Dependency analysis as a heat map for architecture standardization

Johannes Becker[1], Mark Gilbert[1], Armin Förg[2], Matthias Kreimeyer[3], Donna Rhodes[4], Markus Lienkamp[2]

Abstract Heavy duty trucks are high variant products with a comparably small production volume per product family. A high degree of specialization regarding utilization scenarios and transportation tasks, as well as strong spreading of functional variability generate increasing numbers of offered variants. The continuous introduction of new legal, technical and customer requirements combined with long product life cycles as well as the need for prolonged technological backward compatibility causes a complexity problem. Architecture standardization is a key lever in reducing complexity by deliberately cutting the number of variants and defining stable interfaces. However, at this point standardization potentials do not seem to be fully exploited.

This paper proposes an architecture standardization method using two approaches complementing product architecture development. First, a prescriptive approach predicts direct and indirect change propagation paths within a generic truck architecture, based on component dependencies. Secondly, a descriptive approach identifies geometrical conflicts in the product concept phase and facilitates the introduction of architectural standards, which in turn resolve these conflicts and decouples dependencies within the architecture. Applying these methods serves as a heat map that helps to identify the hot spots for potential standardization in product architectures. It is outlined and illustrated in two examples of change-related conflicts between physical components and product functionality.
Keywords: product architecture, change propagation, dependency analysis, complexity management, architecture standardization,

---

[1] Johannes Becker (j.becker@mytum.de)
Mark Gilbert (m.gilbert@mytum.de)
Technische Universität München, Garching, Germany

[2] Armin Förg (foerg@ftm.mw.tum.de)
Prof. Dr.-Ing. Markus Lienkamp (lienkamp@ftm.mw.tum.de)
Institute of Automotive Technology
Technische Universität München, Garching, Germany

[3] Dr. Matthias Kreimeyer (matthias.kreimeyer@man.eu)
MAN Truck & Bus AG, Munich, Germany

[4] Dr. Donna H. Rhodes (rhodes@mit.edu)
Systems Engineering Advancements Research Initiative (SEAri)
Massachusetts Institute of Technology, Cambridge, MA, USA

1 Initial situation and outline of paper

Heavy duty trucks (HDT) are products with a tremendously high number of different operational scenarios and use cases. Their functionality and technical requirements vary greatly depending on their individual purpose (e.g. long-haul logistics, distribution, construction, etc.). The operational demand for a truck is characterized by maximization of payload, reliability, efficiency and uptime. [18, p. 8]

This paper discusses *product architecture* standardization in HDT using the example of MAN Truck & Bus AG (MAN), a leading German commercial vehicle OEM. MAN's large portfolio of highly configurable vehicles relies on a modular architecture allowing for easy mass customization. The structure of this paper is shown in figure 1.1.

