UMR IRISA

Activity Report 2022

Team ARCHWARE

Architecting Software-intensive Systems and Systems-of-Systems

D4 – Language and Software Engineering

Contents

This activity report covers the year of 2022 of the ArchWare Research Team.

1 Team composition

Researchers and faculty

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

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
Maykel Mattar, Research Engineer (IGR contractuel, projet VOODOO-METADATAS)

PhD students

Philippe Charton
Ahmed Elmarkez
Elia Christy Fikany
Brendan Le Trionnaire
Paul Perrotin
Monica Buitrago Ramirez
Jesus Antonio Sanchez Ramos
Jeisson Andres Vergara Vargas
Thierry Waszak
Sidbewendin Yameogo

External collaborators in particular in cotutelles of PhD theses

Thais Batista, Tegawendé F. Bissyandé, Everton Cavalcante, Djamel Meslati, Mohamed Ahmed Nacer, Jair Leite, Marcel Oliveira, Chouki Tibermacine

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 component-based and service-oriented 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.

Besides, in SoSs, architectures are designed to fulfill specified missions. Indeed, an important concern in the design of SoSs is the systematic modeling of both global and individual missions, as well as all relevant mission-related information. Missions play a key role in the SoS context since they define required capabilities of constituent systems and the interactions among these systems that lead to emergent behaviors towards the accomplishment of the global mission of the SoS.

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 targeting year 2020 and beyond.

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 2014, 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). In 2015, the CSA CPSoS presented a research agenda for developing cyber-physical SoSs.

All these roadmaps show 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 (transversely to application domains).

It is worth to note that software-intensive system-of-systems is an emergent domain in the research community. The systematic mapping of the literature shows that 75% of the publications related to the architecture of systems-of-systems have been published in the last 5 years and 90% in the last 10 years. Furthermore, most of these publications raise open-issues after having experimented existing approaches for architecting systems-of-systems.

Overall, the long-term research challenge raised by SoSs calls 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 2022, addressed applications areas include:

1. Internet-of-Things (IoT), Industrial Internet-of-Things (IIoT), Internet-of-Vehicles (IoV);
2. Intelligent Transportation Systems;
3. Battlefield Engineering;
4. 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), Mission Specification, 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 run-time for developing the SoS based on discovered constituents.
- Systems architectures often evolve offline. In the opposite, SoS architectures always evolves online.

In 2022, the effort in this line of research was mainly dedicated to the implementation of SosADL as a software IDE (as presented in the Software section afterward).

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, the behavior, and the execution of a software architecture of a software intensive-system.

From the architectural perspective, in single systems, the controlled characteristics of components are under the authority of the system architect, and the stable notion of connectors linking these components is mostly decided at design time. 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 line of research produced in 2022 are presented.

3.2.1 Formalizing SysADL: model-driven approach for generating formal software architecture descriptions: from SysADL to π-ADL

Keywords: Model-Driven Development, Model Transformation, Architecture Description Language, Formal Verification, SysML.

The critical nature of many complex 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 have preferred using standard notations, in particular SysML and their 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 track developed 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 is providing formal semantics to SysADL, a SysML-based language to describe software-intensive system architectures. The formal semantics is provided by π-ADL, a formal architecture description language. A model-to-model transformation was defined and implemented to concretize the mapping between the elements of these languages and hence automatically generate formal architecture descriptions in π-ADL from SysADL. This work developed a proof-of-concept to validate the mapping between SysADL and π-ADL and an exploratory study on the transformation performance. The first results were published in [4], and the current results are planned to be published in a journal article to be submitted in 2023.

3.2.2 Empowering SysADL with formal verification: from SysADL to CSP

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

One of the many purposes of software architecture descriptions is to contribute to an early analysis of the architecture with respect to quality attributes. The critical nature of many software systems calls for formal approaches aiming at precisely verifying if their designed architectures can meet important properties such as consistency, completeness, and correctness. In this context, it is worthwhile investigating the role of architecture descriptions to support the formal verification of software architectures to ensure their quality, as well as how such a process happens and is supported by existing languages and verification tools. To evaluate the research landscape on this subject, we have carried out in this research track a systematic mapping study in which we collected and analyzed studies available in the literature on formal verification of architecture descriptions. This research track contributed with a structured overview and taxonomy of the current state of the art on this topic as well as the elicitation of key open questions. The first results were published in [7], and the current results are planned to be published in a journal article to be submitted in 2023.

3.3 Addressing design and operational aspects of software-intensive systems-of-systems

Keywords: Software Architecture, Modeling, Simulation, Reconfiguration, Systems-of-Systems.

