

Capturing, Structuring, and Accessing Design Rationale Across Product Design and FEA

Morteza Poorkiany^(✉), Joel Johansson, and Fredrik Elgh

School of Engineering, Jönköping University, Jönköping, Sweden
{morteza.poorkiany, joel.johansson, fredrik.elgh}@ju.se

Abstract. Implementing design automation systems to automate repetitive and time consuming design tasks enables engineer-to-order manufacturers to perform custom engineering in minimum time. To maintain a design automation system, regular updating of design information and knowledge is necessary. Consequently, there is a need of principles and methods to support capturing and structuring associated knowledge, specially, design rationale. In this paper a method for capturing, structuring, and accessing to design rationale in order to support maintenance of design automation systems is presented. The method is tested through a design automation system that develops FEA (finite element analysis) models automatically. The results are evaluated in a case company which is a supplier to the automotive industry serving many brands and car models which each more or less requires a unique solution.

Keywords: Design automation system · Computer supported engineering design · Design rationale

1 Introduction

For many companies it is important to innovate and launch new products according to new customer demands. Because of intense competition in the market and the need to being faster in developing new products, the firms attempt to control the quality of the product and manage the product information. PLM as a backbone to manage product-related knowledge, and technology used to access to this knowledge, through different domains of product lifecycle, helps the organizations to overcome the engineering challenges and complexities during developing new products. A PLM system is a discipline emerged from different methods and tools across different phases of product and can work as a hub for product development systems in order to increase reliability and facilitate exchange of product information [1]. Design process is an important element in PLM because making proper decisions at this early stage could optimize the development process and support the organizations to deliver their business initiatives and strategic goals. Design work encompasses a wide range of methodological approaches in order to solve different technical problems that require different solution strategies [2]. Regarding the time-intensive nature of many design tasks, up to 80 % of design time is spent on routine tasks [3] such as modifying,

adapting or redesigning already existing and proven solutions. Such activities that do not rely on high creativity can be performed automatically with the help of skillful automation tools executing existing design knowledge.

Design automation systems may have long life-span and require maintenance for proper function. Since the systems mainly manipulate and process design information, to maintain a system over time, frequent updating of design information is a necessity. Consequently, there is a need of principles and methods to support management of associated knowledge, specially, design rationale. Design rationale mainly explains the reasons behind the design decisions and developed solutions and helps to understand the effects of decisions on other aspects of design.

This study aims to investigate how design rationale can be introduced to support the use and maintenance of design automation systems. A focus is set on the need to share knowledge between FEA and product design when introducing design automation which automates the process of developing FEA models of customized designs. In order to manage the required design rationales, a method addressing capturing, structuring, and accessing to design rationale has been presented. The applicability of the method has been tested in a company acting as a supplier in the automotive industry providing unique product solutions for different car models.

2 Frame of Reference

The importance of access to design rationale was studied in a survey [4]. Approximately 80 % of the respondents said they fail to understand the reason of a design decision without design rationale support. Lee [5] defines design rationale as, "... not only the reasons behind a design decision but also the justification for it, the other alternatives considered, the tradeoffs evaluated and the argumentation that led to the decision". The realization of a design rationale system encompasses methods and tools to capture, structure, and share design rationale across organizations, processes, systems and products. The challenges regarding capturing, structuring, and accessing to the design rationale are discussed in [6]. One of the problems for design rationale capture is to define how and what type of knowledge should be captured. Indeed, any type of design information (e.g. product specification, geometries, design tables, rules, and catalogues) can potentially be a part of a design rationale. Capturing design rationale, especially when predefined templates and schemas are needed to be filled in, could intrude the design process and disturb the designer. Embedding the capture process in the current practice of the engineers would be a solution to overcome such problem. In addition, structuring and representing the design rationale because of the variety in type and format could be a challenge for the system developers. Well-structured design rationale makes it easy for the users to maintain and update the contents of the system. Moreover, the design information is stored in different software applications, therefore, the interaction between the design rationale system and the software applications at the company should also be considered by the system developers.

2.1 Related Work

In a recent research [7, 8] that was performed by the authors, a method for design rationale capture, structure, and access was introduced. The idea was to provide an integrated design environment constituted of traditional software applications commonly used by the designers. A computer-supported engineering system could be developed for administration and management of information flow between the software. The method enables the engineers to concurrently capture, structure, and access to the design rationale in different software. The discussed method identifies design rationale with three elements described as follows:

- *Design rationale targets* are references targeting underlying design information. Targeting a feature in a CAD model, specific bookmarks in a word processor documents, selections in plain text documents, and a range of cells in spreadsheets.
- *Design rationale description* is an explicit description (preferably in web page format) explaining different design issues. The aim is to record the part of the design rationale that is not captured in other software.
- *Design rationale connection* is a set of design rationale targets that includes any of the software applications.

