by acting only on their connections, without being able to act in the constituent systems themselves.

Formal architecture descriptions provide the framework for the design, construction, and dynamic evolution of SoSs.

From the architectural perspective, in single systems, the controlled characteristics of components under the authority of the system architect and the stable notion of connectors linking these components, mostly decided at design-time, is very different from the uncontrolled nature of constituent systems (the SoS architect has no or very limited authority on systems) and the role of connection among systems (in an SoS, connections among constituents are the main architectural elements for enabling emergent behavior to make possible to achieve the mission of an SoS).

The nature of systems architectures (in the sense of architectures of single systems) and systems-of-systems are very different:

- Systems architectures are described by extension. In the opposite, SoS architectures are described by intention.
- Systems architectures are described at design-time for developing the system based on design-time components. In the opposite, SoS architectures are defined at runtime for developing the SoS based on discovered constituents.
- Systems architectures often evolve offline. In the opposite, SoS architectures always evolve online.

In the sequel, the main results of this research line produced in 2023 are presented.

3.1.1 (Crisp) SosADL: Generating concrete architectures from abstract architecture descriptions

Keywords: Systems-of-Systems, Architecture Synthesis, Model Transformation, Architecture Description Language.

Based on the work on SoS architecture synthesis previously carried out, the effort in this line of research was mainly dedicated to terminating the design and implementation of the generation of concrete architectures (from abstract architecture descriptions) in SosADL Studio.

3.1.2 Fuzzy SosADL: Mediators as control systems for automating coordinated actions under uncertainty

Keywords: Systems-of-Systems, Uncertainty, Fuzzy Theory, Architecture Description Language.

The work on Fuzzy SosADL has continued and Fuzzy SosADL has been applied to supporting opportunistic SoS, while mastering uncertainty and interactions under uncertainty. In particular, a case study of Fuzzy SosADL, in terms of Fuzzy Mediating Control Systems for safely supporting vehicle overtaking maneuvers on two-way roads, one of the hardest applications of driving automation systems, has been carried out. In fact, a key challenge in automating vehicle driving maneuvers is to address uncertainty, mainly related to the interactions between the ego vehicle and all the surrounding traffic as well as the road infrastructure. In the case of an overtaking maneuver, the neighboring vehicles include the vehicle(s) moving in front of the ego vehicle - in particular, the vehicle(s) to be overtaken - and possibly other vehicles moving in the adjacent lane. In this context, the underlying research question addressed was: how a driving automation system of an ego vehicle shall deal with occurring uncertainties to safely perform overtaking maneuvers, knowing that each of its neighboring vehicles is managerially and operationally independent, their disposition has been evolutionarily formed, and by their interactions, they jointly raise emergent behaviors. For details, see: [6]

The work on Fuzzy SosADL regarding the extension of SosADL based on Fuzzy Theory for dealing with uncertainty in SoS architectural descriptions has now ended.

3.2 Formal approaches for systems architectures: the Systems Architecture Description Language (SysADL)

Keywords: Architecture Description Language (ADL), Architecture Modeling, Executable Architecture Specifications, Verifiable Architecture Specifications, Software-intensive Systems (Sys).

Participants: Flavio Oquendo, Camila Araujo, Thais Batista, Everton Cavalcante, Fagner Dias, Jair Leite, Marcel Oliveira.

The architecture provides the right abstraction level to address the complexity of Software-intensive Systems (Sys). This research line addresses the research challenges raised by architecture modeling based on the ISO SysML notation. We defined an Architecture Description Language, named SysADL, as a specialization of the ISO-OMG SysML standard to software architecture description.

SysADL brings together the expressive power of software architecture description languages (ADLs) for architecture description, with a standard language used by the industry (SysML). SysADL defines viewpoints to describe the structure and the behavior of the software architecture of a software-intensive system.

The main research challenge is thereby to enforce correctness-by-design associating on the one hand a software architect-friendly notation with on the other hand formal representations for supporting formal refinement and analysis.

In the sequel, the main results of this research line produced in 2023 are presented.

3.2.1 Formalizing SysADL in terms of π-ADL for supporting run-time verification

Keywords: Formalization, Model Transformation, Architecture Description Language, Formal Verification, SysML.

