Computer-Aided Design 45 (2013) 764–776

Contents lists available at SciVerse ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

SysML-based uniform behavior modeling and automated mapping of design and simulation model for complex mechatronics

Yue Cao [a], Yusheng Liu [a,*], Hongri Fan [a], Bo Fan [b]

[a] *State Key Lab. of CAD&CG, Zhejiang University, Hangzhou, 310027, PR China*

[b] *School of International and Public Affairs, Shanghai Jiaotong University, Shanghai, 200030, PR China*

ARTICLE INFO

Article history:
Received 21 August 2011
Accepted 5 May 2012

Keywords:
SysML
Hybrid behavior modeling
System-level behavior
Mechatronic system
Model transformation

ABSTRACT

Model-based systems engineering (MBSE) is becoming a promising solution for the design of mechatronic systems with the increasing of complexity. To facilitate system design engineers to express their various behavioral requirements, a uniform behavior modeling approach is proposed based on SysML. A set of new constructs are proposed which compose the uniform behavior modeling profile. Based on the profile, the hybrid behavior of the mechatronic system can be uniformly represented in a multi-view and hierarchical way. Moreover, the language- and tool-dependent simulation related information is also described based on SysML as the complement part of the uniform behavior model, which can be transformed to different simulation platforms based on the triple graph grammar (TGG). In this way, the system-level design of the mechatronic system is simulated and verified automatically in the early design stage. Finally, the method is implemented and an example is given to illustrate the whole process.

1. Introduction

The complexity of mechatronic systems is increasing which integrates multiple physical processes and multiple singular technologies together. System-level design, which considers mechanical, electrical and software components simultaneously, is needed for the development of complex mechatronics at the initial stage [1]. Recently, model-based systems engineering (MBSE) [2] has attracted many researchers' attention and is becoming a significant method for the system-level design of mechatronic systems. And the standard system modeling language, SysML, has been proposed to support the specification, analysis, design and verification and validation of a broad range of complex systems [3]. Based on the formalized definition for the design requirements from the initial design stage, it provides unambiguous semantic description, supports early-stage simulation and gives a common communication platform between different stakeholders and designers. However, it is obvious that there are great differences between the mechatronic system and other complex systems. The most remarkable two are the multi-domain integration and discrete/continuous hybrid behavior. The intertwined hybrid behaviors of various domains result in the complexity of behavior modeling. Some research has been conducted to adapt MBSE for mechatronic system modeling [4]. However, according to our knowledge, there is no a satisfactory method for SysML based mechatronic system behavior modeling and its integration with different simulation tools. The deficiencies lie in:

(1) Only the physical part of the mechatronic system is considered now. Essentially, the mechatronic system is comprised of two parts: the control subsystem and the controlled physical subsystem. The control and software subsystem has become more and more important with the requirement of increasing the product intelligence and flexibility.
(2) There is still no uniform behavior modeling framework defined in SysML. Specially, several diagrams are defined to represent the discrete behavior whereas the parametric diagram (PAR) can be used to represent the continuous behavior. However, there is still no uniform framework for complicated discrete/continuous hybrid behavior modeling [5,6]. The consequence is that different system engineers will devise different behavior representations for the same requirement and thus it is not easy to transform them into the simulation models automatically.
(3) The integration between system design and simulation of mechatronic systems is still point-to-point, which puts a great burden on the developer. In fact, there is great similarity between different simulation-oriented languages. It is possible to develop a formalized mapping mechanism between the design and different simulation languages.

Based on the above analysis, a SysML-based uniform behavior modeling language(UBML) is proposed as a SysML profile for

* Corresponding author. Tel.: +86 57188206681.
E-mail address: ysliu@cad.zju.edu.cn (Y. Liu).

0010-4485/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2012.05.001

Nomenclature

CA	Command amplifier
CP	Command potentiometer
DAE	Differential algebraic equation
DP	Detection potentiometer
EC	Execution control
ECC	Execution control chart
FB	Function block
IBD	Internal block diagram
MBSE	Model-based system engineering
PA	Power amplifier
PAR	Parametric diagram
SA	Signal amplifier
SPD	Sequenced parametric diagram
SysML	System modeling language
TGG	Triple graph grammar
UBML	Uniform behavior modeling language
UML	Uniform modeling language

facilitating systems engineers to define behavioral requirements uniformly and to help automate the mapping from design model to simulation model.

