World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:17, No:10, 2023

An Extended Domain-Specific Modeling Language for Marine Observatory Relying on Enterprise Architecture

Charbel Geryes Aoun, Loic Lagadec

Abstract—A Sensor Network (SN) is considered as an operation of two phases: (1) the observation/measuring, which means the accumulation of the gathered data at each sensor node; (2) transferring the collected data to some processing center (e.g. Fusion Servers) within the SN. Therefore, an underwater sensor network can be defined as a sensor network deployed underwater that monitors underwater activity. The deployed sensors, such as hydrophones, are responsible for registering underwater activity and transferring it to more advanced components. The process of data exchange between the aforementioned components perfectly defines the Marine Observatory (MO) concept which provides information on ocean state, phenomena and processes. The first step towards the implementation of this concept is defining the environmental constraints and the required tools and components (Marine Cables, Smart Sensors, Data Fusion Server, etc). The logical and physical components that are used in these observatories perform some critical functions such as the localization of underwater moving objects. These functions can be orchestrated with other services (e.g. military or civilian reaction). In this paper, we present an extension to our MO meta-model that is used to generate a design tool (ArchiMO). We propose constraints to be taken into consideration at design time. We illustrate our proposal with an example from the MO domain. Additionally, we generate the corresponding simulation code using our self-developed domain-specific model compiler. On the one hand, this illustrates our approach in relying on Enterprise Architecture (EA) framework that respects: multiple-views, perspectives of stakeholders, and domain specificity. On the other hand, it helps reducing both complexity and time spent in design activity, while preventing from design modeling errors during porting this activity in the MO domain. As conclusion, this work aims to demonstrate that we can improve the design activity of complex system based on the use of MDE technologies and a domain-specific modeling language with the associated tooling. The major improvement is to provide an early validation step via models and simulation approach to consolidate the system design.

Keywords—Smart sensors, data fusion, distributed fusion architecture, sensor networks, domain specific modeling language, enterprise architecture, underwater moving object, localization, marine observatory, NS-3, IMS.

I. Introduction

APPLICATIONS dedicated to monitoring different environments depend extensively on the deployment of a network of sensors to collect scientific data. A sensor network is a group of specialized sensors with a communication infrastructure designed to monitor and record terms and scientific data at various locations. Gathered data are relayed to dedicated workstations in real time for analysis to become useful information. Marine Observatory (MO) applications are not different in concept, though they differ in the technology used [1]. They provide new opportunities to sea surveys such as a continuous observation of the sea [1]. The MO uses Underwater Acoustic Sensor Networks (UASNs) to operate underwater. These sensors collect data collaboratively and send it to the dedicated workstations. To achieve their objective, sensors make full use of the Sensor Networks' (SN) main advantages. These advantages include, but are not limited to, the autonomous nature of the SN and the operation of SN as a distributed information system. Autonomy is based on the ability of the SN to adapt to the characteristics of the ocean environment. Whereas the operation of the SN as a distributed information system allows it to process queries regarding any service requirements, to be able to satisfy all of these aforementioned features, the sensor network has to be complex [2], [3].

Although it can be applied to any MO project, we chose the Marine e-Data Observatory Network (MeDON) as our research scope [4]. As any MO project, the MeDON project allows the continuous observation of the sea. MeDON uses different communication protocols (REST, SOAP and proprietary ones) to connect its different components (Hydrophones, Fusion Servers, Object Localization Algorithms) together [4], [3]. It seems obvious by now that the implementation of such distributed system is considered as complex [2], [3]. This complexity originates from different sources. However, in this paper we focus on two of them: (1) the architecture of a distributed system by itself is a complex system since it comprises many heterogeneous components (underwater sensor network and the rest of the information system); (2) deployment of such a system requires high level value accuracy of several properties of these components.

Our scope in MeDON project is in the design of the Smart Sensor Network and the system to localize the underwater objects. Designing an information system such as the one of MeDON may generate errors while performing the design activity. These errors which will affect negatively the deployment activity by resulting in inappropriate and inapplicable designs of Sensor Networks. According to [4]

M. Aoun is a Research Associate Professor at the Department of Electrical and Computer Engineering, ICAM (Institut Catholique d'Arts et Metiers) School of Engineering, Toulouse Campus, France. He is also in association with Lab-STICC, CNRS UMR 6285, ENSTA (Ecole Nationale Superieure de Techniques Avancees), Brest, Bretagne, 29806, France (e-mail: charbel.aoun@icam.fr).

M. Lagadec is a Full Professor and Research Head of the IT Department at the Lab-STICC, CNRS UMR 6285, ENSTA, Brest, Bretagne, 29806, France (e-mail: loic.lagadec@ensta-bretagne.fr).

and [3], the complexity of the design is a result of: (1) the different domains of experience (business process modeling, Information system modeling, and the underlying infrastructure modeling) that are required from the designer(s) to be able to model and describe such systems; (2) distributed software structure of MeDON Information System (Fig. 1) since each component (e.g. Data Fusion Server, Smart Sensor, etc) is responsible to perform set of specific tasks.

Our global objective is to help the MO designers to reduce the complexity of the design activity. The deployment of set of sensors (Sensor Network) is a costly operation due to: the necessary equipment such as specific boats, marine cables, sensors (hydrophones), Data Fusion Servers, and experts in diving, etc. Additionally, we cannot ignore that the deployment operation is risky and the placement of sensors is also crucial for Underwater Sensor Networks (USNs) [2], [5]. More specifically, the hydrophone can be placed horizontally or vertically [5]. For example, to cover a specific underwater area, we install the hydrophone with a vertical tilt angle of 15 degrees to 25 degrees instead of installing it horizontally with a specific angle. Thus, this may result in covering the most important area of the targeted one. This is due to the number and quality of signals detected in this vertical position. Thus, a complexity factor must be taken into account before setting up the networking platform which incorporates physical and logical aspects. For this purpose, [2] considers this complexity factor as an essential requirement of USNs.

