NordDesign 2012
August 22 – 24, 2012
Aalborg, Denmark

An Architecture Framework for Multi-Product Portfolio Management in the Commercial Vehicle Industry

Simon Plaikner
Institute of Product Development, TU München

Maximilian Kissel
Institute of Product Development, TU München
kissel@pe.mw.tum.de

Matthias Kreimeyer
Product Architecture, MAN Truck & Bus AG

Udo Lindemann
Institute of Product Development, TU München

Abstract
In the field of commercial vehicle manufacturing, the functionality and complexity of products rises steadily while innovation cycles become considerably shorter. In addition, customer orientation causes rising external complexity of product portfolios. Without the moderating and managing role of an architecture department within a corporation these diversified markets drive a multi-product development towards increased inner complexity which cannot be handled efficiently anymore.
In this paper, we address the role, necessary tasks and methodologies of a product architecture department at a commercial vehicle manufacturer. A so called Architecture Framework is presented that helps to cope with the organizational complexity in the field of product development by structuring work processes and assigned tools. For that, we reflected the requirements of a commercial vehicle manufacturer towards an Architecture Framework to existing solutions in literature. The advantages of each concept were consolidated. The suggested framework was prototypically tested in an industrial environment.

Keywords: Product Architecture Design, Design Process, Architecture Framework for Truck Industry

Introduction
In automotive industry, complexity of products and processes rises amongst others through interweaving of different disciplines in cross-company processes, individualization of products, and rising demand towards new functionality. While more and more subsystems are outsourced to external specialists, the integrating role of a product architecture department gains in importance for the competitiveness of a company. In order to provide an attractive product portfolio ("high external complexity") the internal complexity [10] must be mastered efficiently. A product architecture department should make relevant dependencies in these complex networks transparent, align the processes of an organization, and manage the development and the changes of variants, modules, and interfaces.
Commercial vehicles are commonly based on a highly modular architecture that supports a wide range of applications, e.g., trucks for different uses (e.g., wood transport, military, etc.) and market segments (such as long-haul, distribution or traction). This is especially done so as almost all commercial vehicle manufacturers offer a wide variant portfolio with limited sales volumes, especially in comparison to passenger vehicle industry.

To systematically foster this methodology, the methodical design of product architectures is an important enabler. However, with the growth and internationalization of a company, this can be less and less handled by the individual engineers but needs to be more and more run by a central organization within the company. At the commercial vehicle manufacturer considered in this study, therefore, the product architecture department was put in place to handle a centralized variant management with the goal to develop and document only those parts that are, in fact, needed from a customer’s point of view. To this end, new processes to support the early phase are being introduced, namely a specification phase to generate a set of product specifications in a formal manner, a product architecture phase to translate the functional specifications into a consistent vehicle architecture, and a package phase to ensure the product architecture is collision-free for the intended variant models.

As these tasks, in such a central format, are new to the company, roles, responsibilities, tasks, methods, tools and such are not initially defined; at the same time, the high complexity of “product architecture” as such is hard to handle, as it relates to many already existing processes and organizational units within the company.

The idea presented in this paper is to develop a so called Architecture Framework which aims to support the work of a product architecture department. The framework should be designed to cope with complexity in the field of product development by structuring work processes and assigned tools. It aims to ease the product developers' work and to guide the workflow in an efficient way.

Frameworks are common practice, mainly in fields of research which have to deal with complex problems. Thereby, frameworks are able to reduce the complexity of a problem by providing or creating self-complexity which Weber defines as “structuring of a task by reducing the number of possible solutions” [12]. With a framework that provides the right amount of self-complexity, complex problems become solvable without restraining the organizational process of solution finding.

There are extensive suggestions in literature how to design a framework which describes these aspects (e.g. [1], [3], [5], [7], [11]). Matthes [7] by himself identifies over seventy such frameworks. However, all the approaches we found could not be transferred one to one to the special situation and background of a commercial vehicle manufacturer. The central question of our work is therefore:

How can an architecture framework be designed to efficiently fit the needs of a commercial vehicle manufacturer with a highly variant product portfolio in terms of defining the major task, roles, methods, and responsibilities?