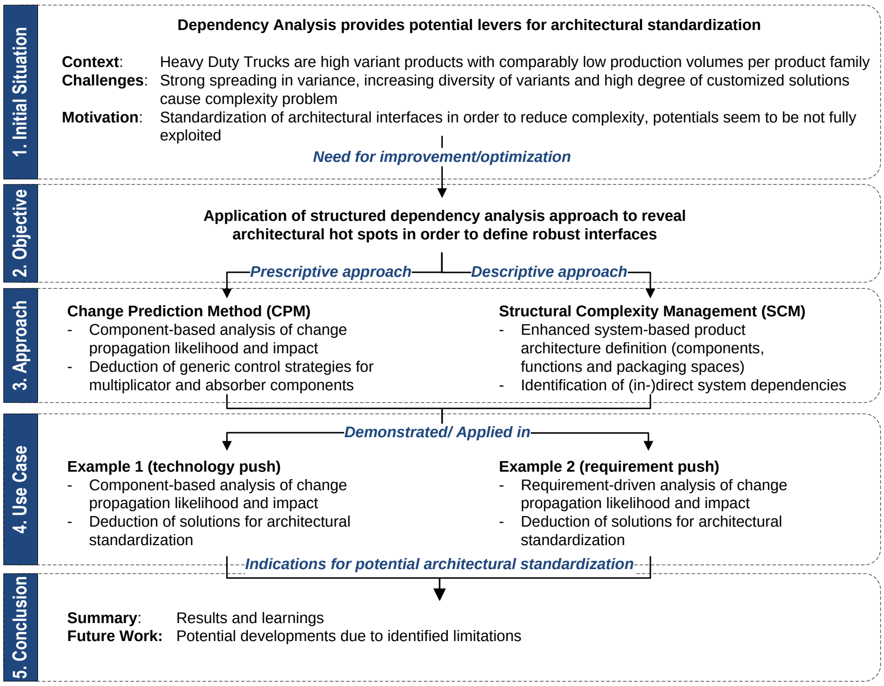

Fig. 1.1 Outline of the paper structure

1.1 Challenge

The *transport solution* needed by a customer is always a combination of the truck and its services and the body or trailer [18, p. 9]. Every single truck is a very individualized product (mass customization [18, p. 6]) serving as a platform for further extensions and modifications by equipment and body manufacturers, i.e. the truck product architecture has to offer adaptability and flexibility beyond the influence of the OEM. MAN's truck product architectures allow functional variants of up to 10^{46} [12, p. 1].

Truck manufacturers cannot benefit from economies of scale due to significantly lower production volumes compared to passenger cars [18, p. 6]. Instead, the focus lies on the modularity and versatility of HDT product architectures.

The high compatibility of components in conjunction with an ever-increasing diversity of variants causes complexity problems. This is due to an increase of functionality provided by new technologies (e.g. hybridization). Additionally, the complexity further rises with increasingly sophisticated national and global legal requirements (e.g. introduction of the EURO VI norm [16, pp. 172-175]). Lastly, HDT product architecture has to remain backwards-compatible and still support systems introduced decades ago, despite the continuous incorporation of new technologies and components (e.g. concurrent use of EURO II/III in developing countries and EURO IV+ in Europe).

The balancing act (to be solved) lies in the generation of new variants based on a structure, which has grown over time and cannot be modified in a radical way in order to retain backwards compatibility. At the same time, it is necessary to ensure that the product architecture is future-proof against upcoming and long term changes in technology and other requirements.

1.2 Motivation

This complexity problem is caused by growing numbers of subsystems or combinations thereof, which have to be incorporated in the product architecture.

Standardization of interfaces as well as geometric boundaries of package configuration were identified to be key levers to mitigate or control this complexity problem. Through standardization potential variant combinations are deliberately excluded from the desired solution space while standardized interfaces facilitate controlling arising change propagation.

However standardization potentials are not fully exploited with regards to HDT product architectures. A considerable amount of manpower is still involved in clarifying and solving variance issues and further research on modular kits in commercial vehicle design is ongoing [12, p. 2].

Product architecture standardization could resolve this complexity problem and reduce conflict potentials in the early stages of product development.

2 Objective

The objective of this paper is to systematically analyze the *product architecture* of HDT by applying a structured dependency analysis approach to reveal architectural *hot spots* in order to define robust and stable interfaces. Hot spots are understood as the ideal elements to leverage the revealed standardization potentials.

Furthermore, this approach identifies *change propagation* paths as well as deduces means of controlling occurring change propagation. For an unambiguous description of different *packaging space* constellations, a qualitative formalization of packaging spaces is proposed, enabling systematic assignment of components into geometric sections of the physical product structure [12]. Lastly, the objective is to identify interfaces for potential standardization.

The novelty of the paper is constituted by proposing an extended product architecture definition and by applying a combination of two approaches to truck product development.

3 Approach

The approach used in this paper combines prescriptive and descriptive methods in order to identify key elements for architecture standardization. In general, prescriptive methods have the goal to advance the state of the practice using theory-based knowledge while descriptive methods use information and constraints of the state of the practice in order to advance the theory-based state of the art. [20, p. 2]

The *Change Prediction Method* (CPM) [2] allows for prescriptive identification of change-related risks in a product and provides a framework for handling change-critical elements of the system (section 3.2). *Structural Complexity Management (SCM)* [14] allows the definition of extended product architectures from a systems point of view and acts as the descriptive part of our approach.

The interaction between the prescriptive method, product development process, descriptive method and architecture standardization is illustrated in figure 3.1.

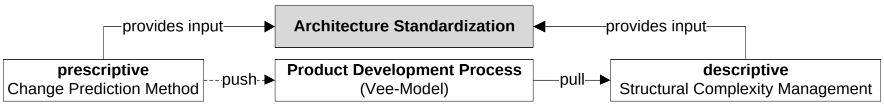

Fig. 3.1 Utilization of prescriptive and descriptive methods for architecture standardization

3.1 Theoretical Background

The classical definition of *product architecture* originates from the early 1990's when the question was raised whether product architectures may be used to simplify product development and to reduce its complexity. This product architecture definition is based on three characteristics: [24, p. 420]

- arrangement of functional elements or functional structure
- mapping of functional elements to physical elements
- specification of the interfaces among interacting physical elements

Product architecture can also be described as a *system*. Definitions of systems have a long history [26, p. 52; 9, p.34]. In this paper, a system is understood as a combination of the definitions in [13, pp. 23-24] and [15, p. 8], considering both system boundaries as well as its inputs and outputs.