A prototype system embedding SolidWorks, Word, Excel, and wiki pages was developed for evaluation of the method (see Fig. 1). The aim was to choose different wide spread software to represent the design rationale in different formats. The system was coded in VB.net environment and could be installed as extension in Word, Excel, SolidWorks, and a web browser. It is also possible to target more software applications if beneficial. So, the design rationale for an artifact in the prototype system consists of a set of targets such as a feature in SolidWorks, a table in Excel, a number of lines in Word, and a description such as a web-page in wiki. These all are connected together via the “connection” functionality provided in the system (see Fig. 1 left side). The system provides similar user interfaces containing three windows for displaying the targets, connections, and descriptions in all four applications (see Fig. 1 right side). This makes it possible to select a target (e.g. a feature in SolidWorks) and see the related targets (e.g. a design table in Excel), or descriptions (wiki pages) concurrently in all the applications via the user interface.

3 Management of Design Rationale in Product Design and FEA

Taking into account the purpose and intended use of design rationale is necessary when developing a design rationale method. It should be clarified what knowledge needs to be captured and how structure it in order to share the right details of the knowledge for the users.

Further improvements to the previously discussed design rationale method are provided in this section.

Generally, a customized product is constituted of assemblies that each assembly includes a number of components or maybe another sub-assemblies. Depending on the

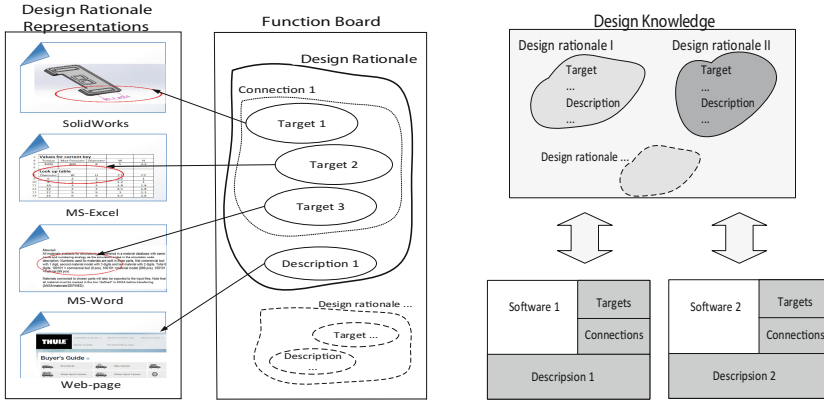


Fig. 1. The generic architecture of the design rationale method.

current task of the designer (designing a component or assembling a number of components together), the design rationale can be captured in different details to avoid recording duplicated knowledge. Below is a framework developed for capturing design rationale addressing what should be captured as design rationale and how. The framework targets two important phases of the development process: design phase and finite element analysis which making a decision in one highly effects the other one. FEA is mainly the analysis of the design and shows whether the part or assembly work the way they are designed which can steer the designers toward a more cost-effective product.

1. *What details of design rationale should be captured?*

1.1 Design rationale capture in component design.

- (a) The captured design rationale shall explain why the component is designed as the way it is. It should also address:
 - i. Performance of the component.
- (b) FEM analysis information. Necessary information for finite element analysis of the component.

1.2 Design rationale capture in assembly design.

- (a) The captured design rationale shall explain why the assembly is designed as the way it is. It should also address:
 - i. Performance of the component in the assembly
 - ii. Performance of the assembly
 - iii. Interaction of a component with other components
- (b) FEM analysis information. Necessary information for finite element analysis of the whole assembly.

2. *How design rationale should be captured?*

- (1) Identify design rationale targets
- (2) Attach a description to the target if required
- (3) Connect the target to the related targets in either the current or other software.

The design rationale can be captured by the engineers through design and FEA phases. So, the designer is responsible for capturing the reasons behind the design decisions, while, the FEM engineer captures the analysis information for the design.

Retrieval of the captured design rationale is easier when the design rationale is structured and classified within different categories. Since design rationale has been identified as a set of design rationale targets and a set of design rationale descriptions, one way to structure the content would be classifying the targets and descriptions (i.e. the wiki). The suggestion is to categorize all the targets into a connection if there is any relationship between them (see Fig. 1). In addition, similar to Microsoft Windows or GUI operating systems, folders as virtual locations can be created to help users store and organize the connections. The wiki pages on the other hand, can be classified in different categories.

Finally, all the design rationales shall be accessible and monitored in a standardized way. Accessibility can be supported by providing search techniques enabling the user to search the information according to the needs.

4 Case Study

A system for automatically develop a complete FEA-model by integrating a CAD system and FEA pre-processor was introduced by one of the authors in [9]. The prototype system connected Solidworks, ANSA, and LS-Dyna to automate the crash simulation process by providing structured mesh. An extension was developed and installed in SolidWorks to scan the CAD-model to generate a macro for different features, execute the macro to render the mesh model, appended the mesh model and compile the final model to the format of the targeted FEM-solver. Figure 2 displays generating mesh models in CAD software and FEM pre-processor. When running the system, FEA-models are generated in less than 4 min compared to up to a week when done manually. This increases the amount of tested product proposals ultimately increasing the product quality without increasing the product cost.

