

Holistic Change Management in Automotive Electrical/Electronic Development using an Integrated Engineering Product and Process Model

Daniel Watzenig and Florian Pölzlbauer
The Virtual Vehicle Competence Center (ViF)

Copyright © 2008 SAE International

ABSTRACT

The systematic networking of engineering product and its underlying dynamic development process forms the solid basis of cross-domain change management and product/process optimization at any time during the development. In this article a model-based approach to efficiently integrate engineering product and process during passive safety system development within the vehicle electric/electronic (E/E) domain is proposed. The relevant parts of the system are modeled on their functional level while the corresponding development process is represented by taking into account the information flow between process steps as well as organization structures. By linking both domains, i.e. both models, given their interfaces and dependencies on different abstraction levels, the impacts of product/process changes on the process/product during all design stages are evaluated in either direction.

INTRODUCTION

The automotive industry has encountered significant changes within its competitive environment over the last decade. Automotive OEMs are forced to constantly introduce innovative features and additional functions within their engineering products in order to satisfy customers needs. As a consequence, electrical/electronic (E/E) development has become a key process enabling OEMs to address the required demands. To face this new challenge, the automotive industry has started to gradually introduce development processes for E/E systems. However, current approaches are component-oriented rather than system-oriented meaning that the engineering product is considered completely decoupled from the underlying engineering process in terms of data and information exchange. In the last years a shift towards model-based

product and process development as well as their coupling can be observed. Many different approaches have been investigated and presented such as feature-oriented models, UML models, modular systems and knowledge-based models [1, 3, 4, 7]. However, these approaches suffer from two disadvantages: the do not focus on coupling on all levels of abstraction between the product and the process domain, and the do not support dependency analysis and tracking during the entire development process cycle from the specification phase to the rollout phase. A first global approach to the integration of product and process models (IPPM) can be found in [5]. Within this work, the development of a loader following the IPPM methodology is addressed. By evaluating the development time and cost this approach outperforms the traditional development strategy. The impacts of changes on the product, process, requirements and/or design constraints have not been tackled so far.

In this article main focus is put on the holistic change management and requirements handling in automotive E/E development following the IPPM approach. The E/E system (or parts of it), which is referred to as *engineering product*, is modeled on its functional level. This model can be simulated using standard simulation environments such as MATLAB®/Simulink®. Similar to the product model, the entire E/E development process is modeled by taking into account tasks, tools, human resources, organization structures and the information flow between process steps. Due to the proposed integration of both models on different levels of abstraction possible changes of the product and the process can be tracked at any time during the development. Based on the coherent, consistent and complete functional model developed within the CAPEmaster framework [6], fault clearances and the incorporation of undesirable but inevitable change requests near start of production can be easily managed. The engineering product and process integration for efficient change management will be discussed in detail for a

change request in the late phase for the vehicle airbag control system. Due to the prediction of change impacts on all affected technical parts and organizational units, the responsible management has a solid basis for decision making. Furthermore, the integrated product process model features variant management. By adding, removing or distributing certain functions as it will be demonstrated for the side-airbag device control, the feasibility of different variants can be investigated and their impact on the underlying development process can be determined.

INTEGRATED ENGINEERING PRODUCT AND PROCESS MODEL

The two basic aspects in the model-based development of an engineering product are the modeling of the product itself and the modeling of the accompanying development process. Due to the steadily increasing complexity of products, especially in the E/E domain in the automotive field, interdisciplinary teams have to be concurrently managed in order to optimize the design, implementation and testing phases. Based on a consistent and coherent model capable of representing both product and process, the interrelations and interactions can be investigated and tracked at any stage between specification and maintenance. From the perspective of static and dynamic model behavior, the coupling of the two models – the *model integration* – can be realized in three different ways: static product model and dynamic process view, static process and dynamic product model, and both models are mutual and dynamic [3]. A linking of process and product is essential in product development in order to manage product changes, product variants and requirements that may change during development. Model integration allows to propagate information in either direction between process and product implying conflict forecast, fault tracking and global optimization procedures.

Within this work, the engineering product model forms the kernel of product development, the dynamic process model is continuously adapted in terms of consistency and completeness. In the following the modeling procedure for both product and engineering process are discussed by example using the CAPEmaster approach [2, 6].

THE PROCESS MODEL Fig. 1 illustrates the typical V-model that describes the relationship between customer requirements, product properties, functional/technical requirements, architecture specification, HW/SW implementation and component/module/system test on the complete product level, module level and component level. The underlying engineering process that covers the process steps from the product idea to the start of production (SOP) is represented by the process model. More precisely, the engineering process usually consists of market analysis, product definition & specification, product development, manufacturing, release, rollout, SOP and maintenance. Following the model-based approach, the entire engineering development process is based on a

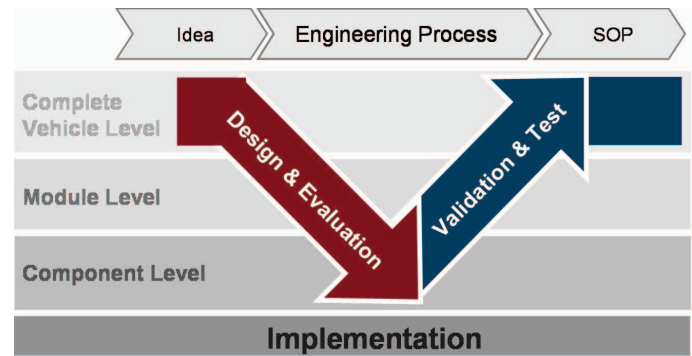


Figure 1: V-model covering the product development from the idea to start of production (SOP). The left branch reflects the phases from definition, requirements and concept down to the implementation level. The right branch denotes the validation and test on component, module and system level.

process reference model. This model is covered by three different levels, the *process level*, the *task level* and the *action level*. Within the process level, the information flow, the workflow and the material flow between different process steps is described. The flow is represented by inputs and outputs of a process step and the connection between them.