The critical nature of many software-intensive systems requires formal architecture descriptions for supporting automated architectural analysis regarding correctness properties. Due to the challenges of adopting formal approaches, many architects prefer to use standard notations, mainly SysML and its derivatives, to describe the structure and behavior of software architectures of software-intensive systems. However, SysML and other semi-formal notations have limitations regarding the sought support for architectural analysis. This research line develops an approach to bridge the rigor of formal architecture descriptions and the ease of use of SysML-based notations widely used elsewhere.

The main concern in this research line is, thereby, to provide formal semantics to SysADL, a SysML-based language to describe software-intensive system architectures.

After having defined a denotational semantics to SysADL in terms of π-ADL, the work carried out focused on the verification of the π-ADL architecture specification generated by the model transformations to demonstrate that it is equivalent to the originally described architecture in SysADL. A case study for a Flood Monitoring System architecture has also been carried out.

This work has now ended. Its final results are planned to be published in a journal article to be submitted in 2024.

3.2.2 Formalizing SysADL in terms of CSP for supporting design-time verification

Keywords: Software Architecture Description, Formal Verification, Correctness Properties, CSP, SysML.

Software architecture analysis requires formal methods and software architecture description languages to verify relevant system properties. Despite being significantly valuable in supporting architectural analysis, a formal ADL is typically complex from the point of view of software architects and developers. This work proposes an approach to ease the task of formally verifying software architectures using models, techniques, and tools. It relies on a supporting tool to deliver a model-checking-based automated analysis for formally verifying software architectures described in SysADL. The developed formal verification uses the semantics based on the Communicating Sequential Processes (CSP) for SysADL and is supported by the FDR4 refinement model checker for CSP. The formal verification guarantees the consistency among elements and viewpoints of an architecture description. Furthermore, it enables automatic concurrency verification, such as deadlock-freedom and livelock-freedom, and safety properties.

After having defined a denotational semantics to SysADL in terms of CSP, the work carried out focused on extending the plugin developed for SysADL Studio to provide a seamless use of the verification features.

This work has now ended. Its final results are planned to be published in a journal article in 2024, the submitted version being now under minor revision.

3.2.3 Enhancing SysADL with an executable architecture viewpoint for supporting scenario-based simulations

Keywords: Software Architecture, Executable Architecture, Simulation, Scenario Modeling, SysML.

Executable software architectures enable architects to validate the correctness of their designs early in the development lifecycle of software-intensive systems, reducing the risk of costly misconceptions. Over the years, the field of executable software architectures has been an important topic of research and has undergone significant evolution. Even though various methods for executing software architectures have developed, there is no broad study assessing the state-of-the-art on the subject nor a proposal to describe architecture execution as a first-class citizen. In the first phase of this work, we conducted a systematic mapping study on executable software architectures by analyzing 45 studies in the last almost 30 years. The findings contribute to understanding the activities involved in the execution of architectures; providing an overview of the tools, approaches, notations, and quality attributes involved; and identifying the main classes of systems targeted, as well as the domains and contexts of study. This mapping study also sheds light on relevant challenges and potential directions for future research on executable software architectures. The second phase, that started by the end of 2023, will develop an executable architecture viewpoint for supporting scenario-based simulations with SysADL.

This systematic mapping study has now ended, and its report is currently under review. Its final results are planned to be published in a journal article to be submitted in 2024.

3.3 Software architecture in product lines

Keywords: Software Architecture, Software Product Line.

Participants: Salah Sadou, Mohamed Lamine Kerdoudia, Chouki Tibermacine.

A large software system usually exists in different forms, with different variants targeting specific business needs and stakeholders. This kind of system is generally provided as a set of "independent" products and not as a "single whole". Engineers use ad-hoc mechanisms to manage variability. In this research line, the proposed vision of software development for this kind of system is to consider a Software Product Line (SPL) architecture where the architecture of each variant of the product is derived before its implementation. In this way, each derived variant can have its own life.

In this work, we have developed a novel approach for Software Architecture Product Line (SAPL) Engineering. It consists of a generic process for recovering an SAPL model which is a product line of "software architectures" from large-sized variants, and a forward-engineering process that uses the recovered SAPL to derive new customized software architecture variants. Our approach was first experimented with thirteen Eclipse variants to create a new SAPL. Then, an intensive evaluation was conducted using an existing benchmark that is also based on the Eclipse IDE. For details, see: [5].

3.4 Steering evolution of systems-of-systems

Keywords: Software Architecture, Systems-of-Systems, Evolution.

