

DISTINCTION OF DOMAIN-SPECIFIC AND CROSS-DOMAIN LINKAGE TYPES FOR ENGINEERING CHANGE MANAGEMENT

Wilms, Robert (1,2); Inkermann, David (1); Cemmasson, Vadym Finn (1); Reik, Michael (2); Vietor, Thomas (1)

1: TU Braunschweig; 2: Volkswagen AG

ABSTRACT

Engineering Changes (ECs) are substantial elements of the design process of technical products and are in particular relevant for companies due to enormous additional costs and time delays they can cause. In order to better understand ECs and realize efficient Engineering Change Management (ECM), different approaches exist. One aspect of ECM are change propagation analysis, which try to analyze knock-on effects of an EC on other product elements or the development process. How ECs can propagate is in particular difficult to assess for complex products realized within different engineering domains (mechanical, electrical and software engineering). To address this challenge, ECs are classified, strategies to cope with ECs are presented and change propagation approaches are analyzed in this paper. Thereby a lack of indicators for cross-domain propagation is identified. To overcome this issue, the distinction of domain-specific and cross-domain linkage types is proposed and a set of linkage types is presented. Further research is motivated to integrate these linkage types in product models while also considering processes and organizational structures as additional dimensions of ECM.

Keywords: Engineering Change Management, Design process, Complexity, Systems Engineering (SE)

Contact:

Wilms, Robert
Volkswagen AG
Group Engineering Processes & IT
Germany
r.wilms@tu-braunschweig.de

Cite this article: Wilms, R., Inkermann, D., Cemmasson, V.F., Reik, M., Vietor, T. (2019) 'Distinction of Domain-Specific and Cross-Domain Linkage Types for Engineering Change Management', in *Proceedings of the 22nd International Conference on Engineering Design (ICED19)*, Delft, The Netherlands, 5-8 August 2019. DOI:10.1017/dsi.2019.118

1 INTRODUCTION

Increasing product complexity and the necessity to integrate different engineering domains (mechanics, electric/electronics and software) lead to significant challenges when handling engineering changes (ECs) within the design process of established companies. Taking automotive engineering as example, this complexity can be illustrated by the following development: in 1989 a Volkswagen Golf (second generation) had two electronic control units (ECUs), for today's vehicles up to 100 ECUs are needed (Automobilwoche, 2018); some vehicles had approximately ten million software lines of code in 2010, until 2016 this number went up to 150 million (Burkacky *et al.*, 2018). A corresponding trend is that up to 90% of today's innovations in automotive engineering are based on electronics and software (Automobilwoche, 2018), which indicates a clear shift in terms of differentiating characteristics for vehicles. This trend is highlighted by Burkacky *et al.* (2018) stating that software can "reach 30 percent of overall vehicle content (around \$5,200) in 2030". The higher relevance of electric/electronics and software for automotive engineering reduces the amount of functions that are solely realized within the mechanical engineering domain (Eigner, 2014). With regard to these numbers and the demand to adapt R&D competences and capacities, a main challenge for established companies is to integrate and coordinate involved engineering domains with varying design processes and principles.

The described complexity regarding products as well as processes and organizational structures have strong implications on the management of ECs, since these can be seen as small design projects within the entire product development process. The relevance of ECs for product development is illustrated by Yildirim and Campean (2013) reporting an average of five ECs per part for the development of a typical new vehicle. In this context, Albers *et al.* (2016) highlight an increasing challenge, stating that "Changes can affect multiple systems and stakeholders. To assess which changes are relevant for which stakeholder is not a trivial task." As a consequence, a clear understanding of linkages and interfaces between different components or systems and resulting possibilities for change propagation is crucial for successful engineering change management (Clarkson *et al.*, 2001; Yildirim and Campean, 2013). However, this understanding does not only refer to the product perspective, it also needs to be identified, if an EC needs to be evaluated from an interdisciplinary perspective or even results in an interdisciplinary EC process. Approaches that are indicating which engineering domains are affected by an EC are missing in literature. Therefore, this contribution aims at highlighting the relevance of cross-domain engineering change management and introducing linkage types to identify and assess cross-domain effects of ECs.

This paper is organized as follows: Section 2 describes the basic concept of engineering change management, presenting generic EC processes as well as classification approaches for and strategies to cope with ECs. Section 3 focusses on change propagation by describing different types of change propagation behavior and existing approaches to manage change propagation. As a basis for future cross-domain engineering change management, the distinction of domain-specific and cross-domain linkage types as well as an initial set of cross-domain linkage types are proposed in section 4. Section 5 finally gives a conclusion of this paper and motivates future work.

