<http://www.diva-portal.org>

Postprint

This is the accepted version of a paper published in *Advanced Engineering Informatics*. This paper has been peer-reviewed but does not include the final publisher proof-corrections or journal pagination.

Citation for the original published paper (version of record):

Poorkiany, M., Johansson, J., Elgh, F. (2016)

Capturing, structuring and accessing design rationale in integrated product design and manufacturing processes.

Advanced Engineering Informatics, 30(3): 522-536

<http://dx.doi.org/10.1016/j.aei.2016.06.004>

Access to the published version may require subscription.

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:hj:diva-28196>

Corresponding author: **Morteza Poorkiany**, PhD student

Affiliation: Jönköping University, Sweden

Mobile phone: +46 (0) 76 58 38 086

Direct phone: +46 (0) 36 10 15 71

Fax: +46 (0) 36 12 53 31

Email: morteza.poorkiany@ju.se

morteza.poorkiany@gmail.com

Capturing, structuring, and accessing design rationale in integrated product design and manufacturing processes

Morteza Poorkiany^a, Joel Johansson^a, Fredrik Elgh^a

^a *Department of Product Development, Jönköping University, Sweden*

Abstract

Developing customized products is the business case for many manufacturing companies striving to fulfill the customers' specific needs. When manufacturing customized products it is often necessary to also develop corresponding customized manufacturing tooling. There is a need to support concurrent development of new product variants along with their manufacturing toolsets. The communication between design engineers and manufacturing engineers is hence a key issue that if solved would enable design engineers to foresee how changes in product design affect tooling design and vice versa.

To understand the correlation between the design of a product and its corresponding manufacturing tools, access to design rationale of the product and the developed tooling is required. Design rationale provides an explanation of why an artifact is designed in the way it is, including statements (textual, numerical or geometrical), argumentations, and decisions. Since design rationale is composed of information scattered all across the company's repositories in different formats (e.g. in type of a geometry, picture, table, and textual document), representing the design rationale is a challenge for many enterprises. In this paper a method is introduced that enables capture, structure, and access to design rationale across product design and tooling design. The system enables representing design rationale in formats such as CAD models, spreadsheets, textual formats, and web pages. The method has been examined by developing a prototype system tested in a case company which develops and manufactures customized car accessories, such as roof racks and bike carriers, for different car models. The company develops and manufactures the products as well as the required tooling equipment. The prototype system includes different software commonly used by engineers during designing a product, for the purpose of making it applicable for other companies.

Key words. Design rationale, computer supported engineering, engineer-to-order, design for manufacture, and knowledge acquisition.

1. INTRODUCTION

Tense markets require manufacturing companies to frequently develop customized products to fulfil new customer specifications. In the early phases of the development, having an overview of the complete product lifecycle enables the best product quality and

makes it possible to optimize cost and time-to-market [1]. In concurrent engineering the product's requirements that need to be met during the product lifecycle are already considered in the project planning and design [2]. Concurrent engineering is "the proactive practice of designing products to be built on standard processes, or concurrently developing new processes while developing new products" [1]. It is important to support concurrent engineering of the product and the related processes including manufacture and production. Teams from engineering design and manufacturing work together to consider the interaction between design and manufacture.

When developing a customized product the product designers should keep in mind what resources are available in the production line and if these resources meet the constraints and properties of the intended manufacturing process. They always have to consider manufacturability of the design [1] and be aware of the required manufacturing toolsets. Sometimes if a product variant is to be redesigned in order to fulfill new specifications, it is necessary to develop new tooling or modify the current ones. The engineers need to realize when and where changes in product design affect the tooling design and vice versa. Thus, a close collaboration between product design and tooling design teams is required.

Most of activities performed when designing a customized product rely on previous design projects and obtained experiences. The importance of reusing design information in order to improve future designs is discussed by Kim et al. [3]. The resulting documentation from product development and tooling design usually describes the results of the design and tooling as they are, answering questions regarding "what" the product is and "how" it is to be manufactured. But if a product is to be redesigned, more detailed information is required regarding the purpose of the design, the reasons behind the design decisions, the evaluated trade-offs, and the design alternatives. This type of information, which is called design rationale, often aims to explain the product in the way it is designed answering "why" questions [4]. A survey was performed in [5] to investigate the value of design rationale and how it is used in decision making processes. Approximately 80% of the respondents said they fail to understand the reason of a design decision without design rationale support. So, capturing design rationale is important, however, Tang et al. [6] claim even when design rationale is captured, it is not well enough organized to retrieve and track it easily.

Making the previously captured design rationale accessible, facilitates reuse and supports the designers to solve similar design problems. According to Dutoit et al. [7], approaches for managing design rationale should mainly address how to implement three basic processes as follows: 1) how to capture design rationale; the process of extracting design rationale from designers, 2) how to formalize design rationale; the process of transforming design rationale into desired representation form, and 3) how to provide access to design rationale; the process of making design rationale available for users. It is further mentioned the possibility to combine the capture and formalizing processes in a single operation. This is also stated in [8] which defines design rationale capture as

a process of recording knowledge as well as organizing the knowledge based on a representation schema, and storing it in a knowledge base. Although both literatures state the need to formalize and organize the rationale based on representation forms, what is not explicitly mentioned is the importance of structuring the information. Structuring design rationales supports the process of design reasoning by “helping designers track the issues and alternatives being explored and their evaluations” [9]. In fact there is a trade-off between the ease of retrieval and structure in design rationale representations. What will be achieved from well-structured information is easier retrieval.

Since design rationale varies in type and format, representing the design rationale can be a challenge for the system developers. Depending on the aim of the system, design rationale can be represented in different ways. For example, it could be an explanation about a design alternative, an argumentation about a decision, or the history of a product design [4, 7, 10]. As seen, any type of design information (e.g. product specification, geometries, design tables, and rules) can potentially be a part of a design rationale. Since the design information is stored in different software applications, the interaction between the design rationale system and the software applications at the company should always be considered.

One important factor to ease information access is the need to develop a natural user interface in order to facilitate intuitive human-computer interaction [11]. User acceptance is of high importance and strongly related to the access of information. The authors’ experiences show that designers are typically reluctant to accept and add new applications for archiving and annotation of design rationale to their current portfolio of software. Therefore, in this research an integrated design environment for representing design rationale is developed. The integrated environment is constituted of traditional software applications that are currently used by the designers and a computer-supported engineering system which administrates and manages information flow between the software.

