# Towards Metamodeling approach For IoT based System of Systems

Sahar Smaali and Rachida Boucebsi
*LIRE Laboratory, Abdelhamid Mehri-Constantine 2 university, Constantine, Algeria*

**Abstract**

IoT based System of Systems is currently attracting an immense interest from service providers, businesses and academic researchers since they take advantage from both paradigms. However, the complexity level has increased, arising significant challenges in their design and development. One of the promising solutions is the Model Driven Engineering, which offers a high level of abstraction and user-friendly notations of these systems key concepts. In this paper, we propose a multi-view metamodeling approach for designing IoT based SoS systems. Our model covers various features of these systems from different perspectives, in particular the SoS organization description, the specification of its missions, a detailed view of constituent systems structure and their internal behavior as well as how they cooperate to fulfill a specific goal. A smart home use case is illustrated to demonstrate the potential of the proposed approach.

**Keywords:**

System of Systems, Internet of Things, Model Driven Engineering, Multi-view, Metamodel.

## 1. Introduction

Software systems complexity has significantly increased; especially with the recent evolution of technology and incessant user requirements, giving birth to a new type of systems such as Systems of Systems (SoS) and Internet of Things (IoT). These systems have emerged in almost all aspects of our daily life comprising transportation, healthcare, defense, agriculture, manufacturing, and more. An SoS can be defined as large-scale integrated, heterogeneous, independent and networked systems cooperating in order to accomplish a common goal that cannot be achieved by the individual systems independently [1]. Whereas, IoT provides a bridge between the physical world and the digital world. In fact, an IoT system[1] consists of a global network of smart devices that are not only engaged in data discovery, but also have computational and communication capabilities, which led to the stimulation of a large-scale deployment of connected devices and sensors.

Due to the fact that SoS and IoT share some major characteristics such as the operational, managerial, evolutionary independence of its constituent systems, their geographic distribution and their emergent behavior. Therefore, IoT may be considered as the integration of heterogeneous and distributed objects in a single SoS, efficiently managing the collection and processing of vast amount of data, generating added-value services and applications to achieve common goals [2]. As a consequence, SoS paradigm can be applied to IoT systems to bring together the strength of the two paradigms by supporting the high dynamic cooperation of the IoT systems.

Moreover, IoT may benefit from adopting a model driven engineering (MDE) approach to support IoT based SoS development in order to cope with issues regarding their design. More particularly, an IoT based SoS model should provide an explicit and a clear description of the following aspects:

*Tunisian Algerian Conference on Applied Computing (TACC 2021), December 18–20, 2021, Tabarka, Tunisia*

EMAIL : sahar.smaali@univ-constantine2.dz (S. Smaali) ; rachida.boucebsi@ univ-constantine2.dz (R.Boucebsi)

- the SoS architecture configuration (its constituent systems, their roles, hierarchy and interconnection).
- the SoS objectives and goals (i.e., global missions) as well as those of its constituent systems (i.e, their individual missions).
- the SoS subsystems while covering all architectural elements of an IoT system (different layers, smart devices, context adaptation and service specification).
- the collaboration of the SoS subsystems to reach a precise goal and accomplish a given mission.

In this context, and with the aim of tackling this challenge, we explore the potential of MDE to propose a metamodeling approach to ease the analysis and the design of this type of system. It consists of a multi-view metamodeling approach providing a common and unambiguous vocabulary that allows the specification of IoT based SoS architectures. Our proposed model is meant to be used by designers in describing IoT based SoS while taking into account the organization, mission, collaboration and IoT perspectives.

The rest of this paper is organized as follows. In section 2, some existing works related to our approach are discussed. In section 3, we outline the principle of our approach and we present its different viewpoints. Sections 4 is devoted to the validation of our approach through an illustrative case study of smart home. Finally, the last section concludes the document with some perspectives.

## 2. Related work

With the attention paid to IoT and SoS in recent years, considerable effort has been made contributing to the specification and description of their architectures. In what follows, we present some of the existing work that focus on the design of these systems.

Among the earliest works that were attracted to SOS, we cite [3] where the authors provide a framework of SoS basic concepts to support engineers in the development process. Other works such [4], [5] and [6] adopt SysML visual modeling language in order to propose a viewpoint-driven approach for modeling SoS elements and their relationships. [7] revolves around the ease of designing SOS while covering their fundamental concepts (missions and roles) through the proposition of a mission conceptual model. In the same context, the authors of [8] introduced a model-driven approach and provided a meta-model allowing SoS graphical specification. [9] focused on security problems by proposing a domain specific modeling language (DSML), inspired by SysML, to model SoS security at the architecture level of the development life cycle to reduce the cost, time and effect of later modifications.