2 BASIC CONCEPT OF ENGINEERING CHANGE MANAGEMENT

Since most products can be seen as evolution of existing designs (e.g. Gerst *et al.*, 2001; Jarratt *et al.*, 2004c), engineering changes are substantial element of the product development process (e.g. Fricke *et al.*, 2000; Clarkson *et al.*, 2005). However, in many industries ECs are rather perceived as unplanned, expensive and time-consuming, neglecting positive consequences of an EC for instance in terms of product quality or innovation (Riviere *et al.*, 2003a). According to Jarratt *et al.* (2004a) an EC can be defined as "alteration made to parts, drawings or software that have already been released during the product design process. The change can be of any size or type; the change can involve any number of people and take any length of time." In particular the second part of this definition highlights the relevance of ECs with reference to the time schedule and costs of a design project, which is why an efficient engineering change management (ECM) can be seen as a main competitive advantage (Wilberg *et al.*, 2015). To gain this advantage Wilberg *et al.* (2015) state that "Engineering change management seeks to forecast possible changes, identify changes that have already occurred, plan preventive measures and coordinate changes across the entire project." ECM also comprises the

implementation of (bundled) ECs, which is why it is closely linked to Release Engineering (RE). Since changed elements need to be released as part of an existing product configuration, [Inkermann et al. \(2018\)](#) emphasize the strong interaction of ECM and RE with Configuration Management (CM). CM tries to ensure the compatibility of changed elements in a product configuration and demands a consistent and transparent documentation of all configurations.

2.1 Generic engineering change processes

To coordinate and control ECs within design projects, companies have established EC processes. Key element and trigger of these processes are change requests ([Maurer et al., 2018](#)). Figure 1 visualizes two generic EC processes based on [Riviere et al. \(2003a\)](#) and [Jarratt et al. \(2004a\)](#). The process described by Riviere *et al.* comprises seven tasks as part of three main phases named “EC proposal”, “EC investigation” and “EC embodiment”. Jarratt *et al.* distinguish between six tasks and three phases named “before approval”, “during approval” and “after approval”. Although both processes might have different focuses, they are similar in particular with reference to task two, three and four. After initializing the change request, these elements represent the three main tasks (MT) towards a technical solution and are the basis for the following implementation of the EC (last phase of the EC process). The content of these main tasks can be summarized as follows (see also Figure 1): defining (elements of) the solution space (MT1), analyzing possible solutions and impacts (MT2) and defining the solution of an EC (MT3).

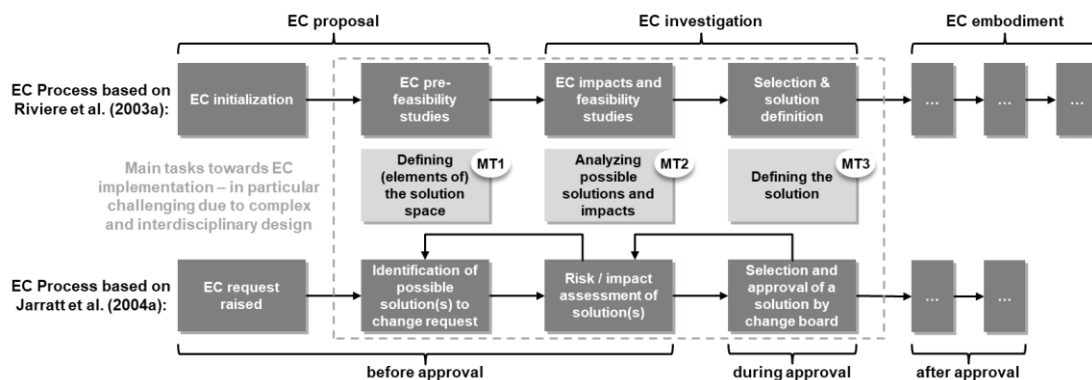


Figure 1. Generic EC processes and challenging main tasks (MT) due to higher complexity

Performing the three main tasks described before becomes challenging with regard to products of higher complexity and interdisciplinary design processes, since domain-specific technical decisions more frequently cause cross-domain effects. This can be illustrated using the following example: due to problems with the vehicle’s stiffness, an EC request with reference to the windshield - today a structural part of the vehicle’s body - is raised. In order to achieve the required stiffness, modified characteristics of the windshield (e.g. differing thickness) are proposed by the body department in main task one (MT1). Since the EC request is raised and a technical solution is found within the mechanical domain, no cross-domain impacts are considered in main task two (MT2) and the solution is approved in main task three (MT3). After the implementation of the EC (last phase of the EC process), problems with the vehicle’s rain sensor - a component developed in the electric/electronics domain - occur. These problems are caused due to a neglected cross-domain dependency visualized in Figure 2.