Participants: Salah Sadou, Jérémy Buisson, Nicolas Belloir.

The architecture provides the right abstraction level to address the complexity of 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.

In the sequel, the main results of this line of research produced in 2022 are presented.

3.3.1 SoS for the battlefield

Keywords: Military SoS, Battlefield Engineering, Model-Based Engineering, Operation Orders, Systems-of-Systems.

Digitalization of the whole society will change the way SoSs have to be considered. Remaining independently operated and managed, SoSs increase their collaboration skills using shared or cooperated information systems. People can be seen as particular digital subsystems due to smart equipment they can use. Military operations, which are considered as typical SoS, are no exception to this fact. Foreseen evolution with the current French programs is towards mixing command post, human soldiers, their equipment, remotely operated robots and autonomous robots interacting in a single SoS. New operational doctrines have to be created to take advantage of those new capabilities. In this research track, we developed new methods supported by tools inspired by software engineering to create new automated capabilities in battlefield engineering. They support the direction which should be considered in the area of battlefield engineering in order to deal with those new capabilities. Inspired from Model-Based Engineering, we developed a metamodel inspired by the French PROTERRE doctrine (soldier to platoon levels) in such a way that the metamodel is suitable in a multimodel interaction scenario, including both human and robotic systems. For details see: [2].

3.4 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, Régis Fleurquin, Jesus Ramos, Salah Sadou, Sidbewendin Yameogo.

The architecture provides the right abstraction level to address the issue of security-by-design in software-intensive systems and SoSs.

This research line addresses the research challenges raised by architecting secure software-intensive systems and SoSs. Different aspects of cybersecurity have been addressed at different levels, directly or indirectly related to systems & software architectures.

In the sequel, the main results of this line of research produced in 2022 are presented.

3.4.1 Security measures: measuring security in architecture descriptions

Keywords: Software Architecture Description, Cybersecurity, Patterns, Metrics.

When designing a software architecture, some well-established patterns are known to improve the security of the software system being designed. For instance, having a well-identified single entry point to a subsystem eases the interactions with the policy enforcement point and policy decision point that implement access control to the services and resources of that subsystem. But whether such patterns are actually and suitably used in software design has still to be measured. Our first outcome to this issue is a literature review. It confirmed that the architecture design affects the security hardening. We found several metrics that attempt to assess the susceptibility of vulnerabilities at the code level. But, our literature review also shows the lack of existing metrics at the architecture level to address the specific question of design-time security.

3.4.2 Cybersecurity risk assessment: towards the integration of cybersecurity risk assessment into model-based requirements engineering

Keywords: Model-Based Systems Engineering, Security requirements, Risk assessment, System and security co-engineering..

Engineering projects require to consider the increasingly significant needs and constraints regarding expected behaviors, services, quality and security. These requirements are introduced into system and software engineering projects as functional and non-functional properties. Satisfying such properties implies rigorous processes that steer the project, from the requirements identification and definition to the system deployment and maintenance. Model-Based System Engineering (MBSE) is an effective approach to address security requirements and risk assessment at the early stages of the development life cycle, which enables cost-efficient fixes. The aim of this work is to investigate how cybersecurity risk assessment could be integrated into model-based requirement engineering. We propose a Model Based Cyberisk Assessment (MBCA) method, that comprises: (1) A semantic alignment between risk assessment concepts and system modeling concepts and (2) A modeling language extension to represent security concepts and metrics throughout the system modeling life cycle. To illustrate our approach, validate its applicability and evaluate its expressiveness, we applied it to an industrial in-flight entertainment system, presented in [8].

3.4.3 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.

Its 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 started.

3.4.4 Vulnerability analysis: analyzing vulnerability using the architecture of Socio-Technical Systems-of-Systems

Keywords: Human Vulnerability, Human Models, Cybersecurity, Vulnerability Propagation.

Today, most complex and large systems integrate the physical and human aspects of computing and networking. They constitute a system-of-systems (SoS) with a socio-technical part. The human element is nowadays one of the most important attack vectors in the context of SoSs. In this context, the estimation of the impact of human vulnerability on these systems enables a more secure design and an integration of the system development cycle with security concerns according to the security-by-design principles. To improve the resilience of these SoSs with respect to the vulnerability that humans bring, it is necessary to be able to estimate the impact that an individual can have on the system. In this line of research, we have developed an approach that enables to assess the impact of human vulnerability on an SoS composed of humans, which in this case will be a social-technical SoS (STSoS). We propose to use behavioral models to model the propagation of a human vulnerability in a STSoS as well as to use different models in order to capture the plurality of attacks on STSoS. For details see: [10].