An important aspect of product architecture research is to identify means of reducing *complexity*. In systems, complexity is manifested by connectedness, characterized by relationships, and variety, represented by elements. Their diversity and quantity further adds to complexity [19, pp. 22-24]. The complexity of systems can be represented using graph theory or matrix-based approaches [13, pp. 43–61].

Furthermore, complexity can be classified by linking the different dimensions of complexity to strategic technical aspects of a product (see figure 3.2) [25].

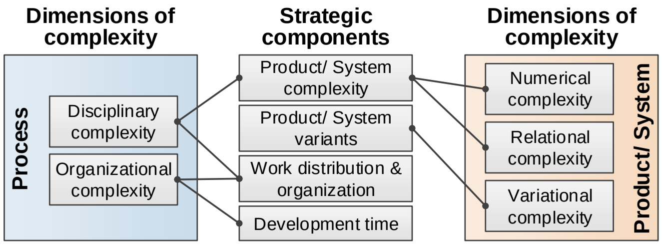

Fig. 3.2 Coherence between dimensions of complexity and strategic components of a system

However, complexity is not an undesired state for a system or a product per se. It comes with opportunities (e.g. capability to control a diversity of variants) and obstacles or negative effects (e.g. numerous changes due to lack of transparency). In competitive market environments it is advantageous to have the ability to cope with complexity. Many prosperous companies work on the edge of manageable complexity, and are successful for exactly that reason [13, p. 20].

Literature discusses two fundamentally different approaches for handling complexity: (1) to avoid and mitigate it, and (2) to manage and control it [13, pp. 31-35].

This paper utilizes matrix-based approaches to model and analyze product architectures as a system. The *Design Structure Matrix (DSM)* is a sort of intradomain matrix which maps elements of the same nature to each other [6; 11, p. 2; 22]. A domain contains elements of the same nature [13, p.49].The mapping between DSM elements represents a specific dependency (e.g. physical, spatial, energy, or information) [13, p. 49].

A domain mapped to another domain is known as *Domain Mapping Matrix (DMM)* [3]. DMM was introduced as a complementary form of DSM to overcome its characteristic single-domain limitations. [4, p. 304]

A combination of DSM and DMM complemented with computation logics to derive indirect relationships was introduced as *Multiple-Domain-Matrix (MDM)*. The MDM [17] enables the division of a complex system into subsystems, which are represented by the different domains within the MDM [13, p. 78].

Hence, the *domains* of *components* and *functions* suffice to fully describe product architecture according to the definition mentioned above.

Modularity is one of the most important aspects of product architecture. It defines the way *chunks* of the product architecture are mapped to functions. There are two archetypes of architectures: (1) *modular* and (2) *integral* architectures. [10]

In reality, product architectures are not purely modular or integral but can be classified into different degrees of modularity: (1) *Component Sharing Modularity*, (2) *Component Swapping Modularity*, (3) *Cut to Fit Modularity*, (4) *Bus Modularity*, (5) *Sectional modularity* and (6) *Mix Modularity*. [7, p. 350]

When represented in DSM form, different types of modularity can be visually identified as shown in figure 3.3. *Integral product architectures* (DSM_a) have a very dense DSM. *Bus modular architectures* (DSM_b) have vertical and horizontal lines identifying their bus elements. *Fully integrated product architectures* with serially connected elements have band DSMs (DSM_c). [10, p. 6]

$$DSM_a = \begin{bmatrix} & \bullet & \bullet & \bullet & \bullet \\ \bullet & & \bullet & \bullet & \bullet \\ \bullet & \bullet & & \bullet & \bullet \\ \bullet & \bullet & \bullet & & \bullet \\ \bullet & \bullet & \bullet & \bullet & \end{bmatrix} \quad DSM_b = \begin{bmatrix} & \bullet & \bullet & \bullet & \bullet \\ \bullet & & & & \\ \bullet & & & & \\ \bullet & & & & \\ \bullet & & & & \end{bmatrix} \quad DSM_c = \begin{bmatrix} & \bullet & & & \\ \bullet & & \bullet & & \\ & \bullet & & \bullet & \\ & & \bullet & & \bullet \\ & & & \bullet & \end{bmatrix}$$

Fig. 3.3 Different types of modularization of product architectures according to [10, p. 6]

Standardization can be used to reduce complexity, costs and lead times in product development. Modular architectures facilitate standardization [24, pp. 431-432].

