Mécanique & Industries 11, 401–406 (2010)

DOI: 10.1051/meca/2010052
www.mecanique-industries.org
Mécanique & Industries

Integrated design methodology of a mechatronic system

Régis Plateaux[a], Olivia Penas, Jean-Yves Choley, Faïda M'henni and Alain Rivière

Higher Institute of Mechanics of Paris (ISMEP-Supméca), Mechanical Systems and MAterials Engineering Laboratory (LISMMA), 3 rue F. Hainaut, 93407 St Ouen Cedex, France

Received 7 October 2009, Accepted 19 July 2010

Abstract – Current literature provides many studies on the mechatronic design process. However, they only focus on one level of the design V-cycle. This study deals with the entire downward side of the cycle in view to processing it globally. To achieve this, we propose a hybrid methodology based on several tools, languages and methodologies such as SADT, SysML, Modelica and CATIA Systems®. The method was validated successfully in a Modelica/Dymola framework. It is now possible to partially automate the process.

Key words: Integrated design methodology / Modelica / mechatronics

Résumé – Méthodologie de conception d'un produit mécatronique. Aujourd'hui, de nombreuses méthodologies et outils de conception mécatronique existent. Mais leur approche est partielle au sens où ils ne permettent de passer continûment des spécifications au prototype virtuel. Dans cet article, la première partie descendante du cycle en V est intégrée. Pour faire cela, une méthodologie hybride, basée sur différents outils, langages et méthodologies tels que l'analyse fonctionnelle, SADT, SysML, Modelica et Catia Systems® est proposée. La méthode a été implémentée et validée avec le langage Modelica sous l'environnement Dymola, et il est possible de l'automatiser partiellement.

Mots clés : Méthodologie de conception intégrée / Modelica / mécatronique

1 Introduction

Design phases of mechatronic systems are numerous. Prior to the physical prototyping phase of a mechatronic system in the V-cycle, many steps whose consistency is not always guaranteed have to be carried out. First, the requirements are collected and translated into parametric specifications, then the system is broken down into a functional and a structural models and, finally, simulations are run during the preliminary design.

A major reason for this problem is the discontinuity of modelling between design phases, due to data losses or changes during data transmissions between the different design tools. Updates are not automatically passed onto the models. There is a numerical fracture between them.

Mechatronic systems are often highly complex, due to their high number of components, their multi-physical characteristics and their couplings. Consequently, it appears judicious to define the design needs of these systems.

Chakrabarti and Bligh [1] propose the three following requirements to define an ideal design approach:

- to support routine and innovative design;
- to be able to carry out synthesis;
- to provide variable design solutions depending on levels of detail.

To fulfil the first condition, the continuity of the different levels of modelling – functional, logical and physical – is fundamental. This continuity not only has the advantage of limiting data losses, it also makes it easier to ensure consistency between the different specifications, parameters and models of the different design levels. It reduces errors, omissions and redundancies and ensures that the final prototype satisfies the initial specifications. Moreover, it makes traceability easier by automating follow-ups and updates of common data between the different modelling levels, and by recording them in the design project documentation. The choices of solutions will therefore be automatically and fully documented since initial requirements will be explicit and accessible.

However, these conditions are not sufficient for mechatronics. Indeed, the multidisciplinary aspect of

[a] Corresponding author: regis.plateaux@supmeca.fr

mechatronic systems requires a single framework for all the domains involved, enabling the modelling of mechanics, electronics, control engineering, etc. and thus collaborations between services. Furthermore, during the preliminary-design phase, coupling calculus may require a simplified model built by other technical teams and thus require granular model representation in order to provide a multi-scale view as a function of user needs. Indeed, during the early stages of the preliminary-design stage, the more models can be simulated, the less it is necessary to iterate and the shorter the development time. In order to facilitate mechatronic design, visualisation early in the process design (before detailed geometry) of 3D parts, even simplified ones, would enable taking into account not only boundary box definitions, but also proximity and contact interactions. Moreover, for geometric preliminary design, the integration of the size and position of objects in 2D/3D logical tools makes it possible to handle geometric objects that incorporate their behaviour.