The process steps of the E/E development process are successively refined ending up in basic process steps that are described by basic process elements or basic process types. Within this framework, the basic types are the task types (attributes are man hours, period to complete a task), the tool types (attributes are licenses and software platforms to complete a task) and the human resources represented by engineers, technicians, merchants and project managers required to complete a task. On the task level the connection to the product level is established in terms of engineering objects and associated attributes. The third level of abstraction of the process reference model – the *action level* – represents the company organization structure of the companies involved in the development process of the E/E system. Typical elements are companies, departments, groups, teams, employees, product data management (PDM) systems or the technical infrastructure.

THE PRODUCT MODEL Within the last decade a paradigm shift from application-driven ECU-based development to development of reusable functionalities has taken place. Typically, the functionality of ECUs is defined by the OEMs in requirements documents, while the implementation is subsequently performed by the suppliers. Beside the fact that this document-based development process often leads to misunderstandings resulting in a reduction of product quality and increase of development time, the resultant solutions are technology-dependent. This counteracts the OEM desire of reusability due to permanent technology advancements. In order to overcome

this drawback, several organizations started to establish methods in order to increase the reusability of automotive electronics. One typical example is the AUTOSAR® standard. Functional engineering introduced to automotive industry in the last decade features the development of a library of functional solutions, reusability, separation of functions and technical implementation and supports consistency throughout the entire model. In this context, the engineering product is modeled at different levels of abstraction, each fitting the requirements of the actual development step [6]:

- functional hierarchy
- virtual solution design (VSD)
- technical architecture (ECUs and bus systems)
- topology level (devices, cables, routing,...)

The main idea behind this approach is to decouple the functional description of the engineering product from the technology of implementation. This approach allows to reuse both, the product functions as well as the technology, without affecting each other.

Functional hierarchy After defining the set of product requirements, the entire product is described by its functionalities. Each function of the system can be modeled in terms of its objective (What will this function do?) and its interface, which shows the need of information in order to fulfill this objective (What information does this function need? What information does it provide?). In order to support reusability, the functional model exhibits a hierarchical structure. This is particularly useful when highly networked systems such as automotive control systems have to be managed. Fig. 2 shows the functional model

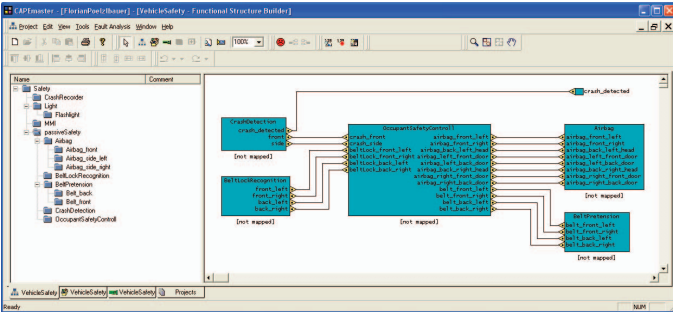


Figure 2: Functional model of the passive safety system, tree-view (left window) network-view (right window)

of the passive safety system of a middle-class car, created with the engineering platform CAPEmaster [6].

Based on this complete and coherent model, inevitable inconsistencies (e.g. mismatch of data types for data exchange) that may occur since a lot of engineers work on the functional model at the same time, can be immediately

identified. Furthermore, the functional model can be evaluated in terms of error path analysis in order to investigate conceptional errors [2], as shown in fig. 3. Since the func-

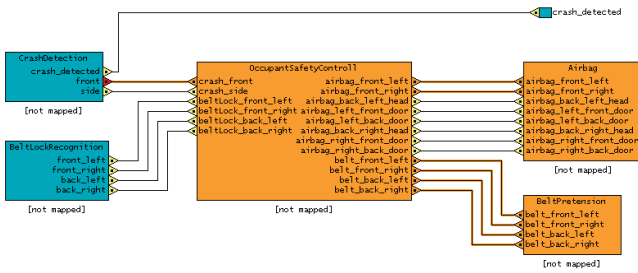


Figure 3: Error path analysis for loss of information "front crash", based on the functional model

tional model is independent of any technology, it can be reused in other projects by creating libraries. This method leads to faster product definition, lower development costs and at the same time increases product quality due to use of already proven elements.

Virtual solution design (VSD) The VSD views the electronic control system as a process chain consisting of sensors, actuators, analog/digital converters (ADC), digital/analog converters (DAC) and software elements. Fig. 4

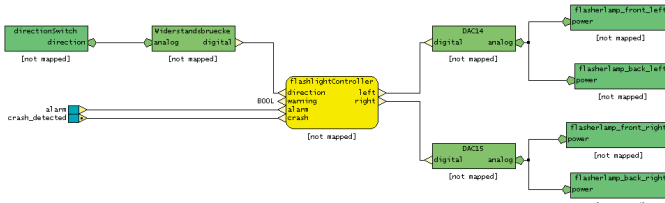


Figure 4: Virtual solution of indicator lights

illustrates the virtual solution design for the flash lights function of a car. The desired direction is measured by the indicator switch. The electrical signal is subsequently processed by hardware channels that will evaluate the electrical encoded direction and transform it into software variables. Based on that information (and additional information like "alarm on" or "crash detected") the software unit will control the hardware channels that will drive the flash light bulbs.

The VSD is based on three basic elements that are able to model every electronic control system regardless of its complexity.

- process type (e.g. sensors and actuators): bidirectional transformation between physical quantities (like