Component standardization leads to the creation of less component variants. Consequently, these fewer variants are used in higher quantities and benefit from economies of scale and quality improvement by experience.

Architecture standardization (e.g. deliberate elimination of certain possible component variants and component assignments) can be used to mitigate complexity and change propagation.

A formalization of *packaging spaces* is proposed to assign components in early phases of product development into defined sectors of the package. The packaging space model is of qualitative nature and constructed using simple geometrically adjoining bodies, since only rough information regarding component dimensions and form is available in these early stages. The different vehicle types are generically divided using number and location of axles to derive a parametric typologization model. It is based on cross sections attached to meaningful longitudinal and lateral locations of the vehicle.

Due to its qualitative nature, the model can be universally transferred to other vehicle architectures (e.g. cars). The result is a flexible grid, which is adaptable in terms of distinctness. An emerging packaging space element is modeled as a rectangular prism defined by its six boundary layers. It can be unambiguously visualized using superposed side and top projections of technical drawings (see figure 3.4).

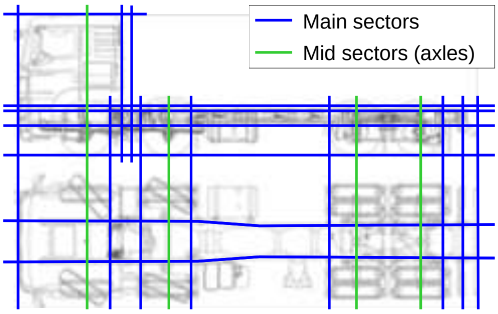

Fig. 3.4 Packaging spaces visualized by superposing qualitative grid and technical drawings in [12, p. 10]

There are other methods for investigating system-to-system interaction (i.e. zonal analysis [1]). However this method is preferably used for aerospace system safety assessment of specific systems, not considering variable system scenarios.

In contrast the proposed model focusses on supporting early decision making by confirming the feasibility to accommodate certain components in emerging packaging spaces.

3.2 *Prescriptive: Change Prediction Method*

The *Change Prediction Method (CPM)*, initially proposed in 2004 [2], can be used for modular kit development in commercial vehicle design with some adaptations. [12, p. 8-9]

The input used for this method is an innovation planning document mapping product requirements to a generic truck product decomposition. This data is used to generate a DSM describing the change propagation likelihood between elements of the product decomposition. These change propagation dependencies can be modeled in different ways. Firstly, as a *dependency score* where multiple occurrences of dependency pairs from the input document are summed up with specific weights indicating the level of dependency. Secondly, in a *probabilistic way* where multiple occurrences of dependency pairs in the input document are treated like a binomial distribution of propagation likelihoods.

The probabilistic approach has the advantage of producing bounded values representing propagation likelihoods from 0 to 1. In a first step, these likelihoods describe direct propagation from one component (C_1) to another (C_2). However, this approach also allows the aggregation of indirect propagation paths from component (C_1) to component (C_2) via a path of other components (C_i) [2, p. 792-793]. The combined likelihood matrix can be multiplied with propagation impacts in order to compute a propagation risk matrix.

This component-to-component DSM is used to classify components into different change propagation behaviors by comparing their indegree and outdegree [21, p. 7; 5, p. 13; 23, p. 73-74]. In this classification, components can act as

- *constants*, which are not affected by change. They neither propagate nor absorb changes nor do they add complexity to the change propagation problem.
- *absorbers*, which can absorb more changes than they initiate. Absorbers reduce the complexity of change propagation.
- *carriers*, which propagate a similar amount of change as they absorb. They do not affect the complexity problem.
- *multipliers*, which propagate more change than they absorb. Thereby they amplify changes and increase the complexity of the problem.

Propagation behavior is not an intrinsic feature of a system component. It can be influenced by increasing or decreasing the *contingency margin* of a part [21, p. 13]. Some components, however, might not be changed due to management policy or strategic implications. Typically, these components are bought-in components or involve long development times; they are called *resistors.* Resistors reflect changes [5, p. 14] and usually cause changes in more changeable parts of their surroundings.

In order to increase robustness of product architectures, different measures can be taken regarding change multipliers: They can be isolated and decoupled from other components in order to reduce their probability of receiving and thus multiplying changes, or equipped with sufficient contingency margins mitigating their multiplying behavior in favor of more absorbing behavior [5, p. 14].

