Towards Model Driven Architecture and Analysis of System of Systems Access Control

Jamal El Hachem

Towards Model Driven Architecture and Analysis of System of Systems Access Control

Jamal El Hachem*
*LIUPPA Laboratory
University of PAU and Pays Adour, Mont de Marsan, France
jamal.elhachem@univ-pau.fr

***Abstract*—Nowadays there is growing awareness of the importance of Systems of Systems (SoS) which are large-scale systems composed of complex systems. SoS possess specific properties when compared with monolithic complex systems, in particular: operational independence, managerial independence, evolutionary development, emergent behavior and geographic distribution. One of the current main challenges is the impact of these properties on SoS security modeling and analysis. In this research proposal, we introduce a new method incorporating a process, a language and a software architectural tool to model, analyze and predict security architectural alternatives of SoS. Thus security will be taken into account as soon as possible in the life cycle of the SoS, making it less expensive.**

***Index Terms*—Model Driven Engineering, Maritime Security, Architectural Alternatives, Simulation.**

I. Introduction

Systems-of-Systems (SoS) are large-scale concurrent and distributed systems, comprised of complex systems [1]. Mair was more specific by defining the SoS as a collection of systems that must have five principal characteristics to differentiate them from complex monolithic systems [2]:
- Operational Independence of the elements: Each system of the SoS constituent systems may possess its own goals, and must be able to operate independently in order to achieve these goals.
- Managerial Independence of the elements: Each system of the SoS constituent systems may belong to different organizations/entreprises and they do operate independently, being managed at least in part for their own purpose.
- Evolutionary Development: The SoS's format is unstable, its development is subject to several insertions, modifications and suppressions of systems, functions and / or goals during its life cycle.
- Emergent Behavior: The main functions and purposes of the entire SoS do not remain in any constituent system, they emerge from the cumulative actions and interactions between these constituents.
- Geographic Distribution: The SoS constituent systems are geographically dispersed, accordingly the exchange between systems can be disrupted by disagreements between different national regulations.

Several other features can describe the behavior of the SoS, particularly [3]:
- Autonomy: Describe the capacity of a SoS's constituents to make decisions and achieve goals independently or together.
- Belonging: A system could be a member of the SoS if it has a role in enhancing the value of the system's objective.
- Connectivity: Constituent systems employing different protocols, vocabularies and data models could be able to exchange their information.
- Diversity: SoS is a complex system benefiting from the diverse and varied functions of its constituent systems.

These characteristics make the SoS a challenging domain with a fast (nearly exponential) growing for the past thirty years [4], in wich Europe is seeking global leadership as indicated by the T-AREA-SoS project[1].
These specific characteristics also impact the non-functional properties of SoS. One of these properties is security, on which we will focus. Traditionally, security deals with confidentiality, integrity and availability of data [5]. How these security properties could be described and verified in the context of SoS where different constituent systems communicate, coalesce and interact?

II. Motivating Scenario

An additional reason of the fast growing interest in SoS is the wide variety of its application domains, notably, defense and national security, intelligent transportation systems, aerospace, cyberspace, healthcare, electrical power grid and many other areas. Among 194 primary studies[2], Klein identifies that defense and national security is the most frequently discussed domain [6]. Therefore, in this paper, we present a motivating scenario inspired from the maritime safety and security case study presented in [7].

A. Scenario Description

This case study describes an SoS composed of geographically dispersed constituent systems: a Maritime Security Center (MSC) and three National Navy systems (Danemark, Netherlands and Italy). We model this SoS as a UML component diagram (Figure 1), to describe not only the SoS constituent systems and their input and output interfaces but also the basic components of each constituent system. As we can see in the diagram, the two main components that interact are the Maritime Security Center and the National Navy. The

[1]https://www.tareasos.eu/.
[2]http://www.andrew.cmu.edu/user/jklein2/primarystudies.pdf/.

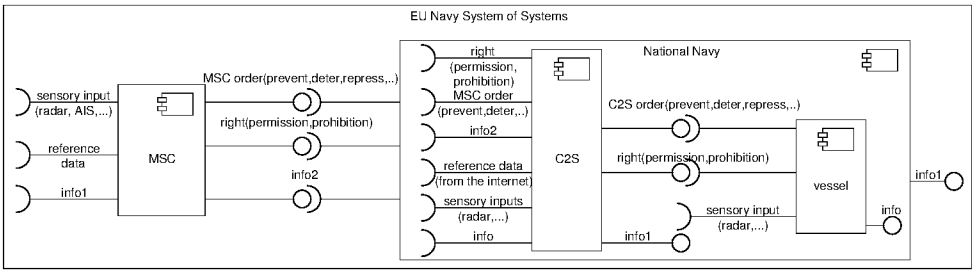

Fig. 1. UML component diagram of the case study.

first system collects information (sensory input, reference data and info1) and sends as output MSC orders, rights and other information (info2); this output is the input of the National Navy system. The latter itself has two interacting components: the Command and Control System and the vessel, each one having as input either transferred information (MSC order, info2, right) or its own collected data (info, sensory input, reference data). However we found it difficult to properly model the exchange of information and transmission of orders between the different components. For example the constituent systems of the National Navy are not accurately attached to each other: the output of the C2S component "info1" is not connected to any input of the vessel component, and "sensory input" of the vessel component is not connected to any output of the C2S component.

