Design Science

Supporting early robust design for different levels of specific design knowledge: an adaptive method for modeling with the Embodiment Function Relation and Tolerance model

Jiahang Li[1], Dennis Horber[2], Patric Grauberger[1], Stefan Goetz[2], Sandro Wartzack[2] and Sven Matthiesen[1]

[1]*Karlsruhe Institute of Technology, IPEK – Institute of Product Engineering, Kaiserstraße 10, 76131 Karlsruhe, Germany*

[2]*Friedrich-Alexander-Universität Erlangen-Nürnberg, KTmfk – Engineering Design, Martensstraße 9, 91058 Erlangen, Germany*

Abstract

Early robust design (RD) can lead to significant cost savings in the later stages of product development. In order to design systems that are insensitive to various sources of deviation in the early stages, specific design knowledge (SDK) plays a crucial role. Different design situations result in higher or lower levels of derivable SDK, which leads to different activities to achieve the development goal. Due to the variety of design situations, it is difficult for design engineers to choose a more robust concept to avoid the costly iterations that occur in the later development stages. Existing RD methods often do not adequately support these differences in design situations. To address the problem, this paper outlines an adaptive modeling method using the Embodiment Function Relation and Tolerance model. The method is developed in two contrasting design situations, each with a high and low level of derivable SDK, and evaluated in another two corresponding case studies. It has a consistent structure with five stages and gates. At each stage, the derivable SDK is taken into account and the individual modeling steps are adapted. This method provides design engineers with concrete support for early robustness evaluation of a product concept in different development scenarios.

Keywords: design methods, product development, robustness evaluation, design situation, design knowledge

Received 07 April 2024
Revised 21 September 2024
Accepted 28 October 2024

Corresponding author
Jiahang Li
jiahang.li@kit.edu

Des. Sci., vol. 10, e46
journals.cambridge.org/dsj
DOI: 10.1017/dsj.2024.48

1. Introduction

Today's market demands fast development cycles with high product functionality. During a product's lifecycle, its design parameters are prone to deviation, for example, geometric deviations caused by manufacturing, wear during usage, or deviations in environmental conditions. Despite these deviations, technical systems must reliably complete their tasks throughout their entire life cycle. Robust

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

design (RD) aims to design systems that are insensitive to various sources of deviation, which originates from the quality engineering framework according to Taguchi et al. (2005). Implementing RD in the early stages can reduce costly iterations during later product development (Jugulum and Frey, 2007). The challenge for early robustness evaluation of a product concept is that often only qualitative data is available, which makes simulation or experimentation for robustness optimization difficult. One possibility for evaluating the product concept without quantitative data is qualitative modeling, which represents the properties and characteristics of a system without detailed parameters (Grauberger et al., 2022). An example of a qualitative model is the Embodiment Function Relation (EFR) and Tolerance (EFRT) model developed by Horber et al. (2022). Using the EFRT model, the robustness of the product concept can be evaluated with appropriate criteria in a given design situation (Li et al., 2023).

In order to perform effective qualitative modeling of a technical system, specific knowledge is required. This knowledge of describing the technical systems in the design process is defined by Hubka and Eder (1990) as specific design knowledge (SDK). Different design situations result in higher or lower levels of derivable SDK, that is, the SDK that can be derived from the data available to design engineers at the beginning of the development process. This leads to different activities to achieve the development goal (Ponn and Lindemann, 2005).

Due to the variety of design situations, it is difficult for design engineers to choose a more robust concept to avoid the costly iterations that occur in the later development stages. To successfully evaluate the robustness of a product concept through qualitative modeling, it is essential to investigate how the existing SDK affects the modeling activities. This paper aims to deal with the lack of consideration given to different design situations by developing an adaptive modeling method, which incorporates the EFRT model, with a focus on the geometric design of mechanical parts within technical systems. This method provides design engineers with concrete support for the evaluation of the product concept through modeling in different development scenarios.

1.1. Early RD

Taguchi et al. (2005) divide RD into three phases along the product development process: system design, parameter design, and tolerance design. Central principles of RD include awareness of the impact of unavoidable deviations, focus on insensitivity to such deviations, and consistent consideration of robustness throughout the product development process (Arvidsson and Gremyr, 2008).

Existing conventional RD methods mainly focus on robustness optimization based on controlled simulations or experiments (Phadke, 1989; Taguchi et al., 2005). These methods need detailed product data in a well-defined shape, which is only available in the later stages of design, that is, late parameter and tolerance design (Hasenkamp et al., 2009). Andersson (1997) argues that the achievable level of product robustness using these methods is determined by the early system design. In addition, the early design stages have a significant influence on product performance and cost (Ullman, 2010). Although early RD is crucial to robust product development, its application in practice remains challenging due to a lack of quantitative information (Eifler and Schleich, 2021; Jugulum and Frey, 2007).

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

Methods, such as Quality Function Deployment (QFD) (Akao, 2004), Variation Risk Management (Thornton, 2004), can be used to raise awareness of the impact of unavoidable deviations on the fulfillment of functions. To evaluate the fulfillment of the functional requirements of the product concept, the concept of key characteristic (KC) is widely used. A KC serves as a quantifiable product specification whose deviation has a major impact on the fulfillment of functional requirements (Thornton, 2004). In order to focus on reducing the sensitivity to such deviations, it is necessary to analyze the deviations and their effects (Ebro and Howard, 2016). For this purpose, methods such as Variation Mode and Effects Analysis (VMEA) (Johansson et al., 2006) and Failure Modes and Effects Analysis (FMEA) (Teng and Ho, 1996) exist, which identify potential failure modes and sensitive design parameters, but provide limited guidance for design tasks.

For design tasks, a common practice for early RD is to use design principles, such as reducing load paths or avoiding overdetermination (Andersson, 1997; Ebro et al., 2012; Ebro and Howard, 2016). Evaluating the robustness of different concepts using design principles is often difficult because of the large number of principles and the lack of connection to the individual product concept. In addition, the influence of the deviation on functional fulfillment cannot be analyzed in detail.

At the system level, qualitative models, such as Design Structure Matrices (DSM) (Eppinger and Browning, 2012) and Characteristics Properties Models (Weber, 2014), can be used to analyze the system structure. The dependencies between the components can be systematically analyzed and the coupling in concepts can be reduced, which improves the robustness of the concept (Suh, 1998). These models support the structuring of the system, but they still lack support for the concretization of the product concept in the early design phase, because the geometric relations between the individual elements and the sum effects in the geometric chain cannot be represented.

In order to support the design engineer in specifying robust concepts, it is essential to model the geometric relationships in a product concept. The graph-based approaches facilitate the representation of the geometric structure of the product concept despite limited product information (Ballu et al., 2010). An example of such an approach is the tolerance graph (Goetz et al., 2018). It facilitates an early-stage analysis of the geometric relations in a product concept, focusing on the analysis of KC, and enabling an initial robustness evaluation of the product concept in its early design stages (Goetz et al., 2018).

The functional fulfillment of the product concept also depends on the system behavior in different states, which cannot be modeled with graph-based approaches. Therefore, a pure analysis of the geometric relationship is often not sufficient. An appropriate approach requires a more detailed modeling and understanding of the EFR in the product concept. EFRs describe how the design parameters of a system in its embodiment affect its behavior and, through that behavior, the desired functional requirements that are fulfilled. A qualitative modeling approach that has already been applied to RD tasks by analyzing EFR is the Contact and Channel Approach (C&C^2-A) (Grauberger et al., 2019; Matthiesen and Ruckpaul, 2012).

In summary, for a comprehensive robustness evaluation of the product concept for design tasks, we need a model that can model both the deviations and their effects on functional fulfillment. In order to analyze the functional fulfillment with

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

geometric deviations for a product concept, the modeling requires that the concept has a certain degree of maturity in its geometry. Therefore, it is necessary to model the geometric relationships in the product concept as well as the system behavior. For this purpose, Grauberger et al. (2020) proposed combining the tolerance graph and C&C²-A in a holistic model to support design engineers with extended modeling aspects for the product concept, which is then described in section 1.2.

1.2. The EFRT model

The EFRT model developed by Horber et al. (2022) integrates the benefits of both tolerance graph and C&C²-A by combining the two approaches. The EFRT model offers great potential for early robustness evaluation by modeling a wide range of qualitative information, such as the system structure for early tolerance design, or the EFR for analyzing the system behavior (Li et al., 2023). The models and the result of the robustness evaluation can be used in subsequent approaches (Horber et al., 2023), for example, for the automated generation of a preliminary Computer-Aided Design (CAD) model (Goetz et al., 2022).

Figure 1 illustrates the combination of both approaches and the core elements in an EFRT model (Li et al., 2023). This figure also shows the relationships between the individual elements and how they can represent the information in the early stages of product development. In an example coining machine, both tolerance graph and C&C^2 models can be derived. With the tolerance graph, the geometric structure of the product concept is modeled. The product concept is first decomposed from assembly to part and then to geometry elements (GEs), facilitating the analysis of the contributors to KC in the concept phase. The C&C²-A can be used to analyze system behavior and functional fulfillment. For this purpose, three key elements are required: the Working Surface Pair (WSP), the Channel and Support Structure (CSS), and the Connector (C) (Matthiesen et al., 2019). Beginning with the WSP, it describes, where exactly an interaction between the components occurs. A WSP is the place of the interface, where parts of the system connect while it fulfills its function; it consists of two working surfaces (WS). The path for the information transfer is defined as a CSS. A CSS runs through parts in the system and connects two WSPs. Finally, the information on the system boundary is stored in the C (Matthiesen et al., 2019). WSPs can be added or deleted in different system states, allowing separate modeling of system states.

As shown in Figure 1, the link between the tolerance graph and the C&C^2 models is at the interface between the GE and the WS. Based on this link, Horber et al. (2022) proposed an initial workflow for building the EFRT model that covers the main modeling steps for building both the tolerance graph and C&C^2 models. However, this workflow focuses on the general connection between the two approaches and the model implementation in System Modeling Language. Different design situations are not considered, and the level of detail in this workflow is limited. To better support design engineers, the workflow has to be further detailed and adapted to the design situation (Horber et al., 2022), since different processes are expected from different design situations in product development. In practice, design engineers often choose inappropriate design processes or methods due to a lack of specific knowledge (Wilmsen et al., 2019). Therefore, it is important to investigate how the specific knowledge from different design situations affects the modeling activities.

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

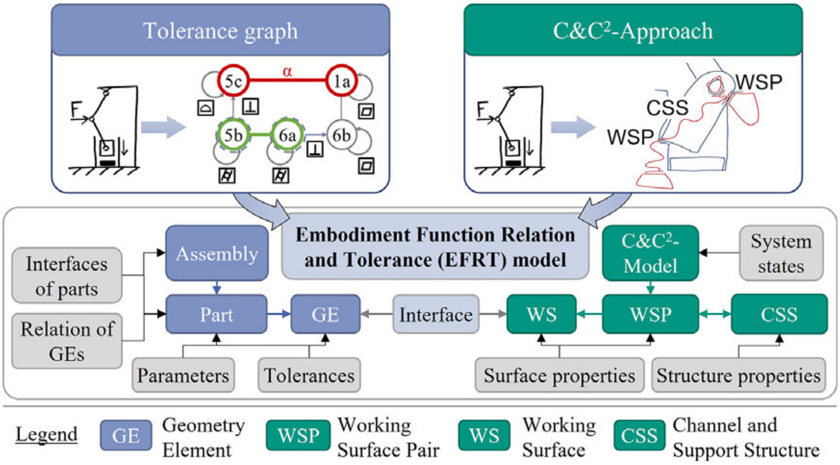


Figure 1. The Embodiment Function Relation and Tolerance (EFRT) model and the summarized information of its model elements according to Li et al. (2023).

1.3. Design situations and SDK

Fundamentally, knowledge can be distinguished by its availability into explicit and implicit knowledge (Dzbor and Zdrahal, 2001). While implicit knowledge is tied to individuals and hard to communicate, explicit knowledge can be formalized and derived from data or research. Such knowledge is accessible in models such as the function–behavior–structure model (Gero and Kannengiesser, 2014) and the organ domain model (Andreasen et al., 2015). This type of knowledge is crucial for product development. To support comprehensive design tasks, SDK is essential. It represents the knowledge about EFR that can be modeled with the EFRT model.