Another option is grouping all absorbers and isolating carriers and multipliers by packing them in separated modules [8, p. 7].

3.3 Descriptive: Structural Complexity Management

Structural Complexity Management (SCM) is a generic and standardized approach to tackle engineering problems in product design and development [13, pp. 62-66].

The procedure is divided into five steps, introduced in table 3.1. General starting point for its application is a sound understanding of the actual engineering problem and where its complexity arises from, even before the system definition is approached.

No.	Step	Activities and operations	Deliverable
1	System definition	A target-aimed system definition is performed by modeling all information within a MDM framework. It involves the definition of a system boundary, an appropriate level of abstraction, identification of domains and the determination of relevant dependencies within the system. The decision about requirement-driven or data based information acquisition is prepared.	MDM Framework
2	Information acquisition	Gathering of native (direct) dependencies between domain elements. To ensure the expressiveness of acquired data, the acquired information must be frequently verified.	Direct system dependencies
3	Deduction of indirect dependencies	To complete the set of required information the different computation schemes are executed to derive the indirect dependencies	Representation of subsets
4	Structure analysis	Graph and matrix-based models are used to carry out a structural analysis of the system. The main objective is to identify meaningful structures of the system and its key elements.	Significant constellations
5	Product design application	Induces learnings from the structural analysis for incremental improvement or redesign of the system as a whole. The reach of incremental improvement is limited while redesign realizes major improvements regarding structure and documented transparency.	Improved system management & design

Table 3.1 Procedure of Structural Complexity Management

For the application of the SCM approach choosing a thoughtful abstraction level is important. It defines the depth of detail within the relevant system and has a high impact on data acquisition effort. The trade-off must be made between the level of detail, uncertainty of information, and its acquisition efforts.

An advantageous characteristic of SCM is the feasibility to compute indirect dependencies based on natively acquired direct relationships within systems.

We, for instance, we gained valuable insights regarding potential packaging space conflicts of components due to their assignment to the same packaging space. The likelihood of a packaging space conflict between two components that are actually unrelated is proportional to the strength of the computed indirect dependency. In addition, without knowledge of components assigned to distinct packaging spaces, it is possible to advise their assignment into selected packaging spaces based on their individual connectedness within the system. With this, the structural characteristic of the system can be considered.

4 Use case

The application of our approach is presented in two separate examples of current product architecture issues.

4.1 System definition

Our research approach enhanced the ‘classical’ product architecture definition by considering emerging *packaging spaces*. A vehicle usually offers limited packaging space to accommodate components. Such installation spaces are represented by the packaging spaces domain. The actual system definition and its dependencies between components, functions and packaging spaces are illustrated in figure 4.1.

MDM Framework — ***Enhanced Product Architecture***

Classical Product Architecture

	F	**C**	***PS***
Functions	interrelate	is enabled by	
Components		(1) connected: geometrical/ physical/ energy-& mass flow (2) propagate changes	*Accom-modated in*
Packaging Spaces			*geom. adjoining*

Fig. 4.1 System definition of an enhanced product architecture using functions, components and packaging spaces

4.2 Example 1: gearbox integration (technology push)

In this case, a new gearbox technology is introduced, which changes size as well as geometrical form of the generic gearbox component. The preexisting gearbox had a wide and flat geometry, whereas the newly introduced gearbox has a narrower but higher design. Thus it exceeds its initial packaging space limits. The new geometry collides with the positioning of the exhaust piping, which previously ran below the gearbox along the truck body frame. As a consequence a change impulse is initiated.

Applying CPM (section 3.2) shows a change in the gearbox has a high likelihood of propagating changes to the exhaust piping. This is visualized in a change propagation matrix (figure 4.2, left). A hot spot in position (1,3) indicates a high propagation likelihood from C_3 (gearbox) to C_1 (exhaust piping). This can be attributed to the fact that gearbox and exhaust piping are very closely spaced, i.e. the geometric contingency margin of the gearbox is very small. This leads to change multiplying behavior in the presence of geometrical modifications of involved components.

Using SCM (section 3.3), the situation analysis is performed in two steps. Initially, the DMM mapping of components to packaging spaces shows no packaging space violation between C_3 and C_1. The geometrical modifications to the gearbox cause a competitive packaging space situation in PS_2 among components C_3 and C_1 (see figure 4.2). Although multiple components sharing a packaging space is not a conflict in itself, it complicates the independent changing of component dimensions amongst others.

