Advanced Engineering Informatics 35 (2018) 17–29

Contents lists available at ScienceDirect

Advanced Engineering Informatics

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

Full length article

Change propagation analysis for system modeling using Semantic Web technology

Haoqi Wang[a,*], Vincent Thomson[b], Chengtong Tang[a]

[a] School of Mechanical Engineering, Beijing Institute of Technology, China
[b] Department of Mechanical Engineering, McGill University, Canada

A R T I C L E I N F O

Keywords:
Model-based systems engineering
System modeling language
Engineering change management
Change propagation analysis
Ontology
Semantic Web technology

A B S T R A C T

Change propagation potentially affects many aspects of a SysML-based system model during the iterative process of Model-Based Systems Engineering (MBSE). However, few authors have addressed the implication of engineering change and its impact. To address having a successful change process, this article analyzes and explicitly represents different scenarios of how a system model is changed from a formal perspective, i.e., how a system model should be changed, and how model elements should be added, deleted or modified in response to design changes. A workflow is introduced to guide the change process taking change propagation into account. Second, change impact relationships among requirements, behaviors, and structures of the system model are formalized by an ontology to make the semantics both human-understandable and machine-readable. Reasoning rules are defined as well in order to improve automation of the change process. Finally, an experiment using a water distiller system showed that the identification of change impact information could help designers complete the change in less time and with higher quality.

1. Introduction

Systems modeling of contemporary products using model-based systems engineering (MBSE) is an iterative process involving representations of multiple views, e.g., requirements, behaviors, and structures. The systems modeling language (SysML) is a key enabler in MBSE approaches for creating a system model [1]. During the modeling process, system design evolves through many modifications until it is feasible. Accordingly, the SysML-based system model, which produces the primary artifacts from a system design, has to be modified frequently. A single design change may lead to unexpected effects on other model elements. For example, an added requirement to the original system design of a water distiller causes changes to the distiller behavior, decomposition structure and internal structure that are tedious to make [2]. So, Engineering Change Management (ECM) of the system model plays a pivotal role for a successful implementation of MBSE.

As one of the ECM methods, the SysML-based methodology can evaluate change situations arising from system design by representing requirements, behaviors, structures and their relationships in a consistent way [3]. However, few authors have addressed the implication of engineering change and its impact or the implication of different approaches to design implementation, such as, technology, modularization and make-or-buy [4]. Although allocation relationships in SysML are able to enhance the traceability of a system model [5], they are insufficient for specifically describing how change is propagated to impacted model artifacts. In practice, it is always time consuming for designers to identify and change impacted model elements manually. One reason is that requirements and scenarios are not always fully defined, and thus, design changes require considerable analysis in order to decide upon feasible approaches. As a consequence, in order to keep the system model compatible with design changes, designers have to spend most of their time finding which model elements should be changed given how changes are propagated throughout the design and which types of changes should be executed for specific model elements. For instance, in order to change the initial system design of a water distiller, more than 60 model elements were required to be added, deleted or modified [2].

Accordingly, limited to a formal perspective, not an engineering perspective where changes to detailed design parameters, such as the dimension, weight, and material of components, are taken into account, this paper addresses an approach that should be used for MBSE modeling when reasoning about change propagation. To determine this approach we focus on three questions:

(1) How can the approach help designers to understand what needs to be changed in the systems modeling process?

* Corresponding author.
E-mail addresses: Haoqi.wang@mail.mcgill.ca (H. Wang), vince.thomson@mcgill.ca (V. Thomson), tangcht@bit.edu.cn (C. Tang).

https://doi.org/10.1016/j.aei.2017.11.004
Received 4 May 2017; Received in revised form 25 November 2017; Accepted 29 November 2017

(2) If a change happens in a specific SysML diagram, how can the approach help designers know how other related diagrams need to be changed?
(3) What procedures can be automated when modifying system engineering models due to change propagation?

In order to answer these questions, first, this paper categorizes different scenarios of the SysML modeling change process into three subclasses and establishes a change propagation model. Second, the Web Ontology Language (OWL) is used to formalize the change propagation information, based on which reasoning rules are defined to help designers to understand which SysML diagrams and model elements need to be changed, and then, guides them through the tedious process of manually changing SysML models. Third, possible automation of the process is investigated.

2. Research background

2.1. Previous work about change impact analysis for the system model

Existing research on the impact of change on the system model in MBSE can be divided into two classifications. One is comprehensive research on using the analytic capability of matrix-based representations. The second is research on predicting the change propagation risk by checking the inconsistency and incompatibility of the changed system model.

