




GUIDING LOCAL DESIGN DECISIONS TOWARDS A FLEXIBLE AND CHANGEABLE PRODUCT ARCHITECTURE

L. Block 

University of Stuttgart, Germany

 lukas.block@iao.fraunhofer.de

Abstract

Flexibility and changeability are crucial when it comes to the design of embedded automotive architectures. However, flexibility and changeability are concepts rooted in the overall product and its objectives, while architectural design decisions may affect local subsystems as well as the overall system. Axiomatic design is applied to bridge this gap: The architecture is decomposed into its entities and changeability is described through the design's information content. Five domains of actions to foster changeability are identified and generic action schemes are derived.

Keywords: product architecture, design structure matrix (DSM), axiomatic design, flexibility, change prediction

1. Introduction

Flexibility and changeability are of growing importance when it comes to the design of automotive software and hardware architectures. Software updates and hardware upgrades can adapt the car's functionality throughout the life cycle to increase customer value (Guissouma et al., 2018). However, basic decisions with respect to the embedded architecture are made in early stages of product development: For example, the feature set, hardware topologies and the mapping of functions to electronic control units are defined about three to two years before start of production (Jaensch, 2012, p. 33-39).

Consequently, the challenge of automotive software and hardware architecture design is two folded: On the one hand, architecture development is concerned with meeting the functional and non-functional requirements, which are known and defined at the start of development. On the other hand, the decomposition of the system into its elements and interfaces must take place with little information about the future state of those requirements and the elements within the system. Hence, the architecture must incorporate flexibility and changeability to deal with possible future changes. However, this is challenging: Flexibility and changeability are concepts rooted in the overall product and its requirements (Schulz and Fricke, 1999). Architectural design decisions can affect local subsystems, e.g. single interfaces and elements, as well as the overall system, for example in terms of design guidelines. Therefore, the optimal degree of change anticipation varies throughout the system, composing a complex problem with multiple influencing variables.

The goal of this paper is to develop a methodology to understand the implications of flexibility and changeability for architectural design decisions in automotive software and hardware systems. It should guide architectural design decisions on a local and global scale and clear the way for the determination of the optimal degree of change risk mitigation.

2. Approach and methodology

Ulrich (1995) defines the product architecture as the union of the functional structure, the product structure and the scheme by which they are allocated to the each other. Within this paper, this definition is adapted and slightly widened: A product architecture is defined as the system of objectives, the scheme by which objectives are mapped to the product structure and the product structure itself, consisting of software and hardware elements (Figure 1). The extension with respect to the requirements is made in order to better represent the scope of architectural design decisions.

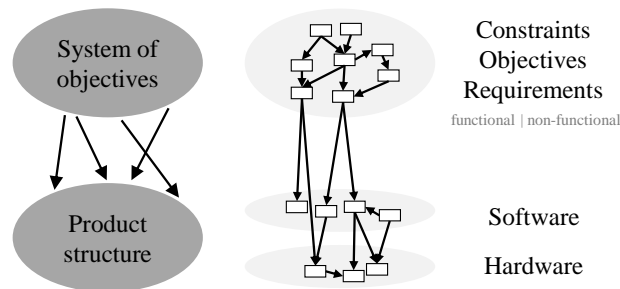


Figure 1. Scope of an embedded automotive architecture

Within the approach, the principles of axiomatic design developed by Suh (2001, 2005) are employed to define flexibility and changeability on a product's overall level. Subsequently, the perspective of axiomatic design is mapped to the above definition of an automotive software and hardware architecture. This allows depicting the influence of architectural design decisions on a local as well as on an overall-product scale. As a result, five domains of actions to foster changeability are identified and generic action schemes are derived.

2.1. Foundations

Axiomatic design was developed by Suh (2001, 2005) as a methodology to guide the design process by universal principles that govern the underlying behaviour of it. As such, axiomatic design views the product's design from four distinct perspectives (Suh, 2005, p. 22-23): The customer domain including the customer demands, the functional requirements (FR), the physical domain containing the so-called design parameters (DP) and the process variables. Usually, the domains are expressed through their characteristic vectors. The domains are related to each other through so-called mapping matrices (Suh, 2005, S. 23-24).

Axiomatic design guides the design process via two fundamental axioms: The independence axiom requires the separation of functional requirements when mapping them to the design parameters: Ideally, each functional requirement FR_i is fulfilled by exactly one design parameter DP_j . Hence, the mapping matrix between FRs and DPs must be of diagonal or triangular shape. Suh (2001, 2005) calls such a design uncoupled or decoupled. Optimal designs, which fulfil the independence axiom, can be generated through an approach called zigzagging (Suh, 2005, p. 27).

