Novel Model-Based approach for instrumentation and control of nuclear reactors

Bassem Ouni[1*], Christophe Aussagues[1], Saadia Dhouib[1] and Chokri Mraidha[1]
[1]CEA LIST, University of Paris-Saclay, 91120 Palaiseau, France
*corresponding author: bassem.ouni@cea.fr

***Abstract*— Technological platforms dedicated for digital instrumentation and control of nuclear reactors are quite complex in terms of functionalities and devices. Hence, the design of these platforms requires high-level abstraction layers able to reduce the complexity, to rise the automation and to check the consistency between different development stages. The development of such systems is a challenging task that requires modeling of various components at different levels of abstraction and viewpoints, notably functional, hardware and software levels. In this paper, a new system engineering methodology is proposed to provide high-level models of different components and inter/intra-communication between them. These models are used for system specification, architecture design, performance evaluation or verification and validation. This approach focuses on the internal behavior of different components at different levels of abstraction in order to enable the interoperability of these components and to enhance cooperation between different stakeholders of the development process. An experimental setup has been carried out to validate this approach by customizing an open source model based engineering tool, Eclipse Papyrus, towards a significant reduction of system development cost in terms of engineering resources and equipment devices.**

***Keywords* — System Engineering; Modeling; Instrumentation and Control; Monitoring nuclear reactors; model driven engineering**

I. INTRODUCTION

With the increasing complexity of Instrumentation and Control (I&C) systems, developers faced challenges in terms of functionalities, architecture definition, and implementation. Consequently, they deployed a Document-Based System Engineering (DBSE) approach carrying out text-based specifications. However, the DBSE approach raised several issues related to product definition and management, system development process, specifications, trace-ability, collaborative work and standards compliance. To overcome these weaknesses, system designers adapt the Model-Driven Engineering approach, known as MDE [1]. MDE attracted a lot of research interest in recent decades due to its ability to overcome the complexity challenges of systems. Moreover, MDE-based approaches allow analyzing the behavior, implementing and checking the safety of these systems where models act as the main development feature. The MDE area covers various standardized modeling languages, such as the Unified Modeling Language (UML) [2], and domain specific languages (DSLs) [3]. DSLs can be based on the UML extension mechanism, called UML profiles, or on meta-models. Considering this, modeling stakeholders harness the capabilities of MDE to reduce a system's complexity in many research fields [4-6].

The architecture of I&C devices dedicated to the control and protection of nuclear reactors is quite complex. To overcome this challenge, we aim to significantly reduce the development costs of these systems in terms of engineering time and equipment aspects. In order to address the cost improvement objectives of the engineering part, a new MDE-based methodology is proposed. To validate this approach, we extend the specific use of the open-source system engineering tool base, Papyrus [7], for the control and protection of nuclear reactors. The integration of this methodology within this tool encapsulates the main stages of use-case system engineering and carries out many automation steps and consistency checks between different design stages. In addition, it also improves the links between different tasks during the specification, design and development of the system.

This paper is organized as follows: next section introduces Model-Driven proposed approach. Section 3 refines the experimental setup. Finally, the conclusion and future works are drawn in the last section.

II. PROPOSED APPROACH

The proposed methodology aims to reduce the development cost of I&C systems, automate, and facilitate the consistency checks at different levels of abstraction. The Papyrus-based system engineering approach will replace the existing process where optimization in terms of resources has been identified. The first step of the approach consists of the description of the system functional architecture and the refinement of its behavior as a sequence of functions, their sub-functions and their interactions. The physical architecture is described subsequently as hardware equipment blocks and the network architecture is specified. After that, the system's physical behavior and the allocation of its functions on the hardware platform are detailed. Throughout different phases of this approach, a database of I/O, including data at different levels, is updated. The developed models are exploited for functional

simulation, document generation and exports to other tools. The modeling language tool used for developing the architectural model-based framework is Papyrus. Papyrus is an open-source model-driven engineering tool and available in the Eclipse platform [8].