From a geometrical point of view, these effects could be remedied by making the cross section of the exhaust piping flat enough for both components to fit next to each other in PS_2. This would, however, negatively influence the vehicle's functionality by reducing the air flow in the exhaust piping. Consequentially, it involves relocating the exhaust piping (as the gearbox is less maneuverable) to avoid collision and a negative impact on performance.

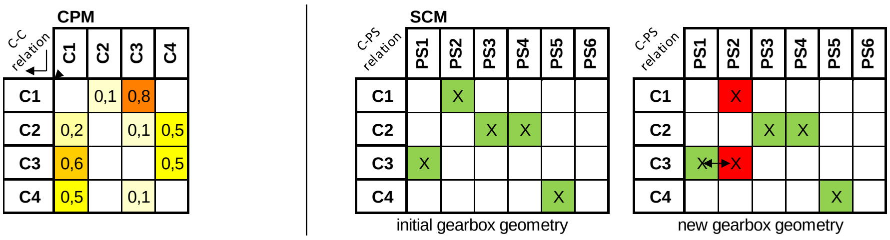

Fig. 4.2 Change Propagation Matrices: left: DSM propagation likelihood, right: DMM component assignment to packaging spaces

Since both CPM and SCM indicate a high likelihood of change propagation between gearbox and exhaust piping, an architectural standard which avoids this interdependency by separating both components from each other is proposed. This could be realized by determining that the exhaust piping always runs from the exhaust silencer at the front right side to the back right side. Introducing this architectural standard, the packaging space conflict is resolved by avoiding any collision with the gearbox regardless of the variant in use. As shown in figure 4.3, components C_1 and C_3 now have significantly lower change propagation likelihood between each other and no longer constitute a hot spot in the propagation likelihood matrix. Moving C_1 to a different packaging space PS_6 (right side of the vehicle) avoids packaging space competition with C_3 regardless of its configuration. This decoupling of the previous component interdependency improves the value robustness [20] of the product architecture by reducing the complexity of future changes to the gearbox.

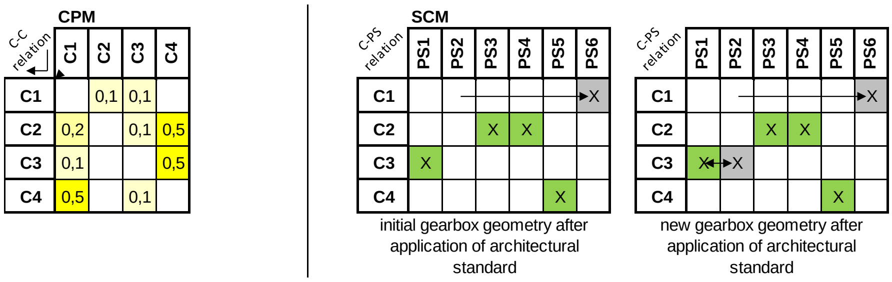

Fig. 4.3. Change Propagation Matrices: left: propagation hot spots eliminated, right: packaging space conflict avoided by relocating components

4.3 Example 2: new legal regulations (requirement push)

In this example, the event chain of altered legal regulations is discussed. Taking effect from 2014, German legislation requires new vehicles to conform to EURO VI emission limits [18, p. 32]. This causes a change impulse.

The EURO IV/V exhaust gas after-treatment system is complemented by two additional components (AdBlue tank and SCR catalyst). Accommodating both components in the given packaging spaces to fulfill the legal requirements causes a packaging space conflict. Thus, the size, position and form of existing components must be carefully considered to solve this conflict. Backward compatibility and carry over components make this a delicate task as component changes might propagate into the component functionality (e.g. tank size correlates with range) or increase the variance of components (i.e. higher costs).

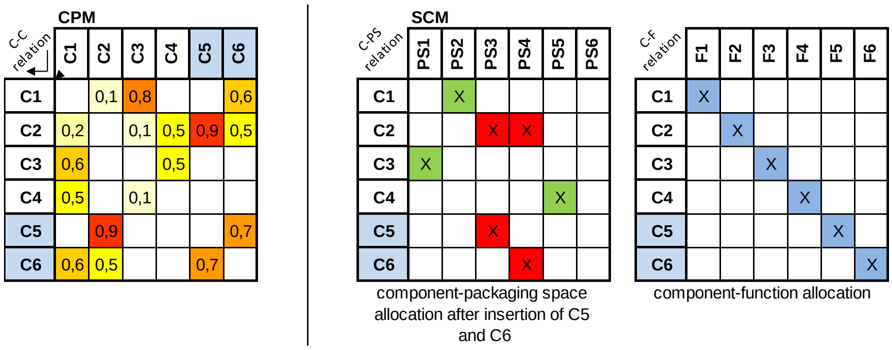

Fig. 4.4 Left: The change propagation likelihood between fuel tank (C_2) and AdBlue tank (C_5) is a hot spot. Right: Both components as well as the SCR system (C_6) compete for volume in packaging spaces PS_3 and PS_4.