In relation to our scope in MeDON (localization of underwater moving objects), the misplacement of the sensors results in complexity and errors while performing the deployment phase. In addition, it influences negatively the performances of sensor algorithm by reducing its optimization (e.g. provide a wrong location of a detected object) [2], [5]. This may occur during the deployment and maintenance phase. Thus, an integration between the information system (Sensors, Servers) and the communication system (e.g. IMS) [6] is needed. For this purpose, our research question is: how to improve the design phase and reduce the complexity of the deployment and maintenance phase? With as a main objective to provide a design tool to the designers of MO that helps them to model their designs taking into consideration enhancing and facilitating the use of design activity, and managing its complexity.

In this paper, we propose an extension to ArchiMO. ArchiMO is a modeling design tool we developed based on ArchiMate that reduces the complexity of MO designs by implementing specific MO concepts and relationships. This tool provides the designer a set of reusable graphical elements and concepts that respect ArchiMate [7] and the MO concepts. This extension brings a high abstract level constraint on the properties of Smart Sensors.

Our approach is based on the concept of domain specific modeling languages (DSMLs), which relies on Model Driven Engineers (MDE) fundamentals [8]. In order to model MO systems, we choose ArchiMate modeling language as it relies on Enterprise Architecture (EA) framework [9], [10] that allows describing a wide range of domains [11]. We use meta-models to generate the tools that belong to different

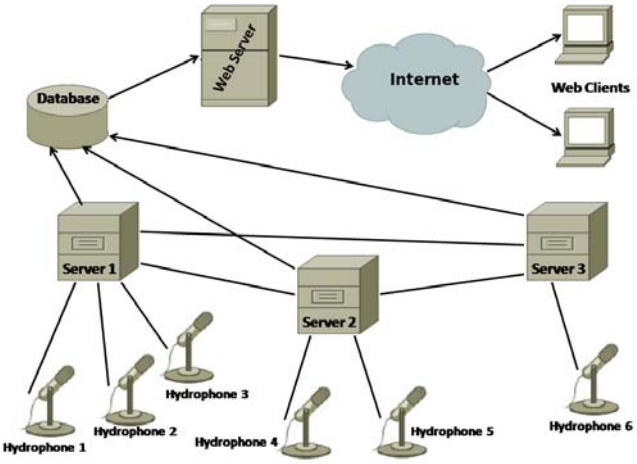

Fig. 1 Structure of MeDON - An example: N = 6, Y = 3

development activities using Eclipse Modeling Framework (EMF) [12].

ArchiMate is proper to model systems from the IT domain [7]. Our proposal extends the ArchiMate meta-model (abstract and concrete syntax) to add new abstract constraint of MO to ArchiMO. We add specific constraint to the grammar of the design tool according to the extended MO meta-model. On one hand, a main feature of (EA) frameworks is sharing the multiple viewpoints [11]. This reduces complexity of one view to a manageable size. EA frameworks introduces interoperability issues between views and their dedicated software [11]. On the other hand, our proposed constraint is extensible, where the developers may extend it and add new sub constraints, concepts and standards according to the progress and needs in MO domain.

Linking our extended MO meta-model to the IP Multimedia Subsystem (IMS) one (proposed previously in [13]) helps to integrate the different smart sensors of the sensor network to the rest of the information system through the core network [6]. We apply our design model to a model compiler to generate simulation code that runs directly in NS-3 network simulator [14].

The paper content is organized as follow, in Section II, we present the related work that is connected to the added constraint on the design tools. In Section III presents MO project. In Section IV, we present MDE fundamentals, DSMLs, ArchiMate, and our proposal constraint for the MO/MeDON. Section V explains the abstract syntax, concrete syntax and semantics of the proposed constraint. In Section VI, we present the added Smart Sensor constraint along with how it is generated with MO design tool, as well as and the simulation approach. In Section VII, we conclude and discuss our future work.

II. Related Work

In this section, we present the related work in relation to the possibility of having high level abstract constraints for a smart sensor component that is added to a design tool.

In relation with the concept of Architectural Description Languages (ADLs) [15], [16], [17], [18] and their design

tools; we are interested in the following concerns that we shall specify and analyze in this section: (C1) error prevention at the design level by invoking language structure or syntax of languages; (C2) multiple viewpoints [1] that are represented in the architectural description [19]; (C3) extensibility of design tool; (C4) diversity of components; (C5) testing/execution platform.

According to the preventing errors concern, the extended design tool early prevents errors that may be made by the designer during design activity, rather than correcting them afterward. This error prevention approach is available in ([20][21][22]). Like in our approach, it's avoided by invoking the abstract syntax (our proposed constraints) where we have defined and added our specific concepts, constraints and relations.

Concerning the multiple viewpoints concern, the extended design tool provides different viewpoints for the designers according to their specialties and domains of experience. In [20], [21], and [22], the design tool provides only one viewpoint in order to fit software development tasks. This design tool does not provide the ability to share the design between different designers. This is due to the non-existent architectural framework which generates a design tool to be fully compliant with the above concern [23]. Our approach considers this issue thanks to the different layers of EA standard that separates between perspectives.

