Mechatronics 21 (2011) 534–547

Contents lists available at ScienceDirect

Mechatronics

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

An architecture model to support cooperative design for mechatronic products: A control design case

A.A. Alvarez Cabrera [a,*], K. Woestenenk [b], T. Tomiyama [a]

[a] Faculty of Mechanical, Maritime, and Materials Engineering, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands
[b] Faculty of Engineering Technology, University of Twente, PO Box 217, 7500 AE Enschede, The Netherlands

ARTICLE INFO

Article history:
Received 18 August 2010
Accepted 26 January 2011
Available online 17 February 2011

Keywords:
Design methodology
Mechatronics
Modeling
Systems engineering
Control systems
Design automation

ABSTRACT

Efficient integration of systems in the mechatronic industry is critical for complex product development and is still challenging. A particular example of this situation can be pinpointed in the development of control software for mechatronic products: design is not carried out in a concurrent way in order to exploit the synergy among domain experts and many "last minute" problems are detected and forcefully solved in the control software domain at an advanced development stage. Unfortunately, industrially applicable research to improve integration in the development process is currently at a stale. This work addresses system architecting introducing a model, a method, and a tool implementation, which aim to help changing this situation by supporting cooperative design, providing usable documentation and improving understanding of the design process by the stakeholders.

1. Introduction

Many design scenarios could benefit from additional support for integration. The control software design/generation scenario is used here to test our proposal. This choice is based on the fact that, probably contrary to the beliefs of some readers, the control software design process entails intensive interaction among design disciplines and is inherently multidisciplinary [1]. The choice also seeks to address part of a current problem in the mechatronic industry: design is not carried out in a concurrent way to exploit the synergy among domain experts and many problems are detected late and forcefully solved in the control software domain at an advanced development stage. These practices compromise the quality of the resulting software and the product. Additionally, model-driven design/engineering methods stimulate designers to formalize models for their specific domains but do not help specifying a common factor among the models which facilitates use across disciplines, besides a common ontology. With the results of this work, we expect to demonstrate how to empower the (control) engineer with a model that can:

- Facilitate him gathering the information from multiple sources to do his job.
- Allow exposing his concerns so he can influence designs to be performed by experts of other disciplines.

Many problems originate at the conceptual design phase, e.g., conceptual solutions from different disciplines are not shared/understood because of lack of flexible common models (see Section 2). What can be done to step towards cooperation and concurrency in (controller) design? Tools and methods to support conceptual design and information exchange at that level are rather scarce and still have to overcome important shortcomings, like the dissociation of information from different sources [3]. The hypothesis handled in this paper is that representing design information lies at the bottom of such issues, and that using the system architecture is the key. This hypothesis responds to the needs rising from the mechatronic industry [4,5], more specifically because of the predominance of (bulky and unstructured) textual information and of models which are not easy to understand and transfer outside their domain or specialty.

The "V" development cycle [2] (see Fig. 1) and other similar methods are intended to guide the design process systematically, but not many tools support their usage, as discussed in previous work of the authors [3]. We propose to use the system architecture to support such methods by providing:

- A base to document the decomposition phase.
- A formalization to capture design interfaces necessary for the integration phase and its analysis.
- A mechanism to trace the effects of requirements to the designed implementations, and vice versa for verification.

This work contains a representation proposal to capture design information at the architecture-level (cf. Section 3). Therefore, the

* Corresponding author. Tel.: +31 15 278 5608; Fax: +31 15 278 4717.
E-mail address: a.a.alvarezcabrera@tudelft.nl (A.A. Alvarez Cabrera).

0957-4158/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mechatronics.2011.01.009