On the other hand, authors of [10] proposed an integrated approach to ease the development of IoT systems through the application of a development methodology that separates process into different concerns and provides a conceptual framework for that purpose. [11] defines a design and analysis process and related framework called, SysML4IoT; a SysML profile based on the IoT-A Reference Model [12], to assist application engineers to precisely specify system models of IoT applications and verify their QoS properties. An extension of this paper is presented in [13] to propose a modeling language for the specification of SOA-based IoT applications named SoaML4IoT. Moreover, [14] FRASAD (FRAmework for Sensor Application Development) is a model-driven approach for improving the reusability and flexibility of sensor software systems. [15] focused on Adaptive IoT Systems and presented a model-driven approach to facilitate the modeling and realization of such systems by adopting SysML4IoT. SysML is adopted and extended also by [16], but this time in order to apply a model-based design approach in the IoT e-Health systems. Authors of this article focus on the Remote Elderly Monitoring System (REMS) use case by identifying its criticalities and modeling them as SysML requirements, while SysML constraints and parametric diagrams are employed to describe and verify quantitative criticality requirements.

Despite the large amount of work destined for both IoT and SoS systems, to our knowledge, there is no work that focuses on designing IoT-based SoS. In this article, we aim to accumulate this lack by defining a multi-view model that covers both of SoS aspects as well as those of IoT.

## 3. IoT Based SoS Metamodel

A metamodel defines necessary elements for a specific domain description and identifies semantic relations among them. It provides a vocabulary for a particular domain to assist designers to establish a common view of this domain and ease their collaboration. The main contribution of this paper consists of a generic modeling approach for IoT based SoS. We propose a multi-view metamodel to describe such systems architecture from different perspectives while taking into account their structure and behavior. Our model is structured in four view-points of IoT and SoS (see Figure 1):

- *Organization view* represents the SoS configuration. In particular, it defines the hierarchy of the SoS constituents and the manner in which they are composed.
- *Mission view* defines what the system is intended to do by specifying its functional objectives (i.e., individual and global missions). In addition, it describes how individual missions (ensured by subsystems) are structured towards a goal mission (ensured by the SoS).
- *IoT view* gives a detailed perspective of constituent systems multi-layered architecture. It describes their components (smart devices, sensors, actuators, services, etc.) as well as their interactions.
- *Collaboration view* models how constituent systems collaborate within the SoS to accomplish a given global mission.

This multi-view approach is adopted to manage the IoT based SoS complexity and to cover its different features and various aspects.

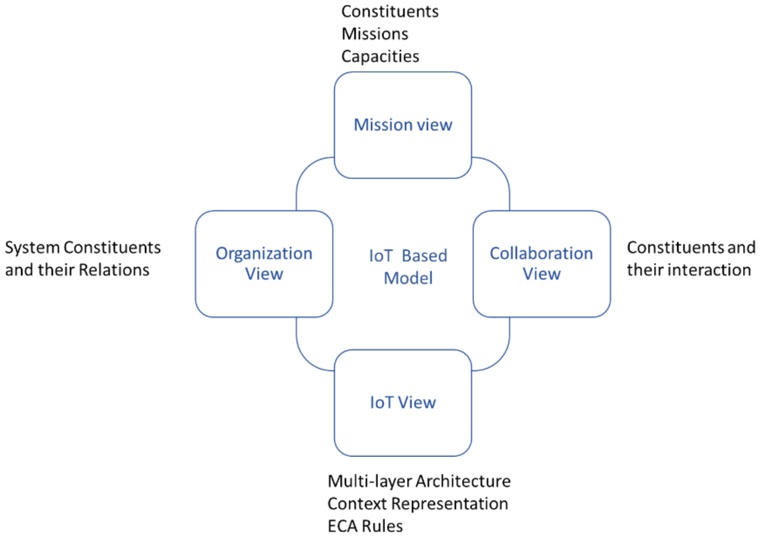

**Figure 1**: Multi-view model for IoT based SoS

### 3.1 Organization View

The metamodel in Figure 2 defines an organization diagram to describe the SoS configuration. They Constituent systems form an organization to achieve a particular objective of an IoT based SoS while remaining managerially and operationally independent. We consider an SoS as a set of organizations where each organization groups a number of systems collaborating to accomplish one or more global missions of the overall system. The diagram allows the specification of the system constituents dependency one on another. In addition, we note that system class may be instantiated to represent both IoT systems and SoSs. The sub-systems composition models the hierarchical organization of an SoS since this later could be refined into numerous constituent systems.

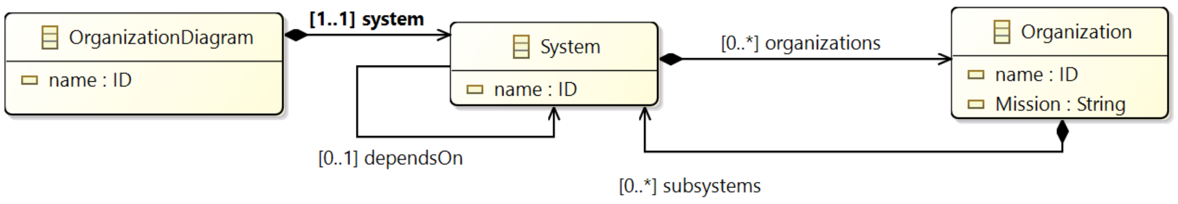

**Figure 2**: SoS Organization Metamodel

### 3.2 Mission View

The metamodel of Figure 3 gives a functional view of IoT based SoS architecture from a mission perspective. An IoT based SoS system is defined as a set of systems and relations. Each sub-system is a useful system by itself, having its own goals and resources, and collaborates within the SoS to achieve its missions. Therefore, its capabilities that are required for a given mission accomplishment should be determined and specified. A system capability is an abstraction of its provided services and the operations that can execute. In addition, every capability is related to one or more missions that participate in their realization. Moreover, the *Relation* class relates an SoS global mission to one or more of its subsystem's missions (individual missions).