The second axiom – the information axiom – states, that out of all optimal designs, the one with the lowest information content must be chosen. The information content is defined in terms of the probability to successfully satisfy the FRs with the current set of DPs (Equation 1) (Suh, 2005, p. 30).

$$I_i = -\log_2(p_i) \quad (1)$$

I_i is the information content of the functional requirement FR_i . It is inverse to the probability p_i of successfully satisfying FR_i with the current set of DPs.

The information content of a complete design can be calculated by the sum of I_i for all FRs. Suh (2005, p. 37) suggests, that the probability distribution $\Pr(X)$, $X \in \{FR, DP\}$ depicts the likelihood of the system's DPs to fall into the design range of allowed FRs (Figure 2 (left)). Helander and Lin (2002) inverse this perspective and describe the functional requirements FR as a distribution of the desired system properties DP. The system's design range is viewed as the supplied range (Figure 2 (right)).

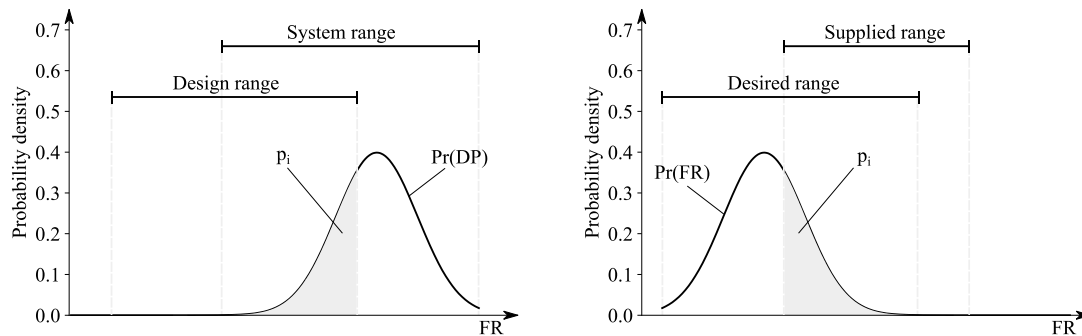


Figure 2. Components relevant for the determination of the information content according to Suh (2001) (left) and Helander and Lin (2002) (right)

Foith-Förster et al. (2016) adapt the approach of Helander and Lin (2002) to define flexibility and changeability: Flexibility is the information content determined by the distribution of the currently known requirements and the actual system design. In contrast, changeability is future-oriented. It is the information content calculated via the distribution of possible future requirements, covered by a system range, which is reachable through a change of the current design.

2.2. Flexibility and changeability in architectural design

In the following, the elements of axiomatic design are related to the components of the automotive software and hardware architecture. This makes it possible to derive implications for architectural design decisions on each layer locally. Overall impacts on flexibility and changeability are depicted via the information content of axiomatic design.

The system of objectives consists of all goals relevant for the product's development and of the constraints, imposed by the environment on the product's design (Muschik, 2011, p. 118). Objectives are under control of the company and are expressed in the form of requirements on the product level. Objectives are derived from constraints and form hierarchical and interactive structures (Muschik, 2011, p. 17). Consequently, the system of objectives OS represents the customer and functional requirement domain in axiomatic design. However, it loosens the strict separation between them and focuses more on the overall structure of objectives. Thus, OS covers the FRs. We denote them with OR (objectives and requirements) in the following, because non-functional requirements are also taken into account.

The product structure PS is divided into software and hardware. The hardware and software elements are connected physically or through a material flow, energy or information (Pimpler and Eppinger, 1994). The properties of these elements equal the design parameters DP of axiomatic design.

Within this paper, a change is understood as the adaption of an element or part of the embedded automotive architecture, which ultimately stems from a misfit between the current properties of the product DP and the desired properties OR. A controlled development process ensures that $OR \subseteq DP$ holds, i.e. that in the end of the development process all requirements are satisfied. However, the product does not fulfil all requirements at all times during development or product parts might fail in tests, causing a change of the product's structure. $OR \setminus DP$ is the set of necessary changes to fulfil the requirements (Figure 3). Inversely, $DP \setminus OR$ describes the overhead of the design: These are the properties or design parameters, which are present because of side effects for example, but are neither required nor forbidden.

Yet, a change can happen because product properties degrade or requirements will change over time: Figure 3 depicts this situation for a future change in the system of objectives at time t_1 . The set $(OR_1 \cap DP) \setminus OR_0$ represents properties that are now necessary and prevent a change, but which were previously labelled as overhead. These resources were intentionally or unintentionally hold available for previously unknown variations of the product. They describe the product's flexibility (Zäh et al., 2005). The area $OR_1 \setminus (DP \cup OR_0)$ is outside of this "flexibility corridor" (Foith-Förster et al., 2016). It can only be satisfied via a change of the product, which includes additional effort. Thus, we define flexibility as the ability of a system to adapt to a change in requirements without effort. In contrast, changeability is the system range reachable with additional effort.