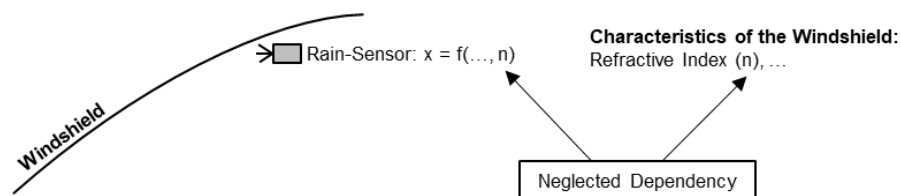


Figure 2. Example of cross-domain dependencies to be considered within ECM

As shown in Figure 2, the ability of the sensor to receive and interpret signals correctly, depends on the refractive index of the windshield. Since the refractive index has changed during the modification of the windshield’s characteristics, the functionality of the rain sensor is influenced. This interrelation

between the mechanical and the electric/electronics domain was neglected in main task two (MT2), since the modification of the windshield seemed to be a domain-specific issue. As illustrated by this example, the ECM of one specific engineering domain is established and an existing competence of many companies. However, understanding and managing cross-domain ECs for highly complex products is an increasing challenge. As a starting point to address this challenge, approaches to classify and cope with ECs described in literature are presented in subsections 2.2 and 2.3.

2.2 Classification approaches for engineering changes

In order to better understand engineering changes, various approaches to describe and classify ECs are used in literature. As a basis for strategies in ECM (subsection 2.3), this subsection provides an overview of identified attributes characterizing ECs (summarized in Table 1).

Table 1. Attributes to classify engineering changes

EC Attribute	Example / Markedness	Relevant Literature (exemplary)
Cause	emergent EC (e.g. insufficient product quality), initiated EC (e.g. new customer demands)	Lindemann and Reichwald, 1998; Eckert <i>et al.</i> , 2004; Jarratt <i>et al.</i> , 2011
Stakeholders causing ECs	external (e.g. customers or legislators), internal (e.g. engineering, manufacturing)	Jarratt <i>et al.</i> , 2011; Chucholowski <i>et al.</i> , 2012
Effects (negative)	additional product costs, changing time schedules	Fricke <i>et al.</i> , 2000; Chucholowski <i>et al.</i> , 2012
Objectives (positive effects)	enhanced product functionality, quality improvements	Riviere <i>et al.</i> , 2003a; Chucholowski <i>et al.</i> , 2012
Lifecycle stage	concept phase, production phase	Fricke <i>et al.</i> , 2000; Eckert <i>et al.</i> , 2004
Urgency	immediate, mandatory, convenience	Lindemann and Reichwald, 1998; Jarratt <i>et al.</i> , 2011

A major attribute of an EC is its **cause**. In this context Eckert *et al.* (2004) and Jarratt *et al.* (2011) differentiate between emergent and initiated changes. Whereas emergent changes are caused by product weaknesses (e.g. in terms of quality), initiated changes are based on changing product requirements. A changing product requirement might result from upcoming innovations (Lindemann and Reichwald, 1998), but also from new customer demands or legal regulations (Eckert *et al.*, 2004) for example.

An additional classification in the context of emergent changes (not explicitly listed in Table 1) made by Deubzer (2005) refers to the question if an EC could have been avoided (by the organization). Considering initiated changes as changes that can be seen as essential for market acceptance and legal conformity of the product, this classification is not applicable. Nevertheless this attribute might be an additional and useful classification criteria with reference to the cause of an EC, because it indicates potential fields of action for the management of emergent changes in future projects.

With reference to initiated and emergent changes, a valid classification of ECs can be made regarding the **stakeholders causing a change**. These stakeholders might be customers, legislators or suppliers. Apart from external also internal stakeholders like marketing and sales, engineering, manufacturing as well as the quality, purchasing or finance department can cause an EC (Jarratt *et al.*, 2011; Chucholowski *et al.*, 2012). Following a survey by Chucholowski *et al.* (2012) the most important stakeholders causing ECs are customers and market trends, R&D, legal regulations as well as marketing and sales.

As described before, ECs are often perceived as negative due to the consequences they have. Frequent **effects** (as third classification approach) are additional product costs, changing time schedules and a lack of consistent information in the development process (Fricke *et al.*, 2000). Chucholowski *et al.* (2012) also describe negative effects with reference to the people involved, since frequent ECs might have an influence on the motivation or reputation of the responsible engineer. Apart from these negative effects, also positive effects - that can also be described as objectives of an EC - can be identified. **Objectives** of ECs are for instance the integration of new technologies, enhanced product functionality, quality improvements, a better positioning of the product with reference to competitors or cost reductions in the long term (Riviere *et al.*, 2003a; Chucholowski *et al.*, 2012). This understanding of ECs corresponds the concept of Release Engineering as proposed by Inkermann *et al.* (2018).

In particular negative effects of an EC strongly depend on the product's **lifecycle stage**, when the EC is supposed to be implemented (fifth criterion for classification in Table 1). Nowadays most of the ECs even for complex products like aircrafts are requested before the customer gets in touch with the product (Riviere *et al.*, 2003b). However, since the product gets more integrated and time is becoming more critical during the ongoing design process, in particular ECs late in the design phase can have a significant influence on the product's costs (Eckert *et al.*, 2004). This phenomenon, stating that the costs of an EC increase by the factor of 10 for each phase in the design process, is described as "Rule of Ten" (e.g. Fricke *et al.*, 2000).