Each National Navy has three types of vessels: Frigate, Patrol and Surveillance And Reconnaissance (SAR) and a Command and Control System (C2S). Frigates are responsable for raiding missions and conveying messages and they may fight in small numbers or singly against other frigates. Patrols conduct protection duties and safeguard the integrity of the territorial waters. SARs help to strengthen regional security by supporting coast guard mission needs. The Maritime Security Center and the Command and Control System collects information (sensory inputs, reference data) and verify if the different ships have the rights to access information. Information is of two types: public and private. Each National Navy is a constituent system which can be managed and operated independently. The emergent behavior of this system is depicted by the LawEnforcementVessel, which is used to model the vessels which, at a certain moment, has the task of preventing/fighting against crime. This hierarchical composition is modeled using a class diagramme (Figure 4).

B. Security Design

In the previously described scenario, information access is performed in a hierarchical manner and the sequence diagram of Figure 2 models these accesses: a vessel (IT1 Patrol) requests information (getInfo(blueStarInfo)) from its Command and Control System (IT C2S), this request is transmitted to the Maritime Security Center (EU MSC), the latter will provide or not the requested information (SendInfo()) depending on the type of the requested information (typeOf(info)) and the applicant's identity (typeOf(vessel)). Hence, there is a necessity to develop rules that express these security requirements resulting from information sharing between the various constituent systems of the SoS.

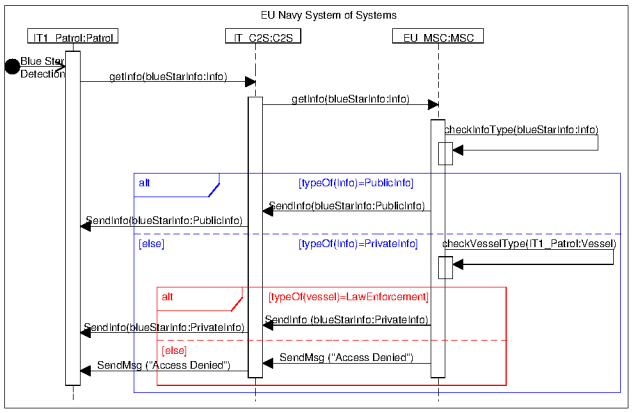

Fig. 2. UML sequence diagram of the case study.

Rule Name	Type	Organization	Role	Activity
MSC1-1	Permission	EUNavy	EUVessel	ReadPublicInfo
MSC1-2	Prohibition	EUNavy	EUVessel	ReadPrivateInfo
MSC2	Permission	EUNavy	LawEnforcementVessel	ReadPrivateInfo

Fig. 3. Security rules of the parties in the scenario.

Figure 3 summarize the rules adopted in our scenario: Operators on vessels of the EUNavy have permission to access public information but they cannot access private information. While operators on LawEnforcementVessels can access private information. Once the rules are defined, Figure 4 describes a possible way to model them in the architecture of the SoS using the UML class diagram and the Role-Based Access Control (RBAC) security pattern [8]. Indeed, the RBAC security pattern describes how to assign rights based on the object and the role. This combination of *object, role* and *right* is an instance of the authorization pattern. In our class diagram (Figure 4), the object is the information, the roles are normal Vessel or LawInforcementVessel and the rights are Permission or Prohibition.

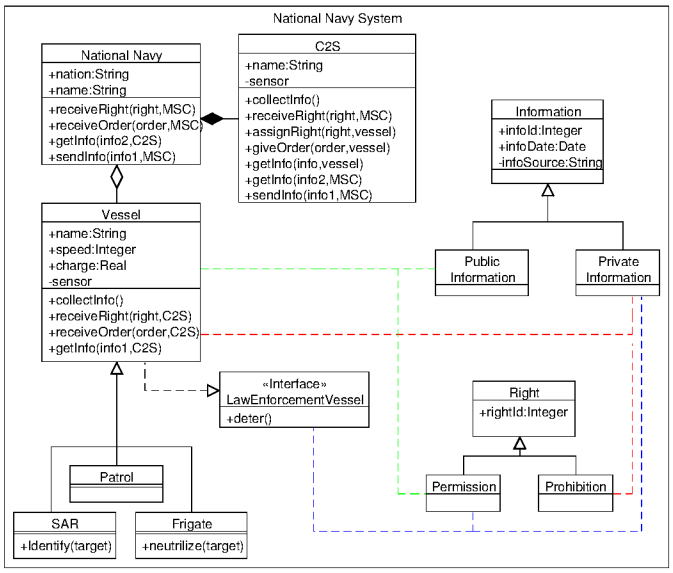

Fig. 4. UML class diagram of the case study.