During the design process the majority of all products on the market are modeled in a CAD system. CAD models provide specific information to understand the layout of the assembly and detailed information for each part. However, this type of information fails to offer high-level semantics [12]. Therefore, the spreadsheets for creating design tables and manipulating data and documents in textual format are broadly used during the design process in order to record and explicitly represent the generated and utilized information. Considering these types of representations together with the fact that a big part of the design information at the company is accessible via internal websites, the target of this study was to enable capture and representation of design rationale in different formats such as CAD models, spreadsheets, textual format, and web pages.

Design is an important process in product development and making proper decisions at this early stage improves the development process and support the organizations to deliver their business initiatives and strategic goals. The design decisions could affect different aspects of the product’s lifecycle, therefore, a design rationale system should enable the users to keep track of design rationales and

the relevant information in different disciplines in order to give insight into the reasons of why design decisions are made. For instance, it should preferably be possible to link the design rationales and the related underlying information together within the product design domain, or even extend the functionality to cover the design rationales and the relations across the product design and the tooling design. This has been addressed in the research presented in this article and a method was developed and implemented in a case company where the development of customized products as well as the development of required tooling and equipment is carried out concurrently within the enterprise. The selected company delivers accessories to cars, both as a sub-supplier and by retailing in the consumer market. A prototype system has been developed for the purpose to investigate the applicability of the proposed method.

This paper is organized as follows. Section 2 gives an introduction to fundamental concepts of design knowledge, design rationale and basic methods and approaches for capturing, structuring, and representing the design rationale as well as related research. In section 3, the proposed method and the constructs for capture, structure, and access to design rationale are described. Also, an information model that outlines the backbone in a computer-based solution is presented in section 3. The case study is described in section 4. In order to examine the applicability of the proposed method a prototype system is developed and implemented in both product design and tooling design processes in the case company, which is described in chapter 4. A discussion over the proposed method and solutions for enabling capture, structure, and access is provided in section 5. Finally, the conclusion and future works are discussed in section 6.

2. FRAME OF REFERENCE

This section introduces the basic concepts of design knowledge and design rationale. Fundamental methods and approaches for capturing and representing design rationale are also described. Next, related work to this study are presented and finally, the knowledge gap and motivation for the research are described.

2.1 Design knowledge

There is a vast number of information and knowledge sources that are utilized throughout the process of designing an artefact [13]. These sources contain product knowledge, process knowledge, technology information, formal and informal information regarding activities, methodologies, discussions and meetings, as well as catalogues, CAD files, assemblies, features, rules, and bill of material. Johansson [14, 15] classifies design knowledge into four types when it comes to the metal forming industry which could also be extended to other industries: 1) heuristic which is generally found in different handbooks or company standards and is

based on skilled engineer's experiences, 2) analytical that derives from fundamental physical laws tends to be more complex than heuristic, 3) numerical where most often the common method is finite element method (FEM), and 4) empirical which is based on experience. The engineer collects and stores this knowledge in different repositories and formats. However, providing an appropriate knowledge representation that enables access to this vast amount of information which differs in type is a big challenge. Owen and Horvath [16] classify representation formats into five categories: pictorial (e.g. sketches, drawings, and pictures), symbolic (e.g. decision tables, and production rules), linguistic (e.g. verbal and textual communications), virtual (e.g. CAD models and simulations), and algorithmic (e.g. computer algorithms, and mathematical equations). Some knowledge representations can fit under more than one category. In the early stages of design the presentations are more linguistic and pictorial in nature, while by evolving the design process, the presentations are predominantly virtual and symbolic [11].

2.2 Design rationale; motivations and challenges

Elgh [17] categorizes design knowledge into two groups: One is design definition describing the results of the design, without any explanation concerning the reasons and argumentations behind the design. Such information can more often answer the "what" question. A CAD model or design table are some examples of design definition which are mostly based upon insights, experience, trade-offs, calculation, simulations, etc. The other type of knowledge is design rationale explaining the purpose and reasons behind the design in more details. Design rationale provides a better understanding for design definition and often aims at explaining the artifact in the way it is designed answering the "why" question. For instance, "why a CAD model looks like as it is?"

The research community has defined design rationale as a way to know the reasons behind a decision [12, 18], but it could also be the justification for it, the design alternatives, and the evaluated trade-offs that led to the decision [9]. In fact anything in a design process that can be represented and that trace a reason can and should be a part of design rationale, for instance in its extreme even the conversations between the design members during lunch or meetings [9] should be acquired.

Lee [9] points out the potential benefits of using design rationale and implementing design rationale methods. The major benefits include: 1) *Support reuse* for a better design. Reuse of design rationale can support the development of a new artefact or modification of existing variants (design changes) as it gives insight into the reasons behind the current design. In addition, Lee states, reuse can be supported in two ways: one is to serve as index to similar designs and problems faced. The other one is to reuse rationale themselves from past activities to suggest potential design alternatives. 2) *Support documentation*. Design rationale systems support knowledge transfer by learning from the past and explicitly linking the requirements to the final solutions. In addition, design rationale includes explanations of what to do and what not to do, therefore, it can be used as an external knowledge source to support training of new

users and keep them up to date. 3) *Support collaboration*. Implementing a design rationale system is a proper way for designers to share their thoughts to other design members. Still, collaboration is not just limited to being among designers, it involves people throughout the complete life cycle of the product (e.g. sales persons and production engineers).

Successfully capturing design rationale requires the identification of the types and details of rationales as well as the means and objective for capturing it [7]. Design rationale can be captured during the design phase or when the design process is finished, directly by the designer or by an agent. Regli et al. [19] discuss two major methods for design rationale capture: 1) *User-intervention-based capture* which is the documentation method created by the designers and the intention is to record the history of design activities as the design process evolves. This type of documentation helps people outside the project to understand the process and activities. 2) *Automatic rationale capture* which records the communication among the involved people to extract design rationale and decisions as they proceed in design project. A drawback for the latter approach would be that some people might not feel confident to record their communications including their e-mails and telephone calls. Dutoit et al. [7] mention capturing design rationale can be an intrusiveness activity for the designers and a reason to resist performing it. Usually, design rationale systems provide some methods for capturing design rationales, for instance, filling out a template. This requires some extra works to be done by the designers which could be a draw back for design rationale systems [20]. So, design rationale capture might actually be detrimental to design process and disrupt designers' thinking. Additionally, design is an intense activity and it should be considered that the focus of a designer shall be more on doing creative tasks not routine and monotonous jobs such as archiving information and documentation.