In product development, design engineers must deal with different types of design situations that require different activities in the design process (Ponn and Lindemann, 2005). At the start of development, there may be some differences in the design situation, such as how novel the design task is, what information can be used, and how mature the present concept is. These differences lead to different levels of derivable SDK. The derivable SDK in this paper is based on the data available to design engineers at the beginning of the current development project. This includes both explicit data and potentially implicit knowledge held by the design engineers themselves. While we recognize that the status of implicit knowledge is difficult to measure and is not the primary focus of this paper, we have chosen to assume that it remains constant and focus our measurements on the differences in the available documentation. For example, in the coining machine shown in Figure 1, the level of derivable SDK is higher when there are existing test documents available as references for the current development project.

In this contribution, we will explore three particular indicators of the design situation – the *level of innovation*, the *usability of data*, and the *design phase* – and their influences on the SDK. Other indicators could be considered, but the ones presented are sufficient to demonstrate this influence. It should be noted that these three indicators are not independent of each other.

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

The first indicator of the design situation we explored is the *design phase*. In product development, design activities are typically divided into different phases. For general design tasks, Pahl et al. (2007) divided the design into four phases: clarification of the task, conceptual design, embodiment design, and detail design. Haik and Shahin (2011) supplemented this division with a solution concept phase. Other divisions of design phases also exist for different development goals (Andreasen et al., 2015; Taguchi et al., 2005). Although there is no generalized separation of design phases, it can be said that early stages can be distinguished by the parameterization of the product. During these phases, the level of maturity of the concept is continuously increasing, and the difference in the concept maturity should be considered during modeling. Therefore, this indicator is particularly related to the different stages in the modeling process.

The level of derivable SDK increases as the design phase progresses and the concept becomes more mature. Niu et al. (2022) collected the key information items at different design phases in an explorative study, showing how the information content grows during the design phase. Capturing and storing the essential information generated throughout the product development process is the task of a design model (Eisenbart et al., 2011). In practice, the stage-gate process based on the Quality Gates approach is often used to store the various information contents. This approach divides a process chain into phases and periodically checks the quality of the process. The process moves to the next stage when the results are mature enough to reach a gate (Hammers and Schmitt, 2009). If we take the four stages from Pahl et al. (2007), the SDK is always being built throughout the design process to achieve the development goal. In clarifying the task, part of the SDK can be derived from the requirements specification. Different activities have to be performed to build the SDK and create different concepts according to the requirements. During the embodiment design and detail design phases, the concept becomes more mature and we gain more knowledge about the functional fulfillment of the selected concept, thus increasing the SDK. The EFRT model is proposed to support the early stages from the initial idea to the early embodiment design phase before the final elaboration of the geometric parameters. The challenge for the modeling is how to deal with the different maturities of the product concept.

The second indicator of the design situation we explored is the *level of innovation*. Henderson and Clark (1990) distinguished between four types of product innovation: incremental innovations, architectural innovations, modular innovations, and radical innovations, where the innovation level increases. Pahl et al. (2007) categorized the design tasks into original design, adaptive design, and variant design, and suggested different strategies for each. Albers and Rapp (2022) stated that every development of a new system is based on references, and they divided the development into principle variation, attribute variation, and carryover variation. The authors have differentiated the design tasks according to the level of innovation. This is because different difficulties are to be expected, and therefore, different design processes are required.

Higher levels of innovation result in lower levels of derivable SDK. Developing a new product concept involves more uncertainty than refining an existing one, which benefits from prior information in the reference systems (Pahl et al., 2007). The reference systems can vary in their level of innovation, such as references from the market, from the company, or previous products. Low derivable SDK occurs

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

with absent or abstract references, for example, in principle variation. High derivable SDK arises from concrete or carryover references, for example, in an adaptive design.

The third indicator of the design situation we explored is the *usability of data.* In addition to considering the art of the reference system, data availability can affect design activities (Ponn and Lindemann, 2005). Information becomes valuable only when it is used, as it is a resource that gains its worth from its utilization (El Hani et al., 2012). Therefore, it is necessary to check the usability of data. Design data can be both qualitative and quantitative, for example, requirements specifications, lists of requirements, sketches, technical drawings, detailed CAD data, or bills of materials. Data processing effort is determined by the abstraction of the data. It is important to distinguish whether existing information in the data can be reused or whether the required information must first be prepared.

A higher level of data usability indicates a higher level of derivable SDK, as more sources and evidence can be used to support and validate the product concept. Higher information usability also enables more in-depth analysis and simulation of the system design, which can enhance the SDK for further development steps.

The design situation determines the level of derivable SDK at the start of the development, but this is not identical to the required SDK for the development goal. The relationships between the design situation, derivable SDK, modeling activities, required SDK, and development goal are illustrated in Figure 2.

As shown in Figure 2, different design situations lead to different levels of derivable SDK. Comparing the derivable SDK at the beginning of the development process and the required SDK for early robustness evaluation, different sequences of activities can be derived. Two contrasting cases deserve special attention. In the first case, the development project starts with a low level of experience and only references that are far away from the product to be developed are available. Therefore, the derivable SDK is insufficient and less than the required SDK. It is necessary to generate information and detail the desired concepts. In the other case, the development project is based on a previous product and starts with extensive

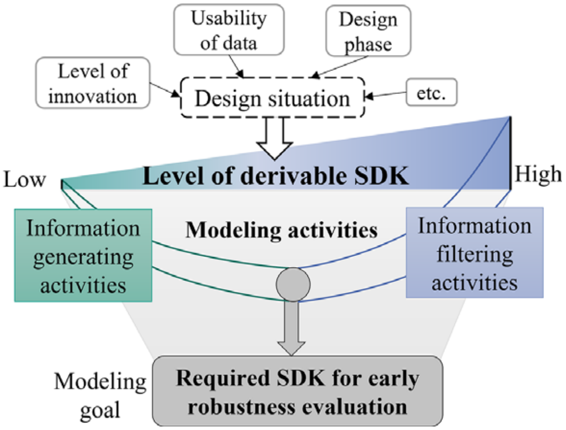


Figure 2. Different design situations influence the level of derivable specific design knowledge (SDK); therefore, different modeling activities are needed to achieve the required SDK for early robustness evaluation.

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

data. Rather than optimizing the robustness of existing components, the focus is on redesigning new subsystems to improve overall product robustness. The derivable SDK is larger than the required SDK. In this case, the information content and the derivable SDK are excessive. The information needs to be filtered. To efficiently model the product concept and obtain the required SDK for early robustness evaluation, this difference must be considered and the different modeling activities need to be examined in different workflows. It is important to note that both cases are in early stages of the current development project, as case 2 involves the further development of the existing product into the new product generation.

1.4. Research question

Due to the lack of detailed product information, for example, a well-defined shape, the robustness evaluation of a product concept in the early stages of product development is challenging. Multiple approaches address this challenge by modeling different aspects of the product concept. Approaches, such as QFD or VMEA, help to identify the deviations in a system but do not support the actual design process. Approaches, such as DSM or axiomatic design support system, design at an abstract level. RD principles can be applied to real design tasks to enhance the robustness of a system, but it can be challenging to select the most robust concept using these principles alone due to the complexity and variability inherent in real-world scenarios. Among various early RD approaches, the EFRT model enables modeling a wide range of qualitative information in a product concept for the analysis of the robustness of the product concept. The combination of early tolerance design and EFR modeling offers a high potential for early robustness evaluation. However, there is still a lack of methodical support for the building of the EFRT model in early RD. Once the model is properly built, the SDK for early robustness evaluation can be better achieved.

Derived from the state of the art described in section 1.3, various design situations result in higher or lower levels of derivable SDK. Different modeling activities are required to achieve the development goal, as shown in Figure 2. However, the initial approach by Horber et al. (2022) was a general workflow without considering design situations. Due to the variety of design situations, there is a risk of either over-detailed modeling or failing to achieve the modeling goal. Therefore, it is challenging for design engineers to choose a more robust concept, which can lead to costly iterations in the later development stages. Existing RD methods in the literature often do not adequately support these differences in design situations. Consequently, methods need to be tailored to the level of derivable SDK due to variety of design situations (Grauberger et al. 2020). This adaptation is crucial, especially in early design stages, to effectively build the required SDK for the development goal. It remains unclear, how the SDK from different design situations affects the modeling activities to build the EFRT model for the early robustness evaluation. Without properly adapting to the design situation, the model would be prone to becoming overly detailed due to the large amount of possible information and content, increasing the possibility that it fails to achieve its purpose through additional and unnecessary effort. To enable successful early robustness evaluation of a product concept with the EFRT model, it is essential to investigate how the derivable SDK affects

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

the modeling activities. As a result, this paper addresses the following research question:

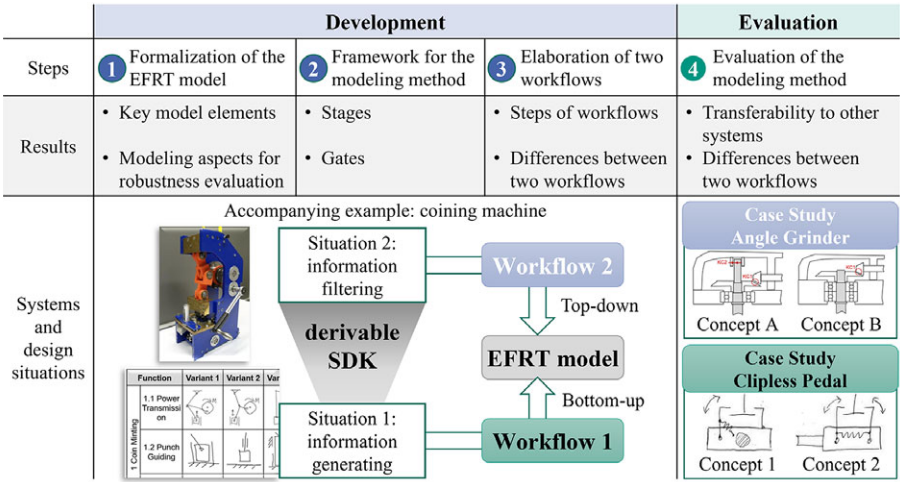
> *How can an EFRT model be built whilst taking different levels of derivable specific design knowledge for early robustness evaluation into account?*

2. Research design for developing the modeling method

To address the research question, we proposed a modeling method based on the EFRT model. This section provides insights into how this modeling method was developed and evaluated in this contribution. The research design for developing the modeling method is illustrated in Figure 3. The research design begins with the formalization of the EFRT model to facilitate the development of the modeling method. To allow for adaptation of modeling steps by different design situations, we have structured the generalized modeling method framework into five stages, allowing flexibility for adapting modeling steps within each stage to suit specific design situations. The required milestones to build the EFRT model can be achieved in each stage. In step 3, we further refine the modeling method by developing two workflows based on contrasting design situations, emphasizing their differences. Using two exemplary but contrasting design situations, we attempt to explore the differences in modeling activities. Step 4 focuses on the initial evaluation of the transferability of the developed modeling method to other technical systems and design situations. This step also evaluates the differences between the workflows. The results of these steps are also illustrated in Figure 3, with detailed explanations provided in section 3 for the results of development steps 1–3 and in section 4 for the results of step 4.

Step 1: Formalization of the EFRT model

To develop a modeling method based on the EFRT model, it is imperative to provide a sufficiently formal description of the EFRT model, which is currently



	Development			Evaluation
Steps	1 Formalization of the EFRT model	2 Framework for the modeling method	3 Elaboration of two workflows	4 Evaluation of the modeling method
Results	• Key model elements • Modeling aspects for robustness evaluation	• Stages • Gates	• Steps of workflows • Differences between two workflows	• Transferability to other systems • Differences between two workflows
Systems and design situations				

Figure 3. Research design for developing the modeling method based on the EFRT model.

