Domain-Spanning Change Propagation in Changing Technical Systems

Bauer, Wolfgang[1]; Nepomuk, Chucholowski[1]; Lindemann, Udo[1]; Maurer, Maik[1]

[1]Institute of Product Development, Technische Universität München

Technical products are facing different changes during their lifecycle. These changes can have drastic impacts to the product, either towards its structure or the effort for the implementation. If platform elements used in various products within a product family, the change impact can be capital. Based on existing methods, we developed an approach to propagate changes, triggered by a specific cause, in technical systems. The approach consists of matrix-based methods and allows the propagation via different domains, for example components, functions and variation attributes. Change causes can be linked more easily to the product architecture and the change impact can be traced via different relevant product domains. The approach is illustrated by an industrial case study shown in this contribution.

1. Introduction

Many external causes for product changes exist: country- or region-specific customer needs, legislation, customer habits, competitive products (Pahl et al., 2007) or different market strategies (Porter, 2008). Company internal triggers for changes can be different production sites with different production technologies, raising dynamics of innovations and technology as well as shortened development and release cycles (Ponn and Lindemann, 2011) as reaction to changes the environment.

Theses causes can be triggers for product changes. Often, changes of technical or functional elements can propagate through a product and affect components that are not directly connected to the initiating changed component (Keller et al., 2005). This leads to undetected changes on further component, causing late changes and high costs. To trace and detect undesired change propagations, product architecture offers a good basis.

The objective of this contribution is the development of an approach for a systematic analysis of the impacts of technical changes to product architecture. Starting with the cause of a change, its impact to the product architecture is analyzed. The impacts are traced via the domain of functions, components and between these domains. This allows an appropriate mapping of changes to the architecture.

Therefore, matrix—based methods from Structural Complexity Management are applied (Eppinger and Browning, 2012; Lindemann et al., 2009).

The paper is structured as follows: starting with an overview of existing methods for the analysis of change propagation in section 2, the developed approach is presented in section 3 and evaluated by an industrial case study in section 4. The paper concludes with a discussion and an outlook for future research.

2. Theoretical background

Changes in technical systems can be analyzed by using different established methods. A short overview over the most relevant methods for the analysis of change impacts and propagation is presented in this section.

The Change Mode and Effect Analysis (CMEA), presented by (Keese et al., 2006; Palani Rajan et al., 2003), serves the assessment of the flexibility of products. The approach developed by (Köhler et al., 2008), CPM/PDD (Characteristics-Property Modeling/Property-Driven Development), is an approach to illustrate and compare the impact of changes. Another method, based on the CPM/PDD-approach and FMEA, is the Change Impact and Risk Analysis (CIRA), developed by (Conrad et al., 2007). Possible solutions for changes during the product lifecycle are analyzed and assessed regarding risk and impact.

The Change Propagation Analysis (CPA) according to (Giffin et al., 2009) serves the analysis of change requests in order to visualize the changes, the affected components and their role in change propagation.

The Change Propagation Method, developed by (Clarkson et al., 2001), determines the change impact on a product by the means of risk assessment. Based on the CPM, (Keller et al., 2005) developed tools for the visualization of the changes and their propagation.

Koh et al. (Koh et al., 2012) developed a change modelling method (CMM) which is based on the CPM and on the House of Quality (HoQ). The CMM assesses the effects of change options on product attributes by tracing change options via requirements to components.

All of the presented methods assess the impact on the components domain. Some of them use also use other domains for a better traceability of the change within the system. However, the use of indirect change propagation via the chain of change - from the change cause via functions to the components – in an integrated model is missing.

3. Approach for Domain-Spanning Change Propagation

The initial situation underlying this research is an existing product family which should be analyzed regarding the question which functions and components have to be changed due to trends or another expected changes in the future. It should be possible to regard the propagation of any kind of changes, e.g. changes due to failures, changing requirements, new market situations, etc. The change analysis should be conducted domain-spanning, meaning the change path should be traced via functions and components. The resulting change paths should also be visualized to create transparency about the change impact to the technical system.