Regarding the extensibility concern, the extension of a meta-model allows the extension of a design tool by adding new concepts and constraints to it [20], [21]. It is realized in our approach by extending the ArchiMate meta-model by new constraints, then generating a new design tool that contains our newly created constraints in the toolbox like in [6] and [11].

Concerning the heterogeneity concern, the existence of different components and communications is related to different contexts and activities. We are facing this heterogeneity in the software components and models in [20], [21] and [22]. In our approach, the diversity of components appears in our MO model which contains a high number of Smart Sensors connected to a large number of Data Fusion servers.

According to the execution test platform concern, we can find an integration between two different platforms to provide an automatic execution test of a given complex model like in [24]. Also, we can find some platforms where the designer is not able to test and verify his models or instances on an executable platform like in [22], [20], [21]. However, based on [6], our approach provide us the ability to validate the created models on a executable platform which is implemented in the same framework where the creation of models occur (see

[1]viewpoint: is a work product establishing the conventions for the construction, interpretation and use of architecture views to frame specific system concerns

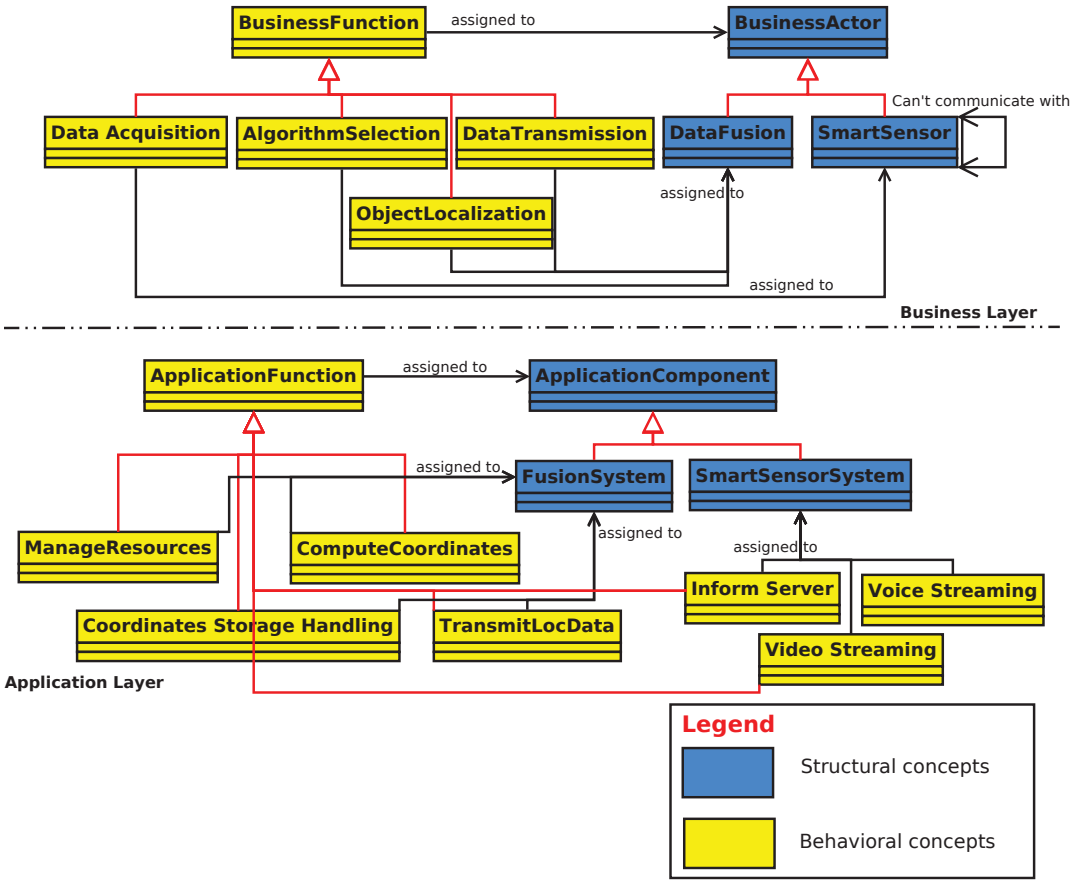

Fig. 2 Extending business and application layers of ArchiMate: proposal of MO Meta-Model

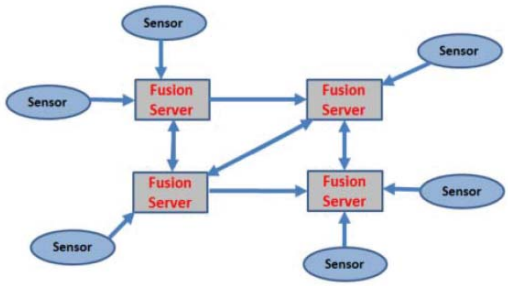

Fig. 3 Distributed Fusion Architecture

Section VI). For example, Smart Sensors and Fusion Servers can exchange messages using (IMS).

III. Marine Observatories

Underwater Sensor networks that aim to environmental data acquisition will play an essential role in the development of future large data acquisition systems [25], [26], [4]. They allow the data to be exchanged and treated between the different devices (servers, sensors). On all these devices, we can have software components to treat and store the data. An example about MO is the project Marine e-Data Observatory Network (MeDON) (Fig. 1). In this context, the designer should be able to include N acoustic sensors that are connected to the Y fusion servers as shown in Fig. 1. These servers treat the acoustic data acquired by the hydrophones then diffuse them on the network. Servers store their data on the same database. The database server provides the treated data to the web server where the configuration of a web application is done. Thus, the web server diffuses the information detected by the hydrophones such the voice of the dolphin to the web clients through a graphical interface.