In addition, the level of responsibility dedicated to the human operator has grown, particularly in Socio-Technical SoSs where humans are considered as a kind of system. Like every system, the human operator can fail by behaving in undesired ways, and consequently have a negative impact on the SoS. Thus, to improve the resilience of the overall SoS, it is necessary to manage the vulnerability of humans. In this research line, we worked on an approach to assess human vulnerabilities in a STSoS through its architecture. We have proposed a model that describes the STSoS, based on human characteristics having a significant impact on human vulnerabilities. We have then defined an assessment metric for each characteristic. Finally, we have proposed an approach allowing not only to assess the vulnerability of a specific human in the SoS, but also to understand how a vulnerability propagates through the SoS. We implemented this approach with a dedicated architecture description language, called Hos-ML, allowing the architect to deal with STSoS vulnerabilities. For details see: [9].

3.5 Other works on software architecture or cybersecurity

Keywords: Software Architecture, Cybersecurity.

In addition to the main research lines described, work has also addressed specific tasks that are in the domains of Software Architecture or Cybersecurity, that pave the way to contribute to future research lines.

3.5.1 Software architecture in product line engineering

Keywords: Software Architecture, Product Line Engineering.

Software systems exist in different forms, as different variants targeting different needs and users. This kind of system is generally provided as a set of independent products and not as a single whole. Developers use ad-hoc mechanisms to manage variability. In this research line, we defend a vision of software development where we consider a Software Product Line (SPL) architecture starting from which the architecture of each variant can be derived before its implementation. Indeed, each derived variant can have its own lifecycle. In this work, we propose a novel approach for Software Architecture Product Line (SAPL) Engineering. It consists of: (i) a generic process for recovering a SAPL model which is a product line of "software architectures" from large-sized variants; (ii) a forward-engineering process that uses the recovered SAPL to derive new customized software architecture variants. Our approach was first experimented on thirteen Eclipse variants to create a new SAPL. Then, an intensive evaluation was conducted using an existing benchmark based on Eclipse IDE. Our results showed that we can accurately reconstruct such a SAPL and derive effectively pertinent variants. Our study provided insights that recovering SAPL and then deriving software architectures offers good documentation to understand the software before changing it. For details see: [3].

3.5.2 Model-driven engineering: interface for tactical operation order of mixed robot-human platoons

Keywords: Model-Driven Engineering, Tactical operation order, Mixed platoon.

Technological advances are enablers for the evolution of modern warfare for armies. But the technological challenges are far beyond building new weapon systems such as semi-autonomous robotic systems and drones. An additional challenge is the elaboration of the necessary infrastructure substrate that will enable the smooth integration of these semi-autonomous systems into teamed human-robot platoons. In this research line, we have explored how one can use Model-Driven Engineering (MDE), borrowed from Software Engineering, to address this specific challenge. We have carried out an experiment on designing a suitable metamodel that reifies the concepts from the PROTERRE tasks (standing for "projection, pour accomplir des missions principalement de protection, de professionnels de l armée de Terre") of the French group to company units. The metamodel is then used as the abstract interface between the chiefs, their human subordinates, and their robots, each using their own modalities. Preliminary results confirmed the suitability of MDE technologies in this context. We also showed that MDE adapts well to modalities that are unusual in Software Engineering, such as gesture communication. For details see: [2].

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 have become an extremely 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, makes it difficult to manage their understanding, identification, and detection. This research 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: [5] and [6].

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.

The alpha version of the SosADL Studio included 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. This dependence with MS4ME requires running the simulation on Windows or Linux.

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 is partially terminated.

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. We produced the SosADL eclipse plugin in its alpha version.

The development of the beta version of the SosADL Studio started in 2020. 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. Generating Coq proofs have not yet been integrated. The generation of concrete architectures has been re-implemented in 2022. This effort aims to prepare the final version of the SosADL Studio 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 leaded 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 (Prof. 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. This project provides funding for a research engineer.

5.3 Collaborations

National research networks:

- Members of ARCHWARE actively participate in the GDR GPL (Groupement de Recherche Génie de la Programmation et du Logiciel - INS2I-CNRS), in particular the team organized the 13es Journées nationales du GDR GPL in Vannes, 7-10 June 2022, https://gdr-gpl-2022.sciencesconf.org/, with ca. 130 attendees from major French CNRS Labs working in a wide range of topics in Software Engineering (initially planned for June 2021 and then reported due to the Covid pandemic to June 2022), jointly with the French-speaking conferences AFADL on Formal Methods and CAL on Software Architectures. For details, see `https://gdr-gpl-2022.sciencesconf.org/`.
- Salah Sadou, Isabelle Borne and Nicole Levy(CNAM Paris) lead the GT GL-SEc (working group on Software engineering and security) in the GDR GPL.
- 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 networks:

- Flavio Oquendo has a collaboration with PRAIA.AI - Applied Research Centers in Artificial Intelligence, Brazil, as part of the PRAIA International Committee.
- Salah Sadou has a collaboration with SnT - Univ. of Luxembourg on the subject of code vulnerability.