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

lacking. In this step, we first provide a formal description of the EFRT model, including its key model elements. Then, we investigate which aspect should be modeled with the EFRT model to evaluate the robustness of a product concept. To illustrate the model, we use a hand-operated coining machine as an example (see Figure 3). This mechanism is described in detail in the study by Horber et al. (2022). The result of this step is a formalized description of the key model elements in the EFRT model and its modeling aspects for robustness evaluation. This formalization forms the basis for the development of the modeling method.

Step 2: Framework for the modeling method

After formalizing the EFRT model, we develop a framework that complies with some boundary conditions for the modeling method. Considering the differences in the design situations, the modeling method must facilitate the adaptation of different modeling activities within it. Despite the different activities, the modeling method should have a consistent structure. This will make it easier for users to follow the modeling method. To keep the modeling process simple, the stage-gate process based on the Quality Gates approach is used. The modeling activities in this paper are named after the summarized activities from Cash and Kreye (2017), partially adapted from Goetz et al. (2018) and Matthiesen et al. (2019). The modeling method to be developed can be considered as a sequence of information action and representation action according to Cash and Kreye (2017). The result of this step should be a framework that has a consistent structure but still allows for different modeling activities. The stages and gates in the framework are then defined after this step.

Step 3: Elaboration of two workflows in the modeling method

This step aims to explore different modeling activities depending on the derivable SDK within the framework in step 2. As derived from section 1.3, two cases are interesting to follow: one, where the design engineers do not have enough information and need to generate it for a new robust concept, and the other is where they have too much information and need to filter it to improve the robustness of the existing solution by adapting the design. These result in different modeling activities, and therefore, different workflows. A workflow can be understood as a specific sequence of activities. Based on the general framework from step 2 and the differences in the two opposite cases, two workflows will be elaborated.

The coining machine described above is used here as an accompanying example to illustrate the solutions (see Figure 3). We use two different design situations to investigate the different modeling activities. The design situations are defined as follows: a company receives an order to develop a coining machine for a customer. The robust and high-quality minting of coins is the primary goal of a coining machine. However, the company has no experience in this area, and the level of derivable SDK is low. In this scenario, a bottom-up method is needed to integrate the collected information into concepts. Later, a concept is selected, detailed, and produced. In the second scenario, a problem occurs during mass production, where the coin is partially minted crooked. The company wants to solve this problem in the next product generation of the coining machine. Subsystems need to be redesigned to improve the robustness of the coining machine.

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

Since the previous product and its documentation already exist and various data can be collected, the level of derivable SDK is high. This scenario requires a top-down method due to the amount of data.

In these two situations, it is important to examine which activities are necessary to build the EFRT model for robustness evaluation of the coining machine. Different modeling activities are integrated into two workflows within the framework developed in step 2. The result of this step should be a unified modeling method with two workflows. Selected modeling steps for the coining machine are shown in the workflows for better illustration. During the elaboration of the two workflows, differences between them should be recorded, as they are essential for the adaptation of the modeling method in individual stages.

Step 4: Evaluation of the modeling method

In this step, an initial evaluation of the developed modeling method is performed using two case studies: a new development of a clipless bicycle pedal (low level of SDK) and an advancement of the next generation of an angle grinder (high level of SDK). First, the transferability of the developed workflows is evaluated to assess their applicability in different design contexts. In addition, the differences in the workflows found in step 3 are evaluated in this step. This evaluation is crucial to determine the extent of adaptation required for the workflows to ensure their effectiveness in different design scenarios. In order to facilitate comparison with the coining machine example, the modeling of these case studies will be performed and evaluated by the authors.

The scenarios of the two case studies are shown in the overview in Figure 3. The first case study involves a company that is new to the design of clipless bicycle pedals and is entering the market due to the growth of the bicycle market. Since the development team has no access to references besides buying other click pedals from the market. This case has lower data usability but a high level of innovation. In contrast, the second case study focuses on developing the next generation of an angle grinder during its concept phase. A significant challenge is the high rejection rate of output shafts due to tight assembly tolerances. With abundant data from previous development projects, this case study demonstrates high data usability but a lower innovation level, resulting in a high level of derivable SDK. Both cases are in the early stages of their respective development projects.

3. Development of the modeling method

This section describes the results from the first three steps outlined in section 2. It is structured as follows: in section 3.1, the EFRT model is formalized. In section 3.2, the framework for the modeling method is derived. In section 3.3, the details of the developed modeling method are elaborated.

3.1. Formalization of the EFRT model

The formalization starts with a formal definition of the EFRT model. An EFRT model is a combined model derived from the tolerance graph and the C&C^2-A. It models the qualitative as well as the quantitative information in a product concept and aims to evaluate its robustness. In this paper, we construct the EFRT model

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

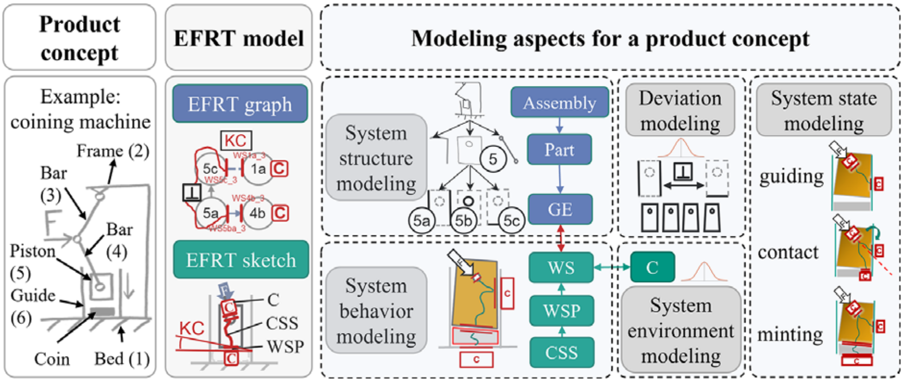


Figure 4. Formalization of the EFRT model (GE: Geometry Element; WS: Working Surface; WSP: Working Surface Pair; CSS: Channel and Support Structure; C: Connector). The EFRT graph and the EFRT sketch are derived from the product concept of the coining machine example. Five modeling aspects are used to evaluate the robustness of the product concept.

into the EFRT graph and the EFRT sketch. Figure 4 shows the EFRT graph and EFRT sketch with the example coining machine. The product concept of the coining machine and its main parts for the coining process are shown in Figure 4 on the left.

In the EFRT graph, the product concept is decomposed into GEs and their geometric relations. To build the EFRT graph, the assembly is first divided into several parts, for example, a piston. The next step is to divide a part into GEs, that is, interacting surfaces. For example, the part piston has surfaces such as the hole, skirt, and crown that serve as GEs, as shown in Figure 4 in the middle (system structure modeling). In the EFRT graph, the GEs are represented as nodes, for example, 5c is the crown and 5a is the skirt. Their relations are labeled on the edges, for example, the perpendicularity of the crown surface to the skirt surface is labeled on the edge between 5c and 5a (see Figure 4 – EFRT graph). The graph is supplemented with key elements of the C&C^2-A, that is, the WSS, CSS, and C. This supplementation has several advantages, such as tracking of load path in the graph.

In the EFRT sketch, a certain area in the product concept, which is considered to be important for function fulfillment, is visualized in a proper sketch. Key elements from C&C^2-A are drawn in the sketch for analyzing the functions of the system. In the accompanying example coining machine, a function for applying the minting force through the part piston on the coin is depicted in the EFRT sketch (see Figure 4 – EFRT sketch).

The concept of KC is used in the EFRT model to evaluate the fulfillment of functional requirements of the product. In an EFRT model, a KC can be assigned into the EFRT graph between GEs, or it can be drawn directly in the EFRT sketch (see Figure 4 – EFRT model).

An EFRT model can be used to model different aspects of a product concept. As described above, the process of building the EFRT graph serves to model the *system structure*, while the EFRT sketch aims to visualize the *system behavior* by finding the active WSP and CSS in the current system state. For a comprehensive view of a system, the *system environment* must be considered. The C is used to model the

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

system environment, which includes information beyond the system boundary, such as external loads.

For robustness evaluation, various *deviations* can be modeled in an EFRT model and their influence on function fulfillment can be investigated. Geometric deviations in the system are modeled in EFRT graph and visualized in the EFRT sketch. Non-geometric deviations, such as material properties or changes in the system environment, can be stored in model elements such as WSP, CSS, and C. To evaluate the influence of these deviations on the fulfillment of functional requirements, the relation between deviations and KCs can be then analyzed using EFRT graph and EFRT sketch.

The analysis is often not sufficient in only one *system state*. Therefore, it is imperative to evaluate the robustness of the product concept in different system states. To model system states, which can be categorized into temporal or spatial states, the C&C^2 sequence models are used (Matthiesen et al., 2019). Figure 4 on the right shows three temporal states and the system behavior of a coining machine relevant to the coin minting function.

3.2. Framework for the modeling method

Considering the requirement of facilitating the adaptation of modeling activities while keeping a consistent structure, we developed a framework for the modeling method corresponding to a stage-gate process. An overview of the framework is shown in Figure 5. The structure comprises five stages: 1. *Define*, 2. *Sketch*, 3. *Structure*, 4. *Model*, and 5. *Decide*. The gates for each stage also follow the common structure: 1. Defined task and data, 2. Sketch, 3. Product structure graph, 4. EFRT model, and 5. Decision. Two workflows are integrated in this common framework in this paper, while the modeling steps of the workflows in each stage are different.

Stage 1 *Define* is the initial processing of the derivable SDK in a database to support the next stages. In Stage 2 *Sketch*, the product concept with its main parts for fulfilling the functional requirements is outlined in an appropriate and

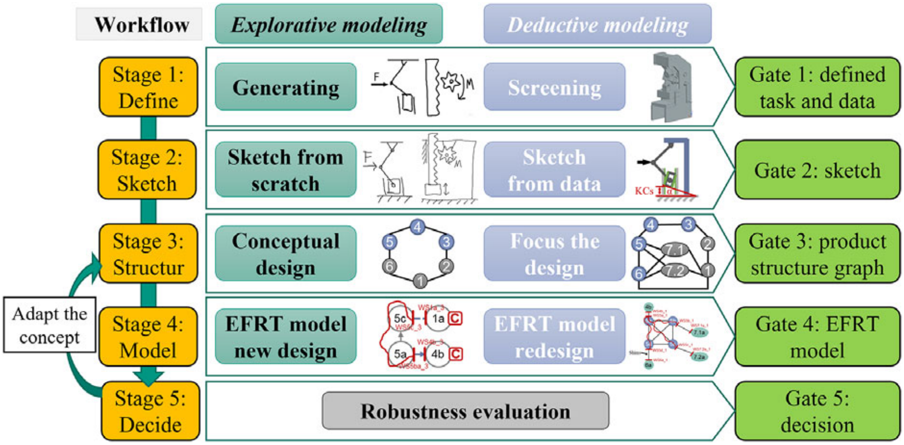


Figure 5. Overview of the framework for the modeling method with two workflows. *Explorative modeling* is shown on the left side and *Deductive modeling* is on the right side

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

simplified sketch. In Stage 3 *Structure*, the parts in the product concept and their geometric relations are structured and integrated into the product structure graph. In Stage 4 *Model*, the EFRT model is built with a sufficient resolution. By analyzing the influences of deviations on the function fulfillment, the robustness of the product concept is evaluated in Stage 5 *Decide*.

As described in section 1.3, modeling activities are different due to the level of the derivable SDK. With a low level of derivable SDK, it is difficult to define the task accurately at the first time, and the system structure must be conceptualized first. While the system environment and system states are mostly unknown, the deviations in the system should be estimated based on prior knowledge and experience. Information such as material properties must be defined or collected for a more detailed analysis of the system behavior. With a high level of derivable SDK, it is imperative to first filter the information to a sufficient amount. Meanwhile, some modeling aspects described in section 3.1 can be found or derived directly from the existing data. The required modeling activities for different levels of derivable SDK are investigated with two cases described in section 2. As a result, the modeling method for the EFRT model is divided into two workflows (see Figure 5):

- *Explorative modeling*: low level of derivable SDK
- *Deductive modeling*: high level of derivable SDK

3.3. Elaboration of two workflows in the modeling method

This section presents the modeling activities through five stages in the two workflows within the modeling method. The steps in the workflows and a first analysis of the differences between the workflows are described in the following subsections 3.3.1–3.3.5.