Participants: Isabelle Borne, Mohamed Hichem Fendali, Labiba Souici-Meslati.

This research line is about a system-theoretic study of how systems-of-systems evolution can be steered towards a desirable direction. For practitioners, this is a non-trivial undertaking because of the potential resistance to change from the independent constituents.

We highlight this subject as a gap in the literature on system-of-systems evolution, but not only. We reviewed the characteristics of the system-of-systems evolution context to establish that, despite the limitations in visibility and control, a system-of-systems will still exhibit many evolutionary tendencies and dynamics, that can be harnessed to steer its evolution. Based on this assumption, we developed a set of prescriptive strategies that practitioners can turn to when they need a system-of-systems to move in any desired direction. The usefulness and applicability of the proposed strategies are illustrated through supporting examples from the case of a commonly recognized real-world collaborative system-of-systems, that of the Internet.

3.5 Addressing cybersecurity in software-intensive systems and systems-of-systems

Keywords: Security by Design, Software Architecture, Vulnerability Modeling, Attack Modeling, Threat Modeling, Risk Analysis.

Participants: Nicolas Belloir, Isabelle Borne, Jérémy Buisson, Monica Buitrago, Jamal El Hachem, Ahmed Elmarkez, Régis Fleurquin, Flavio Oquendo, Jesus Ramos, Salah Sadou, Sidbewendin Yameogo.

The software architecture provides the right abstraction level to address the issue of security-by-design in software-intensive systems and systems-of-systems (SoS). Indeed, the research focus on SoS correctness-by-design, which has been the main focus of ArchWare, has evolved to integrate SoS security-by-design, which will be the basis for the creation of the new SecReizh research team which will succeed ArchWare.

This research line addresses, thereby, the research challenges raised by architecting secure software-intensive systems and SoSs.

In the sequel, the main results that this research line produced in 2023 are presented.

3.5.1 Measuring the conformance to software architecture good practices in a security-by-design approach

Keywords: Software Architecture, Cybersecurity, Metrics.

Security-by-design consists of considering security concerns throughout the whole development lifecycle and, in this way, detecting and fixing potential security-related issues as early as possible. To implement this approach, one of the challenges is to provide the software architect with a means to assess the security level of the software architecture. While several metrics were proposed to measure the attack surface, the attackability, and the satisfaction of security requirements at the stage of the software architecture, proving the correlation between these metrics and the security concern is far from trivial.

In this work, we developed four metrics inspired by security architectural patterns, as well as NIST guidelines and CWE. The metrics we have defined evaluate the conformance of the software architecture to these security patterns and guidelines, focusing on the separation of user/administrative privileges at the entry points and on the isolation of security functions at the leaves. By doing so, the relation to security concerns follows from the expertise of the authors of the patterns and guidelines. The connection between our metrics and these patterns and guidelines intrinsically attracts the architect's attention to solutions to improve the architecture in this regard. For details, see: [4].

3.5.2 Security by design: designing secure industrial systems in the industrial IoT

Keywords: Cybersecurity, Security by Design, SCADA, Industrial IoT (IIoT).

Security by design is rapidly becoming an essential approach in the rapid development of systems, in particular industrial control systems, in the Internet of Things (IoT), and especially in the Industrial IoT (IIoT) towards Industry 4.0. One of the major security challenges in industrial control systems deployed in the IIoT is that they involve both Operational Technology (OT) and Information and Communication Technology (ICT). Indeed, OT encompasses a broad range of programmable systems and devices that interact with the physical environment (or manage devices that interact with the physical environment) while being increasingly distributed or accessible via the IoT in particular the IIoT.

In OT, security-by-design ensures that security controls are built into the design of the OT, in our case focusing on industrial control systems, rather than as an afterthought. Such an approach reduces the likelihood of cybersecurity breaches.

The issue that we address in this research line is how to secure operational technology (OT) according to the security-by-design approach while addressing their unique performance, reliability, and safety requirements.

In this work, the aim is to incorporate commonly used security principles, strategies, tactics, and techniques into the architectural design process of Supervisory control and data acquisition (SCADA) systems, which will ensure the appropriate security measures should the system fall under attack. In reality, traditional security measures like vulnerability assessments and penetration testing are insufficient to guarantee their security. SCADA systems are used to control dispersed assets, where centralized data acquisition is as important as control. A mapping study on security-by-design of software-reliant systems has been realized and is now under review. It will be submitted for publication in 2024.