In the following, the consolidation of the Architecture Framework as well as the framework itself, filled with the information collected in an industrial environment, will be presented exemplarily. Afterwards, it will be discussed which limitations and problems occurred during the practical realization of the theoretical results.

Consolidation of the Architecture Framework

We developed the Architecture Framework in collaboration with the department of Product Architecture at a commercial vehicle manufacturer over a period of four months. Consequently, the framework was, for the best part, derived from the needs and requirements which arise from the product architecture within the commercial vehicle industry. Additionally, the presented system was derived from state-of-the-art frameworks of other fields of research that have to deal with the problem of complexity, e.g. software engineering and business informatics. Figure 1 demonstrates the applied method in developing the framework.

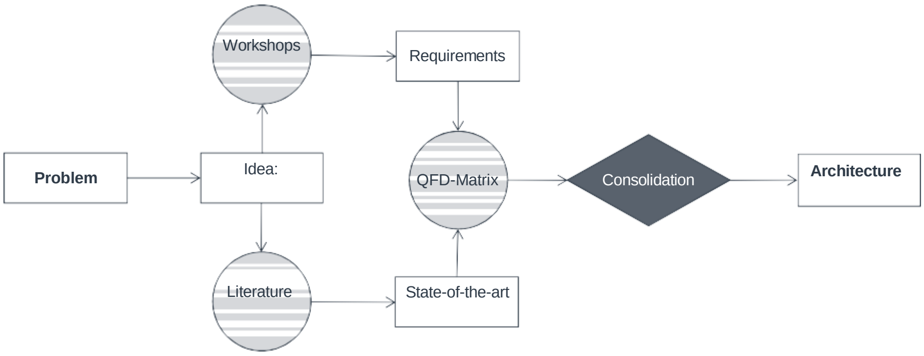

Figure 1: Method of consolidation

Hereby, the search for the framework requirements on the one hand and the reviewing of state-of-the-art solutions on the other hand were carried out simultaneously. Derived from the initial situation at the company, the requirements towards the framework were collected in collaboration with experts of the Product Architecture Department, of departments that collaborate with them (e.g. Requirements Management, Packaging, and Engineering) as well as external experts from process consultancies. The relevant information was assembled and concretized in various workshops and semi-structured interviews together with individuals as well as in small groups. We started out with a series of three workshops, each with an attendance of four to six members, in order to initially identify the product architecture's requirements towards an Architecture Framework. Thereby, we used well-known methods for information acquisition, such as brainstorming and lead questions, in order to guide the discussion and to gather the relevant information in a quick and efficient way. The expertise of external consultants helped to determine the most important requirements and to sort out those that are unrealistic or likely to go beyond the scope of our project. Afterwards, we structured the collected requirements towards the framework thematically and validated them in individual interviews with the members of the Product Architecture Department. The product architects were also asked to rank the obtained requirements in order of importance for their own work. At this opportunity, some of the requirements were once again adapted or stated more precisely. The result was a set of different requirements, ranked according to their importance for the architectural work as shown in table 1.

Table 1: Requirements towards an Architecture Framework

Rank	Requirement	Description	
1	**Process mapping**	The different process steps of the architectural work have to be described in adequate detail.	REQ1
2	**Classification of phases**	The architectural work can be subdivided in structured process phases and assigned to the Product Development Process.	REQ2
3	**Appellation of topics**	The central topics, questions and tasks of the architecture process are stated.	REQ3
4	**Purpose of presentation**	Role and central topics of the Product Architecture Department can be presented in a clear and comprehensible way.	REQ4
5	**Tools and methods**	Tools and methods assigned to the process steps can be linked directly to the framework.	REQ5
6	**Flexibility**	Different amounts of information can be provided.	REQ6
7	**Adaptability**	The openness and receptiveness of the framework allows adjustments and enhancements at any time.	REQ7
8	**Neutrality**	The framework can be applied in different contexts.	REQ8
9	**Process dynamics**	Process-related dynamics such as iterations, alternative solutions and cyclic improvements can be considered.	REQ9
10	**Interfaces/stakeholders**	Interfaces and stakeholders of the process steps are named and described.	REQ10
11	**Input/output**	Input and output contents of the process steps are specified.	REQ11
12	**Responsibilities**	The responsibilities for each architectural process step can be stated.	REQ12
13	**Process maturity**	The framework allows the rating of the process maturity for each process step.	REQ13

Subsequently, these resulting requirements were discussed and validated in an extended circle of internal and external experts.