As last criterion for classification the **urgency** of ECs is identified. Lindemann and Reichwald (1998) differentiate between changes that are necessary or even mandatory (e.g. for safety reasons) and changes that are optional (e.g. enhanced functionality). A similar distinction is made by Jarratt *et al.* (2011) describing the urgency of ECs using categories named "immediate", "mandatory" and "convenience".

Although there might be other classification criteria, the criteria presented in the text and in Table 1 are most common in literature and a valid starting point to determine promising approaches in ECM, which is explained in more detail in section 2.3.

2.3 Strategies to cope with engineering changes

Based on classification approaches summarized in Table 1, strategies to avoid or cope with ECs can be defined. To derive these strategies and the measures they contain, detailed analysis of specific attributes defined in section 2.2 are needed. The relations between EC attributes and strategies can be illustrated using five strategies defined by Fricke *et al.* (2000) as an example. The first strategy called **prevention** in particular is based on analyzing involved *stakeholders*, since the amount of changes in this strategy is supposed to be reduced for instance by improved communication and awareness of these stakeholders. **Front-loading** is a strategy resulting from analysis of the product *lifecycle stage* when an EC occurs and is in particular pursued by companies because of the Rule of Ten. **Effectiveness** predominantly focuses *causes* and *effects* of ECs, demanding more intensive analysis, if an EC is required and beneficial from a technical point of view. **Efficiency** for instance analyzes if the way *stakeholders* (can) communicate and how (their) resources are used can be improved, whereas **learning** is a strategy demanding comprehensive EC analysis.

As exemplary shown before, EC strategies (see also e.g. Terwiesch and Loch, 1999; Deubzer *et al.*, 2006; Lindemann, 2016) can be derived by and defined based on certain attributes of an EC. Resulting measures within a strategy thereby target specific dimensions. Following Browning *et al.* (2006) substantial and highly interrelated dimensions to achieve the companies goals are the product (as "desired result of the project"), processes (as necessary steps "to produce the product") and the organization (as "people assigned to do the work to produce the product"), which can also be seen as key determinants for ECM. Table 2 summarizes the strategies proposed by Fricke *et al.* (2000) as well as measures defined within these strategies. Although these measures are not limited regarding their target dimensions, a main target dimension (product, process, organization) they in particular focus can be defined (see last column in Table 2).

Table 2. Strategies and measures in ECM (Fricke *et al.*, 2000) and their target dimensions

ECM Strategy	Strategy Measures	Target Dimension
Prevention ("number of changes" as target figure)	Improve communication and awareness	Organization
	Make decisions only under high degree of information	Process
	Strengthen decision discipline	Organization
Front-loading ("point of time" as target figure)	Prohibit optional late changes	Process
	Strengthen decision discipline	Organization
	Integrate user early in the design process	Process
	Improve change prediction	Product
Effectiveness ("ratio of effort to benefit of changes" as target figure)	Reduce subjectivity in decisions	Organization
	Improve and use methods for rational decision making	Process
	Enhance effect analysis	Product

Efficiency ("time and cost of a change" as target figure)	Improve (early) information on and documentation of ECs	Process
	Implement adaptive EC processes	Process
	Incorporate changeability into the product	Product
Learning ("quality of product development and change management" as target figure)	Analyze already implemented ECs	Product
	Analyze former projects	Process
	Reuse and transfer knowledge	Process

As illustrated and confirmed in Table 2, all target dimensions - process, organization, product - can be considered relevant for ECM. This also becomes evident, considering that a strategy and its measures in most cases address several dimensions or at least need to involve other dimensions to generate improvements. Since a detailed understanding and a suitable representation of the product is an essential prerequisite to implement adequate EC processes and organizational structures, this target dimension will be focused in this paper. Regarding the product and its representation a key challenge in ECM is change propagation, which is explicitly addressed by two measures in Table 2 (improve change prediction, enhance effect analysis) and will be further discussed in section 3.

3 CHANGE PROPAGATION AS KEY CHALLENGE IN ECM

Already in 1998, Lindemann and Reichwald differentiated ECs only referring to one component - also described as local changes - and interface crossing ECs. The latter so-called change propagation is an increasing challenge in the context of ECM, since the assessment of knock-on effects of an EC on other product elements is complicated for complex products. However, according to a study by Chucholowski *et al.* (2012) only about 50% of the companies try to analyze risks and impacts of their ECs. Although the numbers might differ, there are other references confirming that detailed analysis of change propagation or impacts are rather uncommon for most companies (Jarratt *et al.*, 2011). Considering the fact that companies spend up to one third or even half of the R&D capacity to manage ECs (Chucholowski *et al.*, 2012), the relevance of methods for comprehensive analysis of change impacts between components becomes evident. In order to better understand the phenomenon of change propagation, section 3.1 will present different types of change propagation behavior. Approaches to manage change propagation will then be discussed in section 3.2.