Figure 4.4 shows the AdBlue tank (C_5) and SCR system (C_6) in addition to the existing four components. These components have a very high likelihood of propagating change to the fuel tank (left), as they compete for the same packaging space (middle). For a given wheelbase, packaging space PS_4 only has a limited volume to accommodate all three components. The size of the SCR module is fixed, and the size of the AdBlue tank has to be proportional to the fuel tank (ratio of urea to fuel consumption is 5-7% [16, p. 173]). Thus, given a certain wheelbase, the fuel tank has to be adapted in size to avoid component collisions. Fuel tank volume is not an explicit requirement, though, but rather implied by the need for the greatest range possible. Since the wheelbase limits the total volume, the fuel tank volume is inevitably reduced by the amount necessary to avoid conflicts.

The dimensions of the SCR system and AdBlue tank as well as its connection to the exhaust piping are constant. An architectural standard is defined, which always places the SCR system at the front right side of the truck frame, indicated as PS_4 in figure 4.5. The fuel tank (C_2) is confined to PS_3 and shares this space with the AdBlue tank (C_5). Their respective volumes strongly depend upon each other (see figure 4.5, left) and can be maximized proportionally depending on the available space provided by the given wheelbase. This remaining package space competition and circular change propagation can be handled by always considering fuel tank and AdBlue tank as one coupled system which will be changed altogether if necessary, thereby isolating and internalizing their change multiplying behavior as proposed by [8, p. 4]. The compromise in volume and therefore the maximum range of the vehicle is marked by the asterisks in figure 4.5 (right), indicating altered functionality.

CPM

C-C relation	C1	C2	C3	C4	C5	C6
C1		0,1	0,8			0,6
C2	0,2		0,1	0,5	0,9	0,2
C3	0,6			0,5		
C4	0,5		0,1			
C5		0,9				0,2
C6	0,6	0,2			0,2	

SCM

C-PS relation	PS1	PS2	PS3	PS4	PS5	PS6
C1		X				
C2			X			
C3	X					
C4					X	
C5			X			
C6				X		

component-packaging space allocation after standardization of C6 position

C-F relation	F1	F2	F3	F4	F5	F6
C1	X					
C2		X*				
C3			X			
C4				X		
C5					X*	
C6						X

component-function allocation after standardization of C6 position

Fig. 4.5 Architecture standardization places the SCR module (C_6) in a dedicated location (PS_4). The fuel tank (C_2) and the AdBlue tank (C_5) compete for volume in PS_3 and have to be modified together due to their high mutual change propagation likelihood (middle). The functionality of C_2 and C_5 is limited by the available volume in PS_3.

5 Conclusion and Future Work

As shown, the proposed approach combines elements of Change Prediction Method and Structural Complexity Management as a decision-making aid for product architecture standardization in the early concept phase of product development. The CPM can identify areas of high change propagation likelihood and therefore help eliminate *change mines* by confining their influence with by reasonable standardization guidelines. SCM can be used to generically structure and resolve issues of component placement in the truck package.

The proposed approach is limited by the availability of information and its abstraction level, as higher levels of detail require higher data acquisition effort. Furthermore, the approach does not model quantitative aspects like actual component dimensions. Ongoing and further research combines this approach with early digital mock-ups of vehicle concepts automatically generated from requirements specifications in order to generate architectural standards which also take quantitative data into account based on further expansion of the packaging space model [12, p. 7-11].

Acknowledgments This article contains results of Master's theses from Johannes Becker (Feb to Jul 2013) and Mark Gilbert (Jun to Dec 2012), conducted in a cooperation between Technische Universität München and Massachusetts Institute of Technology. Mr. Becker and Mr. Gilbert would like to thank Dr. Armin Schulz and Dr. Stefan Wenzel of 3DSE Management Consultants for facilitating and supporting their research at MIT. The project was independently funded by the Institute of Automotive Technology at the Technische Universität München, the Systems Engineering Advancements Research Initiative (SEAri) at Massachusetts Institute of Technology, and MAN Truck & Bus AG.