IV. Model Driven Engineering and Domain Specific Modeling Languages

MDE [19] is "a software development method which focuses on creating and exploiting domain models. It allows the exploitation of models to simulate, estimate, understand, communicate, and produce code". MDE helps to manage complexity thanks to the modeling concept and model transformations. Modeling helps to describe the design in a high abstract way and model transformation helps to have a generated design tool.

In our approach, modeling tools follow the constraints and represent the concepts that are defined in the meta-model[2]. It permits to instantiate large number of models that conform to it like in programming languages [27]; numerous of programs can be implemented relying on a specific programming language (e.g. C, C++, Java etc).

Eclipse IDE provides a powerful environment that relies on EMF which facilitates the modeling/meta-modeling activities, it supports many model transformation languages as well.

Model transformations help us to generate design tools and simulation programs directly and automatically considering meta-models and model instances. Every model transformation depends on a set of rules that describe and control the transformation process. The transformation rules may map models that conform to different meta-models (on the same abstraction level) such as ATL [28], or map between different domains using one meta-model for the source model to generate texts/codes (e.g. XPAND [29]).

In our case (Fig. 2), the input model represents the design of highly abstract level, and the meta-model is the extended ArchiMate meta-model which represents the abstract syntax ([19], [13]). Our code generation is an automated process that links directly the design model to the simulation scripts [14]. Thus, it helps to reduce the time of the implementations for large simulation programs, and it minimizes the implementation errors.

A. Domain-Specific Modeling Languages

Domain-Specific Modeling Languages (DSMLs) [30] enable designers from different domains and backgrounds to participate in software development tasks and to specify their own needs using domain concepts. A DSML [31] is comprised of three components: abstract syntax, concrete syntax, and semantics. The abstract syntax defines modeling concepts and their relationships. There are several kinds of concrete syntaxes: visual, XML-based, textual, etc [32]. The concrete syntax is associated with a set of rules which defines the representation of the abstract syntax. Semantics describe the meaning of a model and are related to the abstract syntax. They are well-formed rules for the model and are used to constrain the concrete syntax [31].

Historically, data fusion methods were developed primarily for military applications (e.g. radars tracking a variable object) since fused data from multiple sensors provide several advantages over data from a single sensor [33]. In addition, to cover a specific targeted underwater area, [5] strongly recommends to install at least two hydrophones to acquire the maximum possible number of signals. These signals contain data that are useful and valuable to be treated, analyzed then combined to provide high level services such as the localization of underwater moving objects.

We resume such methodology as combining sets of observations would result in an improved estimate of the target position. Concepts such as information fusion and sensors networks have perforated the research and specially the military research. We distinguish different architecture for data fusion as follows [33], [34]: (1) centralized fusion; (2) hierarchical fusion without feedback; (3) hierarchical fusion with feedback; (4) distributed fusion. According to our context, we have selected the most complex architecture (distributed) to model it (Fig. 3), then simulate it in Section VI. During the design activity, set of constraints and restrictions should be respected by the designer in order to model such architecture [33]. We will present them in the contribution section.

In general, errors caught during the design cycle are much less time consuming to identify and correct than those found during testing. In order to avoid errors in the design activity, we have implemented constraints that are defined in the

[2]The meta-model defines by itself a language for describing a Specific Domain of interest [8]

Fig. 4 Business and Application Layers (Palettes)

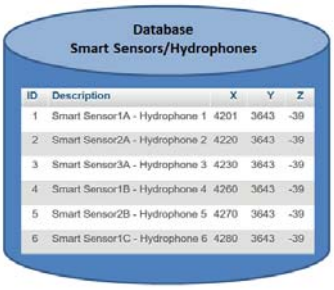

Fig. 5 The Database of the appropriate 3D coordinates of each Smart Sensor/Hydrophone for MeDON

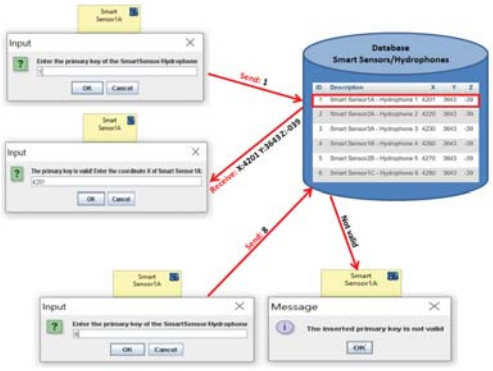

Fig. 6 Connection between ArchiMO and the Database

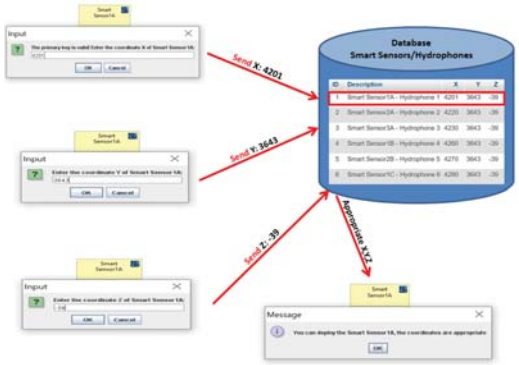

Fig. 7 Verification of the 3D coordinates: X,Y,Z

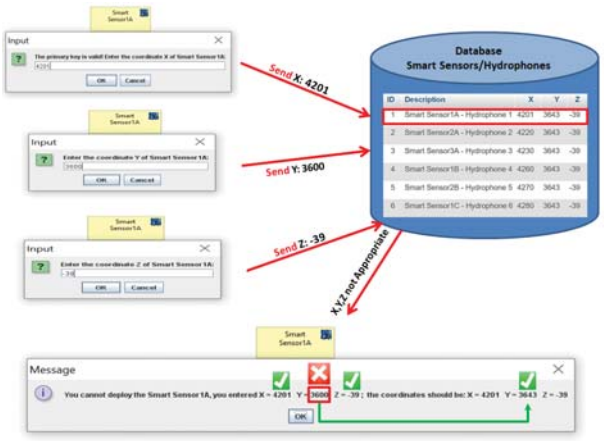

Fig. 8 Y value as inappropriate

abstract syntax of the language (meta-model) (Fig. 2). The concrete syntax that is associated with these added constraints can be implemented in the design tool such as 'ArchiMO' tool in our context. This tool is generated relying on Eclipse-EMF (tool generation concept thanks to model transformations).

