Computers in Industry 74 (2015) 58–70

Contents lists available at ScienceDirect

Computers in Industry

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

Formal computer-aided product family architecture design for mass customization

Martin Bonev [a,*], Lars Hvam [a], John Clarkson [b], Anja Maier [a]

[a] Department of Management Engineering, Technical University of Denmark, Produktionstorvet, Build. 424, 2800 Kgs., Lyngby, Denmark
[b] Engineering Design Centre, Department of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, UK

ARTICLE INFO

Article history:
Received 3 March 2015
Received in revised form 24 June 2015
Accepted 14 July 2015
Available online 9 October 2015

Keywords:
Product family architecture
Design synthesis
Mass customization
Structural analysis
Generic modelling

ABSTRACT

With product customization companies aim at creating higher customer value and stronger economic benefits. The profitability of the offered variety relies on the quality of the developed product family architectures and their consistent implementation in configuration systems. Yet existing methods are informal, providing limited support for domain experts to communicate, synthesize and document architectures effectively. In single product design explicit visual models such as design structure matrices and node-link diagrams have been used in combination with structural analysis methods to overcome the limitation of the informal approach. Drawing on thereto established best practises, this paper evaluates and extends the relevant methods and modelling techniques, to create a consistent and formal approach for the design and customization of entire product families. To validate it's applicability, the approach is tested on a case study at a manufacturing company offering bespoke industrial applications. A generic modelling method termed the *integrated design model* (IDM) is developed and complemented with a computational structural analysis method, to assist domain experts in their daily work. When combined with a configuration system, the presented IDM tool automates the documentation and formalizes the synthesis of architectures, thereby making any decision about a preferred solution explicit and transparent.

1. Introduction

A growing demand towards higher product variety has been reported in many industries [1,2]. Acting upon this trend, companies aim at obtaining higher customer value and stronger economic benefits through rapidly responding to individual needs for customization [3]. Nonetheless, high and diverse product mixes are not always beneficial but often challenge manufacturers with a related increase in operational complexity and decrease in efficiency in sales, design, production and distribution [4]. Platforms and modules built into product family architectures have been reported to facilitate this trade-off [5]. In this context, architectures are defined as an abstract structural representation of the functional units and the corresponding physical components of engineering artefacts [6]. Their development is complex and long lasting and their performance can have wide-ranging effects on the success of manufacturers [7]. Designing architectures suitable for customization raises additional difficulties to organizations, since the right product composition and part compatibility needs to be ensured. With product configuration systems or configurators, manufacturers are able to handle these demanding requirements for information processing, storage and retrieval of feasible variant combinations [3].

Configurators are software-based expert systems that capture the generic architecture of product families in a computer model, through which users are supported in creating feasible product solutions with a minimum number of choices [8]. If combined with well-designed product family architectures, companies can utilize configurators to mass customize their offerings, i.e. to automate operational activities related to product customization and to increase their efficiency to a level which is close to mass production [9]. However, architectures per se are qualitative and current methods supporting their design and documentation are informal and limited [10,11]. Hence, it can be difficult to identify 'good' architectures during product design and to sustain their subsequent implementation in a configuration system. At the same time, configuration software vendors are of no help in this respect, as they are typically not interested in providing a transparent and easy way to create and communicate the

* Corresponding author. Tel.: +45 45256084.
E-mail addresses: mbon@dtu.dk (M. Bonev), lahv@dtu.dk (L. Hvam), pjc10@eng.cam.ac.uk (J. Clarkson), amai@dtu.dk (A. Maier).

http://dx.doi.org/10.1016/j.compind.2015.07.006
0166-3615/

architectures, but rather emphasize consulting services around their modelling and maintenance [12]. Consequently, with the development progress of product families, software experts have problems in keeping an overview of what had been implemented in the computer model and verifying the obtained architectures with domain experts, making it one of the main reasons why designing and mass customizing products is still difficult to achieve [13].