This also supposes that the technical databases between the different designers are interoperable, unless a shared technical database between their different projects exists.

Finally, a last point seems necessary to achieve the quick and efficient design of complex systems, i.e. access to a common database for all the designers of a project, in order to capitalize on previous studies and automate breakdowns of requirements, functions and recurrent architectures.

Therefore, after having defined our approach to ideal design, we shall now describe how we intend to implement it through current methodologies and languages. We also offer a framework in which numerical validation will be possible for each of the mechatronic design stages.

2 Existing methods, tools and languages [2]

Many tools and methodologies in various domains [3–6] already exist. They meet the following designer needs [7–9] to quickly create cheap, basic and reliable products [10, 11] but with often contradictory requirements.

In the field of mechanics the following methods are often used: functional analysis, APTE ("APplication des Techniques d'Entreprise") method, SADT (Structured Analysis and Design Technique) method [12]/IDEF methods [13] (Integrated DEFinition, IDEF0: for Function Modelling Method, IDEF3: Process Description Capture Method and IDEF4: Object-Oriented Design Method), and FAST (Function Analysis System Technique Diagram) method [14].

For electronics, software for hardware description (like VHDL and Verilog) is used to model and simulate the behaviour and architecture of electronic numerical systems. Tools for logical synthesis (Design compiler, RC, Leonardo, etc.) and logical cell libraries are used to build functional descriptions of circuits. Design methodologies with a system approach have begun to emerge [15, 16]. For example, in informatics, UML (Unified Modelling Language) appears to be a good tool that has now been specialized in SysML (System Modelling Language) [17, 18] for a system approach.

However, these tools have several shortcomings. For instance SADT (IDEF0), initially meant to be a descriptive and analysis tool, does not provide any sequential analysis, although this drawback can to some extent be overcome by IDEF4. As far as SysML is concerned, it offers a large number of diagrams for system modelling (requirements, functional, behavioural and structural diagrams), even if they are sometimes redundant. On the other hand, SysML does not provide any methodology. This having been said, the parametric diagram, if implemented, could enable the designer to simulate the physical system even without its geometry. Indeed, this diagram goes as deep as the physical behaviour of the components and can therefore be considered as the interface with the simulation.

Consequently, we notice that the main drawback of SysML and SADT is their inability to simulate the structure generated: there is no effective significant validation before the preliminary design phase. Therefore, even if certain consistency tests are possible, the validation of the final model is directly related to the designer's know-how, since the realistic simulation of the model is impossible for the time being.

Two of the possible solutions can be used to achieve this ideal design framework which, as described previously, includes modelling continuity, consistency, traceability, model data storage, simulatable modelling, initial 3D representation, and multi-level, multi-domain and multi-scale aspects. The different tools can either be interfaced or they can be integrated in a single framework.

The sheer number and variety of tools (multi-level, multi-domain) used for mechatronic design makes it difficult to interface all of them with each other. This is why we naturally turn toward the solution of integration, while choosing the most pertinent tools and methodologies for each level of mechatronic design: SysML and SADT for functional analysis, Modelica [19] for multi-physical simulation [20] and even geometric simulation in the Dymola framework [21], and CATIA V6 for the integrated digital mock-up. Certain studies had already been carried out to integrate several SysML modules in Modelica [22], while others are undergoing development in the laboratory to integrate geometric data of the digital mock-up (3D detailed geometry) in Modelica, to obtain dynamic 3D simulations.