C. Security Design Challenges

From the case study above we analyze the influence of the SoS characteristics on its security:

- Each National Navy and the MSC may operate under different policies, protocols, procedures but they need to cooperate to restrict the access to the private information: incompatibilities and conflicts arising from the *operational independence* of the elements.

- The Command and Control System of each National Navy gives or may give different orders to the national vessels but only the MSC has the role to assign rights (permission, prohibition) to different vessels in the SoS: Managing the specification of rights for SoS activities and ensuring the security concern resulting from the *managerial independence* of the elements.

- In case of purchase or sale of a national vessel, it will be necessary to modify or redefine the roles and rights in the scenario: Security issues should evolve with the *evolutionary development* of the SoS.

- The LawEnforcementVessel is an emergent role assigned to a national vessel in case of emergency and allows the access to the private information and thus introduces new security topics: Security issues need to evolve to accomodate the *emergent behavior* of the SoS.

- Each National Navy and the MSC belong to different countries that may have different national regulations: Difficulties in trying to secure SoS *geographically distributed* systems.

As can be seen from the description above SoS features have a serious impact on its security. Therefore, the main challenge remains in the consideration of security problems as soon as possible in the SoS life cycle. However, if the number of constituent systems and rules hugely increases or if there are conflicts between these rules, it is still possible to model them by a UML class diagram? or other type of diagrams are needed? Is it sufficient to use other existing languages (UMLSec, SysML,) and tools (RBAC,OrBAC)?

III. Proposal

The challenges identified in the previous subsection may be generalized to some challenges for the security architecture of SoS:

- How could incompatibilities and possible conflicts arising from the differences between vocabularies, protocols, policies and regulations be taken into consideration in the security architecture of SoS? How to make this resulting design dynamic and how to incrementally analyze it to discover the emergent behavior?

- How is SoS security architecture conceived? Is it by integrating the SoS architecture and the security architecture? Or by defining a standalone security architecture and linking it to the SoS architecture? Or by extending a security language with concepts specific to SoS architecture?

- How to choose the most consistent SoS architecture(s) for security modeling? How to measure and quantify different SoS security architecture alternatives? And how to analyze and predicts these alternatives?

- Accordingly, existing system modeling and analysis methods and tools will be sufficient? Or new methods and software tools are needed to address all these challenges?

In this research proposal we will introduce a new method incorporating a software architectural language to model, analyze and predict security architectural alternatives of an SoS. The method will be build upon existing approaches for system modeling, such as UML, SysML. It will combine them with analysis and measures in order to quantitatively analyze and predict the security of SoS.

To define the language, a model-driven engineering approach will be used. Indeed, the construction of the language and the associated tools necessitate the use of graphical and/or textual editors, the security analysis requires automatic code generation for one or many architectures, and security prediction needs simulations; all these features can be provided more easily by using an MDE-based approach.

For security, we focus especially on access control, users permissions and interdictions to access resources. These could be analyzed by leveraging existing methods like OrBAC or one of its extensions. Indeed, Organisation Based Access Control is an approach that describes rights (permissions, prohibitions, interdictions) using security rules written as first order logic predicates and enables conflict detection.

For model simulation, an approach based on system execution modeling can be used. In fact, system execution modeling tools can provide quantitative estimation and characteristics to enable the evaluation of systems non functional properties.

IV. Related Work

In the last years, many studies in the field of SoS have been published but few of them discuss their security at the design stage of the software engineering life cycle process. In

this section we review some approaches that aim to handle security architecture engineering of SoS.

[9] address the security coupling/integration into the SoS architecture. In [10] some security design properties like completeness, consistency, etc. are verified throught policies. [11] evoke the necessity to analyze conflicts between security and functional requirements. The possible cumulative effects of a single security incident on multiple constituent systems are examined by [12] and [13]. The considerable number of interactions between users and SoS or its constituent systems increases the number of attacks, hence security mechanisms should accordingly scale up [14]. These works are more oriented towards the SoS security in general or throughout the whole software engineering process not only the design process as we propose to focus.

In contrast, other researches differ from what we propose by the fact that they handle the architecture of SoS without detailing security concerns: [15] and [16] explicit the challenge of modeling SoS in a way to enable security design. In [17] architectural patterns are used to architect and continuously analyze SoS.