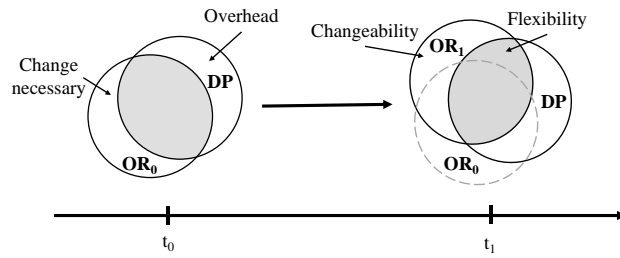


Figure 3. Development of the system of objectives and the product structure over time

As such, changeability is always linked to a level of effort for implementing the new system range. Consequently, multiple changeability system ranges $csr(e)$ exist (Figure 4). Formally, $csr(e)$ is the mapping of a certain amount of change effort e to system design ranges, which can be realized with it. The system's flexibility is described through the information content covered by the system's flexibility range $f_{sr} = csr(0)$.

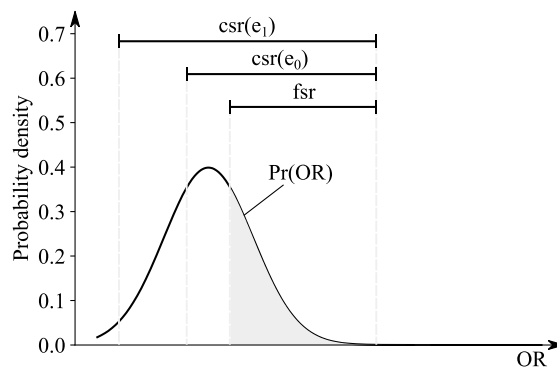


Figure 4. The flexibility range f_{sr} and different levels of changeability $csr(e_0)$ and $csr(e_1)$ with $0 < e_0 < e_1$ over the probability distribution of the objectives and requirements

The differential design matrix ∂A characterizes the $csr(e)$ -mapping. In axiomatic design, the design matrix A relates requirements to design parameters (Equation 2) (Suh, 2005, p. 24).

$$OR = A \cdot DP \quad (2)$$

Therefore, A depicts the product's architecture. A in its differential form ∂A describes the deviation in the design parameters when a change of objectives or requirements happens (Suh, 2005, p. 24). Thus, the following mathematical formulation of the change effort bridges the gap between architectural design decisions, located in A , and the change effort $e(\partial OR)$ (Equation 3).

$$e(\partial OR) = (\partial A^{-1} \cdot \partial OR) \cdot c \quad (3)$$

$e(\partial OR)$ is the total effort for a certain change. c is the vector describing the effort for each unit of design parameter deviation and ∂OR is the change itself. According to the information axiom, the architecture A must be designed in a way, which maximizes the information content of $csr(e_x)$ for each level of change effort e_x . With this in mind, it becomes clear, that there is not only a trade-off in terms of change risk mitigation costs and changeability, but also between different levels of change effort e_x and between the effort of one change and another.

2.3. Implications of flexibility and changeability for architectural design decisions

In the following, the differential design matrix ∂A is decomposed into five matrices. This lowers the above-described complexity and represents the local scale of architectural design decisions. The five matrices represent the components of the embedded automotive architecture: The system of objectives, the product structure, i.e. the hardware and the software layer, and the scheme by which they are all mapped to each other (see Figure 1). Such a decomposition breaks complex change interactions down to

local phenomena. Insights from more local and fine-grained approaches to change risk mitigation can be applied. [Bartolomei et al. \(2007\)](#) give an overview to such research in terms of design structure matrices. Hence, the differential design matrix ∂A is expressed as follows (Equation 4).

$$\partial A = \partial T \cdot \partial P^n \quad (4)$$

T is the traceability matrix, i.e. the mapping of objectives and requirements OR to elements in the product structure DP. T has the same dimensions as A, but differs in its semantic meaning: Whenever a change to OR happens, only the directly affected components can be identified through T.

P is the change propagation matrix (see e.g. [Giffin et al. \(2009\)](#)). P^n depicts the steady state of all change-affected components (row) for which the seed change originates from a certain component (column) (see [Sarica and Luo \(2019\)](#)). The root matrix P itself is further divided into four areas: The software internal part, the hardware internal part, and the two propagation matrices that cross the borders; one for each propagation direction. The areas are differentiated, because the change effort differs fundamentally between them. We expect a change in the hardware to be more costly in general, because prototyping tools must be redone, production facilities have to be adapted or the accessibility must be granted.

However, the perspective of change described through ∂T and ∂P^n is not complete: Architectural design decisions can also utilize objectives, which are under control of the organization, to influence changeability. Thus, the structure of the system of objectives must be investigated, even if not present in the design matrix A.

