Computers in Industry 64 (2013) 499–513

Contents lists available at SciVerse ScienceDirect

Computers in Industry

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

An algorithm for transforming design text ROM diagram into FBS model

Min Wang [a], Yong Zeng [a,*], Lei Chen [a], Armin Eberlein [b]

[a] Concordia Institute for Information Systems Engineering, Faculty of Engineering and Computer Science, Concordia University, 1455 de Maisonneuve Blvd. West, EV 07.633, Montreal, Quebec, Canada H3G 1M8
[b] Department of Computer Science & Engineering, American University of Sharjah, Sharjah, United Arab Emirates

ARTICLE INFO

Article history:
Received 23 August 2011
Received in revised form 15 January 2013
Accepted 12 February 2013
Available online 29 March 2013

Keywords:
Natural language
Design text
Function–Behavior–State (FBS) model
Product–Environment System (PES)
Recursive Object Model (ROM)

ABSTRACT

In this paper, a novel algorithm is proposed to transform a ROM diagram obtained from a design text into a FBS model. Each state of the transformation process is defined by four features: ROM (Recursive Object Model), POS (Part of Speech), PES (Product–Environment System), and FBS (Function–Behavior–State). The transformation algorithm is thus constituted by transition rules which change one transformation state to another, and procedures which apply the transition rules to a given ROM diagram. A software prototype R2FBS is presented as a proof of concept to assist the transformation. Two examples are used to demonstrate how the proposed approach works.

1. Introduction

Design information is usually recorded in various design documents, such as customer requirements, informal notes of phone calls, meetings, e-mails and faxes. However, more formal information can be contained in design patents using a legal technical format, or formal design specifications that provide details of the accomplished design. This means that design information can be contained in various representations, such as text, verbal statements, graphic models and mathematical expressions. Research has been conducted in the processing of different kinds of design information, such as geometric modeling of products [1], sketch representations [2], management of design knowledge [3], and product requirements modeling [4–7]. Among all the representations that express design information, natural language is the most flexible yet ambiguous means; graphic models are the most effective and the most efficient; mathematical language is, however, the most precise.

Along with the advancement of computing technologies, more and more design tasks are directly or indirectly supported by computers. Directly, some design tasks are automated such as geometric modeling, structural analysis and optimization. Indirectly, some design tasks are being conducted through collaboration between human and computers such as drafting, innovation, and requirements elicitation. In order to support the entire design process, emerging CAD/E systems must be able to support the smooth integration of systems with other systems and with their human users [8]. The basis for this integration is a semantic model that can accommodate the communication of systems with each other and with their human users [9].

Technologies supporting this communication range from early efforts on geometric reasoning [10,11] to recent progresses on data/knowledge mining [12,13]. Another application in CAD/E that is closely related to semantics is the acquisition of design knowledge from existing design documents. For example, a design patent may contain novel solutions to a design problem. It is useful to extract from the design document critical design information relevant to a new design or redesign. Critical information can only be identified if the semantics of the document is understood. It must also be pointed out that the design document by itself and in its textual form is only a piece of passive knowledge. Its application depends on how the designer understands and digests the active design knowledge implied in the document so that they can be logically associated with a design situation. By definition, the understanding of a document is the transformation of the document into a formal representation constituted by a set of semantic components and their relationships [14].

A lot of research and development have been dedicated to formal and structured modeling as this would help represent semantic information in the design process. For example, in

* Corresponding author. Tel.: +1 514 848 2424x5801; fax: +1 514 848 3171.
E-mail addresses: zeng@ciise.concordia.ca, yong.zeng@concordia.ca (Y. Zeng).

0166-3615/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.compind.2013.02.007

software engineering, Universal Modeling Language (UML) [15–17] is a widely adopted software modeling notation to specify, construct and document the artifacts [18]. A few semantic approaches have been developed to process class diagrams [19], state machines [20], interactions [21], use cases [19], OCL [22], activity diagrams [23], and so on. However, UML is a software-specific language, and does not support the general needs of design in other domains. Therefore, the system modeling language OMG SysML [24,25] was created and has been steadily gaining popularity in different domains [26]. SysML is used to specify, analyze, and design systems that may include hardware, software, and personnel. It allows engineers to describe how a system interacts with its environment, and how its parts must interact to achieve the desired system behavior and performance. The SysML model provides a shared view of the system, enabling a design team to detect issues early and prevent problems that would otherwise delay development and degrade design quality. Since SysML is based on UML, it also facilitates integration between systems and software development.