3.1 Change propagation behavior

With regard to change propagation, direct and indirect change propagation has to be differentiated. Whereas direct changes propagate from one component to another (from A to B), indirect changes propagate via an intermediate component (from A to C to B) (Clarkson *et al.*, 2001). Considering these examples, Fricke *et al.* (2000) introduced the term "cause-effect net of changes" describing that a component affected by a change can become the cause for the change of another component. With reference to the behavior of components in the context of change propagation, Eckert *et al.* (2004) differentiate between change constants, absorbers, carriers, multipliers and buffers (see Figure 3a).

Whereas **change multipliers** cause more changes than they absorb, **change absorbers** reduce the change problem. **Change carriers** do not enlarge or reduce the change problem, since they cause as many changes as they absorb. **Change constants** do not have an influence on the change problem, because they neither absorb nor cause changes. In contrast to that, **change buffers** can have an influence. Depending on their tolerance range they either become change absorbers or multipliers. An additional type (not visualized in Figure 3a) to describe the behavior of a component in the context of change propagation are change resistors. **Change resistors** are defined not to be changed under any circumstances, because it is foreseeable that changing these components would have significant consequences. (Eckert *et al.*, 2004)

Depending on the change propagation behavior of involved components, the number of changes over time for a specific net of changes develops. As illustrated in Figure 3b, change ripples, blossoms and avalanches can be distinguished in this context (Eckert *et al.*, 2004). While change ripples and blossoms finish within the planned time frame, the number of changes in a change avalanche is not controlled and reduced on time. Due to unforeseen change multipliers, **change avalanches** are difficult to predict and can lead to significant additional costs. Although **change blossoms** involve

many affected components, they - in contrast to change avalanches - finally lead to change absorbers and the process stays under control. In a **change ripple** only a few components or systems are affected, so that the number of changes can be reduced relatively fast.

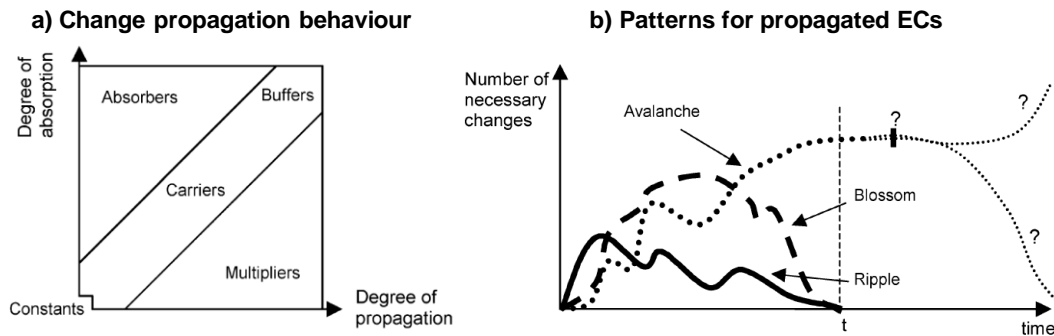


Figure 3. a) Types of change propagation behavior and b) Patterns for propagated engineering changes (Eckert et al., 2004)

In particular regarding possible time delays and additional costs caused by change avalanches, the need to understand and manage change propagation becomes evident. Therefore, existing approaches to manage change propagation are presented in the next section.

3.2 Approaches to manage change propagation

Since the management of change propagation is crucial for ECM (Jarratt et al., 2004c), there are diverse approaches in literature supporting analysis and assessment of change effects. However, managing change propagation is difficult, since (almost) nobody completely understands a highly complex product in detail (Clarkson et al., 2001). In addition to that, adequate approaches also depend on the product type, taking into account that approaches from a software perspective differ from approaches addressing hardware driven products. In order to derive suitable approaches to manage cross-domain change propagation, in particular dependencies between product elements out of the mechanical, electric/electronics and software domain need to be focused and represented.

In the context of hardware driven products, many approaches to manage change propagation use design structure matrices (DSMs) to visualize dependencies between different components (see e.g. Clarkson et al., 2001; Keller et al., 2005; Giffin et al., 2009), because of the simple format a DSM offers. Dependencies and therefore possible ways of change propagation can for instance be illustrated using crosses or numerical values in the cells of the DSM. Numerical values can thereby represent the likelihood that a change propagates from component A to B, the impact this propagation might have or the risk (combination of likelihood and impact) of the dependency between A and B (Clarkson et al., 2001). In order to illustrate these approaches, different types of DSMs are visualized in Figure 4.