3.5.3 Characterizing fake news: a conceptual modeling-based approach

Keywords: Conceptual Modeling, Characterization, Fake News, Explainable Artificial Intelligence.

For some time, and even more so now, Fake News has increasingly occupied the media and social space. How to identify Fake News and conspiracy theories has become an attractive research area. However, the lack of a solid and well-founded conceptual characterization of what exactly Fake News is and what are its main characteristics, make it difficult to manage their understanding, identification, and detection. This work advocates that conceptual modeling must play a crucial role in characterizing Fake News content accurately. Only by delimiting what Fake News is will it be possible to understand and manage their different perspectives and dimensions, with the ultimate goal of developing a reliable framework for online Fake News detection, as much automated as possible. To contribute in that direction from a pure and practical conceptual modeling perspective, this work has proposed a precise conceptual model of Fake News, an essential element for any explainable Artificial Intelligence (XAI)-based approach that must be based on the shared understanding of the domain that only such an accurate conceptualization dimension can facilitate. For details, see: [3].

4 Software development

4.1 The SoS Architect Studio for SosADL: SosADL Studio

Participants: Gersan Moguérou, Jérémy Buisson, Milena Guessi, Elena Leroux, Valdemar Neto, Flavio Oquendo.

SosADL Studio, the SosADL Architecture Development Environment, is a novel environment for description, verification, simulation, and compilation/execution of SoS architectures. With SosADL Studio, SoS architectures are described using SosADL, an Architecture Description Language based on process algebra with concurrent constraints, and on a meta-model defining SoS concepts. Because constituents of an SoS are not known at design time, SosADL promotes a declarative approach of architecture families. At runtime, the SoS evolves within such a family depending on the discovery of concrete constituents. In particular, SosADL Studio enables to guarantee the correctness of SoS architectures.

SosADL Studio includes the following modules:

4.1.1 The type system in Coq, the type-checker and the proof generator

Participants: Jérémy Buisson, Gersan Moguérou.

The type-checker is based on the SosADL type system written in Coq, which covers 2/3 of the SoSADL language. Coq proofs are generated after each successful type checking, enabling the verification of the type-checker according to the type system.

4.1.2 SosADL2Alloy: generating concrete SoS architectures based on SosADL

Participants: Milena Guessi, Gersan Moguérou, Flavio Oquendo.

The concrete architecture generator (SosADL2Alloy) module automatically transforms a SosADL abstract architecture into an abstract architecture in Alloy, and generates a Java class to launch a SAT solver through the Alloy Analyzer. The SAT solutions represent SosADL concrete architectures. During the integration of this module into the SosADL Studio, it has been improved to represent the generated concrete architectures in SosADL.

4.1.3 SosADL2DEVS: generating and simulating concrete architectures

Participants: Valdemar Neto, Wallace Manzano, Gersan Moguérou.

The SosADL2DEVS generator takes one concrete architecture as input and generates a DEVS program, which can be verified using ioSTS/Uppaal and simulated using the MS4ME simulation tool. The simulations generate traces. A client-server link between MS4ME and PlasmaLab enables Statistical Model Checking, by reusing traces of the simulation. The SosADL2DEVS module now generates DEVS programs, which can evolve dynamically during a simulation inside MS4ME.

This module has been integrated in the SosADL Studio. Currently, this module translates SosADL concrete architectures into DEVSNL, the language supported by MS4ME.

4.1.4 SosADL2IoSTS: the SosADL support for architecture verification

Participants: Elena Leroux, Gersan Moguérou.

The SosADL2IoSTS generator takes one concrete architecture, and generates an ioSTS model in order to verify functional properties of SoS. The development of the translator from ioSTS to Uppaal has been carried out for automating that generation.

4.1.5 The SoSADL Studio IDE

Participants: Gersan Moguérou, Jérémy Buisson, Elena Leroux, Milena Guessi, Valdemar Neto, Flavio Oquendo.

The SoSADL Studio provides an Integrated Development Environment (IDE), a simulator, a model-checker, and a statistical model-checker.