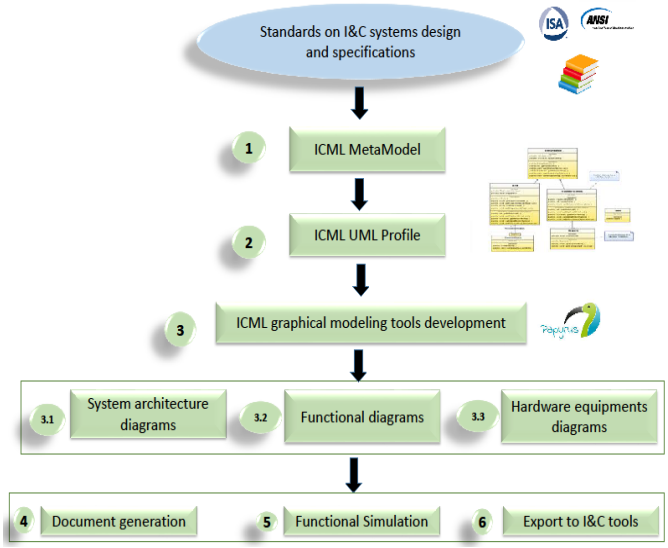

Fig. 1- Model-based tools development process.

The development of proposed tool and the associated graphical editors follows the flow depicted in Figure 1. Considering the I&C specification and the study of existing standards for I&C systems graphical notation [9], the tool is designed based on these two syntaxes:

- An abstract syntax or Metamodel: the metamodel defines different concepts manipulated in the domain and structures the relationships between these concepts and their semantics into a coherent unit. UML is used to formalize these concepts and define the relationships between them. The concept's semantics can be modeled textually or formally via a dedicated language, such as OCL (Object Constraint Language) [10].
- A concrete syntax: This defines the textual or graphical notations for the modeling language. The links between the abstract syntax concepts and the concrete syntax notations are also defined. Several concrete syntaxes can be defined for the same abstract syntax, which allows several representations for the same metamodel. To develop the concrete syntax, the UML [11] profile is used, which is to say, we define stereotypes extending the UML meta-classes. The major advantage of using the UML profiling approach is to benefit from a set of existing tools, both commercial and open-source.

Figure 2 details different levels of abstraction of the proposed framework. The first level of design covers the expected functions and their classification according to their safety category. Then, the behavior of these functions is refined: each system is an arrangement of functions, their sub-functions and their interactions with whatever their technological implementation. Furthermore, the system I/O data are modeled, considering the naming conventions, during this stage. For instance, in the example of Figure 2, we have three systems: system 1, system 2 and system 3. System 3 is not represented at the functional behavior description level for space reasons.

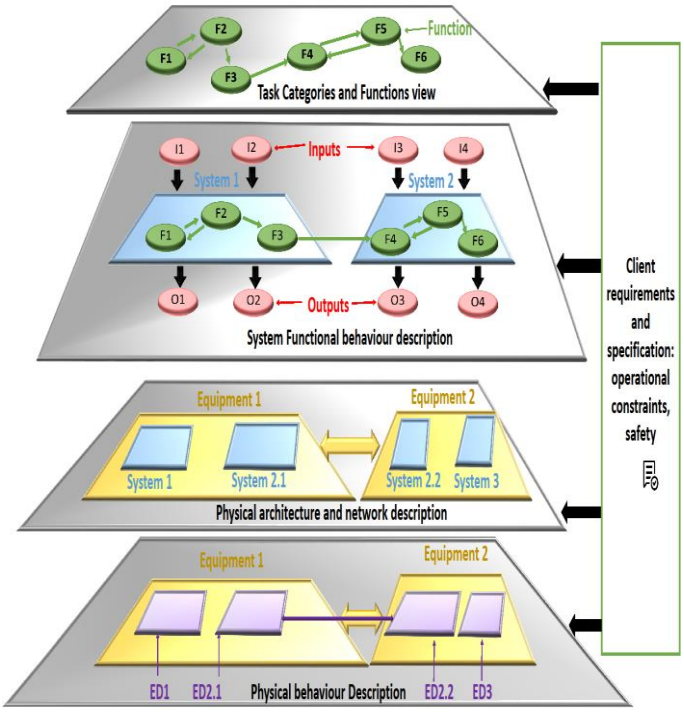

Fig. 2 - Different levels of abstraction of the proposed approach

The third level of abstraction defines the allocation of system functions on the hardware platform. The latter is modeled as a set of equipment. A system can be mapped on two different equipment: for example, System 2 is implemented on two equipment: the sub-systems system2.1 and system2.2 are mapped, respectively, in Equipment1 and Equipment2.