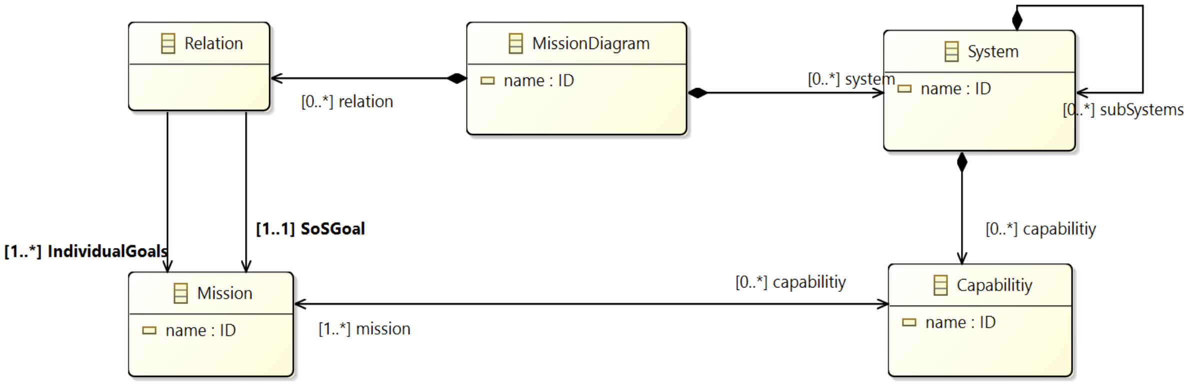

**Figure 3**: SoS Mission Metamodel

### 3.3 IoT System View

We propose a multi-layered architecture for describing IoT systems giving them a hierarchical definition, in such a way that the relationships between the different layers are clearly represented and interpreted. Thus, our model consists of five following layers (see Figures 4 and 5):