References

[1] Caldwell RE, Merdgen DB (1991) Zonal analysis: the final step in system safety assessment [of aircraft]. In: Proceedings of Reliability and Maintainability Symposium, pp 277-279

[2] Clarkson PJ, Simons C, Eckert C (2004) Predicting Change Propagation in Complex Design. Transactions of ASME 126(5): p 788. doi: 10.1115/1.1765117

[3] Danilovic M, Browning T (2004) A Formal Approach for Domain Mapping Matrices (DMM) to Complement Design Structure Matrices (DSM). In: The Sixth Design Structure Matrix (DSM) International Workshop, pp 1-23

[4] Danilovic M, Browning TR (2007) Managing complex product development projects with design structure matrices and domain mapping matrices. International Journal of Project Management 25(3): pp 300-314. doi: 10.1016/j.ijproman.2006.11.003

[5] Eckert C, Clarkson PJ, Zanker W (2004) Change and customisation in complex engineering domains. Research in Engineering Design 15(1): pp 1-21. doi: 10.1007/s00163-003-0031-7

[6] Eppinger S (1991) Model-based Approaches to Managing Concurrent Engineering. Journal of Engineering Design 2(4): pp 283-290. doi: 10.1080/09544829108901686

[7] Fricke E, Schulz AP (2005) Design for changeability (DfC): Principles to enable changes in systems throughout their entire lifecycle. Syst. Engin. 8(4). doi: 10.1002/sys.20039

[8] Greisel M, Kissel M, Spinola B et al. (2013) Design for Adaptability in Multi-Variant Product Families. In: Proceedings of ICED13 volume 4. Product, service and systems design. Design Society
[9] Haberfellner R, De Weck OL, Fricke E et al. (2012) Systems Engineering. Grundlagen und Anwendung, 12th edn. Orell Füssli, Zürich
[10] Hölttä K, Suh ES, De Weck, OL (2005) Tradeoff between Modularity and Performance for Engineered Systems and Products. In: ICED 05. 15th international conference on engineering design: engineering design and the global economy. Engineers Australia, Barton, A.C.T
[11] Kreimeyer M (2012) A Product Model to Support PLM-Based Variant Planning and Management. In: Proceedings of DESIGN 2012, the 12th International Design Conference, vol 3, pp 1741-1752
[12] Kreimeyer M, Förg A, Lienkamp M (2014) Fostering Modular Kits in an Industrial Brownfield Environment. In: Proceedings of TMCE 2014
[13] Lindemann U (2009) Methodische Entwicklung technischer Produkte. Methoden flexibel und situationsgerecht anwenden, 3rd edn. Springer, Berlin
[14] Lindemann U, Maurer M, Braun T (2009) Structural complexity management. An approach for the field of product design. Springer, Berlin
[15] Maier MW, Rechtin E (2009) The art of systems architecting, 3rd edn. CRC Press, Boca Raton
[16] MAN Nutzfahrzeug Gruppe (2010) Grundlagen der Nutzfahrzeugtechnik. Basiswissen Lkw und Bus, Munich
[17] Maurer M, Lindemann U (2007) Facing Multi-Domain Complexity in Product Development. CiDaD Working Paper Series 3(1): pp 351-361. doi: 10.1007/978-3-540-69820-3_35
[18] Nielsen A (2014) Modularization. MAN Truck & Bus AG Lecture
[19] Patzak G (1982) Systemtechnik. Planung komplexer innovativer Systeme : Grundlagen, Methoden, Techniken. Springer, Berlin
[20] Ross AM, Rhodes DH (2008) Architecting Systems for Value Robustness: Research Motivations and Progress. In: SysCon 2008. 2nd Annual IEEE International Systems Conference, pp 1-8
[21] Shah NB, Wilds J, Viscito L et al. (2008) Quantifying Flexibility for Architecting Changeable Systems. In: Conference on Systems Engineering Research
[22] Steward DV (1981) The design structure system: A method for managing the design of complex systems. IEEE Transactions on Engineering Management EM-28(3): pp 71-74. doi: 10.1109/TEM.1981.6448589
[23] Suh ES, De Weck, OL., Chang D (2007) Flexible product platforms: framework and case study. Res Eng Design 18(2): pp 67-89. doi: 10.1007/s00163-007-0032-z
[24] Ulrich K (1995) The role of product architecture in the manufacturing firm. Research Policy 24(3): pp 419-440. doi: 10.1016/0048-7333(94)00775-3
[25] Weber C (2005) What Is 'Complexity'? In: ICED 05. 15th international conference on engineering design: engineering design and the global economy. Engineers Australia, Barton, A.C.T
[26] Weinberg GM (1975) An introduction to general systems thinking. Wiley, New York