The applied process is divided into four major steps (see Fig. 1): first step, possible causes for changes are acquired applying the context model (Langer and Lindemann, 2009). These changes are linked to the product architecture using a Multiple-Domain Matrix (DMM) in the next step. In the third step, the effects of the change to the product architecture are propagated across different domains in order to analyze the change impact using change propagation methods in the last step. The four steps are described in detail in the following sub-sections.

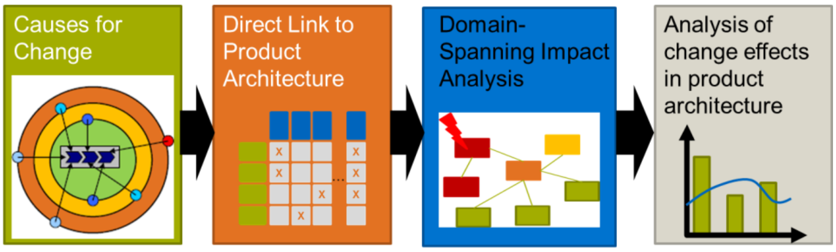

Figure 1 Approach for domain-spanning change propagation

3.1 Step 1: Causes for change

In the first step, possible changes are identified. These changes are triggered by influence factors which can emerge in the environment of a company (e.g. legislation, market, competitors) or within the company (organization, processes, resources). The occurring influencing factors are identified by applying the context model of (Langer and Lindemann, 2009). The context model serves the systematic search and documentation of influencing factors and is structured as a search matrix. The context is categorized into following five elements: environment, market, company interfaces, company, and development process. For every context category, classes of influence “technology / knowledge”, “socio-economics”, “politics / legislation”, “resources” are considered. As many influencing factors as possible are gathered by systematically going through the different search fields of the model. It is best practice to involve experts from different fields of the company to achieve a wide range of perspectives and consequently a high number of influencing factors that may trigger changes to the system under consideration.

3.2 Step 2: Direct link to product architecture

The influence factors triggering changes are linked to the product architecture. According to (Ulrich, 1995) the product architecture is defined as "(1) the arrangement of functional elements; (2) the mapping of functional elements to physical elements; (3) the specification of the interfaces among interacting physical components." In this case, the product architecture model is enlarged by the domain "variation attributes". These attributes represent the differentiating parameters of variants derived from the platform. Examples for variation attributes are size, performance, or color.

To create a link between the changes and the product architecture, again Domain-Mapping Matrices (DMM's) (Lindemann et al., 2009) are applied (see Fig 2). In the first DMM, one axis represents the causes, while the other axis shows the components: a change is connected to a component if the occurrence of these causes changes the component in question. The second DMM connects the causes with functions affected by the cause. In the third DMM, the changes are related to the variation attributes if a differentiation attribute of the product is affected in the means of a change by a certain influencing factor. The three DMMs (see Fig. 2) contain the information about the dependency of the causes and their object of change, the product architecture.

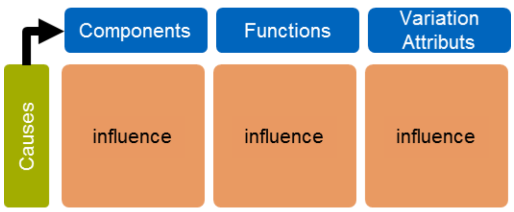

Figure 2 Matrices for creating the link between causes and the product architecture domains

3.3 Step 3: Domain-Spanning Impact Analysis

In this step, the change impact is propagated to the component domain as components are changed by the engineers. The interfaces have to be designed in a way so that changes do not spread within the whole system. But the creation of the direct link from the changes cause to affected components is not always obvious. It is more likely to identify which of the product's functions or variation attributes are affected by a cause. Therefore, these two domains serve as a translation to the components' domain.