At the same time, state-of-the-art solutions of frameworks from different fields of research were reviewed in literature. The most promising ones, in terms of supporting the architectural work, were then chosen. Furthermore, for the selection of the frameworks, we tried to gain a large variety of application fields in order to get different concepts and ideas. We also attempted to collect a selection of well-known and already established frameworks. In the end, eight different frameworks were condensed for further analysis. The Zachman Framework [13] and the Open Group Architecture Framework (TOGAF) [11] are well-recognized frameworks in the field of enterprise architecture, whereas the Rational Unified Process (RUP) [2], the SiFrame [1] as well as the V-Model [8] originate from the field of software engineering. Also ArchiMate [7] is commonly used in enterprise architecture, which provides a well-known language for process modeling, and the European Interoperability Framework (EIF) [3]. In the scope of business informatics, the BIVM [5] offers an interesting approach that helps to cope with the challenges of a complex business intelligence problem.

After the selection of the above state-of-the-art frameworks, the existing solutions were reflected upon, based on the specific requirements towards the Architecture Framework. For that, we used a QFD-matrix, which is shown in figure 2.

Factor of importance	Requirements	Zachman (Zachman 1987)	ArchiMate (Matthes 2011)	EIF (Europäische Kommission 2004)	TOGAF (van Haren 2007)	SiFrame (Bindbeutel 1998)	V-Modell (Pomberger & Pree 2004)	RUP (Bunse & Knethen 2002)	BIVM (König 2009)
3	**Process mapping**	none	strong	none	weak	medium	medium	medium	none
3	**Classification of phases**	none	none	none	medium	none	none	strong	strong
3	**Appelation of topics**	weak	weak	medium	none	none	none	strong	strong
3	**Purpose of presentation**	strong	weak	strong	weak	weak	medium	medium	medium
3	**Tools and methods**	medium	medium	none	none	strong	strong	medium	strong
3	**Flexibility**	weak	weak	medium	strong	strong	strong	strong	strong
3	**Adaptability**	strong	weak	medium	weak	weak	medium	medium	strong
2	**Neutrality**	strong	strong	strong	strong	strong	strong	strong	strong
2	**Process dynamics**	none	strong	none	medium	medium	medium	medium	medium
1	**Interfaces/stakeholders**	none	weak	none	none	weak	medium	weak	strong
1	**Input/output**	none	medium	none	medium	medium	weak	weak	strong
1	**Responsibilities**	none	strong	none	none	medium	strong	strong	strong
1	**Process maturity**	none	weak	none	weak	weak	weak	weak	medium

	Zachman	ArchiMate	EIF	TOGAF	SiFrame	V-Modell	RUP	BIVM
Weighted importance	87	98	72	73	101	119	153	198
Relative importance	0,44	0,49	0,36	0,37	0,51	**0,60**	**0,77**	**1,00**

Correlation	Weight
none	0
weak	1
medium	3
strong	9

Figure 2: QFD-Matrix mapping the requirements and the state-of-the-art solutions from literature research

The matrix shows the correlations between the requirements the framework has to fulfil and the reviewed state-of-the-art solutions. Therefore, we differentiated between a weak, a medium, a strong and no correlation, depending on to what extent a requirement is fulfilled by the respective state-of-the-art solution. Each correlation is assigned to a weight between zero and nine (s. figure 2). For example, the requirement concerning the integration or linking of tools and methods in the framework (s. figure 2) is fully implemented in the SiFrame [1], the V-Model [8], and the BIVM [5] where tools, templates and method descriptions are provided as fix parts of the frameworks. The V-Model, for instance, defines two distinct layers for supporting methods and tools. Consequently, we rated the correlation between this specific requirement and the three mentioned frameworks as strong. Also ArchiMate [7], the RUP [2] and the Zachman Framework [13], where the cells of its grid can contain links to tools and methods, fulfil this requirement adequately and achieve the evaluation “medium”. Only the EIF [3] and the TOGAF [11] do not provide sufficient support in this matter and are therefore rated with a correlation weight of zero.