3.3.1. Stage 1 Define

Figure 6 shows the steps of the workflows in the Stage 1 *Define*. In this stage, the task should be clarified and information should be gathered and documented. At the end of this stage, sufficient data should be available for analysis in the next stages. Gate 1 of the stage is then the defined task and data that will serve as a database for the later modeling process. The letter L in the steps stands for low level of derivable SDK and H stands for high level.

Explorative modeling is shown in Figure 6 on the left. At the start of the development, the level of derivable SDK is considered to be low. In step L1.1, *defining the task*, the task should be clearly defined to establish a focused direction for modeling. In step L1.2, *generating information*, the state of the art and the possible principal solutions from the market are researched. Principal solutions can be created using ideation methods, such as brainstorming (Hatcher et al., 2018). The principal solutions to the tasks are generated methodically, for example, in a morphological box (Zwicky, 1967). With step L1.3, *documentation of gathered data*, Gate 1 is reached. The purpose is to systematically explore and generate diverse solutions when starting with limited SDK.

Deductive modeling is shown in Figure 6 on the right. At the start of the development, the level of derivable SDK is considered to be high. In step H1.1, refining the task, problem areas are identified, identifying specific problem areas that require further development. In step H1.2, *screening available data*, relevant

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

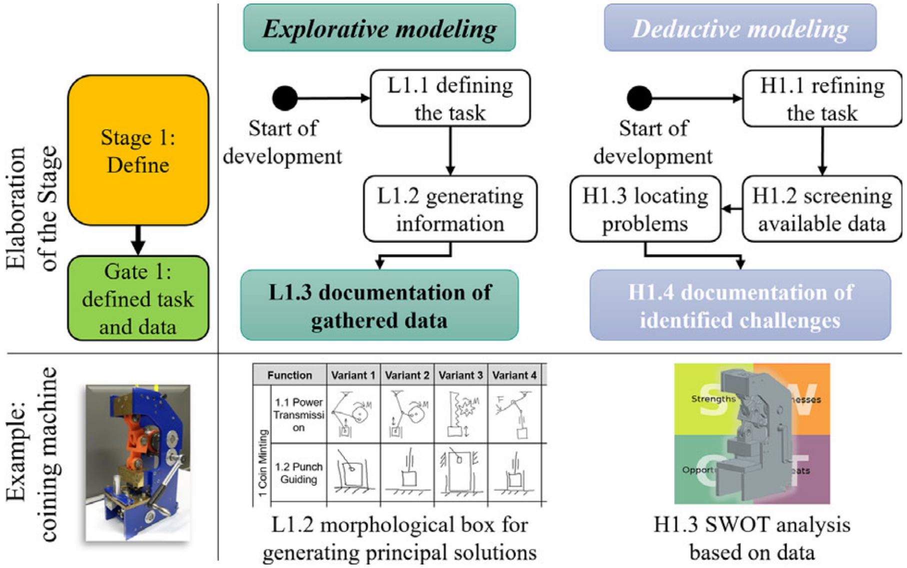


Figure 6. Overview of Stage 1 Define and selected steps of the accompanying example coining machine.

data from references are screened, as the further development is based on these data. References, such as CAD data from the previous product generation, can be used, as well as external references, for example, competitor products. Step H1.3 is to *locate "problem spots" in existing data*. This can be done using problem analysis methods, such as SPALTEN (Albers et al., 2016) or SWOT (Strengths, Weaknesses, Opportunities, and Threats) analysis. Step H1.4 is the *documentation of the identified challenges* to establish a clear understanding of the issues to be addressed. The purpose of this is to systematically refine and address specific challenges or issues leveraging existing knowledge and data.

For the accompanying example coining machine, in *Explorative modeling*, basic information about the system to be developed is gathered. In step L1.1, the task is defined as minting the coin consistently, which results in a constant height and as the target variables. This is also the basis for the later evaluation. In step L1.2, different principal solutions are collected or developed in a morphological box (see Figure 6). In *Deductive modeling*, a situation analysis and problem definition are carried out in this stage, the problem from the previous product generation is that the coin was minted crooked. This means that the actual function (crooked minting) deviates from the target function (high minting quality). Therefore, the task is refined to improve the parallelism from both sides of the coin in step H1.1. In step H1.2, the CAD data of the previous coining machine is screened. A SWOT analysis is conducted to concretize the problem (see Figure 6). As a result of step H1.3, the problem lies with the coining mechanism in the guide area.

The primary difference between *Explorative modeling* and *Deductive modeling* is how the knowledge is structured from the different levels of derivable SDK. The information content of *Deductive modeling* is higher than that of *Explorative modeling* at the start of development. During task defining, *Explorative modeling* requires defining the task if it is unclear, whereas *Deductive modeling* requires refining the task under the consideration of current problems. In the accompanying

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

example, *Explorative modeling* aims to mint a coin, but the solution is not yet known. Conversely, with *Deductive modeling*, the task is more concrete: improving coin quality. In *Explorative modeling*, identifying problems can be difficult due to a lack of information. In *Deductive modeling*, it can be challenging to identify causes of known problems. To reach Gate 1, *Explorative modeling* requires gathering information, while *Deductive modeling* requires screening data and selecting relevant information.

3.3.2. Stage 2 Sketch

Figure 7 shows the workflow steps in Stage 2 *Sketch*. In this stage, the documentation from Stage 1 is used to sketch the product concept. Gate 2 consists of sketches that represent the principal solutions and have sufficient resolution for analysis in subsequent stages.

Figure 7 on the left illustrates the process of *Explorative modeling*. In step L2.1, *selecting solutions*, the collected principal solutions are evaluated and selected with appropriate methods, for example, the weighted sum model (Pahl et al., 2007). The purpose of this step is to identify suitable solutions for the product concept. Bad solutions can also be initially eliminated with a rough estimate based on experience. Step L2.2 involves *defining the temporal and spatial system boundary*. The purpose of this step is to focus on the areas that are relevant to the functional fulfillment. When defining the spatial system boundary, it is necessary to consider the connecting components outside the system boundary that are essential for concept realization. Finally, step L2.3 *involves combining the principal solutions in a product concept and drawing it into a sketch*. The purpose here is to facilitate the further detailing and refinement of the concept.

Figure 7 on the right illustrates the process of *Deductive modeling*. In step H2.1, *analyzing available data with C&C^2-A*, the components required to fulfill functional requirements are identified by finding active WSPs and CSSs. If only 3D data is available, it is necessary to identify suitable intersections to visualize the core

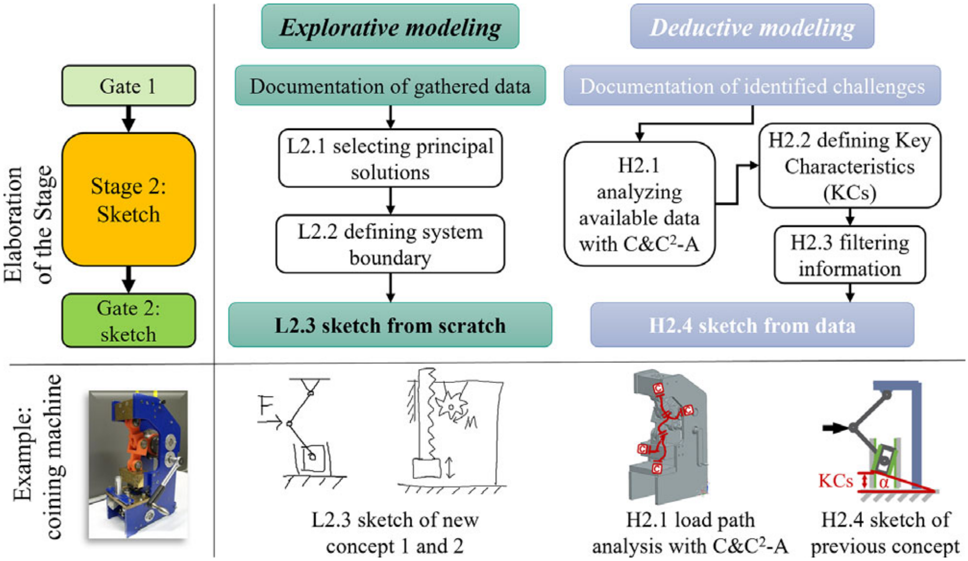

Explorative modeling
Deductive modeling
Gate 1
Stage 2: Sketch
Gate 2: sketch
Elaboration of the Stage
Documentation of gathered data
L2.1 selecting principal solutions
L2.2 defining system boundary
L2.3 sketch from scratch
Documentation of identified challenges
H2.1 analyzing available data with C&C²-A
H2.2 defining Key Characteristics (KCs)
H2.3 filtering information
H2.4 sketch from data
Example: coining machine
F
M
KCs
α
L2.3 sketch of new concept 1 and 2
H2.1 load path analysis with C&C²-A
H2.4 sketch of previous concept

Figure 7. Overview of Stage 2 Sketch and selected steps of the accompanying example coining machine.

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

elements of the C&C²-A. In step H2.2, *defining KCs* in the existing system, the purpose is to highlight critical features that influence the system's functional fulfillment. In step H2.3, *filtering information* is performed to retrieve the principal solution from existing data, such as CAD, aiming to reduce the effort of modeling. Finally, step H2.4 involves *drawing the existing concept that includes the KCs*. The purpose is to provide a clear and simplified visualization of the refined concept based on the previous product generation.

For the accompanying example coining machine, in *Explorative modeling*, the collected information of the principal solutions lead to different concepts for the coining mechanism. In step L2.1, the different principal solutions are first evaluated and selected based on company-specific criteria. In step L2.2, the spatial system boundary is limited to the area through the coining load path without the drive mechanism. The temporal system boundary is one minting cycle. Following the system boundary, two selected concepts are sketched in step L2.3 (see Figure 7). In *Deductive modeling*, the function-relevant parts that are responsible for the crooked minting of the coins are identified from the previous product generation in step H2.1. For this purpose, a load path analysis is conducted (see Figure 7). In step H2.2, two KCs are defined as the angle α and the height *h* of the coin, which can be determined between the bed surface and the piston crown surface during the minting process. The retrieved product concept from step H2.3 and the defined KCs are then drawn in a sketch in step H2.4 (see Figure 7).

Different activities to create a sketch can be identified between the two workflows due to the different databases. Compared to *Explorative modeling*, *Deductive modeling* facilitates a more detailed analysis of the concept due to its high information content. In *Deductive modeling*, analysis with C&C^2-A can help filter the data, while in *Explorative modeling*, many terms still need to be defined to reach Gate 2. Since the problem is assumed to be known in *Deductive modeling*, the KCs can be derived directly from the task definition and defined in the existing product concept. In *Explorative modeling*, the product concept needs to be detailed to define the KCs in the next stage. To reach Gate 2, *Explorative modeling* requires sketching from limited data, while *Deductive modeling* requires extracting sketches from a larger amount of data.

3.3.3. Stage 3 Structure

Figure 8 illustrates the workflow steps in Stage 3 *Structure*. Based on the created sketch from Stage 2, the principal solutions can be decomposed into function-relevant parts and their relations in the product structure graph, which forms Gate 3. A detailed introduction to the product structure graph is described by Goetz et al. (2018).