The SosADL Studio was developed under Xtext/Eclipse. It integrates the above modules into an IDE, which provides a syntactical editor to define an abstract SoS architecture, and then enable the following workflow:

- The type-checker validates the abstract SoS architecture written in SosADL, and generate a Coq proof. This proof can be verified using the Coq proof assistant, according to the SosADL type system written in Coq.
- The concrete SoS architectures are then generated, by the execution of the SosADL2Alloy module.
- Each concrete architecture is transformed into ioSTS, and then into an Uppaal NTA, in order to verify functional properties by Model Checking.
- Each concrete architecture is transformed into a DEVS program, by the execution of the SosADL2DEVS module, and simulated using the MS4ME tool. The traces of the simulation enable Statistical Model Checking in PlasmaLab.

Based on all these modules and supported execution steps of the SosADL workflow, the overall SosADL Studio was implemented.

The development of the integrated alpha version of SosADL Studio started in 2020 and was executed by Gersan Moguérou. The grammar and meta-model have been completely redesigned using Eclipse/Xtext for simplifying its maintenance. The scoping and type-checker have been rewritten following Xtext principles. The generation of concrete architectures has been re-implemented, ending in 2023.

The SosADL Studio software has its alpha version, as an integrated development environment, terminated in 2023.

5 Contracts and collaborations

5.1 National initiatives

- Public-private collaboration on the cybersecurity of large public events between the Université Bretagne Sud, Gendarmerie Nationale and GICAT (industrials in defense) in the domain of the cybersecurity of systems-of-systems. This ongoing collaboration is achieved through an industrial chair led by Salah Sadou. It aims to support the design and operation of large-scale sociotechnical systems-of-systems in open environments. It, in particular, aims to support the 2024 Summer Olympics in Paris. The ARCHWARE team brings to this joint R&D project its expertise on security-by-design for mastering emergent behaviors in sociotechnical SoS architectures.

5.2 Bilateral industry grants

- SEGULA Engineering (Numéro de contrat Ouest Valorisation 2018-01307, renewed for 5 additional years as amendment 2022-01275): Bilateral collaboration on cybersecurity in software architecture for industrial systems and systems-of-systems in the Industrial Internet-of-Things: Bilateral collaboration between ARCHWARE and the R&D division of SEGULA, a multinational systems engineering company developing large-scale systems and SoSs in different domains, including automotive, aeronautics, naval, and railway engineering. This ongoing collaboration aims to support the model-based engineering of critical systems-of-systems in the Industrial Internet-of-Things. The ARCHWARE team brings to this joint R&D project its expertise on formal approaches for the specification and verification of software architectures of SoSs. An additional PhD student has been funded by SEGULA under the scientific supervision of ARCHWARE, led by Flavio Oquendo, in this contract since November 2022 (for 3 years).
- DAWIZZ (Numéro de contrat Ouest Valorisation 2021-01361): "VOODOO METADATAS - Discovery, generation and analysis of relevant metadata from heterogeneous data through machine learning": 2 years project in the context of Post COVID-19 government recovery plan.

5.3 Industrial partners in CIFRE grants

- SEGULA Engineering;
- THALES Fondation St Cyr, Chaire "Cyberdefense-Cybersécurité" St Cyr Thales.

5.4 Industrial academic chairs

- Chaire "Cyberdefense-Cybersécurité" St Cyr Thales (2023-2026), headed by Nicolas Belloir.

5.5 Academic research collaborations

Members of ARCHWARE actively participate in the following academic research collaborations.

National and international research networks:

- Salah Sadou, Isabelle Borne and Nicole Levy (CNAM Paris) leads the GT GL-Sec (working group on Software engineering and security) in the GDR GPL (Groupement de Recherche Génie de la Programmation et du Logiciel - INS2I-CNRS).
- Flavio Oquendo is a member of the INCOSE (International Council on Systems Engineering) System-of-Systems Working Group as well as a member of the AFIS (Association Française d'Ingénierie Système), French chapter of INCOSE, Technical Committee 3SAI (System-of-Systems & Services - Architecture & Engineering).

International research cooperation:

- Flavio Oquendo has a collaboration with UFRN in the framework of the DTArch research action on Digital Twins Architecture for Software-intensive Systems (Contract CNPq-MCTI).

National collaborations with joint publications:

- Flavio Oquendo has a collaboration on systems-of-systems architectures with Khalil Drira (LAAS-CNRS);
- Salah Sadou has a collaboration on reuse of architectural constraints with Chouki Tibermancine (LIRMM).