Furthermore, we gave a factor of importance between one and three for each requirement ranked in order to consider their different significance in supporting the architectural work and based on the results gained in the above described interviews with the product architects at the commercial vehicle manufacturer.

As a result, each framework achieves a weighted importance between 0 and 261 which represents the sum of all its assigned correlation weights (from 0 to 9, s. figure 2), each multiplied by the respective factor of importance (between 1 and 3). We also indicated respective relative importance due to good comprehensibility. Thereby, particularly two out of eight considered frameworks achieved high results: The Rational Unified Process (153) and the BIVM (198).

In a final step, the Architecture Framework was consolidated. Thereby, the QFD-matrix makes it possible to spot these frameworks that have the best potential to fulfil the given requirements in due consideration of their diverse relevance for the product architecture work. Although, the most relevant frameworks are highlighted, it is not exactly prescribed which parts of the selected frameworks have to be chosen or how the Architecture Framework has to be assembled. The QFD-matrix merely helps to identify the most promising state-of-the-art solutions in the variety of different systems existing in literature. Ultimately, the Architecture Framework was composed of parts of selected state-of-the-art frameworks as well as of own results and ideas. Figure 3 illustrates the consolidation of the Architecture Framework exemplarily.

First of all, derived from the Zachman Framework [13] and the BIVM [5], a **2x2-matrix** forms the basic frame of our Architecture Framework. The used modular structure of a grid allows a high flexibility as well as adaptability. The **allocation of the two dimensions** of the grid was carried out according to the RUP [2] and the BIVM [5]: The x-axis represents the dynamic, the y-axis the static dimension. More precisely, on the horizontal axis, the Architecture Framework indicates chronologically the architectural work steps. On the vertical axis, these work steps are grouped thematically into so called “topics” in order to be able to gather the different architecture tasks all together in one system. This axial assignment allows a detailed yet comprehensible and clearly arranged description of the architectural work. Concerning the “topics”, they can be seen as categories of the architectural work and represent the front column of the upper layer. This scheme is based on the RUP where so called **“workflows” structure the process** steps of a software engineering project [2]. Each row, i.e. each topic of the framework, contains the respective activities or rather process steps which are named "process modules". They are each represented by one cell in the according row of the framework. The arrangement of the process modules from the left to the right demonstrates the process-related order of the work steps.