National collaborations with joint publications:

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

International collaborations with joint publications:

- 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);
- 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, 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).

National collaborations with joint PhD supervision:

- Jérémy Buisson:
  - École de l'Air, Salon-de-Provence, France.
- 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, research collaboration with Dominique Blouin on the topic of cybersecurity modeling and analysis.
- Flavio Oquendo:
  - Lab-STICC, France, Ph.D. co-supervison with Pascal Berruet.

International Collaborations with joint PhD supervision:

- Isabelle Borne:
  - University Badji Mokhtar, Annaba, Algeria (Labiba Souci-Medlati).
- Jamal El Hachem:
  - KU Leuven, Belgium, Ph.D. co-supervision (33%) with Yves Wautelet (KU Leuven), Samedi Heng (HEC Liège), on the subject: "Risk governance: issues and solutions associated with digital transformation".

- Flavio Oquendo:
  - UFRN - Federal University of Rio Grande do Norte, Natal, Brazil (Thais Batista);
  - Fraunhofer IESE - Fraunhofer Institute for Experimental Software Engineering, Kaiserslautern, Germany (Pablo O. Antonino, in preparation).
- Salah Sadou:
  - University of Science and Technology of Houari Boumedienne, Alger, Algeria (Mohamed Ahmed Nacer).

6 Dissemination

6.1 Promoting scientific activities

Research and Doctoral Supervising Awards (PEDR)

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

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, 2022;
  - AROSA: IEEE WETICE Conference Track on Adaptive and Reconfigurable Service-oriented and component-based Applications and Architectures, 2022;
  - SESOS: ACM/IEEE ICSE International Workshop on Software Engineering for Systems-of-Systems and Ecosystems, 2022;
  - ECSA Doctoral Symposium: European Conference on Software Architecture, 2022;
  - IWST: International Workshop on Smalltalk Technologies, 2022.
- Jérémy Buisson:
  - ICCS: International Conference on Computational Science, 2022.
- Jamal El Hachem:
  - DeMeSSA @ ECSA: Co-Chair of the International Workshop on Designing and Measuring Security in Software Architectures at the European Conference on Software Architecture, 2022;
  - SOSE: International Conference on System-of-Systems Engineering, 2022;
  - SESoS: ACM/IEEE International Workshop on Software Engineering for Systems-of-Systems and Software Ecosystems, 2022;
  - MPM4CPS: International Workshop on Multi-Paradigm Modelling for Cyber-Physical Systems, 2022;
  - MODELS: ACM / IEEE 25th International Conference on Model Driven Engineering Languages and Systems - Workshop proposals, 2022;
  - ECSA - DE&I: European Conference on Software Architecture - Diversity, Equity and Inclusion Track, 2022.
- Flavio Oquendo:
  - ICSA: IEEE International Conference on Software Architecture, 2022;
  - ECSA: European Conference on Software Architecture, 2022;
  - SOSE: IEEE International Conference on System-of-Systems Engineering, 2022;
  - CPSIOT: International Conference on Cyber-Physical Systems and IoT, 2022;
  - ICSEA: International Conference on Software Engineering Advances, 2022;
  - ICAS: International Conference on Autonomic and Autonomous Systems, 2022;
  - COMPLEXIS: International Conference on Complexity, 2022;

  - AROSA: IEEE WETICE Conference Track on Adaptive and Reconfigurable Service-oriented and component-based Applications and Architectures, 2022;
  - SESOS: ACM/IEEE International Workshop on Software Engineering for Systems-of-Systems and Software Ecosystems, 2022;
  - CAL: French Conference on Software Architecture, 2022.

6.1.1 International journal boards

Member of the Editorial Boards

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

6.1.2 Scientific expertise

- 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 ANR (Agence Nationale de la Recherche) on Software Sciences and Engineering;
  - Scientific Expert acting as evaluator of R&D Projects for the MESRI - CIR on Software Engineering and Technologies;
  - Scientific Expert acting as evaluator of R&D Proposals for the Helmholtz Association of German Research Centers on Software Sciences and Engineering (Germany);
  - Scientific Expert acting as evaluator of R&D Proposals for the FWO (Research Foundation Flanders) on Software Sciences and Engineering, including the SBO Strategic Basic Research Program (Belgium);
  - Scientific Expert acting as evaluator of R&D Proposals for the ESF European Science Foundation on Software Sciences and Engineering (Belgium);
  - Distinguished Professor acting as evaluator of Scientific Standing and Achievements to Full Professorship for different universities in Europe on Computer Sciences and Software Engineering.
- Jamal El Hachem :
  - Scientific Expert acting as reviewer and evaluator of ANR JCJC project.

