Evaluation of the consistency between System Engineering, Dynamic and Safety Modeling views of a Mechatronic System

Aroua Berriche, Jean-Yves Choley, Faida Mhenni, Abdelfattah Mlika

Evaluation of the consistency between System Engineering, Dynamic and Safety Modeling views of a Mechatronic System

Aroua Berriche[a], Jean-Yves Choley[b], Faida Mhenni[c], Abdelfattah Mlika[d]

[a] Quartz Laboratory, Supmeca, 93400 Saint-Ouen, France, aroua.berriche@supmeca.fr

[b] Quartz Laboratory, Supmeca, 93400 Saint-Ouen, France, jean-yves.choley@supmeca.fr

[c] Quartz Laboratory, Supmeca, 93400 Saint-Ouen, France, faida.mhenni@supmeca.fr

[d] LMS Laboratory, ENISo, 4054 Sousse, Tunisie, abdelfattah.mlika@gmail.com

Résumé – Le processus de conception des systèmes complexes implique l'utilisation de langages et outils différents pour la modélisation et la simulation de ses structures et de ses comportements : modèles système (SysML), modèles multi-physiques (Modelica) et modèles sûreté de fonctionnement (Altarica). La conséquence principale est un risque fort d'incohérence entre les différentes vues du système. Dans ce contexte, nous sommes amenés à assurer l'échange entre les différents acteurs interagissant dans le développement d'un système complexe et de garantir la cohérence entre les différents modèles élaborés d'un même système. Dans cet article, nous avons proposé une méthodologie de synchronisation des modèles qui permet de détecter les incohérences entre les différentes vues d'un système. Cette approche est illustrée par un exemple dans le domaine de l'aéronautique qui a montré l'efficacité de cette proposition pour améliorer la coopération des concepteurs d'un système complexes.

Mots clés : Modélisation multi-vues / SysML / Modelica / Altarica / Cohérence des modèles.

Abstract – The design process of complex systems involves the use of different languages and tools for modeling and simulating its structures and behaviors: system models (SysML), multi-physical models (Modelica) and safety models (Altarica). The main consequence is a high risk of inconsistency between the different views of the system. In this context, we need to ensure the exchange between the different actors interacting in the development of a complex system and to verify the consistency between the different multi-view modeling systems. In this article, we proposed a model synchronization methodology to detect inconsistencies between the different views of a system. This Method is composed of two steps: first, the abstraction of entry models to a common representation and second the comparison process which permit to identify the inconsistencies between different views of a system. This approach is illustrated in a case study from the avionics industry, which verifies the effectiveness of this proposal to improve the cooperation between designers developing a complex system.

Key words: Multi-view Modeling/ SysML /Modelica / Altarica / Consistency management

1 Introduction

During the process of developing mechatronic systems, multiples views on the system to be built are often used, these views typically consist of models in different formalisms and to various partial aspects of the overall system. However, this approach has a cost associated with it. As individual view models evolve; inconsistencies between different models are often introduced. We are interested here to suggest an approach aiming at ensuring consistencies between different views.

The remainder of the paper is organized as follows: the next section describes the state of the art, and the proposed methodology is given in section 3, then we present a use case: an electro-mechanical actuator. Finally, a conclusion is given in section 5.

2 State of the art

To alleviate the problem of consistency between models in different languages involved in the design, researchers explored several clues over the last years.

There exists two engineering community researching in this field: The first, concentrate to carry MBSA (Model-Based Safety Assessment) approaches in accordance with system engineering works, authors in [1] suggested to add safety properties on system architecture viewpoint. The second, focus on integrating system architecture and dynamic system models, the author in [2] introduced a SysML profile for Modelica called SysML4Modelica in order to represent Modelica models in SysML. These approaches may be criticized because they consider only oriented relations from system architecture design to safety analysis and dynamic simulation.

In this paper, we have selected three particular but representative modeling tools for presenting our cooperative approach: SysML[3] for engineering system, Modelica[4] for dynamic and control simulation and Altarica[5] for safety analysis.

3 Presentation of the proposed method

In this section, we give an overview of our approach to model multi-view consistency and to automate consistency management during the process of developing complex systems.
The idea consists of two steps:

3.1 Abstraction:

It consists in transforming our source models (SysML, Modelica and Altarica views) in target models with a common representation using graph theory [6] to have the same level of abstraction.
We assume that the abstraction applies to model-to-model transformation [7].
We choose to apply the QVT operational (QVTo)[8], which is an imperative language, define the transformations using mapping. Each mapping can transform one or more element(s) of a source model to the target element(s).

3.2 Comparison:

This steps, we permit to identify the differences between two abstract viewpoints defined by the same metamodel (graph topology).
An algorithm must be developed in order to compare the abstracted models, using three activities:
- Search for components mapping between the two abstracted models.
-Execute a comparison process based on graph properties (label node, number of entering and leaving edges for every node…).
-Arrange the inconsistencies in vectors in different categories.
As a result, we can obtain either coherent models or incoherent ones where the designers must decide whether it is necessary to apply correction operations on their viewpoints or if the inconsistencies do not affect the modeling activity.
To illustrate our approach, we consider a case of an Electro-Mechanical Actuator EMA that is presented in the next section.