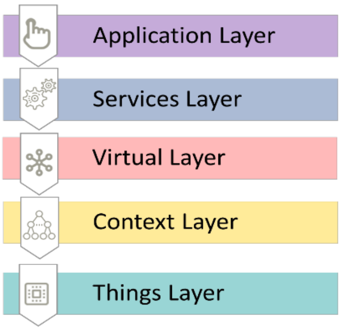

**Figure 4.** Layered IoT Architecture

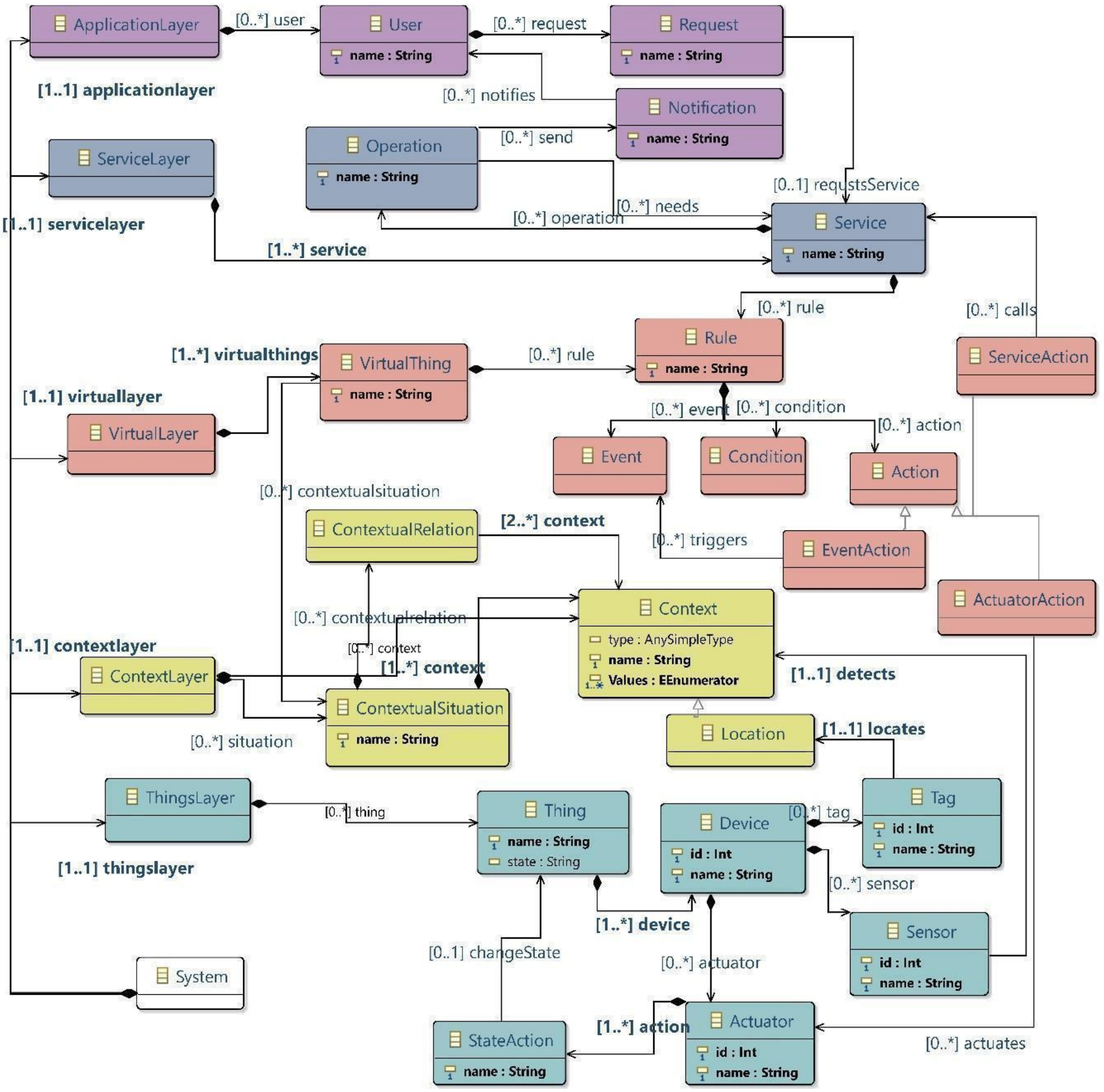

**Figure 5**: IoT System Metamodel.

- **Things Layer**: represents IoT devices intended to collect and digitizes contextual information from different sensors and their transfer to upper layers. This layer groups together a collection of smart things comprising a set of devices. These later represent the heart of a smart thing allowing the management and the control of its behavior. It contains a set of sensors to supervise the environment, tags to locate the objects and actuators to change the state of a smart object via actions.
- **Context Layer**: specifies contextual data relevant to the operation and / or adaptation of things. It makes it possible to define changes in the physical environment in which smart devices are integrated. It describes all situations that may happen and need a reaction or adaptation. A contextual situation is defined by one or more context elements (provided by sensors and tags) with their associated values and their relations.
- **Virtual Layer**: responsible for the virtualization of devices or physical entities. It allows an abstraction and isolation of the resources exposed by the physical entities to the services. In addition, it allows the efficient exchange of information between different things and services. This abstraction manages heterogeneity and interoperability and thus acts as a

bridge between the physical world and services. This layer consists of a set of virtual things abstracting the control of Things Layer devices via ECA rules (Events-Conditions-Actions)

  - Events trigger the invocation of the rule
  - Conditions express what must be necessarily verified for executing a rule.
  - Actions consist of changes in smart devices behavior (Actuator Actions) or invocation of services (Service Actions).

- **Service Layer:** This layer functions as an interface between the virtual layer and the application layer. It is responsible for creating and managing the services offered by smart things. It processes requests for these services by users or by the services themselves. Service composition is modeled by the "needs" association.
- **Application Layer:** describes the necessary APIs that allow end users (human or systems) to access the resources of the smart things through the Service layer.

## 3.4 Collaboration View

The diagram of this view offers a detailed description on how constituent systems of an IoT based SoS cooperate with each other to accomplish a particular global mission (see Figure 6). The capabilities of each subsystem are defined by the services that it provides to other systems and the actions that it can perform to manage its environment and its internal elements. In addition, an action may trigger other actions belonging to the same subsystem and/or require the execution of service operations of other subsystems.

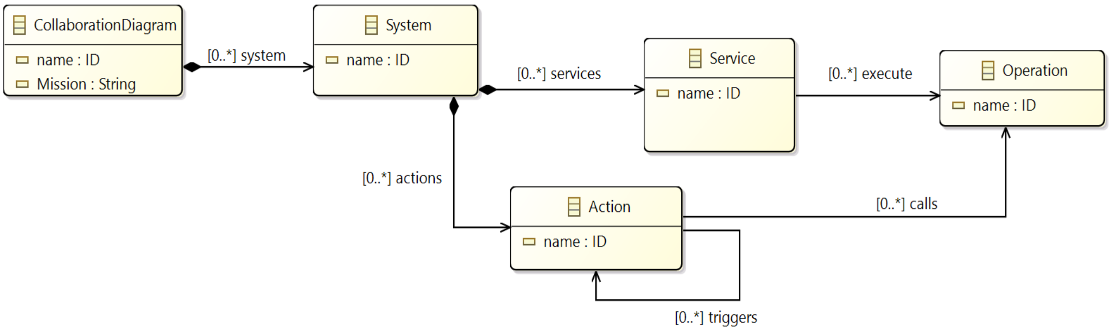

**Figure 6**: SoS Collaboration Metamodel

## 4. Case study: Smart Home SoS

To highlight the ability of our approach to model IoT based SoS, we consider a real-life example of a smart home (SH). This later aims to integrate home automation enabling users to remotely monitor and control the house objects and devices [17]. We can elaborate the SH model by instantiating all the meta-model elements.