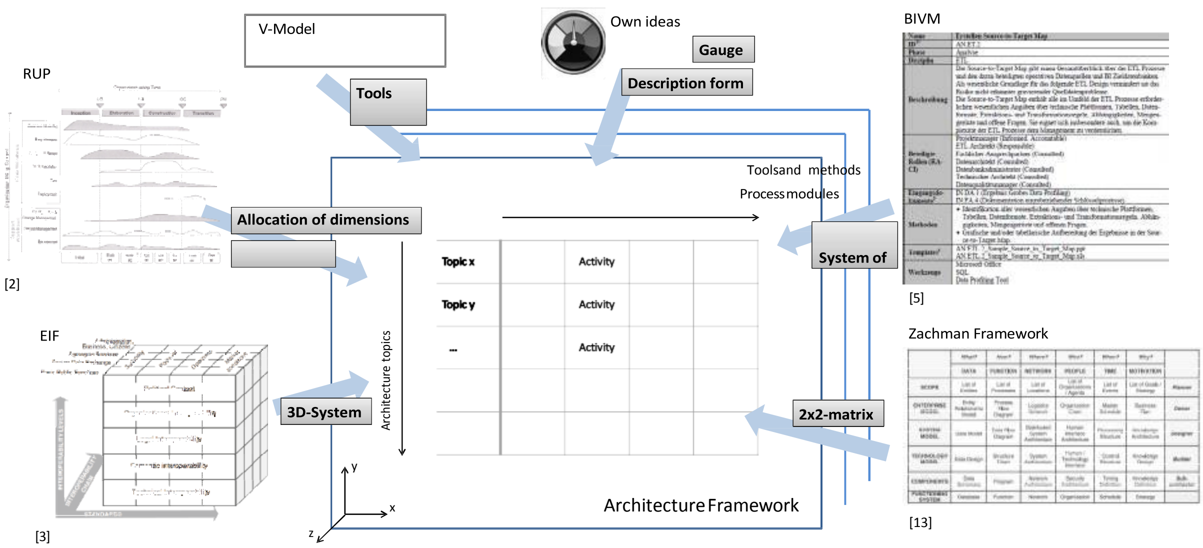

Figure 3: Consolidation of the Architecture Framework

Furthermore and similar to the BIVM [5], each cell of the matrix contains **information** that is **enclosed in two further layers** of the framework lying behind the upper one. Like in the EIF [3], we now have a **three-dimensional system** that makes it possible to clearly arrange the enclosed information. Thereby, the central layer of the framework, as in the BIVM [5], contains further information about the different architectural process steps named in the upper layer, i.e. each cell of the upper grid is assigned to a respective description form in the central layer.
Finally, the lower layer consists of descriptions of the tools and methods used to support the architectural work. Thereby the V-Model [8] had big influence, where, also on its lower layer, **tools** for the software development process and their requirements **are embedded**. Own results and ideas in developing the Architecture Framework include the layouts of the central and lower layer, like the central layer's **standardized description form** for each process step and a **gauge** indicating the process maturity of these work steps (s. figure 5b).
In the end, after synthesizing different frameworks as described, we came up to a three dimensional system consisting of three layers. The upper layer of the Architecture Framework structures the architectural work. The central layer offers detailed descriptions for each architectural process step and contains links to supporting tools and methods which are specified in the lower layer.

Architecture Framework for the commercial vehicle industry

As a first case study the Architecture Framework was filled with practical information collected in the industrial environment at a commercial vehicle manufacturer. The aim of this first implementation was to find out whether the developed framework meets the requirements of the Product Architecture Department in a sufficient way in order to support the product architects' day-to-day work. This commercial vehicle-specific Architecture Framework will be presented exemplarily in the following. For visualizing the framework we implemented it in a software demonstrator.

Upper Layer

The work of the Product Architecture Department includes different tasks, from the systematic structuring of future products to the planning of variance early in the product development process as well as holding a supporting role in the finding of technical solutions for engineering problems. To be able to gather the different architecture tasks together in one scheme, we identified ten major architectural topics presented in figure 4 which shows exemplarily the partly filled upper layer of the Architecture Framework.