Figure 8 on the left illustrates the process of *Explorative modeling*. In step L3.1, *defining parts and relations*, the sketched product concept is detailed with independent and function-relevant parts or subassemblies. This step also involves defining the geometric relations between the parts, that is, types of contact. The types of contact are limited to a few basic interface relations, such as fixed contact, prismatic joint, or cylindrical joint between two parts (Chase et al., 1996). The purpose of this step is to enhance the product concept with realizable parts. In step L3.2, *KCs are defined* and related to the interface relations between the parts, aiming at analyzing the functional fulfillment. Finally, step L3.3 involves creating a

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

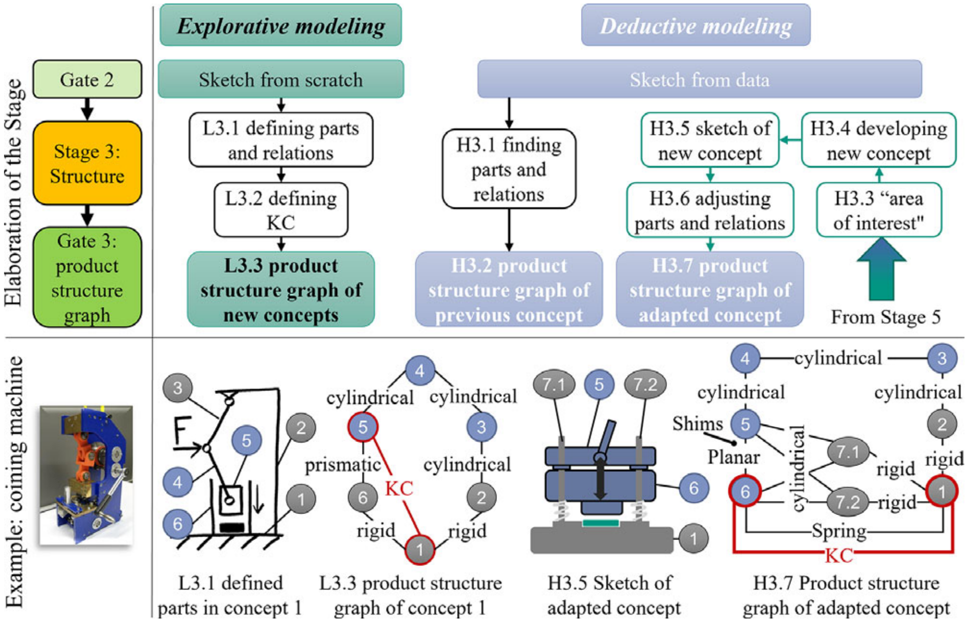

Explorative modeling
Deductive modeling
Elaboration of the Stage
Gate 2
Stage 3: Structure
Gate 3: product structure graph
Sketch from scratch
L3.1 defining parts and relations
L3.2 defining KC
L3.3 product structure graph of new concepts
Sketch from data
H3.1 finding parts and relations
H3.2 product structure graph of previous concept
H3.5 sketch of new concept
H3.4 developing new concept
H3.6 adjusting parts and relations
H3.3 "area of interest"
H3.7 product structure graph of adapted concept
From Stage 5
Example: coining machine
F
cylindrical
prismatic
KC
rigid
7.1
7.2
Shims
Planar
Spring
L3.1 defined parts in concept 1
L3.3 product structure graph of concept 1
H3.5 Sketch of adapted concept
H3.7 Product structure graph of adapted concept

Figure 8. Overview of Stage 3 Structure and selected steps of the accompanying example coining machine.

product structure graph in which each node is a part or subassembly, and each edge is an interface relation between them. The purpose of the product structure graph is to ensure a clear understanding of how parts will interact within the product and provide the basis for further modeling steps.

Figure 8 on the right illustrates the process of *Deductive modeling*. In step H3.1, *finding parts and relations*, the function-relevant parts and their relations already exist in the previous product generation, and they can be found from available data. Based on that, the *product structure graph for the existing concept* is derived in step H3.2. The focus of these two steps is to reduce the model information from the entire dataset to only the necessary amount. The modeling process for the previous product generation then continues in Stage 4. Also at Stage 3, an iteration takes place, as step H3.3 is continued after the complete evaluation of the existing concept in Stage 5. The existing concept will be further developed in steps H3.3–H3.7. Step H3.3 defines the *"area of interest,"* which is related to the localized problem in step H1.3. This step identifies the area that requires adaptation. In step H3.4, *developing a new concept*, the existing concept is adapted in the identified area of interest. The *adapted concept is then sketched* in step H3.5. The adapted concept aims to solve the identified problem, though its robustness has not yet been assessed. In step H3.6, the *defined parts and relations are adjusted* from the existing ones. This adaptation results in a *new product structure graph* for the current concept in step H3.7.

For the accompanying example coining machine, in *Explorative modeling*, the parts that are relevant for coin minting and their relations are first defined and numbered in step L3.1 (see Figure 8). For example, in concept 1, there is a cylindrical joint between the part 4 bar and part 5 piston, while there is a prismatic joint between the part 5 piston and part 6 guide. Two KCs are identified in L3.2 as

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

the angle α and the height h of the coin. Then, the product structure graphs of both concepts are created with this information (see Figure 8). In *Deductive modeling*, the relevant parts and their relations are found from the existing data. Then, the product structure graph of the existing concept is created and evaluated in further stages. After the evaluation, a concept adaptation in the area of minting takes place to improve the minting quality. A new concept with an extended guide is developed in step H3.4 and sketched in step H3.5 (see Figure 8). The adapted concept aims to increase robustness by decoupling the guide and stamping functions in the guide area. Therefore, the piston is divided into two parts, the upper part provides the stamping force and the lower part has a precise and long guide to prevent tilting angle (KC). Based on the sketch, the defined parts and their geometric relations from the previous concept can be adjusted in step H3.6. The product structure graph of the new concept is then created in step H3.7 (see Figure 8).

Differences between the two workflows can be identified. In *Explorative modeling*, the new concepts must be specified with parts and their geometric relations, deriving new product structure graphs. In *Deductive modeling*, the first product structure graph is derived from the existing concept, where parts and relations already exist. Then, a new concept is derived and the product structure graph is adapted in the area of interest. In this stage, *Explorative modeling* extends the information content constantly, while *Deductive modeling* first limits the information content for the existing concept and then extends it for the new concept.

3.3.4. Stage 4 Model

Figure 9 shows the steps of the workflows in Stage 4 *Model*. Based on the created sketches from Stage 2 and the product structure graphs from Stage 3, the EFRT model is built in this stage. Different system states are also taken into account to

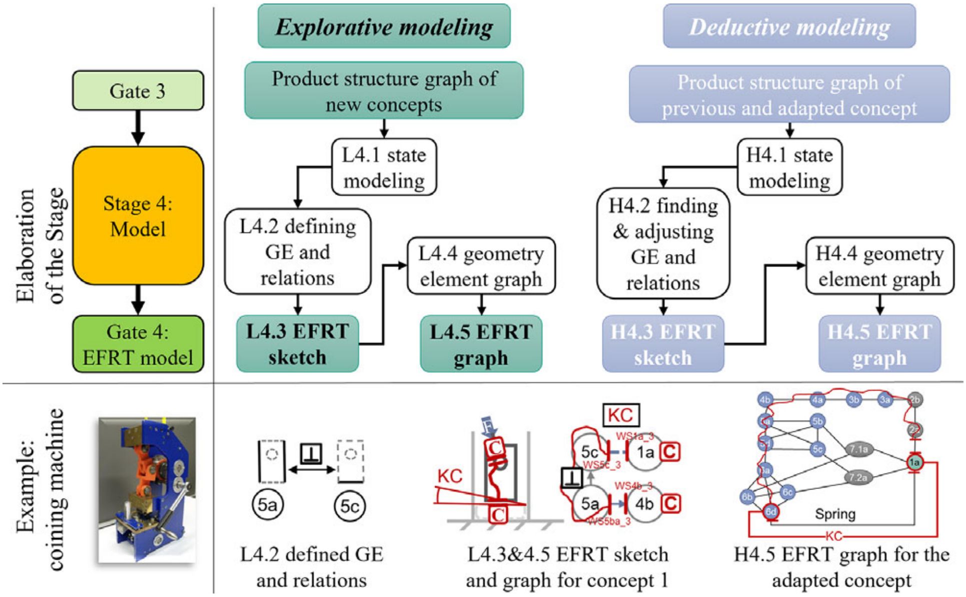

Elaboration of the Stage
Gate 3
Stage 4: Model
Gate 4: EFRT model
Explorative modeling
Product structure graph of new concepts
L4.1 state modeling
L4.2 defining GE and relations
L4.3 EFRT sketch
L4.4 geometry element graph
L4.5 EFRT graph
Deductive modeling
Product structure graph of previous and adapted concept
H4.1 state modeling
H4.2 finding & adjusting GE and relations
H4.3 EFRT sketch
H4.4 geometry element graph
H4.5 EFRT graph
Example: coining machine
5a
5c
L4.2 defined GE and relations
F
C
KC
5c
1a
WS1a_3
WS5c_3
5a
4b
WS4b_3
WS5ba_3
L4.3&4.5 EFRT sketch and graph for concept 1
4b
4a
3b
3a
2b
5b
5c
7.1a
7.2a
1a
6b
6c
6d
Spring
H4.5 EFRT graph for the adapted concept

Figure 9. Overview of Stage 4 Model and selected steps of the accompanying example coining machine.

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

derive state-dependent models. Gate 4 represents the EFRT model, including the EFRT graph and the EFRT sketch.

Figure 9 on the left illustrates the process of *Explorative modeling.* L4.1 is *modeling of the system states.* Based on the sketch, different system states are investigated with the C&C^2 sequence model, which involves variable WSPs. The purpose of this step is to understand how the system behaves under different states. Step L4.2 involves *defining GEs and their geometric relations.* This step includes specifying the interactions between the parts defined in Stage 3, as well as the relative location and orientation of the GEs within a part, such as parallelism or perpendicularity. In step L4.3, the product concept is visualized in different system states using *EFRT sketches.* Step L4.4 involves deriving the *GE graph*, from which the functional tolerance chain is determined. A detailed introduction to the creation of a GE graph is described by Goetz et al. (2018). Step L4.5 involves assigning the elements of the C&C^2-A into the geometric element graph to derive the *EFRT graph.* The EFRT graph, as well as the EFRT sketch in different system states, serves as the basis for robustness evaluation in the next stage.

Deductive modeling is shown in Figure 9 on the right. In step H4.1, *state modeling*, the previously identified problems in Stages 1–3 should be given special attention. In step H4.2, *GEs and their relations* can be found in the existing concept, and they are adjusted for the adapted concept. The description for steps H4.3–H4.5 is identical to steps L4.3–L4.5.

For the accompanying example coining machine, in *Explorative modeling*, different system states around the minting process are modeled in step L4.1. Details of state modeling for the coining machine are shown in section 3.1, Figure 4. Relevant GEs and their geometric relations are then defined in step L4.2, for example, the GE 5a piston skirt and 5c crown as well as the perpendicularity between these two GEs (see Figure 9). The minting process is then visualized with the EFRT sketch in step L4.3. After the derivation of the GE graph in step L4.4, the EFRT graph for the coining machine is created for further analysis. In *Deductive modeling*, the process can be repeated with the previous and adapted concept. Figure 9 shows the EFRT sketch and part of EFRT graph for concept 1 (*Explorative modeling*), while the full EFRT graph is shown for the adapted concept (*Deductive modeling*).

Although the steps in Stage 4 are similar between the two workflows, differences still exist. While the system states have to be first identified in *Explorative modeling*, knowledge about system states already exists in *Deductive modeling.* In addition, existing tolerance information can be used in *Deductive modeling* to help derive function-relevant GEs and their geometric relations. According to Goetz et al. (2021), a faster derivation of the GE graph is possible with available CAD data.

3.3.5. Stage 5 Decide

In Stage 5 *Decide*, the EFRT model built from Stage 4 is used for robustness evaluation. Gate 5 incorporates the design decision made from this evaluation. Since the steps up to the robustness evaluation are identical in both workflows, no distinction is made between L and H in this stage. The steps of the workflow are shown in Figure 10.

Step 5.1 identifies the *critical states* selected from the state modeling in Stage 4. The purpose of this step is to pinpoint the most significant states that need

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

thorough evaluation. In step 5.2, *the EFRT graph is simplified* by deleting longer redundant loops for the considered KC. The method to derive an extracted functional tolerance chain of a KC is described in detail by Goetz et al. (2018). The purpose here is to make the graph more manageable and focused on critical interactions. In step 5.3, *critical deviations* are identified. They are assigned in the nodes and edges of the EFRT graph for analysis of their influence on KC and then visualized in the relevant parts in the sketch. Step 5.4 is *analysis with EFRT sketch*. Due to the identified critical states, a failure mode can now be predicted by analyzing the system behavior under deviations in the EFRT sketch. Using appropriate evaluation criteria, for example, the criteria according to Li et al. (2023), *robustness evaluation* can be carried out in step 5.5. In addition to the robustness evaluation, the *design decision* in Gate 5 should also consider criteria from other disciplines, for example, economy.