## 4.1 SH Organization Diagram

A smart home IoT system could be analyzed as an SoS with various subsystems. Organization diagram in Figure 7 shows the constituents of this system.

- *Lighting System:* relies on lighting technologies to make decisions, follow schedules, offer additional functions such as changing the color of the light, setting reminders and which can be managed remotely.
- *HVAC System:* is a heating and air conditioning system allowing the climate regulation according to people's presence or absence or even by heating or cooling only parts of the house as desired by the owners.
- *Security System:* is a system of interworking components and devices with a central control panel that collaborate to protect a home.

- *Windows and doors Control System:* in addition to controlling the windows and doors, this system alerts the owner if he leaves his home without closing windows or locking doors.
- *Audio System:* allows the music distribution to any area of the home easily, with all the latest internet radio stations, podcasts, and audio streaming services available at the touch of a button.

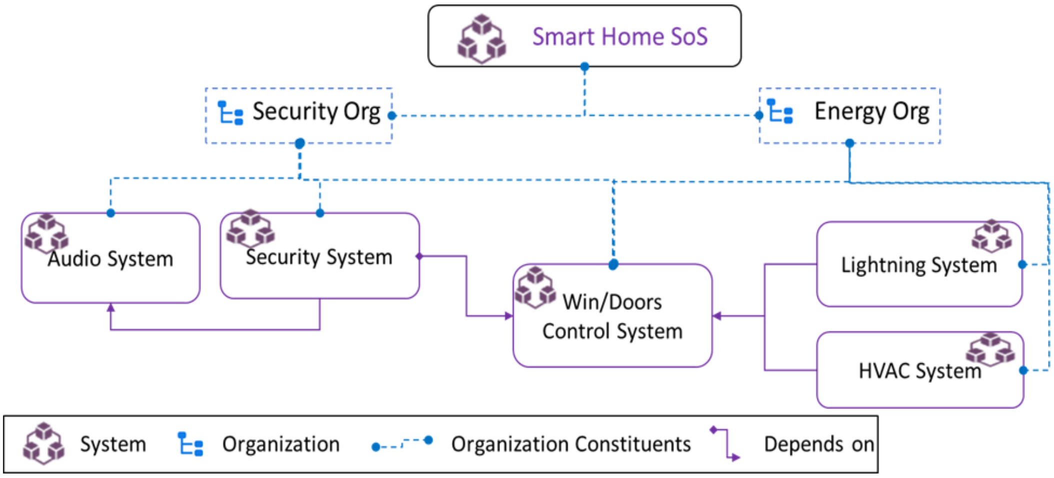

**Figure 7**: SH Organization Diagram

## 4.2 SH Mission Diagram

An SH system has to fulfill a large number of missions. Figure 8 shows a mission diagram specifying some of the SoS global missions as well as subsystems individual missions that are involved in the realization of each global mission. Therefore, each global mission is linked to a number of individual missions which in turn are related to the system capacities fulfilling them. For instance, the Reduce Energy Consumption Mission is achieved by accomplishing three missions; regulating rooms lights, temperature, windows and doors. Each one of these individual missions defines a constituent system responsibility that is covered by its own capacities. As an example, a temperature regulating mission is achieved by the HVAC system that is capable of adjusting the heater and the air conditioner. In addition, some of the individual missions may involve multiple global missions such as triggering the alarm of the Audio system.