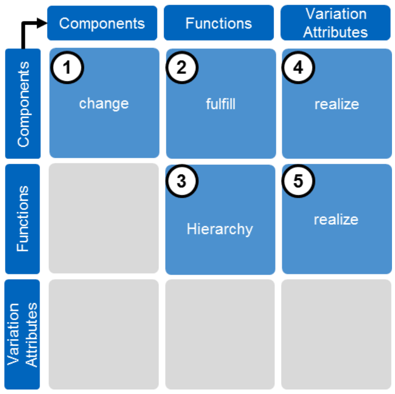

Figure 3 Product architecture model as MDM-representation

By using the product architecture model (see Fig. 3), the tracing of changes is possible via several domains. The model is represented in a Multiple Domain Matrix (Lindemann et al., 2009) and consists of three domains: components, functions, and variation parameters. The interactions of the components are represented by a bidirectional Design Structure Matrix (DSM) (Steward, 1981) in matrix 1. The type of relations between the physical components is "changes", meaning component A and B are connected if a change of one of these components changes the other component. This relation type delineates geometric contact, signal flow, material flow or energy flow (Pimmler and Eppinger, 1994). The second matrix, a Domain-Mapping Matrix (DMM) represents the link between components and functions. The type of relation is "fulfill". The hierarchic dependencies of the functions are captured in DSM 3 which includes main functions as well as their auxiliary functions. DMM 4 and 5 describe the dependencies between components and functions to variation attributes. Here, a dependency exists if a component/function realizes a variation attribute, e.g. component A realizes a certain performance factor.

The effects of a change can now be traced to the affected components by the DMM's in Fig.2 and navigation through the MDM to the components' domain. The change and propagation mechanism is shown in Fig. 4.

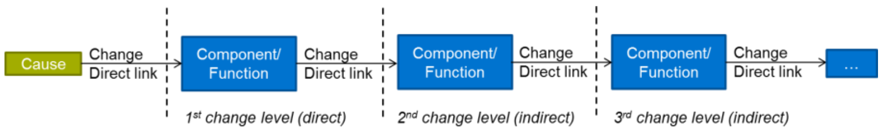

Figure 4 Change and propagation mechanism

The direct link of change causes to influenced elements (independent form the affected domain) is called "1st level influence and is directly influenced by the cause. All affected elements in the further levels are called according to the number of level, e.g. 2nd or 3rd level elements. These elements are indirectly influenced by the change via elements in higher levels.

3.4 Step 4: Analysis of change effects in product architecture

After the affected elements on all relevant levels of propagation are known – especially on the component domain – different analyses can be executed. A change propagates either in different domains or in the component domain. Based on this, it can be derived how strong the change impact to the product architecture is. A qualitative estimation of the change effort can be deduced. Another possibility is to calculate structural criteria such as the active and passive sum (Lindemann et al., 2009) of the elements in order to characterize their contribution to the change effect. The active sum of an element indicates how many other elements are influenced by this element. As the active sum gives notice of the change effect of an element to the system, changes on high active elements should be avoided. In contrary, the passive sum of an element describes how many other elements influence this element. Passive elements should be implemented in a robust way, so changes do not spread via these elements as they are sensitive to changes. This approach is similar to (Giffin et al., 2009) where elements are grouped into absorber, carriers or multipliers.

Moreover, the criticality of an element can be calculated by the multiplication of the active and passive sum. The criticality represents the role of an element within the structure based on the in- and outgoing dependencies. The higher the criticality, the more sensitive this element is against changes and can cause numerous changes to other elements, too.

Besides key figures, different visualization methods, e.g. from (Keller et al., 2005) can be applied for the analysis and interpretation of change impacts. Impacts can be visualized by networks or propagation trees. Such networks represent the dependencies between the elements. The elements can origin form different domains, edges represent the dependencies (inter or intra domain). The length of the edges can characterize different information, e.g. the combined risk or the shortest path (Keller et al., 2005).

For change propagation, not only components but also other domains can be used in different change levels. How many change levels are required for the analysis is dependent of the considered system. A domain-spanning propagation tree is used in this approach to visualize the impacts on different domains and different levels. With this representation, it can be determined how many elements are affected in which change level and when the propagation within the system ends.