The accompanying example coining machine is also illustrated in Figure 10, attached to the steps. Step 5.1 identifies the state "minting" as a critical state because the lateral force from the bar results in a crooked piston orientation. In step 5.2, two loops are identified in the EFRT graph around the KC, and the parts involved in the shorter tolerance loop are focused for detailed analysis. In step 5.3, geometric deviation in the parallelism of the part piston is identified as critical because it affects the WSP between the piston and the guide and thus the angle of minting. Another critical deviation can occur in the perpendicularity between the piston crown surface and the bed. The visualization of the deviations and their effect on the system behavior is then analyzed in step 5.4, which is shown in Figure 10. With the tilting moment due to the force application angle in the example shown, the piston is pressed against the guide during the minting process. Therefore, the coin is minted crooked if the shown deviations exist in the piston. The system behavior is compared among the concepts to evaluate their robustness to the functional fulfillment, that is, the KC angle should be zero. This analysis brings new insight for the concept adaptation, ideas for reducing the tilting moment can be used to adapt

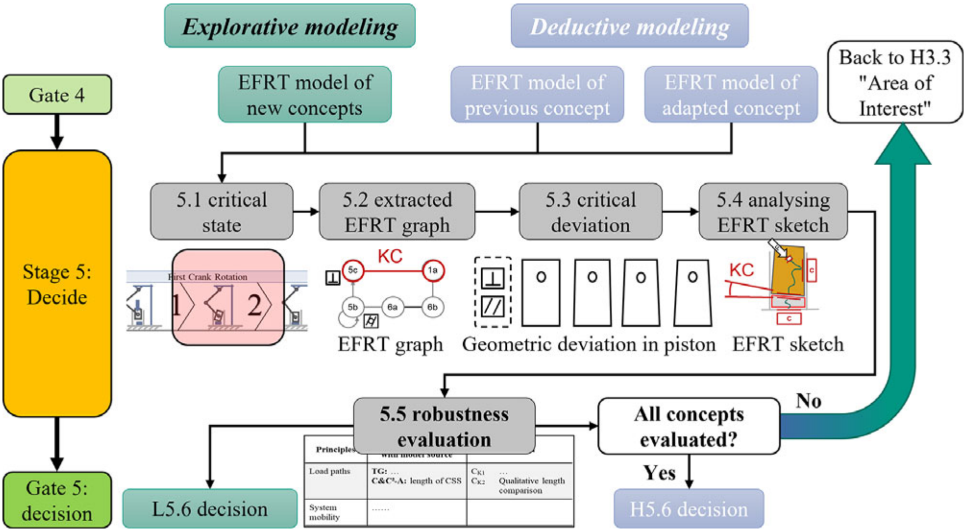


Figure 10. Overview of Stage 5 Decide.

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

the previous concept. For a more comprehensive robustness evaluation, other critical deviations should be considered and the overall effects should be evaluated with appropriate criteria.

In this stage, the robustness evaluation follows identical steps in both workflows. The main difference lies in the information content. The information content in *Deductive modeling* is higher than that of *Explorative modeling*. Another difference lies in the iteration in *Deductive modeling* between Stages 3 and 5, aiming to adapt the existing concept after its initial evaluation.

4. Evaluation of the modeling method with case studies

As described in section 2, the evaluation focuses on the transferability of the developed modeling method and the differences between the workflows with two case studies.

4.1. Transferability of the modeling method

The developed modeling method was implemented by the authors with two different case studies to evaluate its transferability. The implementation of the modeling method with selected steps is illustrated in Figure 11 and briefly summarized below.

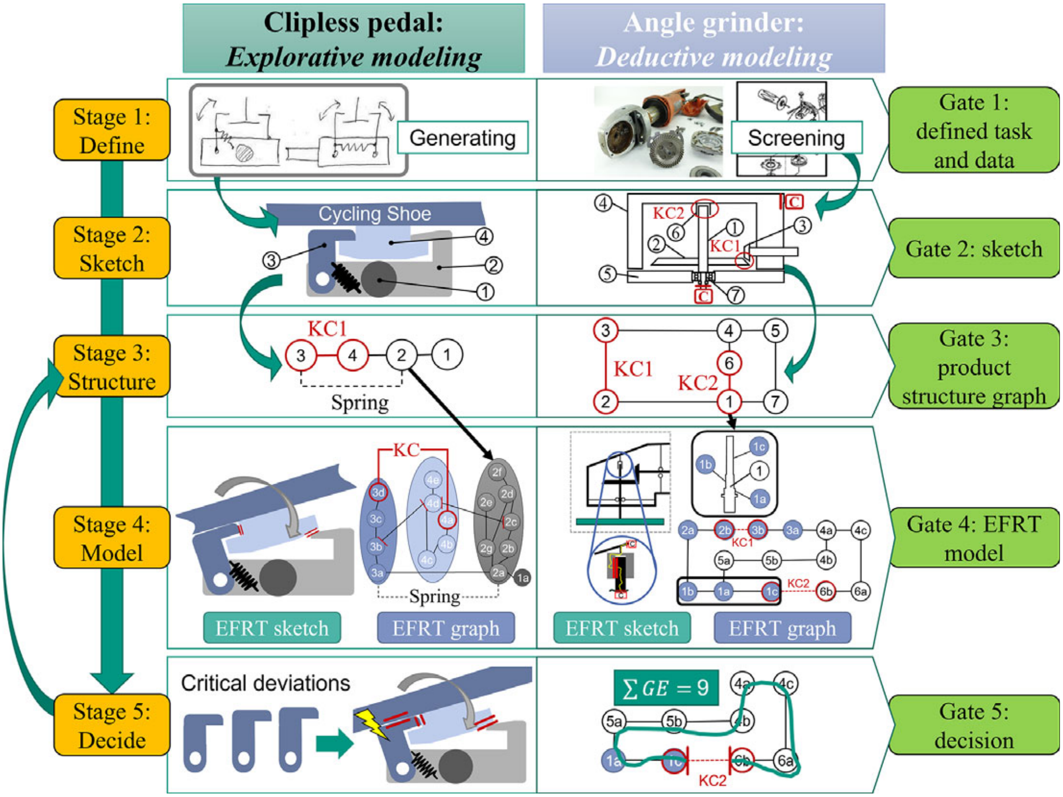


Figure 11. Selected steps for the implementation of the modeling method in the case studies.

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

The first case study with bicycle clipless pedals evaluates *Explorative modeling* for a low level of derivable SDK. Two concepts were created in the project. One is based on an existing concept from a competitor in the clipless pedal market and the other is a new idea (see Figure 11). The first concept has one rotating and one fixed hook, requiring riders to tilt the tip of their foot down to insert the cleat into the mechanism. The second concept has two rotating hooks that hold the pedal cleat in place. In Stage 1, the first step is to define the task. The mechanism must enable the clip-in and clip-out of the pedal plate while ensuring a secure grip of the shoe on the pedal. In Stage 2, sketches are created in the area of pedal axle, pedal body, and pedal plate. In Stage 3, product structure graphs are derived. The gap between the cleat upper surface and the shoe is defined as a KC, as this is important for the function click-in and ergonomics. The EFRT graph and sketch are then built in Stage 4 and used for the robustness evaluation in Stage 5. In this case, a critical state can be identified when considering the clip-in function. The gap between the top of the cleat and the shoe (KC1) must be large enough to allow the hook to close unhindered. Otherwise, there may be a penetration between the hook and shoe by clip-in if there is a deviation in the hook length. This penetration results in undesired WSP and can be visualized with the EFRT sketch. It should be noted that this evaluation is based on only one critical state, and multiple states must be considered for an overall evaluation.

The second case study with an angle grinder evaluates *Deductive modeling* for a high level of derivable SDK. Figure 11 shows the current generation of the angle grinder, where both shafts connect the bevel gear and pinion in a perpendicular arrangement. The bearing concept for the output shaft includes a deep groove ball bearing in the bearing cap and a plain bearing in the housing. In Stage 1, the task is defined as reducing the rejection rate in the assembly. In Stage 2, the angle grinder is simplified in a sketch with the components relevant to the task. Derived from the task, two KCs are defined. KC1 is the perpendicularity between the input and output shaft as it affects the gear function. KC2 is the gap in the plain bearing in the housing, as the problem usually occurs here during assembly. Then, in Stage 3, the relationships of these components are represented in the product structure graph. In Stage 4, the EFRT model of the angle grinder is built. In Stage 5, the extracted EFRT graph for KC is derived. It becomes clear that the number of GEs for KC2 can be reduced. This insight leads to an adaptation of the concept, where the main shaft is supported only by two deep groove ball bearings in the bearing cap. It should be noted that this adaptation only addresses the assembly problem. Other functional requirements must also be considered for an overall evaluation.

In both case studies, the workflows enabled the EFRT models to be built systematically, which served as the foundation for robustness analyses of the respective product concepts. However, certain adaptations were deemed necessary due to differences in available references compared to the coining machine, such as a later redefinition of the KC based on new insights from state modeling in the clipless pedal case study. Considering the information available in the current development project, design engineers are allowed to adapt the steps within the stages as long as the gate can be achieved. Further evaluation of the differences between the workflows is provided in the subsection below.

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

4.2. Differences between the workflows

Of particular interest in evaluating the modeling method are the differences between the two workflows. The differences are initially identified in the development of the modeling method, using the example of a coining machine, and are further elaborated in Table 1. By evaluating these differences through the lens of the two case studies, specific required actions have been derived. These actions reflect necessary adaptations due to different levels of derivable SDK between the case studies and the coining machine example. These required actions are summarized in Table 1 for clarity and comparison.

The workflows differ in *tasks and problems.* In *Explorative modeling*, the task is to design a functional product and the problem is mostly not clearly defined, while in *Deductive modeling*, the product is already designed, and the goal is to refine it with the given problems or requirements. The case studies confirm this difference. In case study 1, the task is defined as enabling the clip-in and clip-out of the pedal plate while ensuring the shoe securely grips the pedal. A problem is first identified when a critical state is found in the state model. In case study 2, reducing the rejection rate to solve the known assembly problem refines the task. To clearly define the task and problems, ideation or analysis methods can be helpful.

The *information content in the gates* shows a different direction of information processing in both workflows. In *Explorative modeling*, the information in the gates is constantly being expanded, while in *Deductive modeling*, the information in the gates is increasingly being precise. Comparing the two workflows, the SDK changes in different directions. The higher the stage, the less difference of the SDK exists in the gates. At the end of Stage 4, the SDK is very similar in both workflows. This flow of information through gates in both workflows is also confirmed by the case studies. In case study 1, information such as parts or relationships is not available, less information can be used directly. More details are added from the original hand sketch in order to build the EFRT model. In case study 2, various information

Table 1. Analysis of differences in the workflows and required actions for each difference

Identified differences in the workflows	Required actions
Tasks and problems	Suggestion of ideation and analysis methods
Information content in the gates	A clear description of the required information content in each gate
Definition of KCs	Iterative process to define KCs
Focus of modeling	Method to limit the system boundary
Sequence of steps	A knowledge-based software to support selection in modeling steps
Modeling of system states	Considering different resolutions for modeling rough and detailed states
Modeling of deviation	Method to select different deviations for modeling
Focus of evaluation	A multi-criteria robustness evaluation method

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

can be reused, such as CAD data for finding GE and their relations. The analysis removes unnecessary information at each stage. This reveals the need for a clear description of the required information content in each gate.

The *definition of KCs* also varies between the two workflows. In *Deductive modeling*, KC can be directly relocated in the given embodiment of the existing concept, considering the key function of the system. In *Explorative modeling*, the concept has to be structured first before KC can be defined. The case studies partially support this difference. In the case study 2, two KCs, the gearing and the gap in the sliding bearing, are defined in Stage 2. They are derived from the problem of a high rejection rate in assembly. In case study 1, the first KC is identified as the gap between the pedal plate and the pedal body, but these parts are not defined until step 3, so the KC is also defined later. During state modeling, another KC, the gap between the clear upper surface and the shoe, is identified as a critical parameter for the clip-in function. This means that new information may emerge during state modeling, which may lead to a new definition of KCs. An iterative process is required to define KCs in the modeling method.