Different approaches exist to derive the above matrices from product descriptions (see e.g. [Käbmeyer et al. \(2015\)](#) or [Block et al. \(2019\)](#) for the automotive domain). They can then be employed to identify elements, which influence change propagation and thus changeability of the system. In the following, we will stick to the approach of [Suh et al. \(2007\)](#), because they define four local change propagation patterns and use them to derive generic action schemes. The four behavioural patterns are: Multipliers, carriers, absorbers and constants. Constants are matrix elements, that are unaffected by change. Their column sum is zero. Carrier elements cause, when triggered by a change, the same amount of change to other system elements. Multiplier elements generate more changes than they receive. Finally, absorbers cause less change, than they receive ([Eckert et al. 2004](#)). Based on these patterns, [Suh et al. \(2007\)](#) derive action schemes to prevent change propagation and incorporate flexibility and changeability. However, they do not differentiate between different element and matrix types. We build upon these action schemes and substantiate them through distinction into the five domains of application, explained above. The results are displayed in Table 1.

Summarizing, the decomposition of A into five matrices yields a breakdown of the complex architectural decisions. System elements, which should be tackled to mitigate change risk, can be identified on a local scale and generic action schemes aid in selecting the right degree of anticipation (Table 1). Global effects, especially unwanted side effects, can be made visible through an investigation of the information content via the relevant csr(e) levels.

3. Application

The methodology is applied to determine the degree of improvement in changeability due to an architectural redesign. The automotive software and hardware system under investigation is part of a test vehicle to research the communication and interaction between pedestrians and autonomous cars. As such, the system requires a high degree of changeability, to support different types of trials and experiments. Each experiment imposes multiple requirements on the hardware interaction devices (e.g. displays and LED stripes) as well as on the system's capability to control the interaction algorithms (Figure 5 (left)). Furthermore, the researcher's skills determine the ease of system control and the requirements towards experiment preparation. The project is chosen, because most of these constraints were uncertain in the beginning, but then manifested within the project or subsequent projects. Retrospectively, the effort for a change, which was uncertain at redesign time, is well known. The architectural redesign that will be investigated in the following introduced a hardware abstraction layer to ease the addition of new interaction elements and to loosen the binding of hardware elements to higher-level software. The change effort is determined through the changed lines of code (LOC). For the hardware elements, the LOC measure is multiplied by 1.5, to account for the hardware's

mounting effort. The probabilities of the requirements' ranges for time t_0 were stated by the project participants at the beginning of development. For today's (t_1) probability distributions, the frequency of the features' usage is taken. The matrices for T and P can be derived from the product's architecture (Figure 5 (right)). The design fulfils the Independence Axiom.

Analysing P yields that the software components SW2, SW4 and SW5 are major absorbers for changes from the hardware layer. According to Table 1, flexibility and changeability can be integrated through a hardware abstraction layer, which was the aim of the architecture's redesign. SW7 and SW8 are change multipliers. They form the hardware abstraction layer, and the dependency is thus hard to change. The architecture's redesign mainly manipulates this part to alter the change propagation behaviour of the hardware elements. Multiple other, possible applications of Table 1 to improve changeability are discovered. However, we will stick to the already implemented change, to evaluate the approach.

Table 1. Generic action schemes to incorporate changeability in the product's architecture

| Change propagation in... | Absorbers | Carriers | Multipliers |
|---|--|--|---|
| ... system of objectives | Absorbers in controlled objectives are welcome in general. However, they might yield inefficiencies in derived objectives if set too high. | Carrying constraints and objectives that are not controlled by the developing entity should be monitored closely. Especially, if they influence a multiplier or receive many changes. | Set the requirements high and create a buffer, to reduce the impact or multiplicity of a change in a controlled objective. Constraints that are not under control, should be monitored carefully, but cannot be adjusted. |
| ... traceability matrix T | Absorbers within the traceability matrix are candidates for over-engineering. Check if changeability is really necessary for this component. | A set of requirements is directly associated with a set of software and/or hardware components. Make sure, information about the state and possible changes of the influencing constraints and objectives are available to the team that is responsible for this component. | Multipliers in T are often quality or non-functional requirements (Pohl, 2008, p. 16). Increase the required property range, to create a buffer and incorporate flexibility into the requirements. |
| ... software layer/hardware layer | Absorbers prevent change propagation, but need to be flexible or changeable themselves. Make sure, that software can be flashed or changed easily (maybe even over the air). | Carrier elements might receive many changes and thus propagate them. Such a carrier element, especially in the hardware domain, might be more expensive than a multiplier and should be handled accordingly (Suh et al., 2007). | Multiplier elements make the system harder to change, the more changes are added (Suh et al., 2007). Traditional approaches from the software or hardware domain (e.g. generic interfaces or SOA) can be utilized to incorporate flexibility. |
| ... between software and hardware/hardware and software | Hardware should be accessible easily or not even change at all. | If a change propagates from software to hardware, they constitute an integrated subsystem. Check if it can be replaced or altered without causing other changes to propagate. If change propagates between software and hardware, product line development is hindered in general. Changeability can be integrated through a hardware abstraction layer. | Hardware is in general more costly to change (see above). Thus, to avoid the propagation of one change in software to multiple hardware elements, make sure, that software can be flashed or changed easily (maybe even over the air). Additionally, incorporate extra resources in the hardware layer. |