Modeling languages are used to describe a system with high level of abstraction (e.g. UML 2.0) [32]. For MeDON/MO, and in relation with our objectives, we describe distributed systems. UML is not enough to cover our needs, as it has only one layer that contains all of the concepts of the design, and these concepts are too general [35]. Thus, we selected ArchiMate modeling language that meets UML in some concepts, but that can describe the systems from IT domain and share multiple viewpoints during the design as it relies on TOGAF framework [19]. Additionally, as of January 2018, the latest version of the NATO Architecture Framework (NAF v4) can be created using The Open Group's ArchiMate meta-model [23]. NAFv4 is a standard for developing architectures.

ArchiMate relies on Enterprise Architecture (EA) framework ([19], [10]). It decomposes the system design into three layers: business, application, and technology. In our approach, we present these layers in the following way:

1) Business layer: specifies the end-user functions and actors. It describes the service activities as perceived by the end-user, and the flow between them;
2) Application layer: specifies the functions and software components of the service. It describes the capability of the system under study, and the way of performing its tasks;
3) Technology layer: specifies the functions, topology, hardware elements, and signaling protocols of the underlying platform. It describes the execution platform that offers functions to be used by the functions of the application layer.

V. Contribution

In general, a meta-model of DSL represents the concepts/ operations and constraints that belong to the domain specificity (MO in our case). A previous work has extended the ArchiMate meta-model in order to take into consideration the domain specificity of MO. As a result, the ArchiMate meta-model includes concepts, elements, relations and

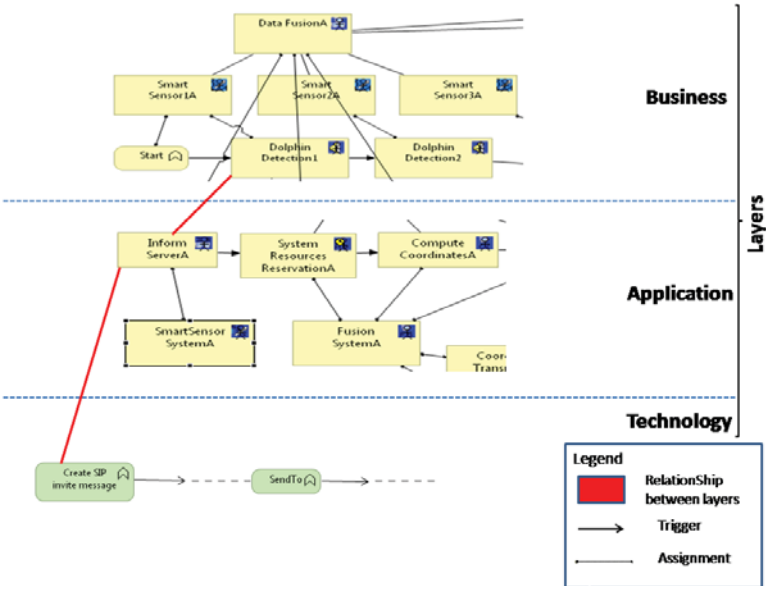

Fig. 9 Object Localization Underwater

constraints related to the MO domain [36]. This section presents our contribution to further extend this meta-model, through its concrete syntax and design tools.

Relying on the distributed fusion architecture (DFA) in [33], our proposed meta-model in [36] (Fig. 2), and according to our context MeDON [4], our MO extended meta-model and ArchiMO generated design tool already accommodate the following components: Smart Sensor(SS), Data Fusion(DF) and several other components (Fig. 2). During the design activity, the SS adheres to the following constraints: (1) two SSs cannot be connected together, a SS can be connected only to a DF [37];(2) a SS can only be connected to a DF component using our already specific extended relationship in [38]; (3) once a relationship is created between SS and DF, a mapping approach between the different layers of EA to throw directly the needed components and relations to each domain expert [39]; (4) a frequency range constraint is required to be inserted by the designer while creating a SS [40].

In standard object localization 2D projects, coordinates are expressed in two dimensions, along two axes: X (horizontal) and Y (vertical). Object Localization 3D projects (e.g. MeDON) have additional information (depth), which is measured along the Z axis (front-to-back).

According to MeDON and the installation instructions of hydrophones [5], the MO environmental experts with seabed conditions define the required and appropriate 3D coordinates of each SS/FS. Relying on these experts, we assume that the required and appropriate 3D coordinates of each SS are stored in a specific database (Fig. 5). These defined coordinates are invoked respectively by the MO designers upon each creation of a new SS in the MO model.