After analyzing the change impact, the results are interpreted. An important step here comprehends the focus on the elements that are not only possibly affected (via dependencies) but in reality triggered by the change cause. If a change cause is a very general one, numerous elements can be affected in theory. By using design parameters (see e.g. (Ponn and Lindemann, 2011),(Eckert et al., 2004; Ehrlenspiel, 2009)) the user can narrow down the real change impacts. If a pressure or temperature is affected by a change cause, only components (or functions) with corresponding design parameters have to be considered. For conducting such a focusing on the affected elements, search strategies such as feed-forward analysis, impact check list, mine seeking or trace-back analysis (Lindemann et al., 2009) can be applied.

When knowing, which domains and their elements are affected by a change, strategies for handling these changes can be addressed. For example, different design strategies such as size ranges, functional integration or differentiation (Pahl et al., 2007), standardization, platform architectures (Ehrlenspiel, 2009), adaption mechanisms (Kissel, 2012) or the encapsulating of changes into modules by change-driven modularization (Bauer et al., 2013b) can be followed. If the change cause is underlying a cyclical behavior, approaches for handling repetitive tasks can be applied, such as a standardization of the change process for a more efficient execution.

If material, engineering, manufacturing and assembly costs are known, the financial effort of change can be assessed. The cost-effectiveness of a change or and potentials for a re-design can be identified.

4. Case study for the domain-spanning change analysis

4.1. Background of the case study

The case study is applied on a product family from the whites goods industry. The meta-model in a MDM-representation is given in Fig. 5.

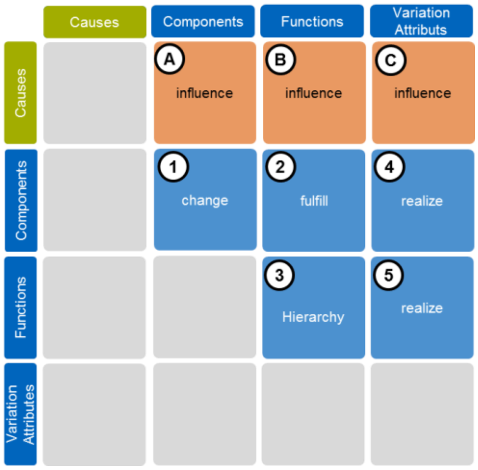

Figure 5 Meta-model of the case study

The different domains contain the following number of elements: in total, 127 causes build the domain “causes”. The domain “components” consists of 94 elements. The domain “function” comprises 11 main functions and 46 auxiliary functions, and the domain “variation attributes” includes 4 elements. The relation types between the domains are given in Fig. 5.

The analysis of the use case was classified according to three criteria:

- Component connectivity: a cause is linked to a component with a low, medium and high connectivity
- Number of directly affected elements: low, middle, high number of directly influenced (1^{st} level) elements (functions, components or variation attributes)
- Number of directly affected common elements: high number of functions and components affected by one or more causes.

In the following, the single steps of the presented approach are applied using the case study. Step 1 and 2 are the same for the use cases shown above; step 3 and 4 are described for each of the three different use cases. Due to reason of non-disclosure, the industrial example and its content is presented in an abstract wording.

4.2 Step 1: Causes for change

In the first step, causes for change for the considered system were gathered. Therefore, the context model (Langer and Lindemann, 2009) was applied in 7 moderated workshops with the industry partner. Each of these workshops, two or more employees from different division (strategic planning, marketing, engineering design, manufacturing etc.) participated. In total, 127 causes were acquired, 98 of them with a direct or indirect link to the technical system (the remaining are e.g. strategic, procedural or organizational causes).

4.3 Step 2: Direct link to product architecture

This step comprehends the direct link of the acquired causes for changes to the product architecture. A detailed process of its acquisition via expert workshops is shown in (Bauer et al., 2013a). All 98 system-related causes were linked to one of the domains “components”, “functions”, or “variation attribute“ using DMM’s. It became true that a direct link to components is not always possible, e.g. as the cause is very abstract. In these cases, especially the access via the product architecture model was very valuable. For example, the cause “development of connectivity of devices” cannot be mapped to components in a direct way, but is related to the function “provide operation”.

4.4 Step 3: Domain-Spanning Impact Analysis and Step 4: Analysis of change effects in product architecture

In this step, two exemplary cases are presented: on the one hand, the direct link of causes to components; on the other hand, the propagation via the link of causes to functions.