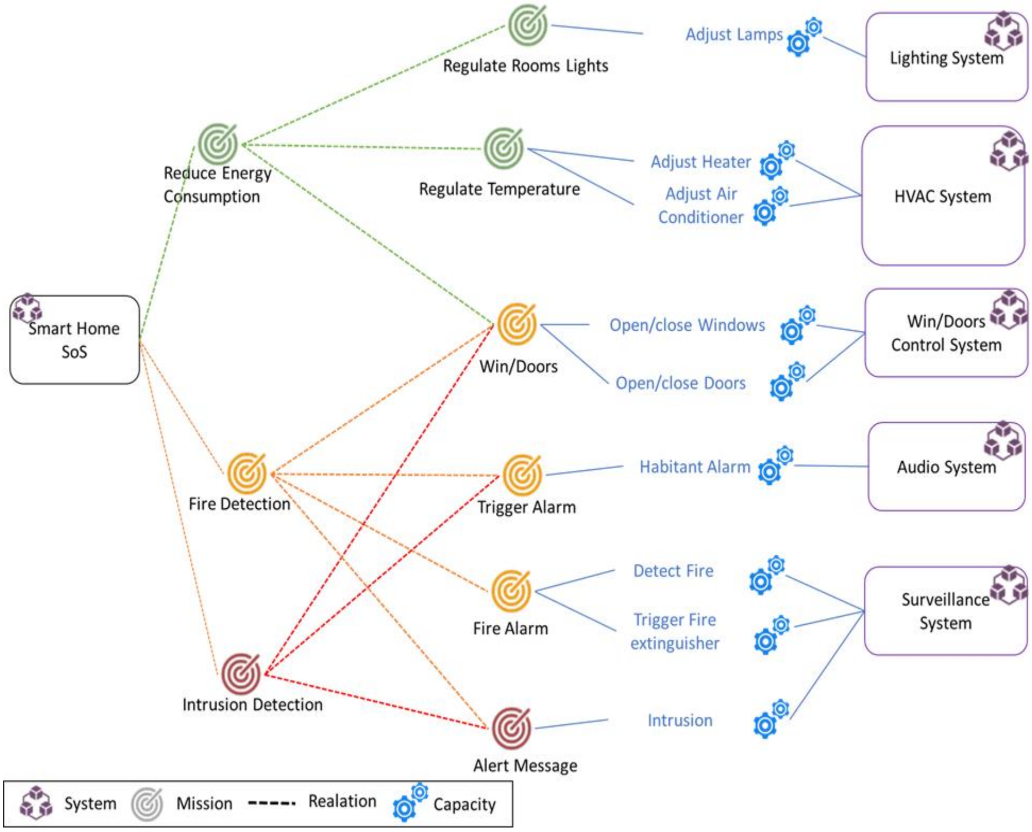

**Figure 8**: SH Mission diagram

## 4.3 SH IoT diagram

In this section, we propose an instance of the IoT metamodel in section 3.2 for HVAC system. Its architecture is shown in figure 9 respecting the correspondence rules summarized in its legend. Things Layer describes an SH room equipped with smart devices namely an inner and an outer box to monitor rooms indoor and outdoor using temperature, humidity and CO2 sensors. Each sensor is linked with room contexts it supervises in the context layer. The combination of sensed contextual data will lead to contextual situations. The IoT diagram describes two contextual situations related to the room temperature and how they are handled by the HVAC system:

- Cold room situation occurs when the indoor temperature is under 22°C and the outdoor temperature is above 20°C.
- Hot room situation occurs when the indoor temperature is above 25°C and the outdoor temperature is above 22°C.

In each situation, the HVAC system needs to regulate and adjust the room temperature to keep it at 24°C. Therefore, virtual layer things control the heater and the AC devices by applying ECA rules corresponding to a particular situation. Each of the heater and AC controls have three rules to turn on, turn off and adjust the room temperature. For instance, the turn on rule of the Heater Control virtual thing is applied whenever the cold room situation happens. The execution of this rule actions will turn on the heater with a temperature of 24°C by means of the heater device actuators. While in the hot room situation, the AC control in the virtual layer will turn on the AC and adjust its temperature.

In both situations, the turn on rules will execute a service action to close the room windows by calling the Windows room service. This latter will send a notification to the Win/Doors control system to close the room windows. Services play the role of an intermediate between the virtual layer things and other subsystems of the SH SoS specified in the application layer. Furthermore, the SH habitants may view the history data of the room by sending requests to the Energy Consumption service. This latter saves the data coming from the virtual things that control the things and context layer.