3 Our integrated approach

A single framework was considered in order to provide a continuous process from requirements to detailed design, thus allowing the full integration of all the modelling design phases. The aim is to build a virtual prototype whose simulation predicts real system behaviour. This is why the Modelica language in Dymola tool has

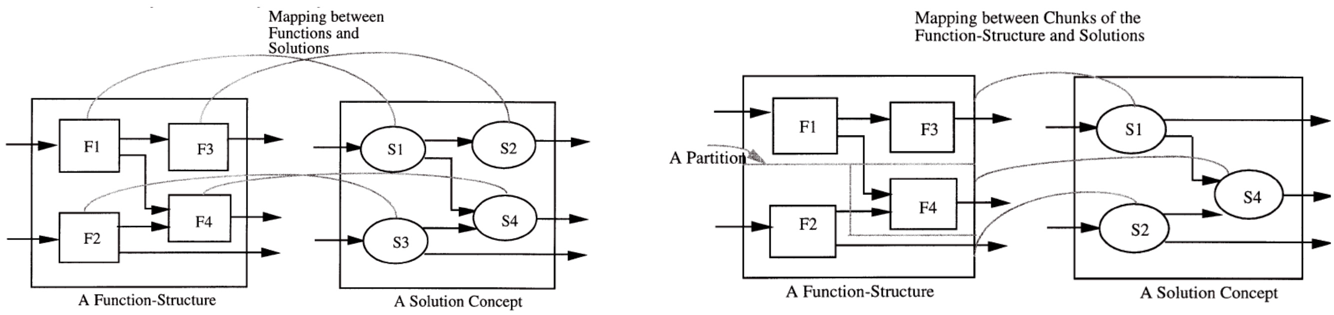


Fig. 1. Modular entanglement level [1] within Modelica.

been chosen for our development. First of all, this framework has to satisfy all the features defined previously for an ideal design framework. It also partially corresponds to the definition of an ideal design process framework defined by Chakrabarti and Bligh [1], due to its ability to support design of any nature and level, and enable design evolutions through different levels of detail.

Indeed, Modelica is an acausal object language that can be used to describe mechatronic (multi-domain) systems, in order to model discrete and continuous phenomena. One of the strengths of this language is that it offers a large number of free libraries (mechanics, electrics, hydraulics, thermal science, etc.) developed by specialists and already implemented in free tools.

This object oriented language makes it easily possible to restructure poor mapping of functions in terms of their structural correspondence (Fig. 1). The level of encapsulation can then be modified.

Whether in routine or innovative design, the modularity permitted by this language enables changing the class of the objects constituting the model. Thus modification of a component type is possible as creation from scratch (either by using a new topology of components or characteristic equations of system behaviour).

Moreover, the continuity of the design process is provided by elements developed in Modelica language within Dymola, in order to achieve modelling continuity through the different levels of the design approach (requirements, functional, components and structural). The design process is like an incomplete puzzle [23]. Requirements, functions, underlying structures and components are the puzzle's macro-pieces. The aim of this ideal process, from requirements to detailed design, is to complete this puzzle by reducing the number of different design models. Throughout this article, an electrical gate will illustrate our approach of this continuous design.

Functional and dysfunctional requirements are directly stored in text format in the Modelica documentation (annotation keyword) with quantitative criteria (with minimum and maximum acceptable values), and a routine in Modelica automatically creates requirement blocks in the graphic interface of Dymola (in the form of a SysML "requirement diagram"). The specifications in the documentation are updated whenever a change is made to the requirements. It will soon be possible to integrate user cases. The documentation is generated live

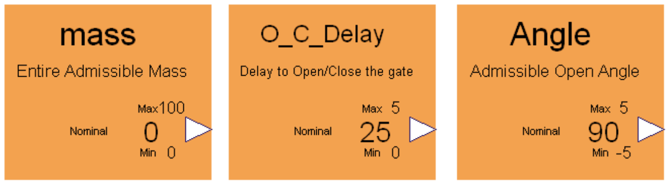


Fig. 2. Requirement objects in Modelica.

and incorporated in the model which includes the specifications, developments and so forth. Online documentation provides reminders on product design, identifying the model's strengths and weaknesses, its assumptions and limitations. This process can be dynamic. Defining specifications recognised by the keyword "constraint" can be considered in the documentation, producing the so-called constraint with minimum and maximum limitations.