In this paper, we extend the SS component by adding the 3D coordinates constraints as properties. Therefore, by extending the abstract and concrete syntax and semantic of ArchiMO, we add to the SS the following constraints: once the designer click on SS icon in the ArchiMO palette, our extended ArchiMO design tool invokes the database which will return the appropriate 3D coordinates and retrieves the corresponding 3D coordinates of the targeted SS (e.g. Fig. 6). At this stage, ArchiMO continues asking the designer to enter the 3D coordinates of the targeted SS in order compare and verify the entered values (X,Y,Z)(e.g. X:4201 Y:3643 Z:-39) with the retrieved values from the database (e.g. Fig. 7). So, the MO designers should enter the appropriate 3D coordinates of each SS that should be deployed.

In case, the MO designer enters one/more inappropriate 3D coordinates, our extended ArchiMO notifies the MO designers by displaying the inappropriate inserted 3D coordinates through a graphical user interface (e.g. Fig. 8).

In order to detect a specific underwater moving object such as the dolphin in our case: it can be detected by deploying SSs (e.g. Hydrophones) that are defined and configured to receive the appropriate 3D coordinates. Otherwise, while implementing a MO project, the possibility of deploying the SSs in wrong places is high. This possibility may result in detecting the wrong targeted object and could lead to major losses in both time and funds. Accordingly, once the SS is created by the designer during the design phase of a MO project, the possibility to deploy the Smart Sensors in wrong places is minimized. Therefore, we ensure that the location of the underwater deployed Smart Sensors is appropriate to detect the required under water moving object.

In order to have a graphical view for the added constraints, we have generated the design tool ArchiMO relying on eclipse EMF. This design tool helps the designer to model the system in a highly abstract way; the elements, relations and constraints are dragged and dropped from the palette. During the model edition, all the constraints specified for the MO extension are checked. These constraints forbid the designer to enter the inappropriate 3D coordinates of SS/DF/Fixed Node.

The extended ArchiMO tool considers different domains of experience, each domain expert works in a specific

layer (Business, application or technology) as the model created in Section VI. We implement our proposed constraint in our already extended MO meta-model (see Fig. 2), and our already generated ArchiMO design tool (see Fig. 6). This implementation is the grammar of the proposed constraint. Our contribution tackles the issues presented in II by: (C1) enhancing the design process by minimizing syntax and relation errors; (C2) providing three layers according to each domain specificity by relying on Enterprise Architecture; (C3) extending MO meta-model to take into account additional constraints by generating ArchiMO that includes these constraints; (C4) deploying several physical components (sensors and servers), and logical components such acquisition/localization algorithms by creating an MO model that contains this variety of components.

VI. Object Localization Case Study

In order to validate our proposed tool, we use it to model the application of Object Localization using the different elements that are proposed in the meta-model (Fig. 2). Then we apply the design model to a model compiler (Fig. 10) that we have developed to perform some error checks and generate automatically simulation code for NS-3. This simulation code runs in NS-3 tool that is as a standard and classical simulator in the networking domain.

A. Design Model

We have modeled a system that localizes an underwater object using our generated design tool ArchiMO. In order to localize this object, sensors should be connected to data fusion servers. We have applied the distributed fusion architecture (DFA) [33] for this design.

The design model is composed of three views regarding to the layers of ArchiMate (Fig. 9): Business, Application, and Technology. In Fig. 9, we present parts of the large model that is designed by ArchiMO. The model contains behavioral elements, in the business layer (Fig. 9). It shows the first activity of the smart sensor which is the dolphin detection1, etc. These activities are assigned to their proper smart sensors and these smart sensors are associated with the different data fusion servers and smart sensors that are required in the DFA [33]. Concerning the application layer, the behavioral elements are the compute coordinates function that is triggered by the resources reservation function, and so on. ArchiMate allows the association between layers, as shown in Fig. 9. For example the InformA Application Function aims to inform the fusion server A of the detection of an object through the smart sensor1A. Regarding the technology layer, a large series of functions are associated in it (e.g. *sendto*) to execute this application function. The *sendto* function forwards/sends a message of type SIP or Diameter from one node to another.

B. Compilation and Simulation

The design tool ArchiMO generates an XMI file to represent the graphical design. This helps to conduct the design model to other tools. We use the XMI file as an input to our self-developed domain-specific model compiler to generate the simulation code (Fig. 10). This hides complexity of constructing simulation programs from the designer and saves considerable time of the development process. The code generator needs both the meta-model including the abstract syntax of DSML for MO, and the input model that is generated from the design tool.

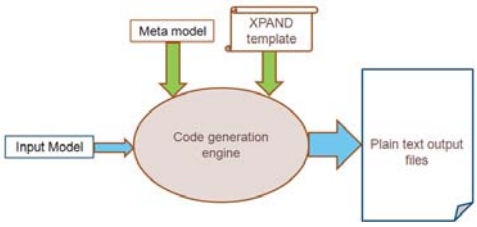

Fig. 10 The code generator workflow in XPAND language

The XPAND template in Fig. 10 contains the mapping rules between the model elements and their representations in NS-3 [14].

We have run the generated code in NS-3 (version 3.13), and the results of compilation and running show that code is error prone. Traces and logs (e.g. PCAP files) were generated to analyze the simulation outputs.

Fig. 11 shows the architecture of the system design that is generated by NS-3 for the mentioned design model. NS-3 generated a hardware representation (nodes, interfaces, wires) for the elements of the design model. The blue colored stream represents a message that is exchanged between two nodes at a given point in time. This confirms that the behavioral elements were mapped as expected.

We have used our approach in different application domains and network simulators (Video Conferencing system [14], [13], and MO context). The common design concept between all these use cases is the underlying platform (IMS) that represents the Platform Specific Model (PSM) [32]. In other words, considering using one tool (e.g. NS-3), we could change the application domain relying on ArchiMate and our extensions (DSMLs) by fixing the underlying platform that is represented in the technology layer. This confirms that our extended design tool (ArchiMO) creates models that follow the same meta-model and domain-specific concepts/constraints.

