CPS simulation models categories in Extended Enterprises

Renan Leroux
IRT Saint Exupéry — ALTRAN, Toulouse
IRIT-UPS/University of Toulouse
Renan.Leroux@irt-saintexupery.com

Marc Pantel, Ileana Ober, Jean-Michel Bruel
IRIT/University of Toulouse
IRT Saint Exupéry, Toulouse
First.Last@irit.fr

ABSTRACT

Simulation based early Validation and Verification is a key enabler for the Model Based Development of complex systems. These activities usually require distinct models for the System of Interest and for its execution environment. For Cyber-Physical Systems, the second kind combines generic environment behavioral models with scenarios that drive specific simulations. When these systems are developed in Extended Enterprises, several sub-systems are developed concurrently and the associated models may not be available when assessing some specific sub-system S developed by a given partner P, or might be cloaked to protect the Intellectual Property of the other partners. These other sub-systems thus become parts of the environment of S and appropriate models might need to be developed by the P partner when conducting simulations. This contribution illustrates these issues relying on the AIDA plane inspection system developed in the IRT Saint Exupéry MOISE project.

Reference Format: Renan Leroux, Marc Pantel, Ileana Ober, Jean-Michel Bruel. 2018. CPS simulation models categories in Extended Enterprises. In *Proceedings of GEMOC workshop at the ACM/IEEE MODELS conference (GEMOC@MODELS'2018).* 3 pages.

1 INTRODUCTION

Many complex Cyber-Physical Systems (CPS) are currently built in Extended Enterprises (EE) where stakeholders try to protect their know-how by minimizing the amount of data they share with the other stakeholders. Model Based Systems Engineering (MBSE) and early simulation based Validation and Verification (V & V) have been shown to significantly improve the efficiency of the development and the quality of the resulting products. In this context, stakeholders protect their know-how by cloaking parts of the models they have built when others need to simulate them during the V & V of their part of the system (called the System of Interest (SoI)). When these systems are built using Concurrent Engineering (CE), some models are even not available when a stakeholder need to conduct V & V activities for a SoI thus requiring him to build intermediate coarse models used for simulation. Our work targets an efficient methodology for building simulation models and associated tools for the V & V of complex CPS in EE.

GEMOC@MODELS'2018, October 2018, Eindhoven, Netherlands

This contribution first describes the MOISE project where our work takes place and the AIDA use case. Then, it provides first insights on the various categories of simulation models that must be built for AIDA and conclude on planned future activities.

2 THE MOISE PROJECT

The Technological Research Institute (IRT) *Antoine de Saint Exupéry* groups industrial and academic partners to transfer research results to industrial practice in the domain of Aeronautic and Space industries. The MOISE (MOdels and Information Sharing in Extended enterprises) project experiments EE aware MBSE Methods and Tools where simulation is used for models V & V. MOISE relies on the Arcadia method [7] and the Capella toolset [6] with 4 phases: *operational Requirement Functional Logical Physical* (RFLP).

Our work focuses on improving the use of simulation in EE relying on co-simulation standards like FMI [1]. It targets Methods and Tools to harness the development of simulators built in EE for models also built in EE. Figure 1 shows how simulators are derived from system models (see [2]) and how we use MBSE to build the various simulators needed to assess the models from the various MBSE phases (see [5]) . The second diagram relies on a RFLP method for the development of each simulation activity: the System Architecture provides *Requirements* for the simulation; the Simulation Architecture is built in *Functional* phase; the EE model is built in the *Logical* phase; the Co-Simulation Architecture is built in the *Physical* phase. Our proposal involves several actors working in EE: a) the *System Architect* (SyA) builds and assesses models of the SoI using simulation. It provides requirements for the simulations (scenarios, expected quality, etc); b) the *Simulation Architect* (SiA) designs the co-simulation platform that executes the model simulations ; c) *Simulation Model Developpers* (SMD) builds the various model components (Functional Mockup Units – FMU) that complete the SoI models to build a fully executable model.

To protect the confidential data and know-how of the various stakeholders in the same project, the various parts of the models developed by a stakeholder will only be partly available to the other ones. The FMI standard [1] provides such cloaking facilities in the execution of the model co-simulation [3].