In the design of engineering systems, especially mechanical and architectural systems, function is recognized as the bridge between human desires and physical behavior of artifacts. Among various function-based models [27–31], the Function–Behavior–State (FBS) model was proposed by Umeda and Tomiyama as a framework to represent a design object hierarchically and to define a function as an association of human intention and behavior [27]. The FBS model has drawn a lot of attention in design research as it provides a knowledge representational scheme for conceptual design [27], and for the knowledge intensive engineering framework [32]. Erden et al. based on a review of various Function Modeling (FM) approaches in the fields of artificial intelligence, design theory, and maintenance, propose a general framework of FM [30]. This general Function Modeling framework extends the FBS model to function–behavior–physical phenomenon–state model (FBPhPhS) by combining the FBS model with a qualitative reasoning system (QRS) [30].

FBS modeling requires that its users understand the product requirements thoroughly and distinguish different functional stages and relationships between the functions. This could be a challenging task for a complex engineering project. The design document for a complex engineering product or process may include a large amount of information, which is tedious for human processing. A computer-aided FBS modeling system is indispensable in such a case. To support the application of the FBS modeling theory, a software tool – the FBS modeler – was developed to support the conceptual design. The FBS modeler provides a function decomposition method, which includes causal and task decompositions [27]. The decomposition process largely depends on the designer's knowledge and experience with the FBS theory. The purpose of this paper is to generate the FBS model from an important kind of design document – design text in natural language by developing a generic experience-independent method.

In order to generate the FBS model from a design text, the following research issues have to be investigated:

- How to capture the meaning of a text?
- How to generate the function representation scheme from the captured meaning of the text during the FBS modeling?
- How to construct the right FBS model by simulating the function recognition processes?

To the authors' best knowledge, no research results have been reported in the literature for transforming design text into FBS model. However, efforts have been made to develop automatic or semi-automatic processes to bridge unrestricted natural language text and conceptual models. Tjoa and Berger proposed an approach to transforming natural language based requirements specifications into an EER model [33]. Mala and Uma present an approach to extracting the object-oriented elements of the required system [34]. Gnesi et al. developed an automatic tool for the analysis of natural language requirements [35]. Liu et al. proposed a methodology with Use-case language schemas to automate natural language requirements analysis and class model generation based on the Rational Unified Process (RUP) [36]. Due to the difficulties in natural language processing [37] and the huge gap between natural language and structured models [35,38,39], those efforts have achieved very limited success.

This paper attempts to deal with the gap between natural language and structured conceptual model through an intermediate model – Recursive Object Model (ROM) [11], which captures the semantic information of the concerned natural language. Since the application area is engineering design, a FBS model is used as our target conceptual model. Through study and practice in engineering, the ROM based transformation can help to extract system dynamics during the earlier design stage [40] and facilitate the general modeling process [41] and specific design methods such as TRIZ [42]. The proposed approach first generates the ROM diagram of a design text describing, for example, a part of product requirements or design patent descriptions. Then the key elements included in the text, such as product components, product environment, and relations between them are extracted based on predefined rules. Finally, the key elements are transformed into a FBS model. In this paper, we will focus on the transformation from a design text ROM diagram to a FBS model.

The rest of this paper is organized as follows: the next section will introduce the algorithm in terms of input, output and the transformation of the input to the output. Then the algorithm for the generation of FBS model from a design text ROM diagram is described. Two examples are presented to demonstrate how the algorithm proposed in this paper works. The last section gives conclusions and points out future directions.

2. Algorithm structure

According to the traditional understanding, an algorithm is a finite, unambiguous description of an effective procedure for the solution of a class of problems. The procedure in an algorithm is often called a transformation. A transformation is defined by a set of transitions which deal with all the possible cases included in the class of problems for which the algorithm was designed [43].

Fig. 1 shows the transformation process from a design text to a FBS model. This process can be divided into two sub-processes: first, the design text described in natural language will go through a linguistic analysis process using the computer tool ROMA, which generates a ROM diagram for the design text; then, another transformation process is needed to transform the ROM diagram into a FBS model. Since the first process has been dealt with elsewhere [11], this paper focuses on the second process. Therefore, the input of this transformation is the ROM diagram corresponding to a design text whereas the output is a FBS model.

In the following sections, the input and output of transformation from ROM to FBS are addressed. The relations between the input and output are analyzed, and accordingly the transition rules are derived. At last, the algorithms are described. The foundation of this discussion is Axiomatic Theory of Design Modeling (ATDM) [44].

2.1. Axiomatic Theory of Design Modeling (ATDM): an introduction

The Axiomatic Theory of Design Modeling (ATDM) is developed to represent structures of design, especially the conceptual design.