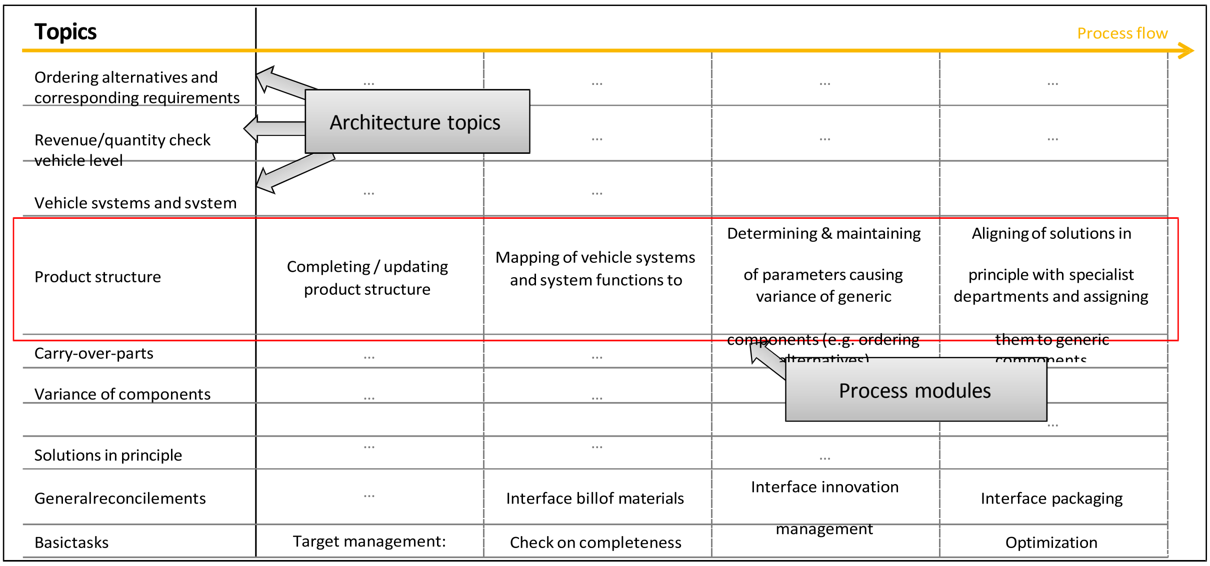

Figure 4: Upper layer of the framework

Ordering alternatives and corresponding requirements means collecting and processing of the requirements from the product management towards the future product. *Revenue/quantity check vehicle level* represents the assessment of the ordering alternatives and their combinations by economic aspects. The topic *vehicle systems and system functions* covers the solution-neutral functional structure of the product, whereas the *product structure* represents its technical counterpart. The *carry-over-parts* cover all technical components that are already being used in existing vehicles and are planned to be reused in the new development project. The *variance of components* signifies the required diversity of a generic component based on ordering alternatives and their combinations. *Solutions in principle* stand for possible technical solutions developed by the engineering department. The topic *functional specification* covers not only the compilation of the functional specification but also a final alignment of the identified solution concepts with the initially stated requirements. Finally, the topics *general reconcilements* and *basic tasks* represent superior architectural tasks which cannot be assigned to a certain process step or architectural topic but are necessary in order to address the major roles and tasks of the Product Architecture Department entirely. Among these tasks count the controlling and target management of weight and costs of a future product, the managing of the interfaces between the packaging and the engineering, as well as steady optimization during the development process and numerous checks on completeness of the project parts. Further, figure 4 shows exemplarily the four process modules assigned to the topic *product structure* (red box).

The format of the upper framework layer as a two-dimensional grid leads to a high system adaptability (REQ7). Its modular composition allows adjustments and enhancements of single process modules (respectively cells of the grid) at any time without having to alternate the adjacent cells at the same time. Also, the design of the framework is not limited to any specific content or problem and therefore must be seen as a neutral system that can also be adapted to other situations (REQ8). Furthermore, the clear structure of the grid makes it possible to state the central topics, questions and tasks of the architecture process in a well arranged and comprehensible way (REQ3). Therefore, the framework meets also the purpose of presentation of the architectural work (REQ4). Finally, the need for an adequate process mapping stated above is fulfilled by the description of the grid's process modules in the central layer of the framework (REQ1).

Central Layer

On the central layer of the Architecture Framework the product architect gets informed about a certain process module in-depth. Thereby, various external parameters affect the process module (s. figure 5a).