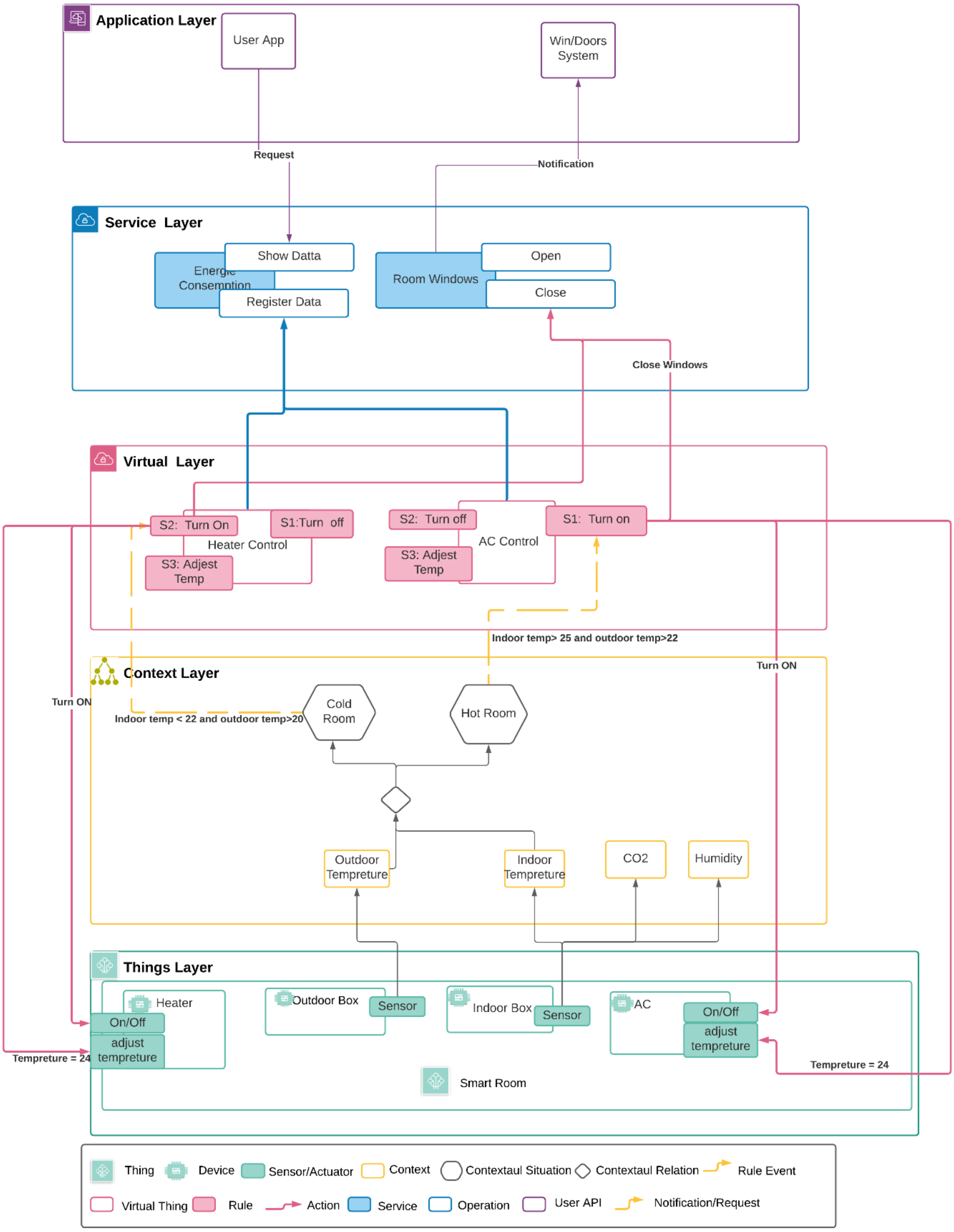

**Figure 9**: IoT diagram of the HVAC system.

## 4.4 SH Collaboration Diagram

Collaboration diagram in Figure 10 represents the interaction between smart home subsystems to realize the fire detection mission in terms of service operation calls and action execution. The surveillance system supervises the house rooms in order to detect any suspicion of fire. In such situation, the system executes the action triggering the fire extinguisher which in its turn calls the room windows service to open the windows. In parallel, the audio alarm service of the audio system is launched and the alert service of the surveillance system sends alert messages to firefighters and police.

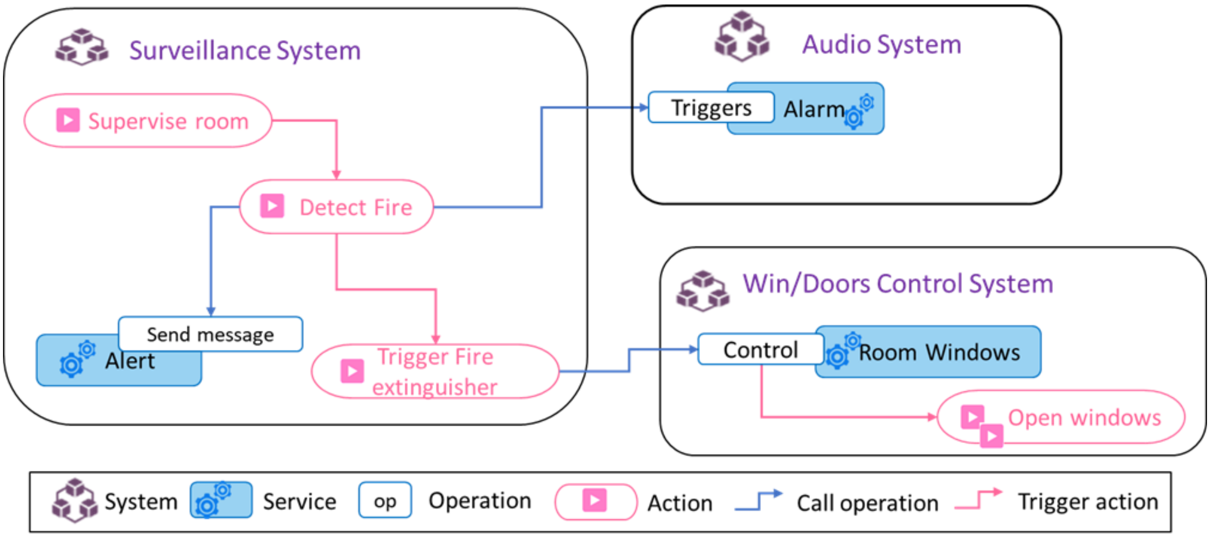

**Figure 10**: Collaboration Diagram of the Fire Detection Mission

## 5. Conclusion

In this paper, we presented a viewpoint-driven approach for designing and modeling IoT based SoS. Our proposal consists of a metamodel that takes into account key concepts and features of both paradigms IoT and SoS. It allows the description of such systems from various perspectives namely the SoS organization description, the specification of its missions, a detailed view of constituent systems structure and internal behavior as well as how they cooperate to fulfill a specific goal. Our approach is able to cover the functional, structural and behavioral aspects of an IoT based SoS. As future work, we intend to extend our model to take into account more aspects and characteristics of both SoS and IoT systems such as emergent behavior, reconfiguration, scalability and dynamic evolution.