3 THE AIDA INSPECTION DRONE USE CASE

The *Airplane Inspection Drone Assistant* (AIDA) use case was developed to illustrate and validate the work conducted in MOISE. A drone moves around a plane on the runway before take off (see Figure 2) to support the pilot in the mandatory pre-flight aircraft inspection. AIDA (i) quickens the pilot inspection task and (ii) improves its quality, by allowing scrutinizing all areas, even the ones not-easily-accessible (e.g., top of the wings, crown of the fuselage, ...), to detect irregularities, such as forgotten caps on sensors, ill

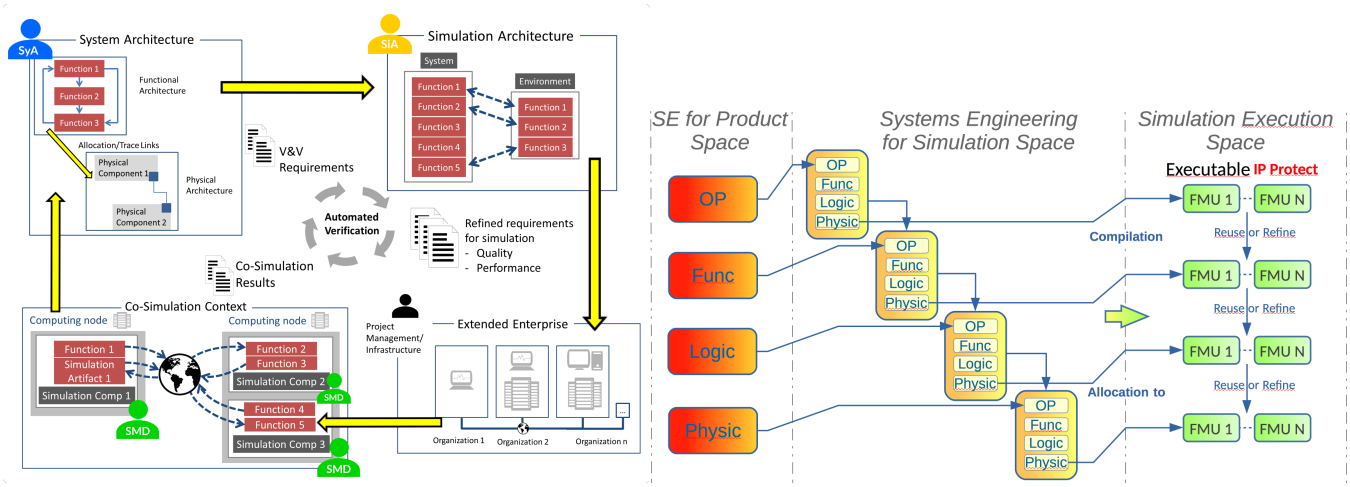

Figure 1: MOISE methodology and MBSE for co-simulation activities

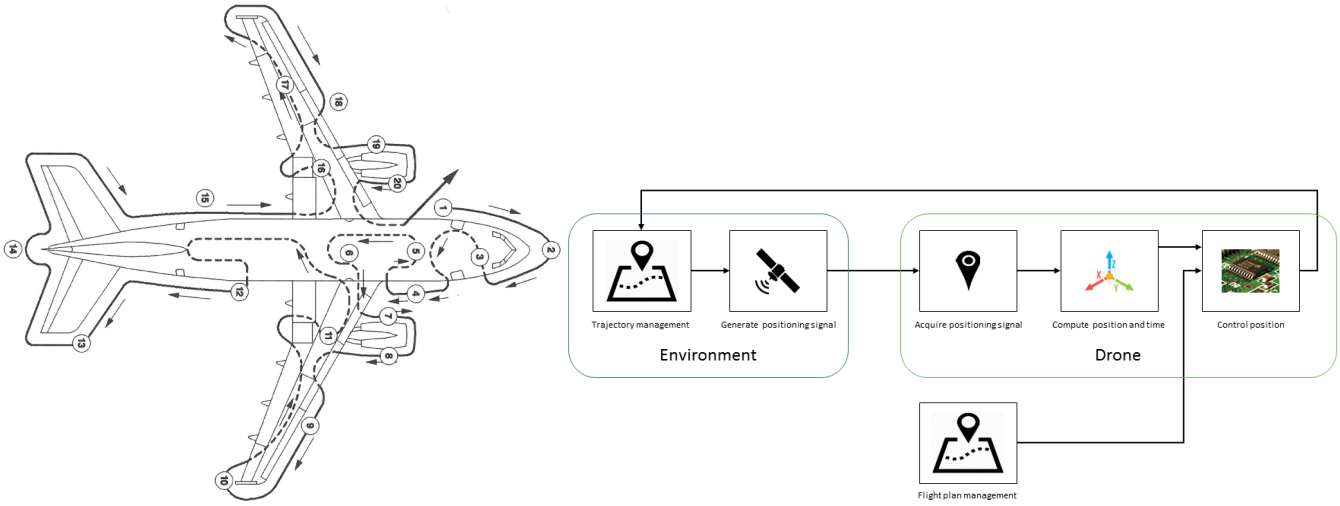

Figure 2: AIDA inspection drone flight plan example and mission sketch

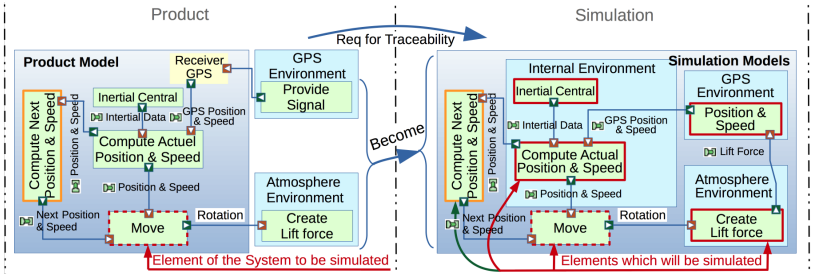

Figure 3: Functional architecture & associated simulation model

closed trap doors, or mechanical defects. AIDA can be manually controlled following predefined paths (flight plans), with enhanced automated safety capacities to avoid hurting ground staff. AIDA is aware of the cartography of the plane and of the location of the points of interest to be scrutinized. It is equipped with various captors: vision system, GPS locator, and a radar, for a greater precision, to ensure a sufficient safe distance with respect to the plane and the ground staff. To enable the diagnostic in case of malfunction, flight

data are saved locally and transferred in real-time to the ground. The operator can watch live images taken by the drone, make sure that control points do not present any irregularities, and adapt the drone flight plan. The drone mission is sketched in Figure 2.