Lee [9] proposed a generic structure to what to be represented by a design rationale system. The structure takes the form of three layers: decision layer containing design alternatives, arguments and issues; design artifact layer containing both decision making steps and related information; and design intent layer representing information underlying design decisions such as objectives and intentions.

Some organizations try to keep and manage the generated design rationales during the design process (e.g. incentives, plans, and procedures) and store them in different sources (e.g. handbooks, and hard drives). However, still the explanation of the reasoning and the way of creating the artefact might be missing in knowledge documentations [21]. Explicitly representing design rationale is a big challenge, because design rationale is usually captured partially and informally, often as natural language in design documents [7]. Therefore, defining the type of the design rationale and how it should be retrieved from these repositories and in what level is a fundamental issue and depends upon the aim and scope of the design rationale systems [19].

Methods for representing design rationale can be divided in two groups: descriptive and prescriptive. The descriptive approach tends to capture the history of the design activities, methods, and strategies, more specifically, what and why decisions are made and by whom [19]. The prescriptive approach on the other hand, aims to improve the reasoning of designers and attempt to make the design process more correct and consistent [7]. However, a design rationale approach might be prescriptive in intent but also has some descriptive objectives. Additionally, Saridakis and Dentsoras [22] discuss computer-based models which enhance features from the descriptive and the prescriptive models. These computer-based models provide a better understanding of design problems by addressing decision-making processes or analyzing the design knowledge.

2.3 Related work

A big challenge when developing a design rationale system is the interaction between design rationale and knowledge repositories. A research project demonstrating the feasibility of applying architectural support to manage interaction with heterogeneous information sources was performed in [23]. The focus of the work was on interaction between the base layer, where the original information resides, and the superimposed layer, where new information or new interpretations are laid over existing information. Because the base-information types might differ, it is not possible to use a common interface for performing navigation and querying with the base layers. Therefore, there is a need for an appropriate architectural application placed between the two layers, allowing the user to navigate and inquire within base layers regardless of their types. The developed middleware framework which is called SPARCE provides plugins that add toolbar or menu to Microsoft Excel, Word and PowerPoint, Adobe Acrobat, Internet Explorer and Media Player. More specifically, SPARCE applications can provide links to a selection in base documents, for example, in spreadsheets or word documents [24]. Although SPARCE can ease communication between the layers, the created hyperlinks to information sources are unidirectional which means they lead out, but not back in [25].

IBIS [26] and DRL [4] are two approaches for representing the reasons and arguments behind the design decisions. They use node elements to represent the design concepts and links to represent the relationships between the nodes. An example of such models is presented in [27]. The model is based on IBIS's schema and provides a semantic representation for design rationale. The study addresses the process of solution evolution when developing an artifact. The process starts by identifying design issues. Issues are specific to particular situations and describe the objectives and motivational reasons for designing an artifact. A solution is to be developed to address the identified issue. The solution can be supported by providing an argument. Argument reflects a designer's view point on the chosen solution and is a statement to support or object to the solution. The provided solution, however, might cause to creating a

new issue which that issue would be addressed by another solution. Again an argument is required to support the second solution. The process evolves until the design requirements are met. The final solution results in at least one artifact.

DRed (Design Rationale Editor) is a tool for capturing design rationale [18] based on the IBIS model. DRed uses the nodes and links concept and allows engineers to record their rationales as the design proceeds and the designers gain understanding of and solve the design problems. An important consideration in designing of DRed is that there is no hidden information and everything is displayed directly on the charts rather than pop-up dialogs or windows. Bracewell et al. [25] developed an extension to DRed in order to enhance the applicability of DRed allowing a more efficient integration environment. In addition, a software prototype is developed and integrated into DRed by Kim et al. [3] which enables searching the required information according to the current task of the designer. The proposed solution is an extension to DRed, employing a bidirectional hyperlink approach (tunnel linking) to allow graphs to be distributed over documents and subsequently navigated freely in any direction from any starting point. A bottleneck for DRed would be that by increasing the number of nodes the graphs easily become more complex.

Developing software tools for annotation has been researched as a means to improve design collaboration. One example is the system developed by Hisarcikililar and Boujut [28]. Further, Sandberg et al. [29] propose an approach that provides knowledge retention and sharing across product development process in CAD environment. The method enables adding design rationale within 3D annotations carried out by designers allowing design and documentation at the same time. The 3D annotations may include general text notes or hyperlinks to other sources of information such as textual documents, figures, spreadsheets and URLs. These systems allow the designers to express specific information with a particular intention. In addition, the systems provide an integrated representation of knowledge, awareness of knowledge sources, access to the knowledge and easy communication among system users. However, in both mentioned systems, the applicability of the systems is limited to CAD environment. A solution to improve the situation can be enabling implementation of more software applications.

2.4 Knowledge gap

Although there has been considerable effort to develop design rationale systems, we are yet to see widespread use of representations to facilitate information exchange especially between knowledge bases and CAD software. One reason is that there are neither appropriate standards available nor widely accessible design knowledge repositories [11]. Therefore, the organizations usually use other tools and applications for capturing and recording the design information beside the CAD tools. Capturing is effective when it is integrated into the current practice of the actors involved. Lundin [30] states capture and representation of design information should derive from design implementations directly and be a natural part of the design process.

Ease of use and access to the information is a major of concern in design rationale approaches. A fundamental problem in access and retrieval of the information is the variety of applications and software to represent the information. A research was carried out in [31], focusing on documentation and management of product related knowledge. The purpose of the project was to reveal problems related to the reuse of design rules in a case company. Investigations in the studied company show that it is difficult for individuals to share their good developed solutions, due to no present system for such documentation covering all organizational tools (in that case office programs, CAD, and programming software). Moreover, they discuss the difficulty in communication among the design engineers and the design programmers and stress that people are, typically, reluctant to add a new application to the tools in hand in order to improve the situation.

Such circumstances and also the need for information exchange between design members, leads the organizations to move towards integration of independent tools for the sake of bettering information representation and contextual communication [30]. Most companies, preferably, look for seamless integration of their tools and applications in order to prevent duplication of effort and need of maintaining several systems. Having a design rationale system that is compatible with organizational platforms and allows interoperation with other systems, is easy for the users to understand and work with, and has low intrusion to the design process is an ideal situation for many companies.