The *focus of modeling* differs between the two workflows. *Explorative modeling* concentrates on the whole system, while *Deductive modeling* targets the subsystems that are modified during development. Besides that, *Deductive modeling* allows for a more comprehensive analysis of the tolerance chain, as it includes more defined elements, such as connecting elements or seals. *Explorative modeling* often omits these elements in the concepts. The case studies confirm this difference. In case study 1, the whole click pedal systems are modeled, but only with the essential parts for function fulfillment. In case study 2, the modeling is carried out in a subsystem around the bearing and relevant connecting elements, but in more detail. This difference is addressed in *Explorative modeling* in step L2.2 “defining system boundary” and in *Deductive modeling* in step H3.3 “area of interest.” A method to limit the system boundary for the analysis should be useful.

The two workflows have different *sequences of steps*. In *Explorative modeling*, the concepts are first generated, then analyzed and modeled. In *Deductive modeling*, the existing system is analyzed and modeled first, and then a new concept is derived. The case studies confirm this difference, and the proposed steps in the modeling method are generally applicable to both case studies. However, some steps are optional for the evaluation, for example, the product structure graph can be omitted if the GEs and their relations are easy to find from the existing concept. Still, the product structure graph serves as a gate stores intermediate information, aiming at automated information processing. It requires knowledge and experience to reduce the effort by selecting modeling steps, a knowledge-based software should support the design engineer in selecting the necessary modeling steps and their sequence.

The *modeling of system states* differs between the two workflows. In *Explorative modeling*, the system states are unknown and need to be investigated, while in *Deductive modeling*, they are given by the previous product. The case studies partially confirm this difference. In case study 1, three states are identified: clip-in, operation, and clip-out. Further analysis shows that there are sub-states that need to be explored through modeling. In case study 2, only one state addresses the assembly problem. It is necessary to select sufficient resolution for state modeling, in order to find the right problem. Rough system states can be determined through experience. For dynamic systems, it may be necessary to model detailed sub-states.

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

The *modeling of deviation* differs between the two workflows. In *Explorative modeling*, the deviation modeling is based on assumptions, and the aim is to find the influence of the assumed deviations on the function fulfillment. In *Deductive modeling*, the problem is given, it must be investigated what is the relationship between the given problem and the deviations. The case studies confirm this difference. In case study 2, the existing assembly problem and the tolerance information allow an early analysis with deviations, while in case study 1, the relevant deviations must be identified through the EFRT graph and sketch. Functional fulfillment is affected by more than a single deviation. Therefore, a method to select different deviations in terms of their impact and frequency of occurrence is required for sufficient modeling.

The *focus of evaluation* varies in the two workflows. In *Explorative modeling*, the entire concepts are assessed, with an emphasis on modeling the critical states in the EFRT sketch. In *Deductive modeling*, the evaluation focuses on the modified parts relative to the unchanged parts, using the EFRT graph as the main tool for comparison. The case studies confirm this difference. In case study 1, the critical states in the clip-in are identified using the EFRT sketch. In case study 2, the evaluation is based on the EFRT graph, which shows the contribution of GEs to the KC and therefore the rejection rate. The evaluation so far is based on single criteria. A multi-criteria evaluation method, for example, the robustness evaluation matrix by Goetz et al. (2020), should be taken into account.

5. Discussion

Based on the results, the research question “*How can an EFRT model be built whilst taking different levels of derivable specific design knowledge for early robustness evaluation into account?*” can be answered as follows.

Motivated by the high potential of the EFRT model for early robustness evaluation and recognizing the insufficient methodical support to build it, a modeling method based on the EFRT model with a detailed description of the modeling steps is provided in this paper. Depending on the design situation at the start of the development, the derivable SDK is often different. Therefore, different modeling activities can be derived to obtain the required SDK for robustness evaluation. The proposed modeling method for the EFRT model takes these differences into account when compared to the previously common qualitative modeling methods, such as Function-Behaviour-Structure (FBS) (Gero and Kannengiesser, 2014), C-K theory (Hatchuel and Weil, 2009), and modeling method with C&C^2-A (Matthiesen, 2021). It allows the sequence of modeling activities to be adapted within a stage-gate process. Using two contrasting cases, we developed two situation-specific workflows for building an EFRT model for early robustness evaluation. These workflows demonstrate the flexibility of the method in adapting to different design scenarios. While the description of RD methods in previous studies is based on a single context (Goetz et al., 2018; Göhler and Howard, 2015; Mathias et al., 2011), our method allows design engineers to tailor the modeling steps to the specific requirements of each case, rather than strictly following a predefined sequence.

For RD, awareness of deviations is emerging as a fundamental step, but effectively dealing with such deviations in the early design stages remains a challenge (Ebro and Howard, 2016; Hasenkamp et al., 2009). The formalization

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

of the EFRT model and the proposed modeling method in this paper allow for early modeling of product concept deviations, enabling a systematic analysis of their impact on functional fulfillment, which is not considered in experience-based approaches based on RD principles (Andersson, 1997; Ebro et al., 2012; Ebro and Howard, 2016). While existing experience-based RD principles assist in generating robust concepts, it is still difficult to determine which principle to apply to a given design task. The modeling method based on the EFRT model complements these approaches by allowing for the systematic modeling and evaluation of the relationship between deviations and functions in the generated concepts. Compared to methods like VMEA (Johansson et al., 2006) or FMEA (Teng and Ho, 1996), which operate at a higher level of abstraction and are less directly applicable in design, the proposed modeling method offers a more detailed deviation analysis of the design object. However, VMEA and FMEA can be integrated into the initial and final stages of the modeling method based on the EFRT model to help identify critical deviations and potential problems in the product concept. Managing such deviations early can save costly tolerance issues in later design stages. For example, in the case of the coining machine discussed in this paper, reducing the tilting moment in the product concept can significantly reduce the cost of guide tolerances. The modeling method based on the EFRT model enhances RD by focusing on the relationship between deviations and functions in the early stages, reducing costly iterations caused by inadequate consideration in RD.

To ensure adequate modeling of the relationship between deviations and functions in early RD, an appropriate method is essential. The modeling method proposed in this paper provides a structured, step-by-step approach for evaluating the robustness of individual product concepts based on derivable SDK. With systematic modeling of geometric relationships and system behavior, the robustness of product concepts can now be evaluated more in-depth. By providing detailed descriptions of each modeling step and guidelines for robustness evaluation, this method enhances support for early RD and helps reduce the risk of selecting inappropriate design methods due to insufficient knowledge (Wilmsen et al., 2019). Therefore, we believe that the modeling method based on the EFRT model provides better support to design engineers throughout the RD process.

This paper investigates two cases of RD, the development of a new robust concept, and the adaptation of a previous concept for robustness improvement. The real design situation is mostly a mixed situation; therefore, the two workflows developed in the modeling method don’t cover all the design situations. With the evaluation of the differences between both workflows in section 4, new insights can be identified for the further development of the modeling method. For example, a knowledge-based software can be helpful for the selection of modeling steps within the stages. Special situations require further investigation, for example, in the case of a large amount of data with a low level of usability. Such investigation shows that the derivable SDK, not only the available data, determines the modeling activities. In many design situations, certain steps in the workflows can be skipped, for example, when an EFRT model from the previous product is available. Another example is the adaptation of the concept in *Deductive modeling*, which is not necessary when comparing the robustness of two existing product concepts. These special situations also reveal the need for flexible adjustment of modeling activities between the two workflows in response to changing situations during modeling.

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

The modeling method based on the EFRT model has the potential to link other methods in different stages. For example, in Stage 2 Sketch, ideation methods can be integrated into *Explorative Modeling*, for example, TRIZ (Altshuller, 1998), the 6-3-5 method (Petersson and Lundberg, 2018), and brainstorming (Hatcher et al., 2018). Analysis methods can be integrated into *Deductive modeling*, for example, SPALTEN (Albers et al., 2016). This potential to link the other methods can be further investigated in the proposed stages of the modeling method.

A limitation of this research is that the development is based on illustrative examples, which leaves open the question of applicability in real industrial cases. It also remains unclear, whether the distinction between low and high levels of derivable SDK serves its purpose in practice, where modeling steps for the EFRT model may be influenced by additional factors. Additionally, the proposed method is limited to the mechanical parts of technical systems, which remain key elements for successful and reliable products.

The method proposed in this paper is developed for design engineers for their design tasks in new development projects. Users of the method are expected to have basic mechanical engineering skills, such as understanding engineering drawings and tolerances. Since the theoretical considerations of these design situations are literature-based, it is recommended that the modeling steps be further investigated in industrial studies.

The authors' evaluation of the modeling method introduces a risk of bias. To address this, further evaluation with subject studies is needed. The initial evaluation of the modeling method in this paper focuses on the transferability to other systems and the need to adapt workflows due to the identified differences. This is necessary to gain preliminary insights into the application of the method. Although the EFRT model and the developed workflows have not been evaluated for success factors such as applicability or efficacy for other users, the formalization in this paper lays the foundation for future investigations. Future work will involve a comprehensive evaluation of the modeling method through empirical studies, including comparisons with other approaches utilized by various users.

6. Conclusion

Different design situations yield higher or lower levels of the derivable SDK at the start of development. Recognizing the lack of consideration given to different design situations in the current RD methods, this paper proposes a novel modeling method based on the EFRT model, which takes the differences in SDK into account. By comparing the derivable and required SDK for early robustness evaluation, different modeling activities can be derived and integrated into a consistent stage-gate process, facilitating the individual adaptation of modeling activities. With the modeling method proposed in this paper, design engineers are systematically supported in the relationship between deviations and functional fulfillment of the product concept to evaluate the robustness in early design stages.

In this paper, two workflows, *Explorative modeling* and *Deductive modeling*, were developed to address low and high levels of derivable SDK, respectively, and were demonstrated using the example of a coining machine. The two workflows were applied to address different cases: one focused on creating a new, robust product concept, while the other aimed to enhance the robustness of existing products by adapting the design. Evaluating the developed modeling method with

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

two further case studies confirms the differences between the workflows and highlighted areas for further development of the modeling method.

Our findings highlight a modeling method that is adaptable to different design situations by allowing adjustments at each individual stage. Currently, the adaptability of our method is being developed and evaluated in two contrasting design situations, each with low and high levels of derivable SDK. This is the first step and the basis for further research on how to cover an even broader range of derivable SDK. It is also a step towards maturing the modeling method in terms of its applicability. It is now possible to build the EFRT model with two different levels of derivable SDK to support robustness evaluation. This can facilitate its use, for example, in bilateral research projects with industrial partners.

Acknowledgements

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) within the project “Holistic robustness evaluation in early design stages” under grant number 467789897. The support is gratefully acknowledged.

Competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

Akao, Y. 2004 *Quality Function Deployment: Integrating Customer Requirements Into Product Design*. Taylor & Francis.

Albers, A. & **Rapp, S.** 2022 Model of SGE: system generation engineering as basis for structured planning and management of development. In Krause, D. & Heyden, E. (Eds.), *Design Methodology for Future Products*. pp. 27–46. Springer International Publishing.

Albers, A., **Reiß, N.**, **Bursac, N.** & **Breitschuh, J.** 2016 15 years of SPALTEN problem solving methodology in product development. In *12th Biennial Norddesign 2016 Conference “Highlighting the Nordic Approach,” NordDesign 2016*, pp. 411–420.

Altshuller, G. 1998 *40 Principles, TRIZ Keys to Technical Innovation: TRIZ Keys to Technical Innovation*, 1st ed., p. 135. Technical Innovation Center.

Andersson, P. 1997 On robust design in the conceptual design phase: a qualitative approach. *Journal of Engineering Design* **8**, 75–89; doi:10.1080/09544829708907953.

Andreasen, M.M., **Hansen, C.T.** & **Cash, P.** 2015 *Conceptual Design: Interpretations, Mindset and Models*. Springer International Publishing.