Our testing approach demonstrates to tackle the issue presented in Section II by: (C5) provide the designer, the ability to test and validate his MO created model on an executable platform which is included in the same framework where the designer creates this model as well. This is provided by generating the simulation code of this model then executing it using NS-3.

VII. Conclusion and Future Work

In this paper, we have presented high abstract level constraints. These constraints are extensions of Domain Specific Modeling Language (DSML) for MO context. We illustrated the proposed MO constraints and design tool, using a marine observatory case study. We presented a defined model for marine observatory showing their different views: business,

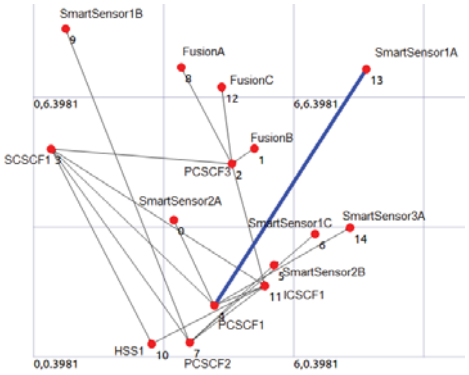

Fig. 11 Snapshot from the animation through NetAnim tool after running NS-3 simulation

application, and technology. These models are designed using our extended version of ArchiMO design tool that contains the new proposed MO constraints relying on MDE fundamentals. Then, the resulting consistent model is simulated using the NS-3 network simulator in order to validate the system model.

Our extended ArchiMO protects the designers from making design errors earlier than the other design activities and the code generation step. We rely on a standard and open tool (Archi) that we extend through developing the modeling language and Java implementations. Another advantage is the extensibility of our proposed meta-model/tool. The developers may extend it and add new concepts and standards according to the progress in MO domain. ArchiMO provides the re-usability of the added MO concepts (e.g. Smart Sensor, Data Fusion, etc) in different applications, activities, models or instances. ArchiMO reduces the time of the design activity as well, by having the specific elements/concepts and constraints in the palette of this tool. Additionally, we conserve the standard constraints in the abstract syntax (meta-model) of ArchiMate since the new added elements inherits concepts from standard ArchiMate elements.

On the other side, representing and meta-modeling the domain knowledge is itself a hard job that needs experience and high level of accuracy, especially when setting the grammar of the DSML according to the meta-model constraints.

As perspectives, we will extend our meta-model in order to satisfy and cover the most possible required operations, concepts and activities in the context of MO.

REFERENCES

[1] O. Zein, J. Champeau, D. Kerjean, and Y. Auffret, "Smart sensor metamodel for deep sea observatory," in OCEANS 2009 - EUROPE, May 2009, pp. 1–6.
[2] S. Fattah, A. Gani, I. Ahmedy, M. Y. I. Idris, and I. A. Targio Hashem, "A survey on underwater wireless sensor networks: Requirements, taxonomy, recent advances, and open research challenges," Sensors, vol. 20, no. 18, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/18/5393
[3] J.-P. Schneider, J. Champeau, and D. Kerjean, "Domain-specific modelling applied to inteegration of smart sensors into an information system," in International Conference on Enterprise Information Systems (ICEIS 2011), Lille, France, Jun. 2011.
[4] MeDON - Acoustic Data. URL: https://keep.eu/projects/7945/Marine-e-Data-Observatory-Ne-EN/.
[5] Marport. https://www.marport.com/.
[6] V. Chiprianov, I. Alloush, Y. Kermarrec, and S. Rouvrais, "Telecommunications service creation: Towards extensions for enterprise architecture modeling languages," in 6th Intl. Conf. on Software and Data Technologies (ICSOFT), vol. 1, Seville, Spain, 2011, pp. 23–29.
[7] The Open Group, ArchiMate 1.0 Specification. http://www.opengroup.org/subjectareas/enterprise/archimate.
[8] J.-L. Pérez-Medina, S. Dupuy-Chessa, and A. Front, "A survey of model driven engineering tools for user interface design," in Proceedings of the 6th International Conference on Task Models and Diagrams for User Interface Design, ser. TAMODIA'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 84–97.
[9] O. Noran, "An analysis of the zachman framework for enterprise architecture from the {GERAM} perspective," Annual Reviews in Control, vol. 27, no. 2, 2003, pp. 163 – 183.
[10] D. Quartel, W. Engelsmanb, H. Jonkersb, and M. van Sinderenc, "A goal-oriented requirements modelling language for enterprise architecture," in Enterprise Distributed Object Computing Conference, 2009. EDOC '09. IEEE International, University of Twente. IEEE, 2009, pp. 3 – 13.
[11] V. Chiprianov, Y. Kermarrec, and S. Rouvrais, "Extending enterprise architecture modeling languages: Application to telecommunications service creation," in The 27th Symposium On Applied Computing. Trento: ACM, 2012, pp. 21–24.
[12] Eclipse Modeling FrameWork. http://www.eclipse.org/modeling/emf/.
[13] I. Alloush, V. Chiprianov, Y. Kermarrec, and S. Rouvrais, "Linking telecom service high-level abstract models to simulators based on model transformations: The IMS case study," in Information and Communication Technologies (EUNICE 2012), ser. Lecture Notes in Computer Science, R. Szabó and A. Vidócs, Eds., vol. 7479. Springer Berlin Heidelberg, August 2012, pp. 100–111.
[14] I. Alloush, Y. Kermarrec, and S. Rouvrais, "A generalized model transformation approach to link design models to network simulators: Ns-3 case study," in International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2013). SciTePress Digital Library, July 2013, pp. 337–344.
[15] B. Jazayeri, S. Schwichtenberg, J. Küster, O. Zimmermann, and G. Engels, "Modeling and analyzing architectural diversity of open platforms," in Advanced Information Systems Engineering, S. Dustdar, E. Yu, C. Salinesi, D. Rieu, and V. Pant, Eds. Cham: Springer International Publishing, 2020, pp. 36–53.
[16] I. Crnkovic, S. Sentilles, A. Feljan, and M. Chaudron, "A classification framework for software component models," Software Engineering, IEEE Transactions on, vol. 37, 11 2011, pp. 593 – 615.
[17] J. El Hachem, Z. Y. Pang, V. Chiprianov, A. Babar, and P. Aniorte, "Model driven software security architecture of systems-of-systems," in 2016 23rd Asia-Pacific Software Engineering Conference (APSEC), Dec 2016, pp. 89–96.
[18] N. Medvidovic and R. Taylor, "A classification and comparison framework for software architecture description languages," 1, vol. 26, Jan 2000, pp. 70–93.
[19] V. Chiprianov, "Collaborative construction of telecommunications services. an enterprise architecture and model driven engineering method," Ph.D. dissertation, Telecom Bretagne, France, 2012.
[20] L. Touraille, M. K. Traoré, and D. R. C. Hill, "A model-driven software environment for modeling, simulation and analysis of complex systems," in Proceedings of the 2011 Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium, ser. TMS-DEVS '11, San Diego, CA, USA, 2011, pp. 229–237.
[21] K. Y. A. Achilleos and N. Georgalas, "Context modelling and a context-aware framework for pervasive service creation: A model-driven approach," Pervasive and Mobile Computing, vol. 6, no. 2, 2010, p. 281–296.
[22] J.-L. Bakker and R. Jain, "Next generation service creation using xml scripting languages," vol. 4, 2002, pp. 2001–2007 vol.4.
[23] NATO Architecture Framework. https://www.nato.int/.
[24] M. Brumbulli, E. Gaudin, and C. Teodorov, "Automatic Verification of BPMN Models," in 10th European Congress on Embedded Real Time Software and Systems (ERTS 2020), Toulouse, France, Jan. 2020. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02441878
[25] J. Sorribas, A. Barba, E. Trullols, J. Del Rio, A. Manuel, and M. de la Muela, "Marine sensor networks and ocean observatories. a policy based management approach," in Computing in the Global Information Technology, 2008. ICCGI '08. The Third International Multi-Conference on, July 2008, pp. 143–147.
[26] NEPTUNE - Ocean Networks Canada. https://www.oceannetworks.ca/.