Requirements are derived in functions connected by links between connectors (Fig. 2) by using an approach of successive refinements, similar to SADT representation. Figure 3 illustrates this representation with the example of the function "operate" of the electric gate. Each function is made up of simple or compound components, identified or not, and readily available (COTS: Component Off The Shelf) or not. There is no need to know the internal structure of the functions to define flows and their associated connectors. Synchronization aspects may be taken into account with further charts by using the set of the structure's declared variables.

Normal behaviours and anticipated dysfunctions are described with Modelica SFC (sequential function charts). Figure 4 shows the example of the opening/closing sequence of the electric gate.

These functions can be linked to structural solutions or to breakdown levels of these functions through a function/solution mapping cross table (Tab. 1). Equations of behaviour can be introduced at this level to identify the flow parameters (as in the "parametric diagram" of SysML) and to begin simulations. Existing solutions can be chosen at each analysis phase to design all or a part of the corresponding sub-assemblies.

This is followed by adopting certain strategies:

- To verify that the function has at least one usable solution without additional developments. Optimal architecture could be investigated in the case of a solution set;

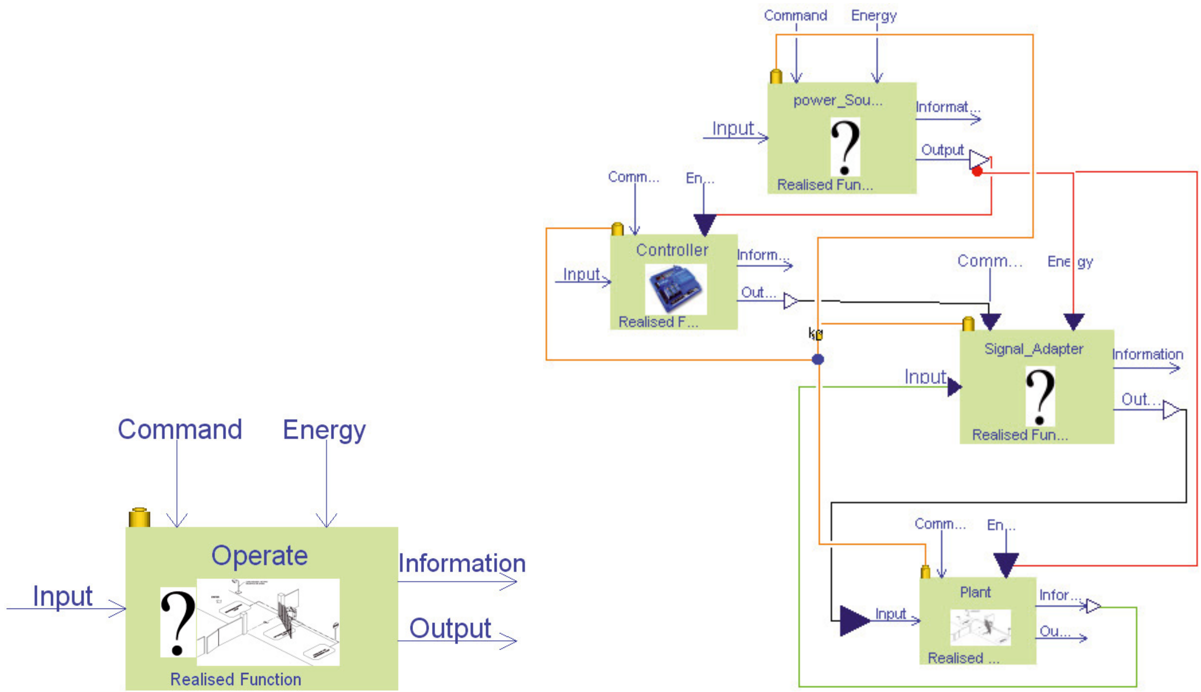


Fig. 3. Examples of SADT representation of a function object and its refinement in Dymola.