International collaborations with joint publications:

- Nicolas Belloir has a collaboration on the conceptual modeling of fake news with Oscar Pastor (Polytechnic University of Valencia, Spain);
- Jamal El Hachem has a collaboration on modeling and analysis of cybersecurity with an application on smart buildings and C3i systems (Command, Control, Communication, and Intelligence (C3i) systems) with Ali Babar (University of Adelaide, Australia);
- Jamal El Hachem has a collaboration on attacks modeling and analysis via Bayesian networks and Game theory, with an application in the field of autonomous vehicles with Elena Lisova (University of Malardalen, Sweden);
- Jamal El Hachem has a collaboration on vulnerability knowledge bases evaluation with Mehdi Mirakhorli (Rochester Institute of Technology, New York, USA);
- Flavio Oquendo has a collaboration on software architecture with Thais Batista and colleagues on software architecture modeling (UFRN - Federal University of Rio Grande do Norte, Natal, Brazil);
- Salah Sadou has a collaboration on architecture security with Felipe Restrepo-Calle (University National, Colombia);
- Salah Sadou has a collaboration on software engineering and cybersecurity with Mohamed Lamine Kerdoudi (Univ. Biskra, Algeria);
- Salah Sadou has a collaboration on code vulnerability with Tegawendé F. Bissyandé (SnT - Univ. of Luxembourg).

National collaborations with joint PhD supervision:

- Nicolas Belloir:

  - Centrale-Supelec, France, Ph.D. co-supervison with Wassila Ouerdane on the topic of improving understanding of misinformation campaigns using semantic analysis of Fake News.

- Jamal El Hachem:
  - IMT Atlantique, Lab-STICC, France, Ph.D. co-supervison with Yvon Kermarrec in the context of the "Chaire de Cyberdéfense des Systèmes Navals".
  - Telecom Paris, France, Research collaboration with Dominique Blouin on the topic of cybersecurity modeling and analysis.

- Régis Fleurquin:
  - Centrale-Supelec, France, Ph.D. co-supervison with Wassila Ouerdane on the topic of improving understanding of misinformation campaigns using semantic analysis of Fake News.

- Flavio Oquendo:
  - Lab-STICC, France, Ph.D. co-supervison with Pascal Berruet on security-by-design of industrial control systems.

- Salah Sadou:
  - CEA, France, PhD supervision on machine learning to improve formal code verification.

International collaborations with joint PhD supervision:

- Isabelle Borne:
  - University Badji Mokhtar, Annaba, Algeria: Ph.D. co-supervision with Labiba Souci-Medlati.

- Jamal El Hachem:
  - KU Leuven, Belgium: Ph.D. co-supervision with Yves Wautelet (KU Leuven) and Samedi Heng (HEC Liège) on risk governance: issues and solutions associated with digital transformation.

- Flavio Oquendo:
  - UFRN - Federal University of Rio Grande do Norte, Natal, Brazil: PhD co-supervision with Thais Batista on executable software architectures for software-intensive systems.

6 Dissemination

6.1 Promoting scientific activities

Research and Doctoral Supervising Awards (PEDR)

- Jamal El Hachem : PEDR (2021-2025);
- Flavio Oquendo: PEDR (2020-2024);
- Salah Sadou: PEDR (2018-2022).

Chair/member of conference steering committees

- Flavio Oquendo:
  - European Conference on Software Architecture - ECSA (Steering Committee Chair);
  - IEEE International Conference on Software Architecture - ICSA (Steering Committee Member);
  - Conférence francophone sur les architectures logicielles - CAL (Steering Committee Member);
  - ACM/IEEE International Workshop on Software Engineering for Systems-of-Systems and Software Ecosystems - SESoS (Steering Committee Member);
  - International Workshop on Digital Twin Architectures - TwinArch (Steering Committee Co-Chair).
- Salah Sadou:
  - CIEL: French Conference on Software Engineering (Steering Committee Chair).