Arvidsson, M. & **Gremyr, I.** 2008 Principles of robust design methodology. *Quality and Reliability Engineering International* **24**, 23–35. doi:10.1002/qre.864.

Ballu, A., **Mathieu, L.** & **Legoff, O.** 2010 Representation of mechanical assemblies and specifications by graphs. In Villeneuve, F. & Mathieu, L. (Eds.), *Geometric Tolerancing of Products*, vol. **21**, pp. 87–110. ISTE; John Wiley & Sons.

Cash, P. & **Kreye, M.** 2017 Uncertainty Driven Action (UDA) model: a foundation for unifying perspectives on design activity. *Decision Sciences* **3**. doi:10.1017/dsj.2017.28.

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

Chase, K.W., **Gao, J.**, **Magleby, S.P.** et al. 1996 Including geometric feature variations in tolerance analysis of mechanical assemblies. *IIE Transactions* **28**, 795–807. doi: 10.1080/15458830.1996.11770732.

Dzbor, M. & **Zdrahal, Z.** 2001 Towards a framework for acquisition of design knowledge. In *Volume 2A: 27th Design Automation Conference. ASME 2001 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. 389–398. American Society of Mechanical Engineers. doi:10.1115/detc2001/dac-21049.

Ebro, M., **Howard, T.** & **Rasmussen, J.** 2012 The foundation for robust design: enabling robustness through kinematic design and design clarity. In *Proceedings of International Design Conference.* DESIGN.

Ebro, M. & **Howard, T.J.** 2016 Robust design principles for reducing variation in functional performance. *Journal of Engineering Design* **27**, 75–92. doi:10.1080/09544828.2015.1103844.

Eifler, T. & **Schleich, B.** 2021 A robust design research landscape – review on the importance of design research for achieving product robustness. *Proceedings of the Design Society* **1**, 211–220. doi:10.1017/pds.2021.22.

Eisenbart, B., **Gericke, K.** & **Blessing, L.** 2011 A framework for comparing design modelling approaches across disciplines. In *DS 68-2: Proceedings of the 18th International Conference on Engineering Design (ICED 11), Impacting Society through Engineering Design, Vol. 2: Design Theory and Research Methodology, Lyngby/Copenhagen, Denmark*, pp. 344–355.

El Hani, M.A., **Rivest, L.** & **Maranzana, R.** 2012 Product data reuse in product development: a practitioner's perspective. In *Product Lifecycle Management. Towards Knowledge-Rich Enterprises*, pp. 243–256. Springer.

Eppinger, S.D. & **Browning, T.R.** 2012 *Design Structure Matrix Methods and Applications*, 334. The MIT Press.

Gero, J.S. & **Kannengiesser, U.** 2014 The function-behaviour-structure ontology of design. In Chakrabarti, A. & Blessing, L.T.M. (Eds.), *An Anthology of Theories and Models of Design: Philosophy, Approaches and Empirical Explorations*, pp. 263–283. Springer London.

Goetz, S., **Kirchner, P.**, **Schleich, B.** et al. 2021 Integrated approach enabling robust and tolerance design in product concept development. *Decision Sciences* **7**, e14. doi:10.1017/dsj.2021.13.

Goetz, S., **Lechner, T.** & **Schleich, B.** 2022 Computer-aided tolerance specification of preliminary designs based on variation analysis. *Procedia CIRP* **114**, pp. 203–208. doi: 10.1016/j.procir.2022.10.028.

Goetz, S., **Schleich, B.** & **Wartzack, S.**, 2018 A new approach to first tolerance evaluations in the conceptual design stage based on tolerance graphs. *Procedia CIRP* **75**, 167–172. doi: 10.1016/j.procir.2018.04.030.

Goetz, S., **Schleich, B.** & **Wartzack, S.** 2020 Integration of Robust and Tolerance Design in early stages of the product development process. *Research in Engineering Design* **31**, 157–173. doi:10.1007/s00163-019-00328-2.

Göhler, S.M. & **Howard, T.J.** 2015 The contradiction index (CI): a new metric combining system complexity and robustness for early design stages. In *27th International Conference on Design Theory and Methodology*. doi:10.1115/DETC2015-47255.

Grauberger, P., **Eisenmann, M.**, **Windisch, E.** et al. 2022. Investigating the generation of specific design knowledge: experimental validation of a qualitative modelling method. *Journal of Engineering Design* **33**, 870–895. doi:10.1080/09544828.2022.2151788.

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

Grauberger, P., **Goetz, S.**, **Schleich, B.**, **Gwosch, T.**, **Matthiesen, S.** & **Wartzack, S.** 2020 A conceptual model combination for the unification of design and tolerancing in robust design. *Proceedings of the Design Society: DESIGN Conference* **1**, 157–166. doi:10.1017/dsd.2020.12.

Grauberger, P., **Wessels, H.**, **Gladysz, B.**, **Bursac, N.**, **Matthiesen, S.** & **Albers, A.** 2019. The contact and channel approach – 20 years of application experience in product engineering. *Journal of Engineering Design* **81**, 1–25. doi:10.1080/09544828.2019.1699035.

Haik, Y. & **Shahin, T.M.M.** 2011 Engineering Design Process, 2nd ed., p. 287. Cengage Learning.

Hammers, C. & **Schmitt, R.** 2009 Governing the process chain of product development with an enhanced Quality Gate approach. *CIRP Journal of Manufacturing Science and Technology* **1**, 206–211. doi:10.1016/j.cirpj.2008.09.005.

Hasenkamp, T., **Arvidsson, M.** & **Gremyr, I.** 2009 A review of practices for robust design methodology. *Journal of Engineering Design* **20**, 645–657. doi:10.1080/09544820802275557.

Hatcher, G., **Ion, W.**, **Maclachlan, R.**, **Marlow, M.**, **Simpson, B.**, **Wilson, N.** & **Wodehouse, A.** 2018 Using linkography to compare creative methods for group ideation. *Design Studies* **58**, 127–152. doi:10.1016/j.destud.2018.05.002.

Hatchuel, A. & **Weil, B.** 2009 C-K design theory: an advanced formulation. *Research in Engineering Design* **19**, 181–192. doi:10.1007/s00163-008-0043-4.

Henderson, R.M. & **Clark, K.B.** 1990 Architectural innovation: the reconfiguration of existing product technologies and the failure of established firms. *Administrative Science Quarterly* **35**, 9–30. doi:10.2307/2393549.

Horber, D., **Brand, O.**, **Li, J.**, **Goetz, S.**, **Matthiesen, S.** & **Wartzack, S.** 2023 Digitale Durchgängigkeit im Toleranzmanagement am Beispiel der ganzheitlichen Robustheitsbewertung in frühen Phasen. In *Tagungsband der 5. Summer School Toleranzmanagement 2023*, pp. 69–76. FAU University Press. doi:10.25593/978-3-96147-665-7.

Horber, D., **Li, J.**, **Grauberger, P.**, **Schleich, B.**, **Matthiesen, S.** & **Wartzack, S.**, 2022. A model-based approach for early robustness evaluation – combination of contact and channel approach with tolerance graphs in SysML. In *Proceedings of the 33rd Symposium Design for X. DFX 2022*, pp. 1–10. The Design Society. doi:10.35199/dfx2022.18.

Hubka, V. & **Eder, W.E.** 1990 Design knowledge: theory in support of practice. *Journal of Engineering Design* **1**, 97–108. doi:10.1080/09544829008901646.

Johansson, P., **Chakhunashvili, A.**, **Barone, S.** & **Bergman, B.** 2006 Variation mode and effect analysis: a practical tool for quality improvement. *Quality and Reliability Engineering International* **22**, 865–876. doi:10.1002/qre.773.

Jugulum, R. & **Frey, D.D.** 2007 Toward a taxonomy of concept designs for improved robustness. *Journal of Engineering Design* **18**, 139–156. doi:10.1080/09544820600731496.

Li, J., **Horber, D.**, **Keller, C.**, **Grauberger, P.**, **Goetz, S.**, **Wartzack, S.** & **Matthiesen, S.** 2023 Utilizing the embodiment function relation and tolerance model for robust concept design. *Proceedings of the Design Society* **3**, 3771–3780. doi:10.1017/pds.2023.378.

Mathias, J., **Kloberdanz, H.**, **Eifler, T.**, **Engelhardt, R.**, **Wiebel, M.**, **Birkhofer, H.** & **Bohn, A.** 2011 Selection of physical effects based on disturbances and robustness ratios in the early phases of robust design. In *Proceedings of the 18th International Conference on Engineering Design. ICED'11, Lyngby/Copenhagen, Denmark*, pp. 324–335. The Design Society.

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press

Matthiesen, S. 2021 Gestaltung – prozess und methoden. In Bender, B. & Gericke, K. (Eds.), *Pahl/Beitz Konstruktionslehre: Methoden und Anwendung erfolgreicher Produktentwicklung*, 9th ed., pp. 397–465. Springer.

Matthiesen, S., **Grauberger, P.** & **Schrempp, L.** 2019 Extended sequence modelling in design engineering – gaining and documenting knowledge about embodiment function relations with the C&C^2-approach. *Proceedings of the Design Society: International Conference on Engineering Design* **1**(1), 1483–1492. doi:10.1017/dsi.2019.154.

Matthiesen, S. and Ruckpaul, A., 2012. New Insights on the contact&channel-approach - modelling of systems with several logical states. In *International Design Conference: DESIGN 2012*, pp. 1019–1028. The Design Society.

Niu, X., **Wang, M.** & **Qin, S.** 2022 Product design lifecycle information model (PDLIM). *International Journal of Advanced Manufacturing Technology* **118**, 2311–2337. doi: 10.1007/s00170-021-07945-z.

Pahl, G., **Beitz, W.**, **Feldhusen, J.** & **Grote, K.-H.** 2007 *Engineering design: a systematic approach*, 3rd ed., p. 617. Springer.

Petersson, A.M. & **Lundberg, J.** 2018 Developing an ideation method to be used in cross-functional inter-organizational teams by means of action design research. *Research in Engineering Design* **29**, 433–457. doi:10.1007/s00163-018-0283-x.

Phadke, M.S. 1989 Quality engineering using design of experiments. In Dehnad, K. (Ed.), *Quality Control, Robust Design, and the Taguchi Method*, pp. 31–50. Springer US.

Ponn, J. & **Lindemann, U.** 2005 Characterization of design situations and processes and a process module set for product development. In *DS 35: Proceedings ICED 05, the 15th International Conference on Engineering Design*, pp 1–15. The Design Society.

Suh, N.P. 1998 Axiomatic design theory for systems. *Research in Engineering Design* **10**, 189–209. doi:10.1007/s001639870001.

Taguchi, G., **Chowdhury, S.**, **Wu, Y.**, **Taguchi, S.** & **Yano, H.** 2005 *Taguchi's Quality Engineering Handbook*, p. **1662**. John Wiley & Sons.

Teng, S.-H. & **Ho, S.-Y.** 1996 Failure mode and effects analysis. *International Journal of Quality & Reliability Management* **13**, 8–26. doi:10.1108/02656719610118151.

Thornton, A. 2004 *Variation Risk Management: Focusing Quality Improvements in Product Development and Production*, p.320. Wiley.

Ullman, D.G., 2010. The mechanical design process, 4th ed. McGraw-Hill Higher Education, New York, 433 pp.

Weber, C. 2014 Modelling products and product development based on characteristics and properties. In Chakrabarti, A. & Blessing, L.T.M. (Eds.), *An Anthology of Theories and Models of Design: Philosophy, Approaches and Empirical Explorations*, pp. 327–352. Springer London.

Wilmsen, M., **Dühr, K.** & **Albers, A.** 2019 A context-model for adapting design processes and methods. *Procedia CIRP* **84**, 428–433. doi:10.1016/j.procir.2019.04.243.

Zwicky, F., 1967. The morphological approach to discovery, invention, research and construction. In Zwicky, F. & Wilson, A.G. (Eds.), *New Methods of Thought and Procedure: Contributions to the Symposium on Methodologies*, pp. 273–297. Springer.

https://doi.org/10.1017/dsj.2024.48 Published online by Cambridge University Press