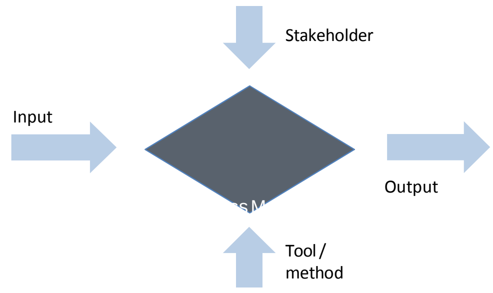

Figure 5a: Process module parameters

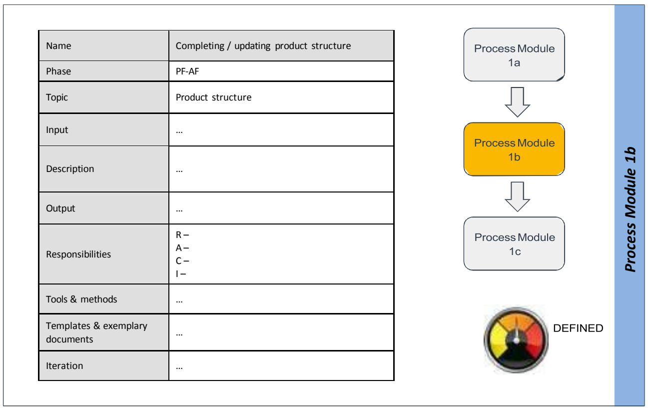

Figure 5b: Central framework layer

For the layout of this layer, we developed a scheme of description (s. figure 5b). Each process module, i.e. process step of the architectural work, is described by a number of so called "attributes" which characterize the respective process step: the *name* of the process step, the *phase* of the development process in which the step is carried out, the respective *topic* in the upper layer of the framework as well as detailed information about the *input and output* of the process module and a textual description about how to carry out the work step. Furthermore, the *responsibility* of the process step is documented using the RACI model known from enterprise architecture [4]. Also *templates* and exemplary documents as well as important *tools and mehtods* for handling the process step can be addressed. Where reasonable, links to the respective files or programmes can be directly provided. Last but not least, the attribute *iteration* contains information about the process-related dynamic of the work step, e.g. how many times the step has to be repeated. We also embedded a gauge which indicates the *maturity level* of the process step. Thereby, we distinguish between three different maturity levels: defined, planned and initial.

All in all, by developing this standardized description scheme we could meet the requirements stated above, namely the consideration of process dynamics (REQ9), the description of interfaces, stakeholders (REQ10), input and output (REQ11) as well as the assignment of responsibilities (REQ12) and the definition of process maturity (REQ13). The application of attributes also allows a high flexibility in the amount of information that is provided (REQ6). Finally, the attribute *phase* makes it possible to subdivide the architectural work into structured process phases and assign them to the Product Development Process of the commercial vehicle manufacturer (REQ2).

Lower Layer

Last but not least, the lower layer of the Architecture Framework contains information about tools and methods which are linked in the central framework layer and which support the product architect in carrying out the respective work steps (REQ5). Again, we developed a scheme in order to achieve a clearly structured and standardized description which is very similar to the scheme used in the central layer. Based on the tool- and method-description introduced by Lindemann [6], we specify three aspects for each tool or method: its aim or function, the right proceeding when applied as well as additional tools or advices which support the user of the tool or method. Where possible, the described tool or method is directly linked within this layer of the framework.

Discussion

In order to forge the bridge back to the beginning of this paper, we want to briefly discuss whether the presented Architecture Framework is able to efficiently fit the needs of a commercial vehicle manufacturer in terms of defining the major tasks, roles, methods, and responsibilities.

In the discussion with experts of the product architecture department we discussed the benefits and drawbacks of our approach. One benefit lies in having a high level structure to systematize and navigate the necessary tasks for implementing a product architecture department at the company. This serves as basis for communicating (at various levels of detail) the purpose and role of the product architecture group within the organization (e.g. on the group's intranet website).

Internally in the department, the framework can be used for project management by collecting and tracing the progress of implementation and by seeing what parts of the framework are catered for and which ones remain to be installed. Furthermore it represents a tool to focus and streamline the discussions within the department about roles and responsibilities by providing a common understanding of "what is product architecture at our company and what is it not?". From an IT-perspective in the company it supports checking the completeness of IT support and the completeness of integration of the tasks within the overall design process setup at the company.