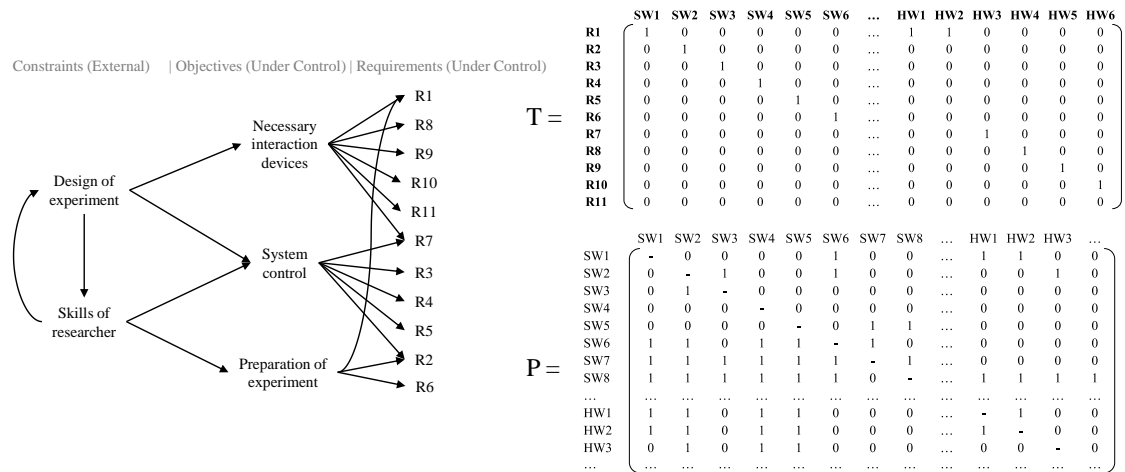


Figure 5. The system of objectives (left), the traceability matrix T (top right) and the change propagation matrix P (bottom right)

The traceability matrix T and the change propagation matrix P constitute the design matrix A. Through them, we depict the different levels of changeability for the overall product and how they were altered by the architectural redesign. For this purpose, all later changes are also implemented in a system where the architecture's redesign was not applied. A comparison of the efforts then yields the increase in changeability. For each objective (see Figure 5 (left)), the changeability before and after the redesign is expressed through the information content (Figure 6). The probability distribution over the possible range as well as the flexibility and changeability ranges after the redesign are plotted in Figure 7. Respectively, the average changeability after the redesign amounts to $1.31 \cdot 10^{-3}$ bits/LOC, $0.47 \cdot 10^{-3}$ bits/LOC and $0.36 \cdot 10^{-3}$ bits/LOC. In total, the architecture's redesign lowered the maximum effort for a full coverage of the uncertain requirements by 728 lines. The redesign itself caused a rework effort of 210 lines of code, resulting in a changeability gain of $0.03 \cdot 10^{-4}$ bits/LOC to $1.18 \cdot 10^{-4}$ bits/LOC.

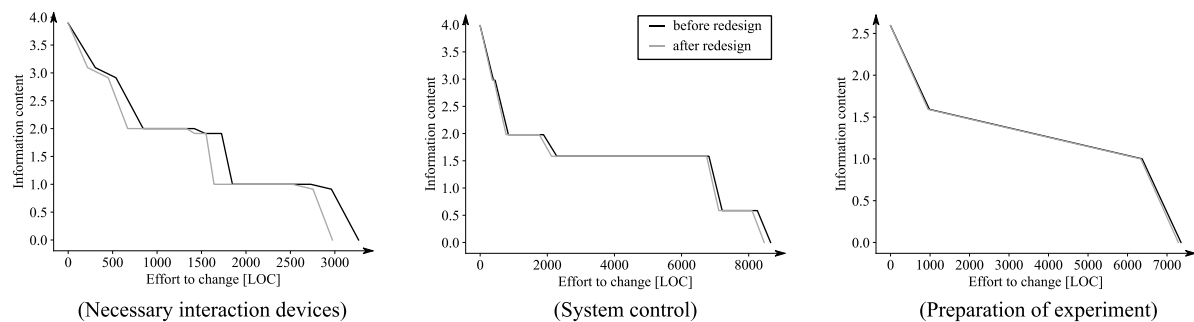


Figure 6. The different levels of $csr(e)$ before and after the architecture's redesign