4 Case study :Electro-Mechanical Actuator EMA

The studied system is an electro-mechanical actuator EMA onboard an aircraft. The use of EMA technologies in flight control is a potential solution since they have many advantages, such as better environmental respect, weight saving, maintenance cost reduction, performance increase and speed accuracy.
An EMA is composed of three interconnected equipment, as shown in figure 1: an electric motor, a mechanical transmission and an electronic and software part composed of a calculator that controls the system.

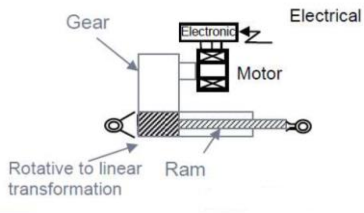

Figure 1 Composite view of EMA
We performed the three viewpoints of the EMA system:

The internal structure of the system was represented with the SysML Internal Block Diagrams (IBD), the dynamic modeling and simulation view was represented using Modelica language, and the safety analysis of the system was specified in Altarica language.
In figure 2, we modeled the IBD SysML view and the Modelica model of EMA.

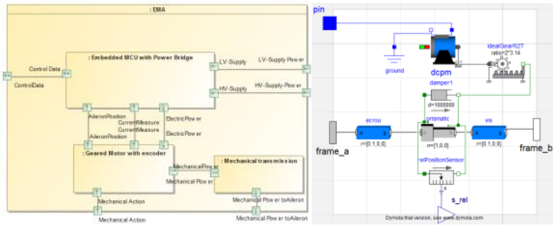

Figure 2 IBD SysML and Modelica models

4.1 Abstraction

To automatically transform the three models in a common representation, we use the Eclipse Modeling Framework platform EMF to implement our transformation using QVTo language according to these steps:
-create the source and target metamodels (as an example: IBD SysML metamodel and Graph metamodel are presented in figure 3)
-Define the rules that permit the mapping from the source to the target metamodel. (Example of the mapping is shown in figure 3)
-Instantiate the source metamodel and run the QVTo transformation.

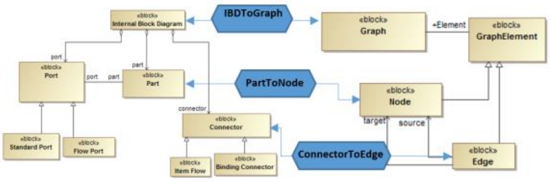

Figure 3 mapping between IBD SysML and Graph metamodels

4.2 Comparison

We compare the models using their topological graph representations as shown in figure 4.

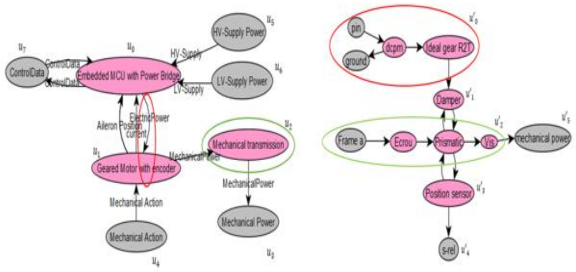

Figure 4 Components mapping between IBD SysML and Modelica graph representations

First, we search for mapping components between the two models (red and green circles in figure 4).
Then, we compare the graphs using a subgraph isomorphism algorithm presented in [9] and we can detect the nodes and edges that are not the same between IBDSysML and Modelica Model.

Thus, we can detect the inconsistencies between the two abstracted models then we inform the designers of its existence to evaluate the problem and to propose a list of correction operations that will be applied to the system.
This methodology can be applied to compare also SysML and Altarica models so; we can detect the inconsistencies that exist between the different views of our EMA system.

5 Conclusion

In this paper, a new methodology to evaluate the consistency of multi-view modeling approach for complex system process is proposed. It is made of two main phases. The first is an abstraction of models to transform the different views of one system in a common representation, at a same level of abstraction. The latter allows defining the mapping of components between models and compare their structures.

The main advantage of our approach is to reduce the time and cost of developing by minimizing the risk of failure in development in an advanced stage due to the misunderstanding between different designers of a system.

6 References

[1] F. Mhenni, « Safety analysis integration in a systems engineering approach for mechatronic systems design », Ecole Centrale Paris, 2014.
[2] T. A. Johnson, « Integrating models and simulations of continuous dynamic system behavior into SysML », Georgia Institute of Technology, 2008.
[3] OMG, « OMG Systems Modeling Language (OMG SysML^{TM}) », 2012.
[4] Modelica Association, « Modelica® - A Unified Object-Oriented Language for Physical Systems Modeling Language Specification Version 3.2 », 2010.
[5] T. Prosvirnova, « AltaRica 3.0: a Model-Based approach for Safety Analyses », 2015.
[6] S. Beretta, L. Denti, and M. Previtali, « Graph Theory and Definitions », in *Reference Module in Life Sciences*, Elsevier, 2018.
[7] T. Mens and P. Van Gorp, « A Taxonomy of Model Transformation », *Electron. Notes Theor. Comput. Sci*,p. 125‑142, mars 2006.
[8] OMG, « Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification ». 2016.
[9] M. Asiler and A. Yazıcı, « BB-Graph: A New Subgraph Isomorphism Algorithm for Efficiently Querying Big Graph Databases », juin 2017.