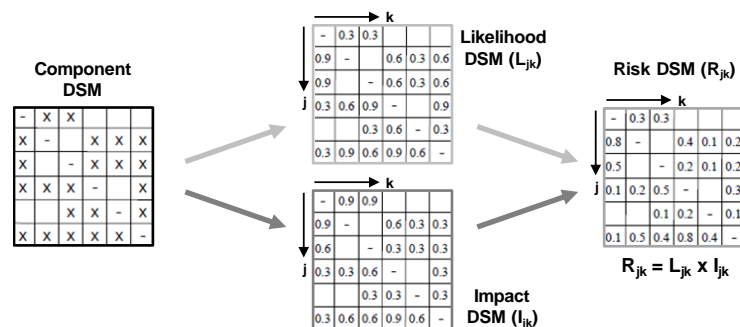


Figure 4. Likelihood, impact and risk DSMs (based on Clarkson et al., 2001)

Although many dependencies can be visualized using DSMs, there are disadvantages in particular with reference to highly complex products. On the one hand, this is due to the fact, that the DSM of a product consisting of many components can become difficult to read because of its size (Jarratt et al., 2004b). On the other hand, DSMs are rather unsuitable to represent indirect links (Keller et al., 2005) and the type of dependency (spatial, thermal, etc.) the components have. In order to consider different interfaces between components, for instance Yildirim and Campean (2013) describe if material (M),

energy (E) or information (I) are exchanged between components. An additional interface type defined as P illustrates if a physical (i.e. spatial) relation between components exists. To properly visualize different interfaces of one component to other components, interface matrices (IM) as shown in Figure 5 can be used.

EXTRACTED INTERFACE MATRIX OF BATTERY PACK		INTERNAL INTERFACES						EXTERNAL INTERFACES								
		A	B	C	D	E	F	E1	E2	E3	E4	E5	E6	E7		
	Battery Pack	Battery Charger	Battery Pack	Driver Interface	Motor Control Unit	Motor	DC-DC Converter	Driver	240 V PIP	Cab	Environment	Chassis Frame	Rear Axle	Fuse Box		
2	Battery Pack	E			E		E	P			P	E	P	E		
		I						I			M	M				

Figure 5. Interface matrix of a vehicle's battery pack (Yildirim and Campean, 2013)

In addition to these matrices, Yildirim and Campean (2013) use interface analysis tables (IAT) to present further information of an interface between two components. In these tables for instance the target value for and a detailed description of the exchange as well as the function of the interface (e.g. transmit electrical power) are summarized.

Although the described approaches can indicate cross-domain dependencies to a certain degree (i.e. interface type E suggests the involvement of the electric/electronics domain), they do not specify these dependencies detailed enough. As a consequence it is not clearly defined, which type of dependency or interface demands a cross-domain interaction and collaboration of two or more engineering domains in the design process. This will be further discussed in section 4.

4 DISTINCTION OF DOMAIN-SPECIFIC AND CROSS-DOMAIN LINKAGE TYPES TO SUPPORT ENGINEERING CHANGE MANAGEMENT

As shown in the sections before, linkages between components and (sub)systems that are relevant for ECM can be direct or indirect. In addition, linkages can be evaluated using values for the likelihood, impact and risk that an EC propagates between components. However, existing approaches do not properly indicate, which engineering domains - mechanical (ME), electrical (EE) or software engineering (SWE) - are affected in the case of change propagation. As described, this lack of information can be reduced using interface matrices and interface analysis tables. Nevertheless, a clear distinction of domain-specific and cross-domain linkage types is missing.

Domain-specific linkage types thereby indicate that solutions for and possible impacts of an EC (see in particular MT 2 in Figure 1) can be evaluated within one engineering domain (e.g. only ME). In contrast to that, **cross-domain linkage types** demand cross-domain communication and collaboration (e.g. between the ME and EE domain) within the EC process. Based on a literature analysis and a linkage type classification (see Wilms *et al.* (2019) for further details), a set of basic linkage types for technical systems distinguishing domain-specific and cross-domain linkage types was derived. Within this set, the linkage types “material transfer”, “transfer of loads, forces or torques”, “translational or rotational movement”, “magnetic field” (all allocated to ME domain), “electrical earth” and “electrical power” (EE domain) can be classified as domain-specific. In contrast to that, “spatial relations or physical connections”, “vibrations and acoustics”, “electric or electromagnetic fields”, “thermal relations” and “information or control signals” are cross-domain linkage types, since component properties or behavior relevant for different engineering domains can be affected by an EC. Although this set of eleven basic linkage types has not been applied and evaluated in detail yet, it was validated using the vehicle as exemplary product (see also Wilms *et al.*, 2019). Based on this validation, an additional cross-domain linkage type named “optical” was identified. To better understand the defined cross-domain linkage types, Table 3 presents these types and exemplary illustrates the need for cross-domain interaction in the case of an EC. Focusing for instance line two and five, it additionally becomes apparent that besides involved domains, another important information regarding the linkage types refers to the question, if a linkage type is needed (line 5) or rather interfering (line 2) to provide e.g. a certain function. With reference to this aspect further details are given for instance by Yildirim and Campean (2013), who use a five-tier scale to rate linkages. A procedure to assess the cross-domain nature of a specific linkage type is discussed in an additional paper by Wilms *et al.* (2019).