Direct link of causes to components

In this first use case, it is differentiated if a cause directly influences a component with a high, medium or low connectivity. For the assortment of suitable components, the components were ranked according their direct number of component interfaces (see Figure 6). The chosen components are highlighted in the figure.

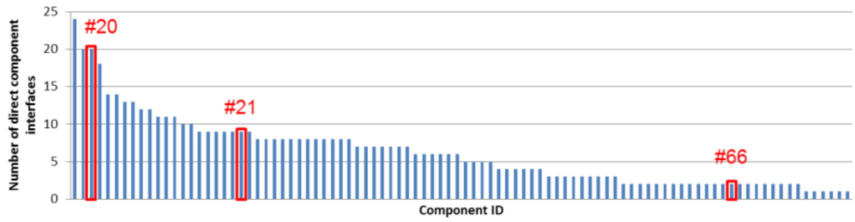

Figure 6 Component connectivity

Component 20 is an insulation foam material. In the considered case, it is changed by the cause “change of energy efficiency by local regulations”. The component has 20 direct interfaces and a criticality of 400, representing a highly connected component. Figure 7 shows the propagation tree for component #20.

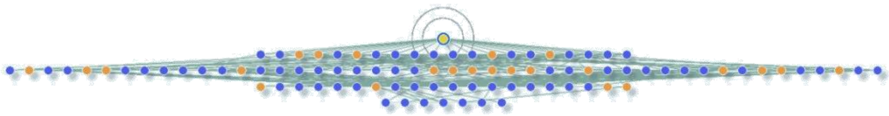

Figure 7 Propagation tree of component #20

The yellow node indicates component #20, the orange ones represent the affected components, blue the connected but not affected ones. It can be seen that this cause can lead to a change to the whole system already on the 4th level. If component #20 has to be changed because of the cause, already 71% on the second and 92% of the total amount of components on the third level are affected.

For the medium component connectivity, component #21, is chosen. This component has 9 direct interfaces and a criticality of 81. The criticality is over the average criticality of 59. This component can be directly changed by the cause “development of equipment”. If component #21 has to be changed, about 70% on the third and 97% of the total amount of components on the fourth level are affected.

Component #66 represents a sensor and a low connected component (two direct component interfaces, criticality = 4). The change triggering cause is e.g. “new development of sensors”. Because of the low activity of the considered component, less than one-third of the components are affected on the 2nd level, and the whole system is affected on the 5th change level.

Figure 8 shows the comparison of the change impact of the three considered components with different degrees of connectivity. The propagation on 1st and 2nd level of the high connected component is double then of the medium and low connected components. At the 3rd level, the impacts converge and on the 5th level at latest, the change has affected the whole system in all cases. A change of highly connected components should be avoided or done purposefully to reduce the change propagation effort. If possible, interfaces of such components should be standardized as the change will stop here and the component can be transformed from a multiplier/carrier to an absorber. The direct and indirect change impacts of low connected components are less than medium or high connected ones. Before implementing such a change, it must be ensured to avoid the impacts in the first and second change level to avoid an enormous spreading of the impact. If a change affects a highly connected component in an indirect way on 2nd level or higher, the same procedure can be executed. The result shows that changes on highly connected components lead to more change impacts via propagation. Therefore, components with a high criticality should not be changed or if it is unavoidable changed with a good coordination and a sophisticated impact analysis.

Deduced from this result, it is obvious that structural criteria such as the degree of connectivity or criticality are a suitable indicator for change propagation.