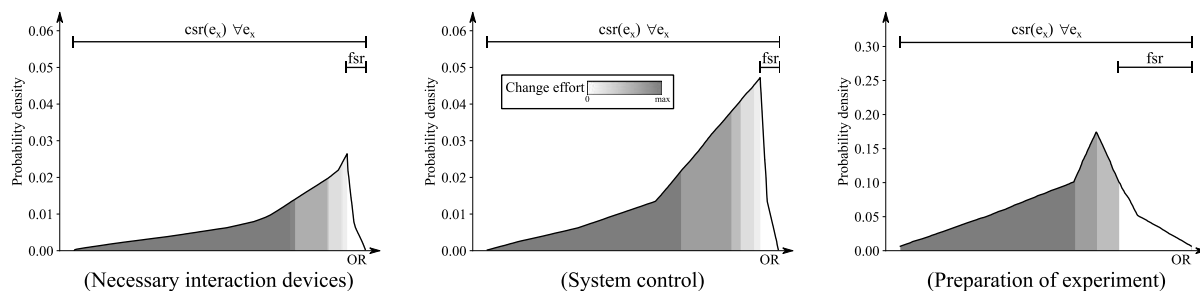


Figure 7. Flexibility and changeability for all three objectives under investigation

4. Discussion and related work

Implementation of flexibility and changeability into an automotive software and hardware architecture is a challenging task: Design decisions are usually taken on the local scale of interfaces and elements. However, the effects of such decisions propagate through the system and influence flexibility and changeability on an overall scale. Changeability is defined as the system range, reachable with a certain effort for rework. Consequently, this paper deals with change, effort and the probability of changes on a local and on a global scale to derive implications for architectural design decisions.

Change risk management in engineering design and product architectures are not new to the scientific community. A lot of literature addresses change from the perspective of the design structure matrix, to estimate change effort and affected components or to improve architectural design decisions. [Eckert et al. \(2001\)](#) for example introduce a generic concept for a better understanding of change propagation and change amplification in the product's structure. [Suh et al. \(2007\)](#) as well as others (e.g. [Eckert et al. \(2004\)](#), [Giffin et al. \(2009\)](#), [Díaz et al. \(2011\)](#) and [Rebentisch et al. \(2018\)](#)) extend this approach to guide architectural design decisions. They aim to increase flexibility and mitigate change risk. Nevertheless, changeability is often not related to the objectives and requirements of the development project. Moreover, scientific literature addresses flexibility and changeability through a methodology called Uncertainty- or Change-Mode-and-Effect-Analysis (e.g. [Rajan et al. \(2003\)](#), [Tilstra et al. \(2009\)](#) and [Zhu et al. \(2017\)](#)). Possible changes are systematically identified and evaluated in analogy to the Failure-Mode-and-Effects-Analysis.

The approach in this paper is based on axiomatic design. [Schulz and Fricke \(1999\)](#) relate their ideas about flexibility, agility, robustness and adaptability to the independence and information axiom. [Matt \(2007\)](#), [Weber et al. \(2015\)](#) as well as [Foith-Förster et al. \(2016\)](#) apply axiomatic design to measure flexibility and changeability in the design and selection process of manufacturing systems. [Padala and Maheswari \(2019\)](#) build a methodology based on axiomatic design to tackle the changing requirements in a construction project. However, no decomposition of the mappings' matrices seems to be present that depicts the granularity of the design structure mapping approaches.

Accordingly, to the best of our knowledge, no methodology exists in literature so far that bridges the gap between the element-wise perspective and analysis of the design structure matrix approaches and the more global view of axiomatic design. The suggested methodology fills this void. It supports design decisions with respect to a definition of flexibility and changeability that is closely coupled to the overall objectives of product development. It takes into account that also the controlled objectives and requirements can be employed, to equip the product for future changes. Additionally, it maps the local design actions to alternations of flexibility and changeability. However, application of the methodology in practice is limited: It builds upon the definition of multiple matrices, which represent the product architecture. Thus, the measures can only be calculated, if a formalized description of the product is present and can be turned into the required matrix format. This might often not be the case in reality. Furthermore, change efforts must be known or estimated. The methodology itself employs the relatively simple concept of probability to model uncertainty, while other mathematical concepts have proven to be more expressive and more accurate (see e.g. [Wood and Antonsson \(1990\)](#)). Nevertheless, it can serve as a mental model in decision situations. It supports the derivation of action schemes to incorporate flexibility and changeability in practice (see Table 1). The retrospective application to an embedded automotive system supports the claim that the methodology is able to reveal the implications of flexibility and changeability on the architecture's design and to guide related design decisions.

Nevertheless, the investment in flexibility and change risk mitigation must be carefully balanced ([Suh et al., 2007](#)). Strategies for prohibitive investment may positively or negatively interfere with objectives, not modelled within this approach (e.g. company-wide goals). Table 1 must thus be seen as a set of guidelines, which are valid under the stated perspectives. However, it does not bear the full set of constraints, relevant for decision making in engineering design.

Finally, some remarks have to be made, regarding the validity of the application's results. The author of this paper was also part of the development team for the system under investigation. As such, he was also involved in stating the expected probabilities. However, the development project was finished, before work on this paper begun. An unintentional and subconscious influence on the development results can therefore be excluded.