Chair/member of conference program committees

- Isabelle Borne:
  - SoSE: IEEE International Conference on System-of-Systems Engineering, 2023.
- Jamal El Hachem:
  - DeMeSSA: Co-Chair of the International Workshop on Designing and Measuring Security in Software Architectures at the European Conference on Software Architecture, 2023;
  - SoSE: International Conference on System-of-Systems Engineering, 2023;
  - SESoS: ACM/IEEE International Workshop on Software Engineering for Systems-of-Systems and Software Ecosystems, 2023;
  - MPM4CPS: International Workshop on Multi-Paradigm Modelling for Cyber-Physical Systems, 2023;
  - ECSA-DE&I: European Conference on Software Architecture - Diversity, Equity and Inclusion Track, 2023.
- Flavio Oquendo:

- ECSA: European Conference on Software Architecture, 2023;
- SoSE: IEEE International Conference on System-of-Systems Engineering, 2023;
- ICT4S: International Conference on Information and Communications Technology for Sustainability, 2023;
- ICSEA: International Conference on Software Engineering Advances, 2023;
- ICAS: International Conference on Autonomic and Autonomous Systems, 2023;
- TwinArch: International Workshop on Digital Twin Architecture at the European Conference on Software Architecture (PC Co-Chair), 2023
- DeMeSSA International Workshop on Designing and Measuring Security in Software Architectures at the European Conference on Software Architecture at the European Conference on Software Architecture (Co-Organizer), 2023;
- SESoS: ACM/IEEE International Workshop on Software Engineering for Systems-of-Systems and Software Ecosystems at the International Conference on Software Engineering, 2023;
- ICSE-SEET Track: International Conference on Software Engineering - Track on Software Engineering Education and Training, 2023.

• Salah Sadou:
  - ECSA: European Conference on Software Architecture, 2023;
  - SoSE: IEEE International Conference on System-of-Systems Engineering, 2023;
  - ICECCS: International Conference on Engineering of Complex Computer Systems, 2023;
  - SESoS: ACM/IEEE International Workshop on Software Engineering for Systems-of-Systems and Software Ecosystems, 2023;
  - AISC: Australasian Information Security Conference, 2023;
  - ACSW: Australian Computer Science Week, 2023;

6.1.1 International conference keynotes

• Flavio Oquendo: Keynote Speaker at the 17th European Conference on Software Architecture (ECSA 2023): Software Architecture in the Era of Collective Intelligence: The Rise of Systems-of-Systems.

6.1.2 International journal boards

Member of the Editorial Boards

• Flavio Oquendo:
  - Springer Journal of Software Engineering Research and Development (Member of the Editorial Board).

6.1.3 Scientific expertise

- Jamal El Hachem :
  - Scientific Expert acting as reviewer and evaluator of ANR JCJC.
- Flavio Oquendo:
  - Scientific Expert acting as reviewer and evaluator of R&D Projects for the European Commission (Horizon Europe);
  - Scientific Expert acting as evaluator of R&D Proposals for the ESF European Science Foundation on Software Sciences and Engineering (Europe);
  - Scientific Expert acting as evaluator of R&D Proposals for the Swiss National Science Foundation (Switzerland);
  - Scientific Expert acting as evaluator of R&D Proposals for the FWO (Research Foundation Flanders) on Software Sciences and Engineering (Belgium);
  - Scientific Expert acting as evaluator of R&D Projects for the MESR-CIR on Software Engineering and Technologies;
  - Distinguished Professor acting as evaluator of Scientific Standing and Achievements to Full Professorship for different universities in Europe on Computer Sciences and Software Engineering, e.g., for the Helmholtz Association of German Research Centers on Software Sciences and Engineering (Germany).

6.1.4 Academic councils

- Isabelle Borne: Deputy director of the Doctoral School MathSTIC-Bretagne-Océane;
- Régis Fleurquin: Member of the CAC (Commission recherche du conseil académique) of Université Bretagne Sud;
- Jamal El Hachem: Member of the CL (Conseil du Laboratoire) of IRISA.

6.2 PhD supervision and teaching

6.2.1 PhD supervision or co-supervision

- Monica Buitrago, UBS (since January 2021) by Isabelle Borne and Jérémy Buisson;
- Philippe Charton, UBS (since March 2023) by Salah Sadou;
- Ahmed Elmarkez, UBS (since November 2022) by Flavio Oquendo;
- Mohamed Fendali, UBS/Badji Mokhtar-Annaba University (part-time, since September 2016) by Isabelle Borne;

- Etienne Lemonier, UBS/Saint-Cyr (since October 2023) by Nicolas Belloir, Jérémy Buisson, and Jamal El Hachem;
- Maykel Mattar, UBS/CEA (since December 2023) by Salah Sadou;
- Jesus Ramos, UBS (since January 2023) by Nicolas Belloir, Jérémy Buisson, and Jamal El Hachem;
- Lucas Tesson, UBS (since October 2023) by Jérémy Buisson and Jamal El Hachem;
- Jeisson Vargas, UBS/National University of Colombia (since April 2023) by Salah Sadou;
- Tales Viglioni, UBS/UFRN (since October 2023) by Flavio Oquendo;
- Sidbewendin Yameogo, UBS/CRéA (since December 2022) by Régis Fleurquin and Nicolas Belloir.