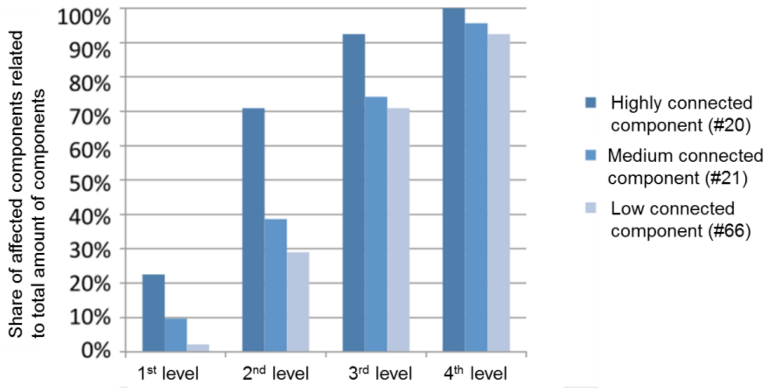

Figure 8 Comparison of three different connected components regarding their change impact

Indirect link of causes to components

This paragraph gives examples for a direct link of a cause to functions and then propagation to components as the target domain. This is done by tracing the direct linked components to the affected function(s). An example represents the cause "development in the field of consumer electronics" which influences the function "provide operation". This function is fulfilled by three components (#85: Human Machine Interface, #88: Local Harness, #89: System Harness).

The procedure for propagation is executed in the same manner as in the examples above. The resulting change impact is again in correlation with the connectivity of the starting element.

A cause which affects more than one function is „technological developments (of the core technology of the device)". 5 of 11 main functions (including 15 auxiliary functions) and in consequence, 46 of 94 components are affected by this cause.

To handle the high possible impacts on the system, the cause can be described in more detail or alternative technical solutions can be derived. It is then possible to isolate the direct link to the main/auxiliary functions, e.g. from 5 to 2 (depending on the situation). Consequently, the different technical alternatives regarding their change impact and consequently the qualitative change implementation effort can be compared.

5. Discussion

The case study shows that the approach is applicable and supports to investigate change propagation via different domains. The underlying architecture models allow an access and propagation between different domains. If a change affects a component, the same procedure can be applied to check if this change affects the functions realized by the component. The procedure can be used iteratively, until all possibilities of change are identified. Therefore, it delivers a solid basis for a change impact analysis and helps to compare different solutions regarding their change effort.

An important aspect regarding a target-oriented propagation is detailing the change impacts. Therefore, design parameters can be included on the edges of the networks to trace the impact more easily. In the presented case, the edges were not further detailed by design parameters as it is a high effort to do so. The cases proved that it is possible for a user of the approach – familiar with the considered system – to trace the change impacts within the system. The dependencies between the system elements help the user to navigate and use them as a kind of check list. By the inclusion of design parameters, the applicability could be enlarged to users not that familiar with the system.

6. Conclusion

The study showed that causes affecting a low number of components, the change propagation through the system is highly depending of the connectivity of the affected component(s). In case of many affected components, the focus of the change and the exclusion of non-changed elements are important to narrow down the analysis.

If a cause is linked to functions, the propagation is dependent of the number of realizing components. The functions can easily be classified regarding their change propagation by calculating their connectivity to components. The procedure for the analysis of domain-spanning propagation is iteratively applicable, independent of the considered domain and uses the connectivity of the elements within the system as a meaningful criterion. Moreover, the results showed that the number of directly influenced elements has an impact on the change propagation, regarding their scope and effort.

The causes were linked to different domains, which allow a reasonable access to the product architecture. Especially the domains “functions” and “variation attributes” are very helpful for creating this link, as a direct link to the affected components is hard to establish.

The reduction of the components from the possible to the actual affected ones was done by a concretization of the change cause, with knowledge of the propagation by the aid of the structural models. Because of the binary information of the structural models, the element connectivity and the change traceability is known.

7. Outlook

The presented approach delivers a qualitative assessment of the change and the consequent effort. Different solution for the implementation of the change can be compared, for example by the number of affected components. To enhance the assessment, the underlying approach and its models can be supplemented by further information: the effort in terms of time and internal costs can be acquired per component or a sum of components (affected by one cause). Therefore, it must be known which department within the company has which amount of effort for implementing the change.

Moreover, the information of the change impacts per cause is currently used to design "change-driven" component modules. Therefore, components are clustered if they are affected by the same cause(s). Mirroring the changes and their temporal behaviour against the company's future strategy, change modules can be established in the architecture for an efficient change, e.g. for a new product variant, and a fast market response.