It can be concluded from the above that there is a need to develop methods for supporting capturing, structuring, and accessing design rationale. The methods should enable easy access to design rationale in the engineering tools to foster their use. This can be achieved by providing an integrated environment constituted of traditional software in order to capture, structure, and access design rationale in the appropriate software.

3. CONSTRUCTS FOR DESIGN RATIONALE CAPTURE, STRUCTURE, AND ACCESS

Lee [9] states a design rationale system should have the potential to support documentation by linking the requirements to the final solutions, and support collaboration by enabling the designers to share their thoughts and decisions to other design members across the development process. Figure 1 shows how a design rationale system can be introduced to support the decision making process across the development process. At the beginning of a product development project there are not much information but some few documents related to the product requirements that are identified in project planning phase. Even though few, these documents are important and affect the downstream decisions. Based on the requirements, the engineering design phase including product design and tooling design is initiated. By starting the product design, the number of documents increase rapidly including test-reports, FMEA, and CAD-models. Soon the manufacturing engineers start to develop tooling design suggestions based on the different design proposals. The documents

are stored in a PLM-system or in project file folders and are treated as files. The concepts, ideas and decisions are however not stored in single files but are fragmented in many files and have different structures than the file structure or even than the product structure. Therefore, design rationale management requires methods and tools to capture, structure, and access information across organizations, processes, systems and products regardless of file and product structures.

As the development process evolves decisions are made in more details, some rationale will be elaborated, some will be modified completely when, for example, a decision made earlier is shown to be incorrect and needs to be altered [32]. The information generated in one phase can be used as input to perform design activities in other development phases. So, first, it is required to enable the engineers to record their decisions, intentions, and argumentations, and second, provide instant access of this information to other design team members. As shown in figure 1, the suggestion is to enable capture and access to the rationale for design members through providing a user interface. Ideally, the user interface is embedded in the current practice of engineers.

Figure 1. Integrating capturing and accessing of design rationale.

The information considered, generated, or analyzed during a decision making process is stored in knowledge repositories in form of test-reports, FMEA, and CAD-models. What is needed to be captured beside that, is information such as intention of activities, and argumentations for or against the considered alternatives. Such information is to be captured in the design rationale database. A function board links the information stored in the rationale database to the related information in the knowledge repositories and enables the engineers to capture and access to this information through the user interface.

In order to support retrieval of design rationale and reuse of design knowledge, a rationale representation should be able to address three questions [12] such as *know-what*: the solutions created, *know-how*: the courses of action, and *know-why*: the reasoning process. It is essential to besides capturing the identified issues, problems, and developed solutions during the design process, capture also how and why these solutions are created. Figure 2 shows a UML model for representing design rationale. The Design Rationale and the Decision classes are central in the model. Design Rationale objects carry general information (text and picture based descriptions) regarding the concept or idea together with a set of Decision objects. The Decision objects carry information regarding decisions taken during the design process such as date, who took the decision and, importantly, the argumentation behind the decision. The decision object also contains pointers to the information the decision and its argumentation was based on. Since that information is scattered these pointers have to be very specific, yet since that information is of many different types the pointers have to be very general. In the proposed model, pointers to documents that are not related to CAD-models are called Statements. Statement objects capture information typically found in test-reports, lists of requirements and FMEA documents. (Note that the pointers target sub-sets of the supporting documents, not the whole files.) Since decisions made during the product development process affect the geometry of the product to a great extent it is possible to make the Decision objects point to Assembly, Part, Feature, Parameter, and Entity objects in CAD-models. A decision may also affect the material selection of components of the product or the tooling and in such cases there are pointers to Material objects.

Design rationale occurs all the time over the product life cycle and should be captured at its origin using the proposed class diagram which can be achieved by providing an integrated environment. In such an environment the tools for capturing design rationale as well as representing it are integrated to software already used by the engineers. A great advantage of such an environment is that the designers can perform design tasks in different software and applications and concurrently capture design rationale. Design rationale can be captured and represented in formats that the designers are already familiar and would prefer to work with.

Besides using the software as representation tools, many companies would like to share their corporate design knowledge via the internal websites to make it accessible for geographically dispersed team members. The Design Rationale objects and the Decision

objects have functionality to produce xhtml-fragments to make flexible web-pages that are generated on demand based on the captured design rationale and underlying information.

The Discipline Model class is introduced to filter the design rationale between disciplines. Product Design and Tooling Design are both derived from the Discipline class in the model presented here, but more such classes could be introduced. A Tooling Design object is associated to at least one product design (which is the common case but one tool could be used to produce several components). A Product Design object is associated with all that is handed over from product designers to manufacturing engineers, including the rationale as shown in the class diagram, see figure 2. When design changes are to be done the related design rationale can be browsed in generated web-pages or accessed through the integrated design rationale environment to see what assemblies, parts, features, parameters, and geometrical entities it will affect in design itself but also on the tooling. The underlying decisions with all connected documents will also show up. Going in the other direction, i.e. when changing the tooling design, is also possible when using the proposed class diagram.

Figure 2. Class diagram supporting capturing, structuring and accessing of design rationale.

Figure 3 shows an example of a decision object and how it can be linked to feature and statement classes. A set of headings are defined in the decision class to provide a better and detailed understanding not only about the final solution and the decision made, but also about the process of making the decision such as date, design members, and arguments for and against the alternatives (both selected and rejected ones). The statements and feature can be linked to the decision object but in a more specific way they can be linked to the exact headings in the decision object.

Figure 3. The correlation between decision, statement and feature objects according to the UML-model.

4. CASE STUDY

To investigate the applicability of the proposed method and the class diagram a prototype system was developed in collaboration with a company. The selected company develops and manufactures customized products and accessories, such as roof racks and bike carriers, for different car models. The development process of a car's roof rack was selected for more investigation. The company acts on the open market competing with car manufacturers and therefore gets no nominal data of car roofs. Instead, the engineers have to collect geometrical information about car roofs by measuring the actual car's roof and develop the roof rack based on that geometrical information. The trend of selling new roof racks is highly related to the time a new car enters the market.

A clamping roof rack is a rack used on cars without rails on the roof (i.e. the rack is standing on the roof and clamped around it where the doors are). Figure 4 shows how a roof rack is attached to a car. The roof rack is developed in a way which enables the engineers to instead of developing a complete new roof rack for a new car, just design the two components called brackets and footpads in figure 4, and accordingly, adapt the roof rack to the new car.