Finally, the detailed specification and equipment behavior (ED) are modelled, taking into account the technological implementation (failure processing, data definition for the monitoring system). A system's ED can describe the behavior of two different equipment. In Figure 2, ED2.1 and ED2.2, respectively, represent the behavior of Equipment1 and Equipment2, implementing System 2.

III. Experimental setup

The use case system is a Reactor Protection System (RPS) gathering various safety and security sensor-based devices embedded in a civil nuclear power plant aiming to safely shut down the civil nuclear reactor if radioactive materials leak. The (RPS) is composed of various safety functions placed on different hardware devices. The use case digital (RPS) is designed by experts from the Tsinghua University of China and described thoroughly in [12]. It is based on electronic boards, sensors, actuators, computers and software tools that are deployed within various sub-systems to ensure the safety and security of a 10MW high-temperature gas-cooled experimental reactor (HTR-10). The first part of this RPS consists of redundant protection logic units in three channels (trains); the second part gathers the surveillance stations and Post Accident Monitoring System located in the main control room; the third part is the monitoring system located in the auxiliary shutdown point.

The system outputs are the protection signals resulting from a

hard-wired 2/3 voting logic operation of the triggering signals in three redundant channels.

The RPS outputs a set of warnings and protection actions for each protection variable. For better safety management, developers divided the protection variables into two groups A and B. In each group, at least one protection variable is set up to protect each initial event, and each group is independently implemented in different processors to act as diversity within each channel. These protection actions are classified into four classes: Protection A (PA) includes the generation of an emergency trip signal to drop the control rods, trip the helium blower and isolate the secondary loop. The protection action B (PB) is referred to the relief of the steam generator. The protection action C (PC) consists of the isolation of the fuel loading/unloading system and the helium purification system from the primary loop. Finally, the protection action D (PD) targets isolating the thermal measurement system from the primary loop.

The proposed tool allows the modeling of a package called "SafetyFunctionLibrary" to define various safety functions; the modeled library has three graphical representations: tree, tabular or class diagram representations. A safety function can include many sub-functions. After that, we define the set of systems. Each system is placed on a rack and runs one or more safety functions. At a higher level of abstraction, the RPS project is modeled as a class with a structural diagram describing its different systems, sensors and actuators, as shown in Figure 3.

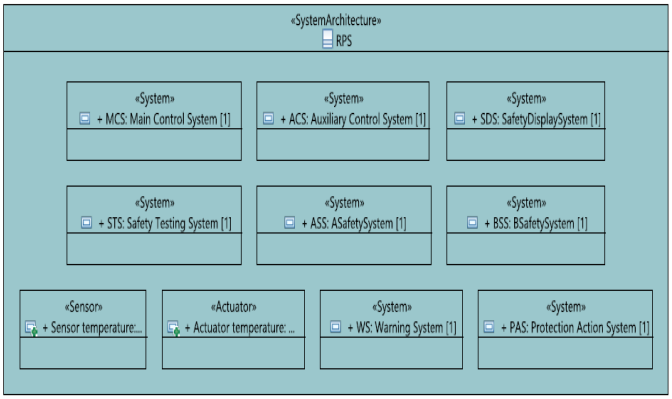

Fig.3- . RPS sub-systems, sensors and actuators.

The input/outputs of all systems are modeled as stereotypical "Sensor" or "Actuator" classes. Their assignment to systems can be edited using an attribute. Communications between systems are represented by links specifying the nature of communication, as depicted by Figure 4. Furthermore, each functional signal is characterized by a set of attributes to specify its type, binary or analog, whether it is an input or output, its I/O board, etc.

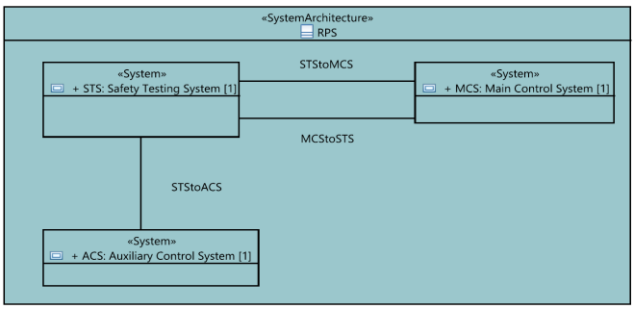

Fig.4- Communications between the RPS' sub-systems.