Acknowledgments

We thank the German Research Foundation (Deutsche Forschungsgemeinschaft – DFG) for funding this project as part of the collaborative research center 'Sonderforschungsbereich 768 – Managing cycles in innovation processes – Integrated development of product-service-systems based on technical products'.

References

Bauer, W., Elezi, F., Maurer, M., 2013a. Application of DSMs for Analyzing Product and Organizational Structures, 15TH INTERNATIONAL DEPENDENCY AND STRUCTURE MODELLING CONFERENCE, DSM 2013. Hanser Verlag, MELBOURNE, AUSTRALIA.

Bauer, W., Elezi, F., Maurer, M., 2013b. An Approach for Cycle-Robust Platform Design, in: Lindemann, U., Venkataraman, S., Kim, Y.S., Lee, S.W. (Eds.), 19th International Conference on Engineering Design 2013 (ICED13), Seoul, Korea.

Clarkson, P.J., Simons, C., Eckert, C., 2001. Predicting change propagation in complex design, Design Engineering Technical Conference ASME, Pittsburgh.

Conrad, J., Deubel, T., Köhler, C., Wanke, S., Weber, C., 2007. Change Impact and Risk Analysis (CIRA) - combining the CMP/PDD theory and FMEA methodology for an improved engineering change management, International Conference on Engineering Design, Paris.

Eckert, C., Clarkson, P.J., Zanker, W., 2004. Change and customisation in complex engineering domains. Research in Engineering Design 15, 1-21.

Ehrlenspiel, K., 2009. Integrierte Produktentwicklung 4ed. Hanser, München.

Eppinger, S.D., Browning, T.R., 2012. Design structure matrix methods and applications. MIT press.
Giffin, M., Keller, R., Eckert, C., de Weck, O., Bounova, G., Clarkson, P.J., 2009. Change propagation analysis in complex technical systems. Journal of Mechanical Design 131, 081001.
Keese, D.A., Takawale, N.P., Seepersad, C.C., Wool, C.L., 2006. an enhanced Change Modes and Effect Analysis (CMEA) tool for measuring product flexibility with applications to consumer products, IDETC/CIE. ASME, Philadelphia.
Keller, R., Eger, T., Eckert, C., Clarkson, P., 2005. Visualising change propagation, 15th international conference on engineering design, Melbourne, pp. 62-63.
Kissel, M.S., P.; Lindemann, U., 2012. Design for Adaptability – Identifying Potential for Improvement on an Architecture Basis NordDesign 2012
Aalborg, Denmark
Koh, E.C., Caldwell, N.H., Clarkson, P.J., 2012. A method to assess the effects of engineering change propagation. Research in Engineering Design 23, 329-351.
Köhler, C., Conrad, J., Wanke, S., Weber, C., 2008. a matrix representation of the CPM/PDD approach as a means for change impact analysis, International Design Conference Dubrovnik, pp. 167-174.
Langer, S., Lindemann, U., 2009. Managing cycles in development processes-analysis and classification of external context factors, 17th International Conference on Engineering Design. Design Society, Stanford University, CA.
Lindemann, U., Maurer, M., Braun, T., 2009. Structural Complexity Management - An Approach for the Field of Product Design. Springer, Berlin.
Pahl, G., Wallace, K., Blessing, L., 2007. Engineering design: a systematic approach. Springer, London.
Palani Rajan, P.K., Van Wie, M., Campbell, M., Otto, K., Wood, K., 2003. Design for Flexibility - Measures and Guidelines, International Conference on Engineering Design, Stockholm.
Pimmler, T.U., Eppinger, S.D., 1994. Integration analysis of product decompositions, ASME Design Theory and Methodology Conference. Alfred P. Sloan School of Management, Massachusetts Institute of Technology, Minneapolis, MN.
Ponn, J., Lindemann, U., 2011. Konzeptentwicklung und Gestaltung technischer Produkte. Springer, Berlin.
Porter, M.E., 2008. Competitive advantage: Creating and sustaining superior performance. Simon and Schuster.
Steward, D.V., 1981. Systems analysis and management: structure, strategy and design. PBI (New York).
Ulrich, K., 1995. The role of product architecture in the manufacturing firm. Research policy 24, 419-440.