Table 3. Cross-domain linkage types (based on Wilms et al., 2019)

Cross-Domain Linkage Type	Example and Domain Relevance (ME, EE or SWE)
Spatial relation or physical connection	Rain sensor (EE) being attached to the windshield (ME)
Vibrations and acoustics	Windshield (ME) transferring vibrations on rain sensor (EE)
Electric or electromagnetic field	Electromagnetic field (EE) attracting mechanical components (ME)
Thermal relation	Temperature exchange between mechanical components (ME) and electric cables (EE) or vice versa
Information or control signal	ECU-Software (SWE) interpreting signals of the rain sensor (EE)
Optical signal	Windshield (ME) refracts the optical signal of the rain sensor (EE)

In order to support engineers in decision making during the design process, a complete set of linkage types for the specific product needs to be derived and validated (exemplary shown before). In a second step, an adequate product model including these linkage types has to be created. Since many different product elements as well as differing linkage types need to be considered, the product model can only be realized using suitable modeling languages (e.g. SysML). Apart from that, the product model needs to be created considering specific requirements of the task it has to support. For instance, a product model supporting cross-domain ECM might therefore differ from a product model supporting consistent requirement specification. This tailoring to derive suitable approaches is also relevant for other target dimensions presented in section 2.3. As a consequence, defining a complete set of linkage types and a corresponding product model is an important step towards an ECM for highly complex products. Nevertheless, other target dimensions (i.e. the process and organizational dimension) not explicitly addressed in this paper also need to be considered to create comprehensive approaches effectively supporting engineers in the design process.

5 CONCLUSION AND FUTURE WORK

Managing engineering changes is a substantial and challenging part of modern product design. However, in particular the increasing relevance of electrics/electronics and software lead to rising product complexity e.g. in automotive engineering. This increasing product complexity is accompanied by the need to adapt R&D competences and to integrate and coordinate different engineering domains in a common design process. As a consequence, companies have to face challenges in the product, process and organizational dimension, whereas in particular analyzing solutions for and possible impacts of an EC is a complicated task. With reference to the product dimension, this task is challenging because of change propagation leading to unconsidered impacts of an EC on other product elements. To identify and evaluate potential impacts different approaches like risk DSMs and interface matrices as well as interface analysis tables are described in literature. Although these matrices and tables offer approaches to assess risks and possible ways of change propagation, they do not clearly indicate when cross-domain interaction and collaboration is needed. As a consequence, the distinction of domain-specific and cross-domain linkage types describing different dependencies between components and potentially affected engineering domains is proposed in this paper. Additionally, an initial set of cross-domain linkage types is presented and illustrating examples are given. However, to derive comprehensive ECM approaches for highly complex products and interdisciplinary design processes, a complete set of linkage types in a suitable product model needs to be created. To implement and use this model, the process and organization as additional dimensions have to be considered. Thereby adaptive EC processes for different types of ECs and linkages as well as corresponding organizational structures need to be derived.

Disclaimer: The results, opinions and conclusions expressed in this paper are not necessarily those of Volkswagen Aktiengesellschaft.

REFERENCES

- Albers, A., Kurrle, A. and Klingler, S. (2016), "The Connected Car: A system-of-systems: Exploration of challenges in development from experts view", *16th International Stuttgart Symposium*, Stuttgart.
- Automobilwoche (2018), "Kennzahlen des Wandels. Unternehmen, Innovationen, Umsätze - schon wenige Ziffern zeigen, wie sich die Autoindustrie verändert", *Automobilwoche*, Vol. 29 No. October, p. 28.