Moreover, communications between the equipment realizing the same functions or different functions are modeled at this stage. Figure 5 highlights the communication links between the equipment implementing different functions, which are the cores of the APIC and a temperature adapter device, through oriented communication links holding physical signals.

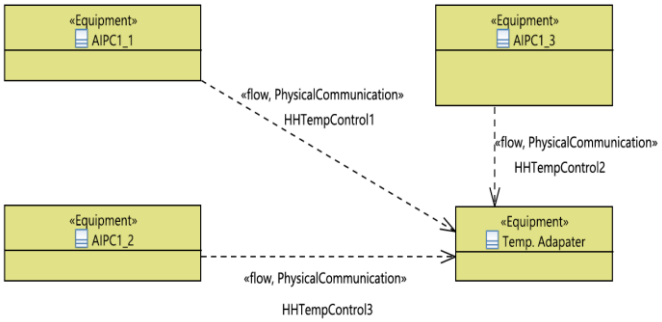

Fig. 5- Communication links between the equipment implementing different functions

IV. CONCLUSIONS

In this paper, we proposed a new model driven methodology to model at different levels of abstraction different I&C components and communication between them. These models are used for I&C system design, performance evaluation, verification and validation. Future works will focus on the enhancement of the synchronization between components at different levels of abstraction, the integration of simulation within the proposed framework to execute UML models, provide control, observation and animation facilities over these executions.

ACKNOWLEDGMENT

This work was supported by Bpifrance institution under the project ORION.

REFERENCES

[1] Adedjouma, M.; Thomas, T.; Mraidha, C.; Gerard, S.; Zeller, G. From Document-Based to Model-Based System and Software Engineering. In Proceedings of the OSS4MDE 2016—Open Source Software for Model-Driven Engineering, Saint Malo, France, 2–7 October 2016; pp. 27–36.

[2] Object Management Group. OMG Unified Modeling Language TM (OMG UML). In Technical Report; Object Management Group: Needham, MA, USA, 2015.

[3] Fowler, M. Domain-Specific Languages; Pearson Education: London, UK, 2010.

[4] Reyes C.R.P.; Vaca, H.P.; Calderón, M.P.; Montoya, L.; Aguilar, W.G. MilNova: An approach to the IoT solution based on model-driven

engineering for the military health monitoring. In Proceedings of the 2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON), Pucon, Chile, 18–20 October 2017; pp. 1–5.

[5] Prasinos, M.; Spanoudakis, G.; Koutsouris, D. Towards a model-driven platform for evidence based public health policy making. In Proceedings of the SEKE 2017—29th International Conference on Software Engineering & Knowledge Engineering, Pittsburgh, PA, USA, 5–7 July 2017; KSI Research Inc. and Knowledge Systems Institute: Pittsburgh, PA, USA, 2017; pp. 566–571.

[6] Van Lingen, F.; Yannuzzi, M.; Jain, A.; Irons-Mclean, R.; Lluch, O.; Carrera, D.; Perez, J.L.; Gutierrez, A.; Montero, D.; Marti, J.; et al. The Unavoidable Convergence of NFV, 5G, and Fog: A Model-Driven Approach to Bridge Cloud and Edge. IEEE Commun. Mag. 2017, 55, 28–35.

[7] Eclipse Papyrus ™ Modeling Environment. Available online: https://www.eclipse.org/papyrus/

[8] Eclipse Environment. Available online: https://www.eclipse.org/

[9] American National Standard. ANSI/ISA-5.1-2009 Instrumentation Symbols and Identification; American National Standard: Washington, DC, USA, 2009.

[10] Object Management Group. Object Constraint Language; Technical Report; Object Management Group: Milford, CT, USA, 2014.

[11] Object Management Gruop. OMG Unified Modeling Language TM (OMG UML). In Technical Report; Object Management Group: Needham, MA, USA, 2015.

[12] Li, F.; Yang, Z.; An, Z.; Zhang, L. The first digital reactor protection system in China. Nucl. Eng. Des. 2002, 218, 215–225.