Figure 4. A virtual model of roof rack.

4.1 Enabling capture, structure, and access to design rationale

The development process of the footpad was selected as the case study. Collaboration and communication between the product designers and tooling designers is significant in the company in order to allow them to realize the downstream effects of their design activities. Access to design rationale explaining the reasons, arguments, and effects of decisions made for both footpad and tooling could support this collaboration and provide a better understanding of the design.

The approach is based on the integration of SolidWorks, Word, Excel, and wiki pages. The reason of choosing different software was to represent the design rationale in different formats. SolidWorks was chosen as representative for 3D modeling, Microsoft Excel for rules definition and drawing tables, and Microsoft Word for specifications and textual usage were selected. Besides the software, wiki pages are chosen to record the decisions. Wiki is a type of content management tool and enables linking, navigation and searching the information. It allows the users to edit and form both the information and the structure to fit their purposes in an organic way. Free available sources and user friendly are the other benefits of using wikis. 18 wiki pages are created explaining a number of decisions.

An information model, displayed in figure 5, was developed in order to form the principles of the introduced method. The information model consists of two types of classes; general and specific. The general classes comprise Design Rationale, Decision, and

Statement classes. The Statement class includes any of the software applications. The specific classes target the software applications, in this case Word, Excel and SolidWorks.

When using the information model, the statement class carry information about where the actual design content is stored. These statements can be viewed as hyperlinks, which can point to a specific file on the hard drive or a web-page at a certain URL, but the statements are more specific than that, pointing to, for instance, a specific feature or a dimension in 3D-modeling software, a specific bookmark in a word processing document, or a certain range of cells in a spreadsheet. It is possible to extend the information model to target other software applications by implementing new types of specific classes, indicated with dashed lines in Figure 5. Additionally, the decision class is used to cluster the statement objects that are related together. For example, some ranges of spread sheets, bookmarks in different textual documents, and some features in some geometrical models that have something to do with each other in a natural way can form a decision class. Such a group can be viewed and utilized as a multilateral hyperlink that makes it possible to go back and forth from one statement object to several others. So, for instance, when selecting a feature in a CAD-model that is targeted in a statement class, all other statement objects in that decision can be monitored to the user making it possible to jump to the connected workbook or spreadsheet entities, or other 3D-modeling entities.

To make the decision classes meaningful, they are stored in Design Rationale objects that can contain a number of decisions that are somehow naturally connected to each other.

It should be considered that the information model for the developed prototype is slightly different from the UML model presented in figure2. Here, the geometrical model class is included in the statement class.

Figure 5. The backbone information model in the prototype system.

By integrating the capture process with other design activities, the designers are able to record the design rationale wherever the need is raised. Access to the information can be strengthened by allowing the user browse and navigate across documents containing design rationale. The suggestion is to provide a common user interface in each software that presents all the rationales, statements, and decisions (see figure 6). Hence, it is proposed to develop add-ins to the targeted software applications in a standardized way so that the users feel comfortable and recognize the system and the functions it stands for.

In the standardized add-in user interface there should not only be functionality for monitoring available design rationales but also to connect the related statements. The user interface can facilitate the user to create a new design rationale, connect the statements together, and associate a wiki to the design rationale. Since the user interface is common for all the software and controlled by the function board, whenever a new design rationale is generated or a new link between the statements is created, all the activities are presented and the content of the windows is updated simultaneously in all the software.

Figure 6. Enabling capture and access to design rationale via a common user interface.

The prototype system was developed in Microsoft .net environment. It can be installed in form of extension to Word, Excel, and SolidWorks, however, it is possible to target more software if beneficial. The system connects SolidWorks, Excel and Word together.

er. In order to embed the wiki into the system, the system allows the user to attach the URL address of a specific wiki page to a design rationale.

The graphical user interface of the prototype system consists of three windows (task panes) added to each targeted software application, see figure 6 (screenshots of SolidWorks and Excel are presented in figures 9 and 10). Two windows, called *Design Rationales* and *Statements*, are added to the right side of each software. The third window is named *Decision* which is shown at the bottom of the software. The properties of each window can be summarized as below:

- The *Design Rationale* window edits and monitors all the design rationales regardless of which software they were first created in. This window shows the created design rationale and is constituted of two tabs named *Active Selection* and *Browse*. The *Active Selection* tab enables add, edit, or remove a design rationale. It is also possible to attach an extra file, for instance, a wiki page to a design rationale. All these functions can be accessed by right clicking in the windows and selecting the desired function. The *Browse* tab monitors all the available design rationales.
- The *Statements* window allows to link the related statements together by forming groups. This window provides functionality to group the related statements together across different software.
- The *Decision* window displays the decision (wiki page) attached to a design rationale.

4.2 Use case example

In order to express the functionality of the prototype system an object diagram is created by showing two examples of decisions made in product and tooling design, see figure 7. Decision 1 is about changing the dimension of the footpad after testing the created prototype. This decision includes one statement in FMEA and one statement in CAD model. Decision 2 concerns applying changes in tooling design after changes occur in footpad design. This model is explained in the following sub-sections. Due to the company's policy, the information presented in the following picture is fictive.

Figure 7. An object diagram illustrating two examples of decisions in product design and tooling design.

Product Design:

An example is given to show how product designers are to work with the suggested system. This example targets the design process of the footpad. According to the requirements, two design alternatives are considered which after some evaluations a decision is made on choosing one of the alternatives. A prototype of the selected alternative is created. The prototype is tested which indicates the need for performing some changes in the footpad design.

Figure 8 shows the created wiki page explaining the decision. It should be considered that this wiki is a sub-page of another page that provides explanation in a higher level of granularity about the product. A template including some headings is provided to make it easy to record, understand and modify the content. The headings are: article name/number, design date, design team members, intention and goals, assumptions, alternatives, and evaluations. As seen in figure 8, the page is not only to address the decision that lead to the final solution but also explains how the solution is created and analyzed, if there were other alternatives considered but rejected, and evaluations of the solution. The evaluation section explains that the final design of the footpad is tested by making a prototype. However, it has been difficult to mount the prototype on the car. This can happen due to the level of accuracy in manufacturing equipment. After a discussion between the product design and tooling design, the designers agree upon changing the tolerance to 1.