This paper presents a formal computer-assisted approach that addresses the requirements for the design of product family architecture as identified by academia and industry. Section 2 first discusses existing approaches from both engineering and software domains, to define a consistent architecture design framework. Next, the challenges with conventional informal approaches are discussed and requirements for a formal support method are developed. Section 3 elaborates and further extends existing modelling techniques and relevant formal architecture support measures. In Section 4 a formal approach is presented and complemented with a case study of a major plant and machinery provider of highly customizable products, to develop a concrete example on a real world problem. The introduced approach combines the capabilities of the utilized configurator with automatically generated grammar graphs representing the implemented architectures. The graphs are modelled with an integrated design model (IDM), using the suggested extended modelling techniques for generic structures. A developed IDM tool is further employed to assist domain experts in synthesising feasible architectures and to evaluate their structural characteristics computationally through a series of metrics, potentially leading to better solutions. Finally, Section 6 concludes with an assessment of the proposed approach.

2. Relevant literature

To evaluate the limitations of existing approaches to architecture design for mass customization, a literature review on relevant frameworks, methods and modelling techniques is performed and requirements for a formal method are developed.

2.1. Approaches to architecture design in engineering and software development

The design of architectures and their subsequent implementation in configurators involves domain experts from different departments and often physically disconnected teams. Several researchers have acknowledged the related organizational challenges and have proposed methods on how to arrange corresponding activities in a more systematic manner [14–16]. In engineering domains Pahl et al. [17] address architecture design on several stages, from formulating customer needs to the construction of embodiment and detailed design. Corresponding to these different phases of development, Jiao et al. [18] argue for an architecture modelling framework which in addition considers several views of a product. At the same time frameworks dealing with architecture design for expert systems typically fall within the area of software systems and base their methods on the life-cycle of object-oriented software development as introduced by Booch [19]. Booch's object-oriented procedure was originally developed to handle the complexity of large software projects by breaking down the development work into phases of object-oriented analysis, design, implementation and maintenance [19]. The transformation from a real world design into a computer model is organized in several steps, where the observed reality is gradually abstracted and formalized [20]. To enable the representation of a large number of physical artefacts with components and variant combinations, related frameworks commonly build upon methods for modelling software architectures using the unified modelling language (UML) [21].

Although the UML standard proved to be particular useful for defining entire product families, its application within engineering design remains limited. In consequence, synergies on coinciding aspects of architecture design are seldom achieved. For example, the challenge of modelling different architecture views has been repeatedly addressed within the two domains and has resulted in comparable outcomes [18,22,23]. Moreover, advancements within engineering design are seldom adopted to software design and vice versa, in particular with regard to the formal computational management of structural properties in complex architectures [24]. Second, the development of a product family architecture for expert systems is often organized within IT and product data management departments. The process is regarded as a liberally new modelling approach which is detached from any preceding design activities of the product development phase [25]. This means that in praxis the design of architectures is not coordinated across the organization, leading to computer models which are very likely to differ from the original design intent of the engineers [13]. Especially for more complex products, this lack of consistency increases the risk of providing undesired product variety to the market [26]. As a benchmark report with more than 300 manufacturers of custom tailored products reveals, the top performing companies with engineering intensive portfolios try to overcome this coordination burden by better involving development engineers in the architecture design process for their configuration systems [27]. This suggests that a more integrated approach to mass customization is needed, which considers equally both the architecture design process and the subsequent implementation into configuration systems.

2.2. Challenges with conventional informal architecture design methods

Fig. 1 illustrates how a consistent framework of the architecture design process and its transformation into a computer model may be organized. The model is based on the Wyatt et al. [11] generic scheme for architecture design in engineering, and combines this with the discussed transformation into a computer model. The focus of this paper is indicated by the grey area in the model, where design aspects from engineering are incorporated with the software domain of configurators. The procedure is initiated by a design or handling problem and ends with a customized solution created by the user of a configuration system. As expressed in the model, supporting methods can be informal, relying on subjective interpretations of domain experts, or formal, involving codable and systematic procedures. The two alternative approaches may be organized along a five phase model of exploration, generation, evaluation, implementation and communication, which is based on the established development model of design science [28].

In analogy to Wyatt et al.'s [11] architecture design framework, the informal approach can be described as follows:

- *Exploration* helps engineers to examine the handling of existing design or the work on a new design problem. Typically, product information can exist in many different formats, such as diagrams, tables, formulas, computer aided design (CAD) files, bills of materials (BOMs) etc. Different departments within a company may even have their own representations of products. By *abstracting* the relevant product information (1), engineers develop an understanding of possible architectures (2).
- Based on a created understanding of possible architectures, engineers *generate* a specific family architecture in the form of an analysis model, which may be the same as previous solutions and further contain errors (3). Discussions on the product architecture