## Acknowledgements

This work was partially supported by the LABEX-TA project MeFoGL:"Méhodes Formelles pour le Génie Logiciel".

## References

[1] M. Jamshidi, Systems of Systems engineering principles and applications. John Wiley and Sons, 2009. ISBN 9780470403501. doi: 10.1002/9780470403501. URL http://onlinelibrary.wiley.com/book/10.1002/9780470403501.

[2] Paolo Azzoni, From Internet of Things to System of Systems. 2020. URL https://www.eurotech.com/en/white-papers/from-internet-of-things-to-system-of-systems

[3] J. Dahmann, J. Lane, G. Rebovich, and K. Baldwin, “A Model of Systems Engineering in a System of Systems Context,” Proceedings of 2008 Annual Conference on Systems Engineering Research, Redondo Beach, CA, 4-5 April 2008.

[4] Mori, M., Ceccarelli, A., Lollini, P., Fromel, B., Brancati, F. and Bon- ¨ davalli, A. (2017). Systems-of-systems modelling using a comprehensive viewpoint-based SysML profile. Journal of Software: Evolution and Process, 30(3), p.e1878.

[5] Mori, M., Ceccarelli, A., Lollini, P., Bondavalli, A., & Fromel, B. (2016). A Holistic Viewpoint-Based SysML Profile to Design Systems-ofSystems. 2016 IEEE 17th International Symposium on High Assurance Systems Engineering (HASE).

[6] MORI, Marco, CECCARELLI, Andrea, LOLLINI, Paolo, *et al.* Systems-of-systems modeling using a comprehensive viewpoint-based SysML profile. *Journal of Software: Evolution and Process*, 2018, vol. 30, no 3, p. e1878.

[7] Cherfa, I., Sadou, S., Belloir, N., Fleurquin, R., & Bennouar, D. (2018). Involving the Application Domain Expert in the Construction of Systems of Systems. 2018 13th Annual Conference on System of Systems Engineering (SoSE).

[8] DRIDI, Charaf Eddine, BENZADRI, Zakaria, et BELALA, Faiza. System of Systems Engineering: Meta-Modelling Perspective. In : *2020 IEEE 15th International Conference of System of Systems Engineering (SoSE)*. IEEE, 2020. p. 000135-000144

[9] EL HACHEM, Jamal, PANG, Zi Yang, CHIPRIANOV, Vanea, *et al.* Model driven software security architecture of systems-of-systems. In : *2016 23rd Asia-Pacific Software Engineering Conference (APSEC)*. IEEE, 2016. p. 89-96.

[10] P. Patel and D. Cassou, "Enabling High-level Application Development for the Internet of Things," Journal of Systems and Software, vol. 103, pp. 62-84, Januray 2015.

[11] B. Costa, P. F. Pires, F. C. Delicato, W. Li and A. Y. Zomaya, "Design and Analysis of IoT Applications: A Model-Driven Approach," in IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, Auckland, 2016, pp. 392-399.

[12] A. Bassi, M. Bauer, M. Fiedler, T. Kramp, R. van Kranenburg, and S. Langeand S. Meissner, Enabling things to Talk. Berlin, Heidelberg: Springer, 2013.

[13]COSTA, Bruno, PIRES, Paulo F., et DELICATO, Flávia C. Modeling SOA-based IoT applications with SoaML4IoT. In : *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. IEEE, 2019. p. 496-501.

[14] X. T. Nguyen, H. T. Tran, H. Baraki and K. Geihs, "FRASAD: A framework for model-driven IoT Application Development," in IEEE 2nd World Forum on Internet of Things , Milan, 2015, pp. 387-392

[15] HUSSEIN, Mahmoud, LI, Shuai, et RADERMACHER, Ansgar. Model-driven development of adaptive IoT systems. In : *MODELS (Satellite Events)*. 2017. p. 17-23.

[16] KOTRONIS, Ch, NIKOLAIDOU, Mara, DIMITRAKOPOULOS, George, *et al.* A model-based approach for managing criticality requirements in e-health iot systems. In : *2018 13th annual conference on system of systems engineering (SoSE)*. IEEE, 2018. p. 60-67.

[17] Stolojescu-Crisan, C., Crisan, C., & Butunoi, B. P. (2021). An IoT-Based Smart Home Automation System. Sensors, 21(11), 3784.