The change of the rectangles' dimension should be applied in footpad CAD model. Figure 9 shows the footpad designed in SolidWorks. First the corresponding feature for the big rectangles in the tree view in SolidWorks is selected. Next, a design rationale object called *design of big rectangles* is created by right clicking in the *active selection window*. By creating a design rationale object, the *statements* window is updated showing the path for the feature in SolidWorks. The user can assign a name to the decision, for instance, *prototype evaluation* in the *statements* window, indicating that this group of statements addresses a decision regarding prototype evaluation. This allows the user to categorize the statements that address the same issue. Moreover, the user can assign a particular wiki page or even a pdf-file as an attachment to the design rationale (shown in *active selection window*). In this example the wiki shown in figure 8 is attached to *design of big rectangles*.

B853-2210

Files

B853-2210

Last edited by Morteza Poorkiaany 0 minutes ago

Footpad Design. Artcile nr. B853-2210

Design date: XXXXX

Design team members: XXXXX

Intention and goals:

This design is a part of the project nr. 1114:22 for developing a roof rack for car type xxxx-4 door sedan.

Assumptions:

Based on the created virtual model of the car's roof, the design team is agreed upon using the previously developed footpad nr B853-1415 as the base design. It is assumed that slight changes in its design will meet the requirements.

Alternatives:

After analyzing the virtual model of the car's roof and the design of footpad nr B853-1415, two scenarios are considered as follows:

- The first scenario is to only change the size of the big rectangles according to the car's roof model. The advantage of this solution is allowing the use of tooling nr T853-1415 with some slight changes in its design. A drawback for this solution is limitation in increasing the maximum load.
- The second scenario which allows a bigger maximum load is to change the dimension of the rectangles, as well as the height and thickness of the footpad. This will cause in major changes in design of both tooling and bracket.

After a discussion with project manager, the first scenario is chosen for detail design.

Evaluations:

A footpad was developed with rectangle's length dimension of 30,1+-0.5 cm. A prototype was made which it was not possible to mount it on the car. Thus, the tolerance has increased to 1.

The following tests have been accomplished before launching the product:

- City crash test according to ISO xxxx.
- Pull test according to DIN xxxx.

Running test is to be performed after launching the product to verify the maximum load.

Figure 8. A decision in footpad design recorded in wiki page.

In addition, the change of the dimension in footpad design should be captured in FMEA. Figure 10 shows a statement in the FMEA captured in Excel that targets certain cells addressing this issue. Since both SolidWorks and Excel are controlled by the function board, the content of the user interface in Excel was updated when *design of big rectangles* was created in SolidWorks. Now, the user can select the corresponding cells in Excel, right click on the *prototype evaluation* in statements window and click on *add statement*. The *statements* window will be updated showing paths to two statements. So, a design rationale object is created called *design of big rectangles* which targets two statements: one a feature in SolidWorks, and one a range of cells in FMEA. The statements are now connected to each other which means if the user clicks on the rectangles in SolidWorks the corresponding cells in Excel document will be highlighted in blue and the other way around, by clicking on a cell in that row in Excel the rectangles in SolidWorks will be highlighted. This makes it easier for the users to find the particular statement that they are looking for that is linked to the current statement they are clicking on.

The wiki page that was attached to *design of big rectangles* is also shown in the decision window in figure 10. It can be maximize or opened separately if it is preferred.

Figure 9. The Footpad modelled in SolidWorks.

The screenshot displays an Excel spreadsheet titled 'Excel' with a table containing FMEA data. The table has columns for 'CHARACTERISTICS', 'ORIGINAL STATE', and 'IMPROVED STATE'. The 'ORIGINAL STATE' column is further divided into 'Potential Failure Effect', 'Potential Causes', and 'Current Process Control'. The 'IMPROVED STATE' column is divided into 'Improvement Action'. The table lists six items, with item 6 highlighted in blue. To the right of the spreadsheet, there is a 'Design Rationales' panel with a tree view showing 'Product Design' and 'Footpad'. Below this is a 'Statements' panel with a list of statements, including 'Prototype evaluation'. At the bottom, there is a 'Decision' panel with a title 'Footpad Design. Article nr. B853-2210', a design date, design team members, and a description of the design's intention and goals. A 3D model of a footpad is shown next to the text. The bottom status bar indicates 'READY' and '100 %'.

CHARACTERISTICS		ORIGINAL STATE		IMPROVED STATE	
#	Dimension	Potential Failure Effect	Potential Causes	Current Process Control	Improvement Action
1	21±0.5	Weaker part	Wrong feeding length	Initial sample	Increase tolerance to ±1
2	22±0.5	None	Tool design	Initial sample	None
3	24±0.2	Not possible to mount	Worn tool	Fixture	Increase tolerance to ±1
4	27±1	Not possible to mount	Bad press setup	Measure	Increase tolerance to ±1
5	16±0.2	Bad function	Wrong feeding length	Initial sample	None
6	30,1±0.5	Not possible to mount	Tool design	Initial sample	Increase tolerance to ±1

Figure 10. The FMEA recorded in Excel.

Tooling Design:

Decision 2 shown in figure 7 is explained in this section. The main process in manufacturing a footpad is using a press machine. A tooling is to be developed to be used in the press machine for forming the footpad. The tooling consists of plates and cylinders that are designed separately and then assembled together. The design team is agreed upon reuse of a tooling that was developed previously for manufacturing a former footpad variant. Figure 11 shows the wiki page created to record the decision. The page is created by following the same template as used in product design.

Due to the changes in footpad design, the tooling needs to be modified. Two alternatives are considered and after a discussion the second alternative which suggests changing the positions of the cylinders in the upper plate is chosen. The arguments for and against each alternative is recorded in the wiki.

When the decision is made, the changes should be performed in the CAD model. Figure 12 shows the tooling modelled in Solid-Works. The upper plate is indicated by an arrow. The upper plate's sketch is modified according to the decision and a design rationale object is created called *upper plate* targeting the new dimension.

Further, the change is to be recorded in the design memo in order to inform the other engineers. Figure 13 shows the design memo recorded in Word document. By opening up the document, all the created design rationales are shown in the user interface. Now, the user can select the lines that express the change, select the *upper plate* in *design rationales* window, right clicks on the corresponding file path and selects *add statement*. So, the dimension in SolidWorks is connected to the lines in Word. By clicking on the lines in Word, the plate in SolidWorks will be highlighted, or vice versa (some lines in Word are highlighted in grey in figure 13).

The wiki page containing the information about the decision is also attached to the *upper plate*, shown in the *decision window*.