The first classification focuses on establishing a Design Structure Matrix (DSM) or a Multiple Domain Matrix based on SysML diagrams, and then, using these matrixes to predict the impact of change [6–12]. Matrix-based representations offer magnificent analytic criteria to achieve system understanding [10]. Clarkson et al. [6] developed a Change Propagation Method using numeric DSMs for the first time. Hamraz et al. [7] extended the Change Propagation Method by introducing Function-Behavior-Structure models. Hamraz et al. [8] further detailed the ontology of the Function-Behavior-Structure linkage method to provide a uniform framework for developing models. Waldman and Sangal [9] were the first to construct a DSM that unites the various views of SysML models. Maisenbacher et al. [10] illustrated the translation of SysML diagrams with network structures to their matrix representation as a DSM or Multiple Domain Matrix, so as to identify indirect dependencies in a system model. Nonsiri et al. [11] proposed a method for identifying the propagation path of changed requirements by integrating a SysML model with a higher order DSM. Fei et al. [12] developed a matrix-based method to analyze change propagation between components of a product based on functional and structural SysML models.

The second classification mainly studies checking the inconsistency and incompatibility of changed system designs to predict change propagation risk [13–16]. By considering mechanical, software and electrical\electronic incompatibility of changed mechatronic systems, an interdisciplinary SysML-based approach, called SysML4*Mechatronics*, for analyzing change influences was proposed by Kernschmidt and Vogel-Heuser [13]. Feldmann et al. [14] formalized the incompatible information of the change influences of mechatronic manufacturing systems by integrating systems modeling with semantic technologies. Kernschmidt et al. [15] extended SysML-based incompatibility checking to the entire system lifecycle. Nejati et al. [16] suggested how to automatically find out the impact of changes to requirements based on a system design expressed by SysML models through computing reachability through inter-block data flow and intra-block data as well as control flow dependencies.

However, as listed in Table 1, change impact analysis of a SysML-based system model, only a few of the works have investigated the change process itself when an engineering change happens, i.e., how the design or system model should be modified, and which model elements should be added or deleted in response to design changes.

Table 1
Change impact analysis of a SysML-based system model.

Approach	In change stage	
	Impact analysis	Implementation
Matrix-based approach [9–12]	Yes	No
Inconsistency and incompatibility contrast approach [13–16]	Yes	No
Analyzing the change workflow [17]	Yes	Partially done
Specifying the change situations of system functions, behaviors, and structures [18]	Yes	Partially done

Two such studies partially involve change implementation [17,18]. Lin et al. [17] proposed a workflow to combine change request management with model-driven engineering for industrial automation software. In the technical review phase, affected model elements of the change requirements were identified, and a change solution was proposed. In the impact analysis phase, processes that guide designers to change SysML models were provided. However, several limitations still exist. First, the method for identifying impacted model elements is only available for restricted situations where the impact of complex changes to requirements, behaviors, and structures cannot be fully represented by SysML cross-cutting relationships alone. In SysML, a cross-cutting relationship is a kind of mapping relationship which can be established between any two model elements. There are two types of mapping relationships, ① the intra-domain relationships, i.e., the links between elements within a given domain, and ② the inter-domain relationships, i.e., the links between elements across two domains [5]. A cross-cutting relationship is the latter, i.e., an inter-domain relationship. Second, besides the existing system model, two more SysML-based models should be created to represent impacted model elements and the corresponding change solution. For a system model with large SysML diagrams, this is obviously time-consuming. Jiang et al. [18] established a function-behavior-structure model of product design history to predict the effect of function change on the final product structures. More importantly, what should be changed during the system design is clearly defined. However, in contrast with SysML, the function-behavior-structure model covers only partial information of a system model. For example, for the structure layer of the function-behavior-structure model, only assembly relationships are described. The items, such as signals, energy or data, which flow across different components, are not specified.

Based on the above analysis, this paper emphasizes the need to improve the SysML model change process itself. For enabling designers to understand which SysML diagrams and model elements should be changed, scenarios for each change type and their change propagation information should be specified and formalized.

2.2. Semantic Web technology

With regard to the many information representation technologies, Semantic Web technology [19] is being accepted by an increasing number of industries due to its strong semantic and logic expression capabilities. Semantic Web technology provides the foundation for the representation of an ontology, which defines specific vocabularies of domain concepts and their semantic relationships [20]. According to Zhang et al. [23], “the World Wide Web Consortium has contributed much towards standardizing the specification necessary for Semantic Web technology” by introducing the Resource Description Framework (RDF) and the RDF Schema [21]. In order to enhance knowledge representation, the World Wide Web Consortium finalized the OWL [22] by extending the expressive power of the Resource Description Framework and the RDF Schema. However, there are many semantic relations that cannot be specified explicitly by the OWL ontology