This paper is organized as follows. Section 2 introduces the related works. Sections 3–6 describe the main parts of this method. Section 7 illustrates the implementation of the whole process with a case study. Section 8 presents the conclusion and future work.

2. Related work

In this study, the hybrid behavior modeling and model integration of different languages for system-level design and simulation are two key parts. The following gives their related works respectively.

2.1. Hybrid behavior modeling

Mechatronic systems often exhibit both discrete behavior and time-continuous behavior. Therefore, hybrid behavior modeling is an essential work for mechatronic systems. Generally, the theoretical basis for hybrid behavior modeling is hybrid automata [7], in which the hybrid system is specified as a finite automaton. CHARON extended the hybrid automata to a modular hierarchic automata model [8]. The behavioral hierarchy is represented by modes. However, CHARON is a textual modeling language without standard visual representation.

Hybrid UML [9] is a UML extension for the specification of hybrid systems which lifts the constructs with formal semantics of CHARON to the associated constructs in UML syntax. Since there are no continuous behavior modeling constructs provided by UML, the continuous behavior in hybrid UML is still modeled on a textual level. Moreover, the state in this approach has only control interfaces through which it can connect with other states by transitions. However, the data interfaces are not provided. Mechatronic UML [10] is a well-structured component-based specification technique for complex, reconfigurable mechatronic systems. The system behavior is modeled by the hybrid reconfiguration automaton which contains locations connected by transitions. Inside each location, the dynamic internal structure of the hybrid component is modeled by the composition structure diagram. This approach has achieved a tight integration of structure and behavior. However, the continuous behavior has not been explicitly modeled.

2.2. Model integration

Since there are a wide range of domain-specific models and corresponding languages for modeling different aspects of the system, a model integration framework is needed for managing the various modeling languages [4]. Here, the languages for the system-level design and simulation, i.e. SysML for system-level design and Modelica [11] and Simscape [12] for system-level simulation are considered. SysML is a general modeling language with strong extensibility, while Modelica and Simscape are typical object-oriented languages for simulating the dynamic behavior of technical systems.

Vanderperren and Dehaene discussed two possible integration methods between UML/SysML and Matlab/Simulink [13]. However, only the discrete behavior was considered and the association between the design and simulation model was not considered. Turki and Soriano [14] represented bond graph-based continuous dynamic behavior based on SysML activity diagrams. Pop et al. [15] proposed the ModelicaML as an integration of Modelica and UML which supports specifying the requirements, behavior, and Modelica simulation in a uniform way. Similarly, Nytsch-Geusen [16] presented the UML^H. These approaches created UML constructs representing the same semantics of Modelica ones and used the new created constructs to model the system behavior.

Johnson et al. [17,18] introduced a formal approach for modeling continuous system dynamics in SysML based on language mapping between SysML and Modelica. Besides representing the continuous dynamics modeling constructs and method of Modelica in SysML, an approach for integrating the continuous dynamics model with other SysML constructs was also proposed. The dynamic behaviors in SysML and Modelica can be transformed into each other by the model transformation framework VIATRA [19]. Cao et al. also discussed the integration of system-level design and simulation of mechatronic system behavior [5,6] in which the design model in SysML and the simulation model in Simscape can be transformed into each other. Since this kind of approach is based on language mappings, they are restricted to certain simulation languages or tools. The uniform approach applicable to multiple simulation tools has not been proposed.

3. Uniform behavior model

As mentioned above, the complexity of behavior modeling of mechatronic systems results from the following two intertwined aspects: (1) *Hybrid behavior modeling*: mechatronic systems often exhibit discrete behavior, time-continuous behavior, and hybrid behavior. (2) *Multiple domain integration*: the various involved domains are connected with each other with their own special characteristics. Moreover, the two aspects of complexities are not independent. Therefore, a general set of SysML constructs should be defined to express the domain-specific semantics.

In this section, the UBML is defined based on the extension of SysML. The benefits of the UBML are two-fold. The first is to represent the above three types of behavior uniformly for both the control subsystem and the controlled physical subsystem. Therefore, system engineers can express their design intents for behavioral requirements uniformly. The second is to help automate the model transformation between system-level design and simulation models.

3.1. Definition of the UBML

There are two extension mechanisms for SysML: the heavy-weight method that defines new meta-classes and the light-weight method that creates stereotypes by extending existing constructs. Here, the latter is chosen since it is easy to be implemented by the existing SysML modeling tool.