The last two papers that we analyze here target both security and design of SoS: [18] discuss a design for evolution to maintain operations regardless of the SoS state, while [19] describe the importance of designing solutions that consider the security issues without clearly detailing these solutions. In conclusion, none of the papers presented above really address the security challenges at the architectural stage of the software engineering life cycle process of an SoS. In contrast, in our proposal we discuss methods and tools to model and analyse security of an SoS taking into account its design evolution.

V. Conclusion

The proliferation of SoS in the last years make it an important research field with a wide application in many domains. The specific characteristics of SoS impose many security challenges that need to be properly addressed and described in the design of SoS. In this research proposal we modeled a motivating SoS scenario in the domain of maritime safety and security using UML component, sequence and class diagrams and we presented a possible way to add security policies to the model, based on existing security design patterns. Then we extracted specific security design challenges and generalized them into more generic security architecture challenges. We also discussed modeling languages, architectural tools to model and deal with the specific security characteristics of SoS. An important contribution of our work is to take into account security at an early stage in the life cycle of the SoS to minimize the effects/costs of the later changes. It might also be useful to treat other security aspects, in addition to access control, such as identification or authorization. The ultimate validation of the approach will be performed by applying it to a case study similar to the described motivating scenario.

References

[1] M. Jamshidi, "System of Systems - Innovations for 21st Century," in *ICIIS, IEEE Region 10 and the Third intl Conf on Industrial and Information Systems.*, 2008.

[2] M. Maier, "Architecting principles for systems of systems," *Systems Engineering*, 1998.

[3] J. Boardman and B. Sauser, "System of systems - the meaning of OF," in *IEEE/SMC International Conference on System of Systems Engineering*, 2006.

[4] G. You, X. Sun, M. Sun, J. Wang, and Y. Chen, "Bibliometric and social network analysis of the sos field," in *Proceedings of the 2014 9th International Conference on System of Systems Engineering (SOSE)*, 2014.

[5] A. Avizienis, J.-C. Laprie, Randell, B., and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," in *Dependable and Secure Computing, IEEE Trans. on, vol.1, no.1, pp.11-33*, 2004.

[6] J. Klein and H. van Vliet, "A systematic review of system of systems architecture research," in *Proceedings of the 9th International ACM Sigsoft Conference on Quality of Software Architectures*, 2013.

[7] D. Trivellato, N. Zannone, M. Glaundrup, J. Skowronek, and S. Etalle, "A semantic security framework for systems of systems," *Int. J. Cooperative Inf. Syst.*, 2013.

[8] E. Fernandez-Buglioni, *Security patterns in practice. Designing Secure Architectures Using Software Patterns*, J. W. . S. Ltd, Ed. Wiley, 2013.

[9] D. Bodeau, "System of systems security engineering," in *Computer Security Applications Conference, Proceedings., 10th Annual*, 1994.

[10] R. Delmas and T. Polacsek, "Formal methods for exchange policy specification," in *Advanced Information Systems Engineering*. Springer, 2013.

[11] P. Meland, J. Guerenabarrena, and D. Llewellyn-Jones, "The challenges of secure and trustworthy service composition in the future internet," in *6th International Conference onSystem of Systems Engineering (SoSE)*, 2011.

[12] P. Kotzanikolaou, M. Theoharidou, and D. Gritzalis, "Interdependencies between critical infrastructures: Analyzing the risk of cascading effects," in *Critical Information Infrastructure Security*. Springer, 2013.

[13] J. Dahmann, G. Rebovich, M. McEvilley, and G. Turner, "Security engineering in a system of systems environment," in *Systems Conference (SysCon), IEEE Intl*, 2013.

[14] M. Ciampi, G. Pietro, C. Esposito, M. Sicuranza, P. Mori, A. Gebrehiwot, and P. Donzelli, "On Securing Communications among Federated Health Information Systems," in *Computer Safety, Reliability, and Security*. Springer, 2012.

[15] I. Eusgeld, C. Nan, and S. Dietz, "System of systems approach for interdependent critical infrastructures," *Reliability Engineering & System Safety*, 2011.

[16] D. Farroha and B. Farroha, "Agile development for system of systems: Cyber security integration into information repositories architecture," in *IEEE Systems Conf*, 2011.

[17] R. S. Kalawsky, D. Joannou, Y. Tian, and A. Fayoumi, "Using architecture patterns to architect and analyze systems of systems," *Procedia Computer Science*, 2013.

[18] M. Duren, H. Aldridge, R. K. Abercrombie, and F. T. Sheldon, "Designing and operating through compromise: Architectural analysis of ckms for the advanced metering infrastructure," in *Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop*, 2013.

[19] M. Merabti, M. Kennedy, and W. Hurst, "Critical infrastructure protection: A 21st century challenge," in *International Conference on Communications and Information Technology (ICCIT)*, 2011.