The system allows the users to group a number of design rationales together by mapping them into a virtual folder. For instance, *Tooling* is a folder containing three design rationales such as *upper plate*, *press*, and *cylinder* (see the *Browse* tab in figure 13).

Tooling T853-2210 ★
Files > Tooling T853-2210
Last edited by Morteza Poorkiaany 0 minutes ago

Tooling Design, changing the design of the plate

Artcile nr. T853-2210.01

Design date: XXXXX

Design team members: XXXXX

Intention and goals: This design is a part of the project nr. 1114:22 for developing footpad, article nr. B853-2210.

Assumptions:
The design team is agreed upon using the previously developed tooling nr T853-1415 as the base design. It is assumed that slight changes in its design will meet the new requirements.

Alternatives:
Two design alternatives are applicable as follows:

- The first alternative is to change the dimension of the length and width of the plate in order to provide a better fit on the footpad. This solution will be costly since the other plates need to also be adapted to the new size.
- The second alternative is to re-position the cylinders to provide enough space for sitting the footpad. This solution will cost less than the other one since only the plate is to be modified not the whole tooling.

After a discussion within the design team the second alternative was selected.

Evaluations:
The new design has been tested for manufacturing a prototype footpad and it is approved to be used in production of footpad.

Figure 11. A decision in tooling design recorded in wiki page.

Figure 12. The tooling modelled in SolidWorks.

Figure 13. The design memo recorded in Word.

4.3 Case study evaluation

The prototype system was evaluated in the case company through workshops and an evaluation session. The product development manager, the chief engineer for racks, and a production engineer participated in the evaluation. After presenting the system, a number of questions were distributed to be answered by the participants. Overall, the participants affirmed the possibility to employ such a system into the design implementations in the company. They stated the great advantage of linking the related information (statements) across the software, especially between SolidWorks and Excel. It was acknowledged that such functionality would highly facilitate tracing the effected knowledge when updates are required across the product documentation in the company.

Although the participants would like to test the prototype system in the organization across the board in a larger scale, they are optimistic that the benefits would outbalance the efforts. However, maintenance of an additional system, and the need for learning and understanding the system by the users were the concerns stressed by the participants. Providing a view of the CAD model in Excel in order to be used by the practitioners who do not need to have access to SolidWorks, were the suggestions for further improvements.

5. DISCUSSION

A method to support capture, structure, and access to design rationale was introduced in this paper. The contribution of the proposed method to each of these processes is explained as follows.

Capturing design rationale

In order to reduce the intrusion of capturing design rationale, it is suggested to embed capturing design rationale as closely in the design activities as possible. This can be achieved by enabling the users to capture the design rationale in the software that they are already using to do their work. Thus, the user does not need to open up an additional tool for the sole purpose of capturing the rationale.

Since accomplishing the design activities entails using different independent software, integrating the software allow the users to capture the design rationale in a range of formats that the available software in the organization support. The suggestion for creating such integration is to implement extensible applications into the software to enable capturing the rationale in the appropriate software wherever the need is raised. The application should have the ability to target the information in each software and connect the related information together in different software. An application with the intention of presenting the information in an explicit

way (in this study wiki was used to represent information in form of text and picture) could also be embedded in this environment in order to record the argumentations that is usually not captured in the statements, for instance, the rejected design alternatives.

One limitation in the introduced method is the lack of a shared base to allow the engineers to record the tacit knowledge from their minds in a common form that is understandable for all other users, not just some texts and comments that can be understood only by the persons who have recorded them.

Structuring design rationale

Structure is achieved by the proposed class model as presented in the UML diagram and the template provided for the wiki pages. The decision class presented in figure 2 makes it possible to group and map the related statements across different software. This facilitates traceability to pursue the effected information when a part of the knowledge needs to be updated.

What needs to be explored more is the granularity of the recorded design rationales and the ability to structure and represent the information in different layers of details as discussed in section 2.2.

Access to design rationale

Access to design rationale can be strengthened by enabling the user to quickly list, view, and browse the design rationales across the documents. Providing instant access would more likely encourage the designers to capture the rationale, because they know the information can benefit them or their colleagues now, not just during the future design projects. The suggestion is to provide a user interface to facilitate intuitive human-computer interaction. This can be supported by developing extensible applications integrated in the software that the designers are using, to make the information instantly accessible whenever it is required. However, a bottleneck in such user interfaces is to keep their efficiency and simplicity when the number of the statements and decisions are increasing.

The proposed method was examined by implementing a prototype system in the case company. One request from the designers was to find and expose the related design knowledge in one software when changes occur in the content of the design knowledge in another software. Therefore, access to the design rationales and statements through a common user interface was recognized as a big advantage of the method in order to fulfil the designers' needs at the company.

The literature review of design rationale systems provided in chapter 2, presents some similar contributions to this study. For instance, providing annotations to explicitly comment an argument in a CAD software, or hyperlinks to knowledge sources. What is new in this study is enabling the users to simultaneously work in an integrated environment of software in which the design rationale statements are interconnected in a way making it possible to go back and forth. For example, when selecting a cell in Excel the system noti-

fies the user about other interconnected Excel cells, but also interconnected SolidWorks-features, word bookmarks, or wiki-pages. Additionally, if found in random access memory the interconnected entities will interactively highlight in its software.

Currently, the prototype system has been tested on local computers. Therefore, it is not possible to exchange the information across networks. Centralizing the prototype system on a server and providing a database enables the engineers to communicate through interactive selections. Such an architecture would facilitate collaboration and communication among a group of engineers especially when one performs an activity in one software and the other one can simultaneously see the activity and trace the effected knowledge in another software. This allows the engineers to assimilate the relevant cues and examine the potential downstream effects of their decisions even if the engineers are not co-located.

6. CONCLUSION AND FUTURE WORK

The focus of the research presented in this paper was on capturing, structuring, and accessing design rationale by integration of different software that are commonly used in the design process. The research resulted in a method applicable to both product design and tooling design. The proposed method based on the introduced class diagram was tested in a case company developing car roof racks. It is essential that the product design conforms to the constraints and properties of the intended manufacturing processes, therefore, a close collaboration and support for knowledge exchange between product design and tooling design is important.

The feedback from the evaluation sessions and workshops in the case company shows the potential benefits of the presented method. Some of the advantages of the method are: integrating design rationale capture and access in the design implementations, representing design rationale in a proper format, and structuring the related statements into a decision. The practitioners at the case company found the proposed method helpful and that it could considerably save time and cost during developing new product variants. However, at this early stage, it is not possible to quantify the benefits of the method or if the benefits outbalance the effort. There are still issues remain regarding the consistency, quality, and version control of the captured design rationale and documentation.