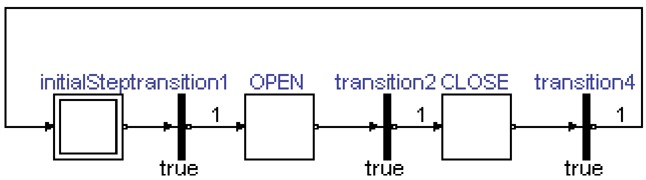


Fig. 4. "Sequence function chart" in Modelica.

- To break functions down into elementary pieces, in order to perform a direct transition to structural design;
- To seek the solution that fulfils the most functions, verify if all the functions are considered, or else repeat the first step with the remaining functions.

Different levels of structural modelling (Fig. 5) can then be used to simulate system operation. This fills out the geometry commonly used for the model, where the design intends are materialised by dimensioning.

Structural elements reveal the first geometric elements. They enable a 3D representation (sometimes simplified) of components which, by integrating geometric parameters, make multi-physical preliminary design simulations more realistic. Some COTS can provide 3D detailed geometry from CATIA.

Flat representation of Modelica objects (Fig. 6) with their positions and dimensions is purely graphic. Geometric data are not coupled with these 2D icons. That is why we propose to extend the representation of Modelica objects in 3D, so as to integrate their geometric behaviour. This shift in paradigm places the geometry at the heart of the simulation. During the design phase, 3D representation will guarantee geometric consistency of the model. In addition to realistic visualization during the simulation phase, collision would modify the behaviour of the system through combined use of 3D and multi-physical solvers. This aspect is detailed extensively in another article [24].

One last important aspect of this fully integrated approach is the transition from the continuous domain (simulation results) to the discrete domain (components). Indeed, a simulation can be processed in the continuous domain before seeking adequacy between the simulated ideal component and a real instance such as COTS (discrete domain). It is then essential to use a database to preserve consistency and coherency. The Dymola integration in CATIA System (V6) enables access to the CATIA PDM module, thereby allowing storing, structuring, plotting, sharing and promoting the reutilization of existing standards and data. Thus capitalizing on different models is possible.

4 Conclusion

This paper proposed an integrated design approach for the design of mechatronic systems within the Dymola/Modelica framework. Its properties make it possible to ensure many of the conditions necessary for an ideal design framework for mechatronic products: the guarantee of consistency and traceability of parameters between levels of modelling is automatic due to the uniqueness of the parameters, language and information within a single framework. Multi-level, multi-domain and multi-scale aspects are already present. The requirements blocks, functional, dysfunctional and structural graphs, and 3D vision can be developed in Modelica. Finally,

Table 1. Cross-functional mapping of the electric gate example.

	Solutions								
Functions	S1 Microcontroller	S2 Transistor	S3 Motor	S4 Gear Reducer	S5 Levers	S6 Screw/nut	S7 Incremental Rotated Sensor	S8 Absolute Rotated Sensor	S9 Motor Sensor Reducer Incremental Sensor
F1 Control System	X								X
F2 Adapt the electric power		X							
F3 Transform electric energy into mechanical energy			X						X
F4 Adapt rotation mechanical power	(X)	(X)		X	X	X			X
F5 Measure the state of the system							X	X	X