Figure 3 is a part of the drone functional architecture model. The left diagram illustrates different kinds of functions in the product model: the SoI function assessed by the simulation is in red dotted line; an already developed function whose model is available for the simulation is in orange full line; and the other undeveloped functions whose models are needed by the simulation. The right diagram explained hereafter corresponds to our work in MOISE: how to build the simulation models and yield the executable simulator taking into account EE constraints.

4 MODEL CATEGORIES FOR CPS

Edward Alan Lee advocates [4] that there exist two fundamental kinds of models: *science* models that describes the observed system behavior and *engineering* models that prescribes the expected system behavior. Cyber Physical Systems mix *product engineering* models and *environment science* models. Both kinds of models can be continuous, discrete and even hybrid but Lee advocates that they should not be handled in the same manner. Product models are *prescriptive* and should be as simple as possible and always deterministic. Then the product must comply to the models. Whereas environment models are *descriptive* models needed to assess the product models. They should also be as simple as possible regarding the purpose of the assessment. Their correctness will be checked with respect to the physical behavior of the real environment. One key point is that level of details in the various models should be consistent in order to ensure an efficient and meaningful simulation. Indeed, in some cases, combining model with different scales (i.e. precision of the physical phenomena or numerical algorithms) can lead to incorrect simulation behavior.

In a single enterprise, with a theoretical V lifecycle, the environment is well known, all models are fully shared between all actors and can be built in the best order to ensure the correctness of the model based V & V activities (i.e. the environment models are correct with respect to the real environment and all the needed product models are available when the assessments are conducted).