- Browning, T.R., Fricke, E. and Negele, H. (2006), "Key concepts in modeling product development processes", *Systems Engineering*, Vol. 9 No. 2, pp. 104–128.
- Burkacky, O., Deichmann, J., Doll, G. and Knochenhauer, C. (2018), "Rethinking car software and electronics architecture", available at: <https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/rethinking-car-software-and-electronics-architecture> (accessed 4 May 2018).
- Chucholowski, N., Langer, S., Ferreira, M. and Forcellini, F.A., Maier, A. (2012), "Engineering change management report 2012", *CiDaD-Working Paper Series*, 08th ed.
- Clarkson, P.J., Keller, R. and Eckert, C.M. (2005), "Multiple views to support engineering change management for complex products", *3rd International Conference on Coordinated & Multiple Views in Exploratory Visualization - CMV 2005*, London.
- Clarkson, P.J., Simons, C. and Eckert, C.M. (2001), "Predicting change propagation in complex design", *2001 ASME Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Pittsburgh.
- Deubzer, F. (2005), "Der Änderungsmanagement Report 2005", *CiDaD-Working Paper Series*, 1st ed.
- Deubzer, F., Kreimeyer, M. and Lindemann, U. (2006), "Exploring strategies in change management: Current status and activity benchmark", *International Design Conference - Design 2006*, Dubrovnik.
- Eckert, C.M., Clarkson, P.J. and Zanker, W. (2004), "Change and customisation in complex engineering domains", *Research in Engineering Design*, Vol. 15 No. 1, pp. 1–21.
- Eigner, M. (Ed.) (2014), "Modellbasierte Virtuelle Produktentwicklung", Springer-Verlag, Berlin Heidelberg.
- Fricke, E., Gebhard, B., Negele, H. and Igenbergs, E. (2000), "Coping with changes. causes, findings and strategies", *Systems Engineering*, Vol. 3 No. 4, pp. 169–179.
- Gerst, M., Eckert, C.M., Clarkson, P.J. and Lindemann, U. (2001), "Innovation in the tension of change and reuse", *International Conference On Engineering Design - ICED 2001*, Glasgow.
- Giffin, M., Weck, O., de Bounova, G., Keller, R., Eckert, C. and Clarkson, P.J. (2009), "Change propagation analysis in complex technical systems", *Journal of Mechanical Design*, Vol. 131 No. 8.
- Inkermann, D., Huth, T. and Vietor, T. (2018), "Towards cross-domain release engineering: potentials and challenges for automotive industry", *10th International Conference on Adaptive and Self-Adaptive Systems and Applications - ADAPTIVE 2018*, Barcelona.
- Jarratt, T., Eckert, C.M., Caldwell, N. and Clarkson, P.J. (2011), "Engineering change. an overview and perspective on the literature", *Research in Engineering Design*, Vol. 22 No. 2, pp. 103–124.
- Jarratt, T., Eckert, C.M. and Clarkson, P.J. (2004a), "Engineering Change", in Clarkson, P.J. and Eckert, C.M. (Eds.), *Design process improvement: A review of current practice*, Springer-Verlag, London.
- Jarratt, T., Eckert, C.M. and Clarkson, P.J. (2004b), "Development of a product model to support engineering change management", *International Symposium on Tools and Methods for Concurrent Engineering 2004*, Lausanne.
- Jarratt, T., Eckert, C.M. and Clarkson, P.J. (2004c), "Pitfalls of engineering change: change practice during complex product design", *14th CIRP Design Seminar*, Cairo.
- Keller, R., Eger, T., Eckert, C.M. and Clarkson, P.J. (2005), "Visualising Change Propagation", *International Conference On Engineering Design - ICED 2005*, Melbourne.
- Lindemann, U. (Ed.) (2016), "Handbuch Produktentwicklung", Carl Hanser Verlag, München.
- Lindemann, U. and Reichwald, R. (1998), "Integriertes Änderungsmanagement", Springer-Verlag, Berlin Heidelberg.
- Maurer, M., Winner, H. and Prokop, G. (Eds.) (2018), "Automotive Systems Engineering II", Springer International Publishing AG.
- Riviere, A., DaCunha, C. and Tollenaere, M. (2003a), "Performances in Engineering Changes Management", in Gogu, G., Coutellier, D., Chedmail, P. and Ray, P. (Eds.), *Recent Advances in Integrated Design and Manufacturing in Mechanical Engineering*, Springer, Dordrecht, pp. 369–378.
- Riviere, A., Féru, F. and Tollenaere, M. (2003b), "Controlling product related engineering changes in the aircraft industry", *International Conference On Engineering Design - ICED 2003*, Stockholm.
- Terwiesch, C. and Loch, C.H. (1999), "Managing the process of engineering change orders. The case of the climate control system in automobile development", *Journal of Product Innovation Management*, Vol. 16 No. 2, pp. 160–172.
- Wilberg, J., Elezi, F., Tommelein, I.D. and Lindemann, U. (2015), "Using a systemic perspective to support engineering change management", *Procedia Computer Science*, Vol. 61, pp. 287–292.
- Wilms, R., Cemmasson, V.F., Inkermann, D. and Vietor, T. (2019), "Identifying Cross-Domain Linkage Types to Support Engineering Change Management and Requirements Engineering", *29th CIRP Design Conference (submitted Paper)*, Póvoa de Varzim, Portugal.
- Yildirim, U. and Campean, F. (2013), "An Enhanced Interface Analysis Method for Engineering Change Management", in Abramovici, M. and Stark, R. (Eds.), *Smart Product Engineering*, Springer-Verlag, Berlin Heidelberg, pp. 191–200.