From an industrial perspective the implementation and integration of a system is a critical process and of high importance. Understanding the relationships between the system and its external environment is essential to establish system functionalities and the way the system communicates with other tools and systems. Actions are required to ensure system alignment with the company's IT infrastructure, more specifically, elaborating on multiplatform solutions such as Linux, IOS, and windows. Additionally, the applicability of the method for a group of users, the communication among the users and the maintenance of the system realization are issues for future research.

ACKNOWLEDGMENTS

This research was a part of the IMPACT project financially supported by KK Foundation in Sweden. The authors gratefully acknowledge Thule Sweden for technical guidance regarding the case product and for being enthusiastic about the proposed method.

REFERENCES

- [1] D.M. Anderson, Design for Manufacturability & Concurrent Engineering: How to Design for Low Cost, Design in High Quality, Design for Lean Manufacture, and Design Quickly for Fast Production, CIM Press, 2004.
- [2] M. Bonev, Enabling mass customization in engineer-to-order industries, A multiple case study analysis on concepts, methods and tools, DTU Management Engineering, 2015.
- [3] S. Kim, R. Bracewell, K. Wallace, Improving design reuse using context, in: ICED07: 16th International Conference on Engineering Design, 2007.
- [4] J. Lee, K.-Y. Lai, What's in design rationale? Hum. Comput. Interact. 6 (3–4) (1991) 251–280.
- [5] A. Tang et al., A survey of architecture design rationale, J. Syst. Softw. 79 (12) (2006) 1792–1804.
- [6] A. Tang, Y. Jin, J. Han, A rationale-based architecture model for design traceability and reasoning, J. Syst. Softw. 80 (6) (2007) 918–934.
- [7] A.H. Dutoit et al., Rationale management in software engineering: concepts and techniques, Springer, 2006, 1–48.
- [8] S. Szykman, R.D. Sriram, W.C. Regli, The role of knowledge in next-generation product development systems, J. Comput. Inform. Sci. Eng. 1 (1) (2001) 3–11.
- [9] J. Lee, Design rationale systems: understanding the issues, IEEE Intell. Syst. 12 (3) (1997) 78–85.
- [10] J. Johansson, M. Poorkiany, F. Elgh, Design rationale management—a proposed cloud solution, in: Moving Integrated Product Development to Service Clouds in the Global Economy, 2014.
- [11] S.K. Chandrasegaran et al., The evolution, challenges, and future of knowledge representation in product design systems, Comput. Aided Des. (2012).
- [12] H. Wang, A.L. Johnson, R.H. Bracewell, The retrieval of structured design rationale for the re-use of design knowledge with an integrated representation, Adv. Eng. Inform. 26 (2) (2012) 251–266.
- [13] B. Hicks et al., A framework for the requirements of capturing, storing and reusing information and knowledge in engineering design, Int. J. Inform. Manage. 22 (4) (2002) 263–280.
- [14] J. Johansson, A flexible design automation system for toolsets for the rotary draw bending of aluminium tubes, in: ASME, American Society of Mechanical Engineers, New York, United States, 2007.
- [15] J. Johansson, D.C. Brown, Automated Computer Systems for Manufacturability Analyses and Tooling Design: Applied to the Rotary Draw Bending Process, School of Engineering, Jönköping University, 2011.
- [16] R. Owen, I. Horváth, Towards product-related knowledge asset warehousing in enterprises, in: Proceedings of the 4th International Symposium on Tools and Methods of Competitive Engineering, Hubei, China, 2002.
- [17] F. Elgh, Modeling and management of product knowledge in an engineer-to-order business model, in: ICED11: 18th International Conference on Engineering Design, 2011.
- [18] R. Bracewell et al., Capturing design rationale, Comput. Aided Des. 41 (3) (2009) 173–186.
- [19] W.C. Regli et al., A survey of design rationale systems: approaches, representation, capture and retrieval, Eng. Comput. 16 (3–4) (2000) 209–235.
- [20] Y. Nomaguchi, K. Fujita, Knowledge representation framework for interactive capture and management of reflection process in product concepts development, Adv. Eng. Inform. 27 (4) (2013) 537–554.
- [21] M. Poorkiany, J. Johansson, F. Elgh, A case study on implementing design automation: identified issues and solution for documentation, in: 20th ISPE International Conference on Concurrent Engineering, 2013.
- [22] K.M. Saridakis, A.J. Dentsoras, Soft computing in engineering design—a review, Adv. Eng. Inform. 22 (2) (2008) 202–221.
- [23] S. Murthy, D. Maier, SPARCE. Superimposed Pluggable Architecture for Contexts and Excerpts, Technical Report CSE-03-010, Maseeh College of Engineering and Computer Science, 2003.
- [24] S. Murthy et al., Putting integrated information in context: superimposing conceptual models with SPARCE, Proceedings of the first Asian-Pacific Conference on Conceptual Modelling, vol. 31, 2004.
- [25] R. Bracewell et al., In: ICED07: 16th International Conference on Engineering Design, Paris, France.

- [26] W. Kunz, H.W. Rittel, *Issues as Elements of Information Systems*, vol. 131, Institute of Urban and Regional Development, University of California Berkeley, California, 1970.
- [27] Y. Zhang et al., A semantic representation model for design rationale of products, *Adv. Eng. Inform.* 27 (1) (2013) 13–26.
- [28] O. Hisarciklilar, J.-F. Boujut, An annotation based approach to support design communication, in: *ICED07: 16th International Conference on Engineering Design*, Paris, France, 2007.
- [29] S. Sandberg et al., Supporting engineering decisions through contextual, model-oriented communication and knowledge-based engineering in simulation-driven product development: an automotive case study, *J. Eng. Des.* 24 (1) (2013) 45–63.
- [30] M. Lundin, *Computer-Aided Product Development: Using computer-aided technologies for efficient design capture and representation for reuse*, Luleå University of Technology, 2015.
- [31] F. Elgh, M. Cederfeldt, Documentation and management of product knowledge in a system for automated variant design: a case study, in: *New World Situation: New Directions in Concurrent Engineering*, Springer, 2010, pp. 237–245.
- [32] J. Burge, D.C. Brown, *Design Rationale Types and Tools*, AI in Design Group, 1998.