In an EE, the various stakeholders want to protect their know-how and confidential data. In MBSE, these elements can be revealed by the models. Thus, the stakeholders do not want to share their models and they wish to hide as much as possible their content. The FMI standard [1] was designed partly to provide such masking techniques during co-simulation. System requirements for the environment behavior shared by all stakeholders are usually high level (e.g. AIDA should be able to fly with gusts of wind up to Xxx kms/h.). They are refined by each stakeholder according to the V & V requirements for the system parts he is designing. Thus, the targeted refinement level for environment models depends on the SoI models' one and on the numerical algorithms used for their simulations. These ones can reveal elements from the ones used for the SoI models and leaks some information about the product design that the developper want to protect. For a given stakeholder P, we will distinguish *internal* models that are built by P and *external* models that are built by the other stakeholders. They can be considered as the environment for the models built by P but they usually describe both parts of the product and its environment, thus are both prescriptive and descriptive. As the content of models from the other stakeholders involved in a simulation is partially hidden (i.e. black or gray box models), it is more difficult for P to build the most appropriate model needed for the assessment as it will interact during the simulation with these cloaked models. It is thus mandatory to provide requirements regarding the model expected qualities (see [2]).

Things can be even worse when using agile processes, where CE is used to maximize the efficiency of the development, it is mandatory to early conduct model based V & V activities even if some of the required *external* models are not available. The stakeholder that assesses an *internal* model he has designed must then build approximate *external* models that are neither fully *prescriptive* nor fully *descriptive*. Such models describe the behavior of the system parts that other stakeholders will build using their own *prescriptive* models. These models might combine parts of the system and parts of its environment. Then, he also wants to limit the level of detail of this model (as it is not *prescriptive*) to the one needed for a meaningful assessment. The right diagram in Figure 3 illustrates such functions using full red lines. These functions are tagged as *internal environment* as they are not part of the final models used to build the product. Recall that the function in full orange line is a prescriptive *internal* model that has already been developed and is thus not tagged in the same way. In these cases, additional verification activities must be conducted to compare the intermediate models that were built by other stakeholders to assess the models they were building, with the final models built by all intended stakeholders.

5 CONCLUSION AND FUTURE WORKS

This contribution illustrated the need for intermediate simulation models when developing a system using agile Concurrent Engineering in an Extended Enterprise. These models are used only to conduct early model based V &V activities and are neither fully prescriptive nor fully descriptive with respect to the product. They must be the subject of additional verification activities. We plan for the future both to extend this work to the full AIDA model and other use cases and lift it to the level of an ontology of models categories for developing Cyber-Physical Systems; and to study the meaning and constraints regarding the work done by Lee.

REFERENCES

[1] T. Blochwitz, M. Otter, J. Åkesson, M. Arnold, C. Clauss, H. Elmqvist, M. Friedrich, A. Junghanns, J. Mauss, D. Neumerkel, H. Olsson, and A. Viel. 2012. Functional Mockup Interface 2.0: The Standard for Tool independent Exchange of Simulation Models. In *Proc. of the 9th Intl Modelica Conference*. The Modelica Association.

[2] B. Bossa, B. Boulbene, S. Dubé, and M. Pantel. 2018. Towards a co-simulation based model assessment process for system architecture. In *Proc. of the 2^{nd} Workshop on the Formal CoSimulation of Cyber Physical Systems, part of the SEFM conf.*

[3] C. Gomes, C. Thule, D. Broman, P. Gorm Larsen, and H. Vangheluwe. 2017. Co-simulation: State of the art. (2017). http://arxiv.org/abs/1702.00686

[4] E. A. Lee. 2016. Fundamental Limits of Cyber-Physical Systems Modeling. *TCPS* 1, 1 (2016), 3:1–3:26. https://doi.org/10.1145/2912149

[5] R. Leroux, M. Pantel, I. Ober, and J-M. Bruel. 2018. Model-Based Systems Engineering for Systems Simulation. In *Proc. of the 13^{nd} International Symposium On Leveraging Applications of formal methods, verification and validation.*

[6] Polarsys. 2018. Capella. http://www.polarsys.org/capella/.

[7] J-L. Voirin. 2018. *Model-based System and Architecture Engineering with the Arcadia Method* (1st ed.). Elsevier.