In addition, some of the limitations of this study should be pointed out. First of all, the given problem is of socio-technical nature. It was asked for an abstract system in order to support the practical work of product architects in the truck industry. For this kind of question, there is no one single best solution, rather there are many different possibilities to realize the designated interaction between employees and the architecture process. Rittel & Webber [9] call such a problem "wicked problem [where] solutions are not true-or false, but good-or-bad". It is the nature of such problems, that there always exist various solutions of diverse quality.

Also, the presented work should be understood as a pragmatic answer to the problems of the Product Architecture Department at a commercial vehicle manufacturer. Repeatedly, there were situations in the developing of the Architecture Framework, where we had to decide between a practical, well achievable solution and a more extensive, more detailed one. It should therefore be pointed out that the aim of the presented study was to develop a solution, eligible for practical application in the architectural work.

Finally, it should be mentioned that the introduced framework, developed to fit the needs of product architects in the truck industry, can also be easily adapted to other industries. Also the method of deriving the framework from state-of-the-art solutions by reflecting them towards given requirements is arguably not limited to certain industry sectors. Therefore, by adapting the Architecture Framework or the used method accordingly it should be possible to structure and facilitate any kind of complex work process which is supported by software tools, certain methods or other resources.

Conclusion and Outlook

To conclude this paper, some of the key findings are summarized. We reflected the requirements of a commercial vehicle manufacturer towards an Architecture Framework to existing solutions in literature. The advantages of each concept were consolidated using a QFD-matrix. The suggested framework consists of three layers. The upper layer structures the architectural work by categorizing it into topics. The central layer offers detailed descriptions for each process step and contains links to supporting tools and methods which are specified in the lower layer. The Architecture Framework was prototypically tested in an industrial environment.

As next step, we propose the Architecture Framework to be adapted to and tested in other industrial environments in the form of case studies. It still is to be shown that the Architecture Framework can be positively applied, independent of the industry sector.

References

[1] Bindbeutel, K.: Engineering-Rahmensystem für den integrierten Produktentwicklungsprozeß. München: Utz 1998. ISBN: 9783896753748.

[2] Bunse, C.; Knethen, A. von: Vorgehensmodelle kompakt. Heidelberg: Spektrum 2002. ISBN: 9783827412034.

[3] Europäische Kommission: European interoperability framework for pan-european egovernment services: Version 1.0. Bruxelles: Office for Official Publications of the European Communities 2004. ISBN: 92-894-8389-X.

[4] Jacka, J.; Keller, P.: Business process mapping: Improving customer satisfaction. 2. Aufl. Hoboken: Wiley 2009. ISBN: 9780470444580.

[5] König, S.: Ein Wiki-basiertes Vorgehensmodell für Business Intelligence Projekte. In: Forschungskolloquium Business Intelligence 2009, CEUR Workshop Proceedings (2009) 542, S. 33–51.

[6] Lindemann, U.: Methodische Entwicklung technischer Produkte: Methoden flexibel und situationsgerecht anwenden. 3. Aufl. Berlin: Springer 2009. ISBN: 9783642014222.

[7] Matthes, D.: Enterprise Architecture Frameworks Kompendium: Über 50 Rahmenwerke für das IT-Management. Berlin: Springer 2011. ISBN: 9783642129544.

[8] Pomberger, G.; Pree, W.: Software engineering: Architektur-Design und Prozessorientierung. München: Hanser 2004. ISBN: 3-446-22429-7.

[9] Rittel, H.; Webber, M.: Dilemmas in a General Theory of Planning. Policy Sciences 4 (1973) 2, S. 155–169.

[10] Schuh, G.; Schwenk, U.: Produktkomplexität managen. Strategien, Methoden, Tools. München: Carl Hanser 2001.

[11] Van Haren (Hrsg.): TOGAF 2007 Edition (Incorporating 8.1.1). Zaltbommel: Van Haren Publishing 2007. ISBN: 9087530943.

[12] Weber, H.: Konzept eines Modells zur Produktentwicklung. Berlin: IPK 1998. ISBN: 9783816752059.

[13] Zachman, J.: Enterprise Architecture: The Issue of the Century. <http://www.zachman.com/images/ZI_PIcs/ibmsj2603e.pdf> - 21.09.2011.