6.1.3 Academic council

- 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;
- Isabelle Borne: Deputy director of the Doctoral School MathSTIC-Bretagne-Océane.

6.2 Teaching

6.2.1 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.

6.2.2 Teaching responsibility

- Jérémy Buisson: Deputy head of the Mastère Spécialisé "Conduite des Opérations et Gestion des Crises Cyber" of Académie Militaire de Saint-Cyr Coëtquidan;
- Jamal El Hachem: Temporary Head of the Cyberdefense engineering degree of the ENSIBS School of Engineering, May-August 2022;
- 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.

Doctoral dissertations and “Habilitation” theses

[1] P. Perrotin, *Analyse de la vulnérabilité humaine dans les systèmes de systèmes sociotechniques*, Theses, Ecole nationale supérieure Mines-Télécom Atlantique, December 2022, `https://theses.hal.science/tel-03969366`.

Articles in referred journals and book chapters

[2] N. Belloir, J. Buisson, L. Touseau, “Model-Driven Engineering as the Interface for Tactical Operation Order of Mixed Robot/Human Platoons”, *in: Developments and Advances in Defense and Security, Smart Innovation, Systems and Technologies, 255*, Springer Singapore, October 2022, p. 205–214, `https://hal.science/hal-03418759`.

[3] M. L. Kerdoudi, T. Ziadi, C. Tibermacine, S. Sadou, “A novel approach for Software Architecture Product Line Engineering”, *Journal of Systems and Software 186*, April 2022, p. 111191, `https://hal.sorbonne-universite.fr/hal-03885616`.

Publications in Conferences and Workshops

[4] C. Araújo, T. Batista, E. Cavalcante, F. Oquendo, "Generating Formal Software Architecture Descriptions from Semi-Formal SysML-Based Models: A Model-Driven Approach", *in: ICCSA 2021 - 21st International Conference on Computational Science and Its Applications*, Cagliari, Italy, September 2021, `https://hal.archives-ouvertes.fr/hal-03584963`.

[5] N. Belloir, W. Ouerdane, O. Pastor, É. Frugier, L.-A. de Barmon, "A Conceptual Characterization of Fake News: A Positioning Paper", *in: 16th International Conference on Research Challenges in Information Science (RCIS'22)*, Barcelone, Spain, May 2022, `https://hal.science/hal-03679073`.

[6] N. Belloir, W. Ouerdane, O. Pastor, "Characterizing Fake News: A Conceptual Modeling-based Approach", *in: ER 2022 - 41st International Conference on Conceptual Modeling*, Hyderabad, India, October 2022, `https://hal.science/hal-03697974`.

[7] F. M. Dias, M. Oliveira, T. Batista, E. Cavalcante, J. Leite, C. Araújo, F. Oquendo, "Empowering SysML-Based Software Architecture Description with Formal Verification: From SysADL to CSP", *in: 14th European Conference on Software Architecture (ECSA)*, L'Aquila, Italy, September 2020, `https://hal.archives-ouvertes.fr/hal-03585038`.

[8] D. Naouar, J. E. Hachem, J.-L. Voirin, J. Foisil, Y. Kermarrec, "Towards the Integration of Cybersecurity Risk Assessment into Model-based Requirements Engineering", *in: 2021 IEEE 29th International Requirements Engineering Conference (RE)*, p. 334–344, 2021.

[9] P. Perrotin, N. Belloir, S. Sadou, D. Hairion, A. Beugnard, "HoS-ML: Socio-Technical System ADL Dedicated to Human Vulnerability Identification.", *in: 2022 26th International Conference on Engineering of Complex Computer Systems (ICECCS)*, IEEE, p. 1–6, Hiroshima, France, March 2022, `https://hal.science/hal-03980301`.

[10] P. Perrotin, N. Belloir, S. Sadou, D. Hairion, A. Beugnard, "Using the architecture of Socio-Technical System to analyse its vulnerability", *in: 2022 17th Annual System of Systems Engineering Conference (SOSE)*, IEEE, p. 361–366, Rochester, France, June 2022, `https://hal.science/hal-03980317`.