5. Conclusion

Flexibility and changeability are crucial when it comes to the design of automotive software and hardware architectures. Software updates and hardware upgrades can adapt the car's functionality throughout the life cycle to increase customer value (Guissouma et al., 2018). The architecture must thus incorporate flexibility and changeability to deal with such possible future changes. However, architectural design decisions can affect local subsystems, e.g. single interfaces and elements, as well as the overall system. Flexibility and changeability are concepts rooted in the overall product and its requirements (Schulz and Fricke, 1999). Therefore, the optimal degree of change anticipation varies throughout the system, composing a complex problem with multiple influencing variables.

A methodology is developed to bridge the gap between the local, less complex perspective and the global view. Axiomatic design is applied to define and measure flexibility and changeability on a product's overall level. Flexibility is defined as the ability of a system to adapt to a change in requirements or elements without effort. In contrast, changeability depends on the level of effort for the rework. Subsequently, the perspective of axiomatic design is mapped to our definition of an automotive software and hardware architecture. This allows describing the influence of architectural design decisions on a local scale. As a result, five domains of actions to foster changeability are identified and generic action schemes are derived. These action schemes on the element and interface level are then connected with the previously developed perspective of axiomatic design, to bridge the gap between architectural design decisions and changeability. The retrospective application to the development history of an automotive system supports the claim that the methodology is able to reveal the implications of flexibility and changeability on the architecture's design.

Future research might deal with the extension of the approach to generic mechatronic systems and a theoretical enhancement to incorporate different uncertainty models. In addition, it can investigate the application of the methodology in an ongoing development project instead of an ex post perspective.

References

- Bartolomei, J. et al. (2007), *Analysis and applications of design structure matrix, domain mapping matrix, and engineering system matrix frameworks*, Massachusetts Institute of Technology, Engineering Systems Division.
- Block, L., Riedel, O. and Herrmann, F. (2019), "A Lifecycle Model to Support Continuous Component Evolution in Embedded Automotive Systems", *19. Internationales Stuttgarter Symposium: Automobil- und Motorentechnik*, Stuttgart, March 19-20, 2019, Springer Vieweg, Wiesbaden. https://doi.org/10.1007/978-3-658-25939-6_94
- Díaz, J. et al. (2011), "Change Impact Analysis in Product-Line Architectures", *Software architecture: Proceedings of the 5th European conference, ECSA 2011*, Essen, Germany, September 13-16, 2011, Springer, Berlin, pp. 114-129. https://doi.org/10.1007/978-3-642-23798-0_12
- Eckert, C., 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. <https://doi.org/10.1007/s00163-003-0031-7>
- Eckert, C., Zanker, W. and Clarkson, P.J. (2001), "Aspects of a better understanding of changes", *Proceedings of the 13th International Conference on Engineering Design (ICED 01)*, Conference Centre, Glasgow, UK, August 21-23 2001, Design Society.
- Foith-Förster, P. et al. (2016), "Axiomatic Approach to Flexible and Changeable Production System Design", *Procedia CIRP*, Vol. 53, pp. 8-14. <https://doi.org/10.1016/j.procir.2016.05.001>
- Giffin, M. et al. (2009), "Change Propagation Analysis in Complex Technical Systems", *Journal of Mechanical Design*, Vol. 131 No. 8, pp. 081001-1-081001-14. <https://doi.org/10.1115/1.3149847>
- Guissouma, H. et al. (2018), "An Empirical Study on the Current and Future Challenges of Automotive Software Release and Configuration Management", *44th Euromicro Conference on Software Engineering and Advanced Applications (SSEA 2018)*, Prague, August 29-31, 2018, IEEE, Piscataway, NJ, pp. 298-305. <https://doi.org/10.1109/SEAA.2018.00056>
- Helander, M.G. and Lin, L. (2002), "Axiomatic design in ergonomics and an extension of the information axiom", *Journal of Engineering Design*, Vol. 13 No. 4, pp. 321-339. <https://doi.org/10.1080/0954482021000050794>
- Jaensch, M. (2012), *Modulorientiertes Produktlinien Engineering für den modellbasierten Elektrik/Elektronik-Architekturentwurf*, [PhD Thesis], Karlsruhe Institute of Technology. <https://doi.org/10.5445/KSP/1000028687>
- Käßmeyer, M., Schulze, M. and Schurius, M. (2015), "A process to support a systematic change impact analysis of variability and safety in automotive functions", *Proceedings of the 19th International Conference on*