6.2.2 PhD juries

- Jamal El Hachem:
  - Examiner in Jury of PhD defense by Islam Debicha at Université Libre de Bruxelles, Belgium;
  - Examiner in Jury of PhD defense by Zakarya Kamagate at IMT Atlantique, France;
  - Examiner in Jury of PhD defense by Rodrigue Goran at IMT Atlantique, France;
- Flavio Oquendo:
  - *Rapporteur* in Jury of PhD defense by Hector Rengifo at University of Groningen, The Netherlands;
  - *Rapporteur* in Jury of PhD defense by Mouhamad Almakhour at Université Paris-Est-Créteil-Val-de-Marne (Paris XII), France;
  - Member in Jury of PhD defense by Camila Araujo at Universidade Federal do Rio Grande do Norte (UFRN), Brazil.

6.2.3 University teaching

- Academics of ARCHWARE teach at the Research Master on Computer Science of Université Bretagne Sud which is part of the regional SIF master administered by a consortium of the main computer science universities and graduate schools in Brittany: Université de Rennes 1, Université de Bretagne Sud, ENS Rennes, National Institute of Applied Sciences, Rennes (INSA) and CentraleSupélec.

6.2.4 Teaching responsibility

- Elena Leroux: Study Director of the Engineering Degrees of the ENSIBS School of Engineering;
- Flavio Oquendo: Head of the Research Master Degree on Computing of Université Bretagne Sud (part of the regional SIF research master in Computer Science headed by Université Rennes 1);
- Salah Sadou: Head of the Engineering Degree on Software Cybersecurity of ENSIBS School of Engineering.

Books and Monographs

[1] *Proceedings of the 2nd International Workshop on Digital Twin Architecture (TwinArch 2023) at the 17th European Conference on Software Architecture (ECSA 2023)*, *LNCS*, Springer, September 2023, `https://hal.science/hal-04463278`.

[2] *Proceedings of the 3rd International Workshop on Designing and Measuring Security in Software Architectures (DeMeSSA 2023) at the 17th European Conference on Software Architecture (ECSA 2023)*, *LNCS*, Springer, September 2023, `https://hal.science/hal-04463302`.

Publications in Conferences and Workshops

[3] N. Belloir, W. Ouerdane, O. Pastor, "Caractérisation des fausses nouvelles : Une approche basée sur la modélisation conceptuelle", *in: 41st INFORSID 2023*, p. 103–104, La Rochelle, France, May 2023, `https://hal.science/hal-04453332`.

[4] M. Buitrago, I. Borne, J. Buisson, "Deriving Metrics for Software Architectures from the Protected Entry Points Security Patterns", *in: 38th ACM/SIGAPP Symposium on Applied Computing*, ACM, p. 1473–1475, Tallinn, France, March 2023, `https://hal.science/hal-04121611`.

[5] M. L. Kerdoudi, T. Ziadi, C. Tibermacine, S. Sadou, "A Novel Approach for Software Architecture Product Line Engineering", *in: ICSA 2023 - 20th IEEE International Conference on Software Architecture*, L'Aquila, Italy, March 2023, `https://hal.science/hal-04445840`.

[6] F. Oquendo, "Fuzzy Mediating Control Systems for Automating Vehicle Driving Maneuvers: The Overtaking Case", *in: 2023 R10 International Conference (TENCON 2023)*, IEEE, IEEE, p. 789–794, Chiang Mai, Thailand, October 2023, `https://hal.science/hal-04463117`.

[7] F. Oquendo, "Software Architecture in the Era of Collective Intelligence - The Rise of Systems-of-Systems (Keynote)", *in: 17th European Conference on Software Architecture (ECSA 2023)*, *LNCS 14212*, Springer, Istanbul, Turkey, September 2023, `https://hal.science/hal-04463266`.

Internal Reports

[8] S. Taillat, D. Danet, S. Haddad, J. Nocetti, N. Belloir, K. Limonier, "Cognitive Security: an Analytical Framework", *research report*, Service Européen d'Action Extérieure, 2023, https://hal.science/hal-04253257.