[27] J. Bezivin, “In search of a basic principle for model driven engineering,,” Novatica Journal, vol. vol. 2, 2004, p. pp. 21–24.
[28] Atlas transformation language. http://www.eclipse.org/atl/.
[29] Eclispe Modeling. http://www.eclipse.org/modeling/.
[30] M. M. T. Zekai Demirezen, Barrett R. Bryant, “Dsml design space analysis,” in UAB, Birmingham, AL 35294, USA, 2011.
[31] H. Cho, J. Gray, and E. Syriani, “Creating visual domain-specific modeling languages from end-user demonstration,” in Modeling in Software Engineering (MISE), 2012 ICSE Workshop on, June 2012, pp. 22–28.
[32] I. Kurtev, J. Bézivin, F. Jouault, and P. Valduriez, “Model-based DSL frameworks,” in Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications, ser. OOPSLA ’06. New York, NY, USA: ACM, 2006, pp. 602–616.
[33] M. E. Liggins, D. L.Hall, and J. Llinas, Multisensor Data Fusion, Theory and Practice, S. edition, Ed. Taylor & Francis Group, LLC, 2009.
[34] I. Liggins, M.E., C.-Y. Chong, I. Kadar, M. Alford, V. Vannicola, and S. Thomopoulos, “Distributed fusion architectures and algorithms for target tracking,” Proceedings of the IEEE, vol. 85, no. 1, Jan 1997, pp. 95–107.
[35] I. Sommerville, Sofware Engineering, Ninth Edition, M. Horton, Ed. Pearson, 2011.
[36] I. Alloush, C. G. Aoun, Y. Kermarrec, and S. Rouvrais, “A domain-specific framework for creating early trusted underwater systems relying on enterprise architecture,” in 2014 IEEE 22nd International Symposium on Modelling, Analysis Simulation of Computer and Telecommunication Systems, Sep. 2014, pp. 120–125.
[37] C. Aoun, I. Alloush, Y. Kermarrec, O. Zein, and J. Champeau, “Domain specific modeling language for object localization in marine observatories,” SENSORCOMM 2014 - 8th International Conference on Sensor Technologies and Applications, 11 2014.
[38] C. Aoun, I. Alloush, Y. Kermarrec, J. Champeau, and O. Zein, “A modeling approach for marine observatory,” Sensors & Transducers, vol. 185, 02 2015.
[39] C. G. Aoun, I. Alloush, Y. kermarrec, J. Champeau, and O. K. Zein, “A mapping approach for marine observatory relying on enterprise architecture,” in OCEANS 2015 - MTS/IEEE Washington, Oct 2015, pp. 1–10.
[40] C. Aoun, L. Lagadec, J. Champeau, J. Moussa, and E. Hanna, “A high abstraction level constraint for object localization in marine observatories,” 12 2017, pp. 605–611.