- Software Product Line*, Nashville, Tennessee, July 20-24, 2015, ACM, New York, NY, pp. 235-244. <https://doi.org/10.1145/2791060.2791079>
- Matt, D.T. (2007), "Design of Changeable Assembly Systems: A Complexity Theory Based Approach", *IEEE International Conference on Industrial Engineering and Engineering Management 2007 (IEEE IEEM 2007)*, Singapore, December 2-4, 2007, IEEE, Piscataway, NJ, pp. 738-742. <https://doi.org/10.1109/IEEM.2007.4419288>
- Muschik, S. (2011), *Development of Systems of Objectives in Early Product Engineering*, [PhD Thesis], Karlsruhe Institute for Technology. <https://doi.org/10.5445/IR/1000023768>
- Padala, S.P.S. and Maheswari, J.U. (2019), "Axiomatic design framework for changeability in design for construction projects", *Asian Journal of Civil Engineering*, Vol. 9 No. 1, p. 15. <https://doi.org/10.1007/s42107-019-00187-1>
- Pimmler, T.U. and Eppinger, S.D. (1994), "Integration analysis of product decompositions", *Design Theory and Methodology (DTM '94)*, Minneapolis, MN, September 11-14, 1994, ASME, New York, NY, pp. 2-10.
- Pohl, K. (2008), *Requirements engineering: Grundlagen, Prinzipien, Techniken*, 2nd ed., dpunkt-Verlag, Heidelberg.
- Rajan, P. et al. (2003), "Design for flexibility: Measures and guidelines", *Proceedings of the 14th International Conference on Engineering Design (ICED 03)*, Stockholm, August 19-21, 2003, Design Society.
- Rebentisch, E. et al. (2018), "Assessment of changes in engineering design using change propagation cost analysis", *Proceedings of the 21st International Conference on Engineering Design (ICED 17): Design Methods and Tools*, Vancouver, Canada, August 21-25, 2017, Design Society, Red Hook, NY, pp. 69-78. <https://doi.org/10.1016/j.procir.2016.11.083>
- Sarica, S. and Luo, J. (2019), "An Infinite Regress Model of Design Change Propagation in Complex Systems", *IEEE Systems Journal*, pp. 1-9. <https://doi.org/10.1109/JSYST.2019.2899988>
- Schulz, A.P. and Fricke, E. (1999), "Incorporating flexibility, agility, robustness, and adaptability within the design of integrated systems: Key to success?", *Proceedings of the 18th Digital Avionics Systems Conference (DASC): Gateway to the New Millennium*, St Louis, MO, USA, October 24-29, 1999, IEEE, Piscataway, N.J., 1.A.2-1-1.A.2-8. <https://doi.org/10.1109/DASC.1999.863677>
- Suh, E.S., de Weck, O.L. and Chang, D. (2007), "Flexible product platforms: framework and case study", *Research in Engineering Design*, Vol. 18 No. 2, pp. 67-89. <https://doi.org/10.1007/s00163-007-0032-z>
- Suh, N.P. (2001), *Axiomatic design: Advances and applications*, MIT-Pappalardo Series in Mechanical Engineering, Oxford University Press, New York, NY.
- Suh, N.P. (2005), *Complexity: Theory and applications*, MIT-Pappalardo Series in Mechanical Engineering, Oxford University Press, Oxford.
- Tilstra, A.H. et al. (2009), "Industrial Case Studies in Product Flexibility for Future Evolution: An Application and Evaluation of Design Guidelines", *20th International Conference on Design Theory and Methodology, 2nd International Conference on Micro- and Nanosystems*, Brooklyn, New York, USA, August 3-6, 2008, ASME, New York, NY, pp. 217-230. <https://doi.org/10.1115/DETC2008-49370>
- Ulrich, K. (1995), "The role of product architecture in the manufacturing firm", *Research Policy*, Vol. 24 No. 3, pp. 419-440. [https://doi.org/10.1016/0048-7333\(94\)00775-3](https://doi.org/10.1016/0048-7333(94)00775-3)
- Weber, J. et al. (2015), "Design of Changeable Production Units within the Automotive Sector with Axiomatic Design", *Variety Management in Manufacturing. Proceedings of the 47th CIRP Conference on Manufacturing Systems*, Vol. 34, pp. 93-97. <https://doi.org/10.1016/j.procir.2015.07.006>
- Wood, K.L. and Antonsson, E.K. (1990), "Modeling imprecision and uncertainty in preliminary engineering design", *Mechanism and Machine Theory*, Vol. 25 No. 3, pp. 305-324. [https://doi.org/10.1016/0094-114X\(90\)90031-E](https://doi.org/10.1016/0094-114X(90)90031-E)
- Zäh, M.F., Möller, N. and Vogl, W. (2005), "Symbiosis of Changeable and Virtual Production: The Emperor's New Clothes or Key Factor for Future Success", *Proceedings of the 1st International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV 2005)*, Munich, Germany, 2005, Utz.
- Zhu, G.-N. et al. (2017), "Change mode and effects analysis by enhanced grey relational analysis under subjective environments", *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 31 No. 2, pp. 207-221. <https://doi.org/10.1017/S0890060417000099>