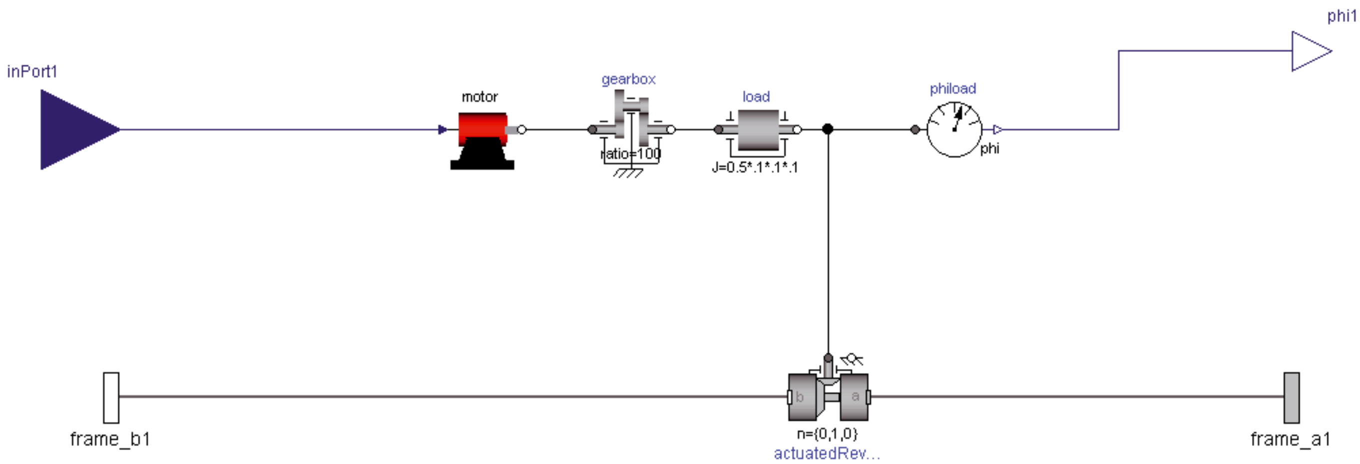
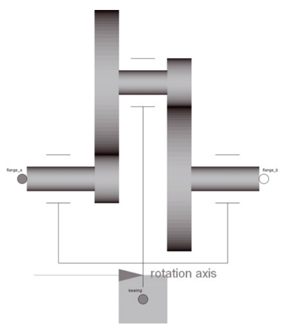


Fig. 5. Choice of structural component within Dymola.




```
model IdealGear "Ideal gear without inertia"
  extends Interfaces.TwoFlangesAndBearing;
  parameter Real ratio=1 "Transmission ratio (flange_a.phi/flange_b.phi)";

  a;
equation
  phi_a = ratio*phi_b;
  0 = ratio*flange_a.tau + flange_b.tau;
end IdealGear;
```

Fig. 6. Example of graphic object and its code in Modelica.

capitalization of models and information may be provided by integrating Dymola with Catia System® V6.

This ideal framework does not yet exist but Catia System® V6 (Dassault Systems) could be the framework in which this development takes place. Dymola is now integrated in the CATIA structure and a Requirement, Functional, Logical and Physical (RFLP) process is proposed. This does not strictly correspond to the approach suggested, but the modules exist within the same development tool. This new framework simply needs electronics and data processing to meet the needs of complete integrated mechatronic design. Then it may allow global optimization in order to provide a set of fitting parameters.

However, mechatronic design also copes with manufacturing variations and operating constraints. Study of the robustness of this methodology is necessary to completely achieve a robust integrated design. In computer science, formal techniques, such as fuzzy testing, robust control, and detailed fuzzy logic are essential. Neurocontrol by artificial neural networks and adaptive control based on neural network modelling could be taken into account to provide an effective industrial process.

References

[1] A. Chakrabarti, T.P. Bligh, A scheme for functional reasoning in conceptual design 22 (2001) 493–517

[2] A. Pop, D. Akhvlediani, P. Fritzson, Towards Unified System Modeling with the ModelicaML UML Profile, Proceedings of the 1st International Workshop on Equation-Based Object-Oriented Languages and Tools

[3] H.-S. Yan, A methodology for creative mechanism design, Mech. mach. Theory 27 (1992) 235–242

[4] J.-Y. Choley, Mécatronique: une nouvelle démarche de conception des systèmes complexes, Technologies et Formations. Delagrave Editions S.A., 2006, Vol. 127 pp. 29–35

[5] R. Sell, M. Tamre, Integration of V-model and SysML for advanced mechatronics system design, Annecy, 2005, pp. 276–280, REM05

[6] M. Bernardi, H. Bley, B. Schmitt, New approaches for developing mechatronical products in multidisciplinary teamwor, Seoul, 2002

[7] G. Pahl, W. Beitz, Engineering Design – A Systematic Approach, 2nd ed., London, Springer, 1996

[8] T.J. Teorey, D. Yang, J.P. Fry, A logical design methodology for relational databases using the extended entity-relationship model, New York, ACM Comput. Surv. (1986) 197–222, doi: 10.1145/7474.7475

[9] L.T.M. Blessing, A process-based approach to design, London, 1995

[10] J. Van. Amerongen, Mechatronic Design, J. Mechatronics 13 (2003) 1045–1066, http://www.ce.utwente.nl/rtweb/publications/2003/pdf-files/125CE2003.pdf

[11] J. Van. Amerongen, Mechatronic Design: A Port-Based Approach, American University of Sharjah, United Arab Emirates, 2007, pp. 1–8

[12] M. Rausand, A. Høyland, System Reliability Theory: Models, Statistical Methods, and Applications, Wiley-IEEE, 2004. pp. 81–83. ISBN 047147133X

[13] Command, Air Force Systems, Information Integration for Concurrent Engineering IDEF4 Object-Oriented Design Method Report. [Online] 1995, http://www.idef.com/

[14] Del L. Younker, Value Engineering: Analysis and Methodology, CRC Press, 2003, 350 pages. ISBN 082470696X

[15] D. Hubaux, L. Guedria, L. Vandendorpe, M. Verleysen, J.D. Legat, Nouvelles méthodes de conception de systèmes, Hammamet Tunisie, 2002, pp. 341–345

[16] David Maliniak, Don't be a separatist–think "System". [Online] 01 02 2007. [Cited: 02 03 2009.] http://electronicdesign.com/Articles/Index.cfm?AD=1&ArticleID=14723

[17] SysML [16] System modeling language (SysML) specification. Version 1.0, Draft. OMG document ad/2006-03-01, SysML [Online] 2006. http://www.sysml.org/ http://www.omgsysml.org/

[18] Georgia Institute of Technology, INCOSE MBSE Challenge Team for Mechatronics/Interoperability. Georgia Institute of Technology [Online] http://www.pslm.gatech.edu/projects/incose-mbse-msi/

[19] MODELICA, Modeling of Complex Physical Systems, Modelica [Online] http://www.modelica.org/

[20] H.J. Hadj-Amor, Contribution au prototypage virtuel de systèmes mécatroniques basé sur une architecture distribuée HLA-Expérimentation sous les environnements OpenModelica-OpenMASK, Toulon, 2008, Thèse du Laboratoire LISMMA (EA 2336) Supmeca Toulon

[21] Dassault Systems, Dymola multi-engineering Modelling and simulation, [Online] http://www.3ds.com/products/catia/portfolio/dymola

[22] S. Turki, Ingénierie système guidée par les modèles: Application du standard IEEE 15288, de l'architecture MDA et du langage SysML à la conception des systèmes mécatroniques, Université du Sud Toulon-Var, 2008

[23] R. Plateaux et al., Towards an Integrated Mechatronic Design Process, ICM, Malagà, IEEE, 2009, doi: 10.1109/ICMECH.2009.4957237

[24] R. Plateaux, et al., Vers un environnement intégré pour le prédimensionnement – Modelica 3D, Marseille, CFM, 2009

[25] J.E.E. Sharpe, Computer tools for integrated conceptual design, Design Studies 16 (1995) 471–488