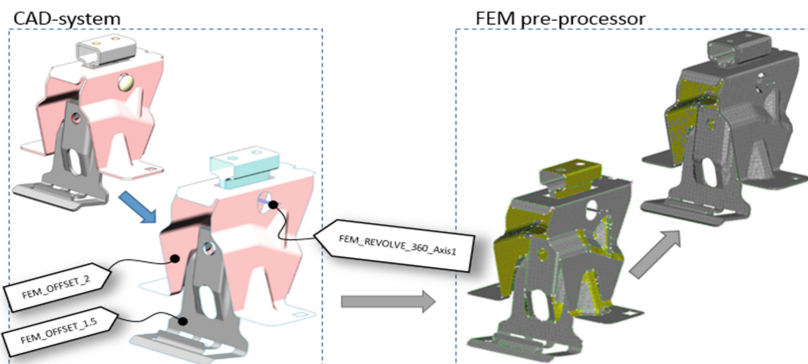


Fig. 2. Generating mesh-models in the design automation system.

Figure 3 shows a screenshot of a component (Footpad) in SolidWorks with both the design automation system on top, and the design rationale system on right. In this picture two windows of the design rationale system are shown: design rationale targets are displayed on the top and design rationale connections in the lower window. Footpads are part of the solution which are developed in the case company for attaching a roof rack on different car models. Generally, a new Footpad has to be designed for every new car model. Since the Footpad is directly adjusted on the car’s roof, analyzing the safety of the Footpad is strongly advised to the engineers and a crash simulation should always be performed on the roof racks.

To create the FEA model and run a crash simulation for the Footpad, a number of features in the design automation system such as defining material properties, thermal expansion, and surface contact need to be determined by the FEM engineer.

The features are accommodated in SolidWorks (e.g. Material feature in the left panel in SolidWorks in Fig. 3). Simulating a new Footpad variant with new properties requires updating these features and creating a new FEA model.

In addition, some decisions made in the design phase might also effect the FEA model and the crash test. For instance, the crash analysis might be valid for a range of Footpad alternatives which a designer could consider it during designing a new variant. Thus, capturing and sharing the design rationale behind both the FEA model and geometry would support the FEM engineers to update the FEA model and run the crash simulation faster and more efficient.

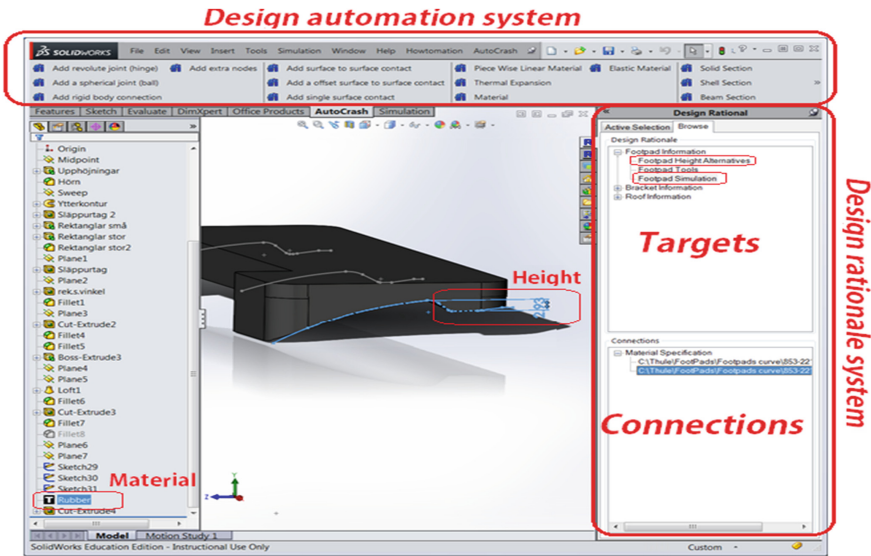


Fig. 3. Design automation and design rationale systems embedded in Solidworks.

4.1 Design Rationale Capture

This section shows how the required design rationale for updating a FEA model can be captured. Two design rationale targets are displayed in Fig. 3. One targets the material

property (Rubber) which is determined by the FEM engineer. The other one targets the height of the highlighted curve in the CAD model. The shape of the curve which is an important factor in finite element analysis varies in different Footpad variants and depends on the roof’s geometry.

The Connection window in Fig. 3 displays two documents connected to the targets. The documents are Excel and Word shown in Fig. 4. Excel document on the left side contains information about design alternatives for the Footpad. Word document on the right side contains information about material properties of the Footpad variants. The highlighted cells in Excel are connected to “Footpad Height Alternatives” target and the highlighted text in Word is connected to “Footpad Simulation” target. So, the designer captures information explaining the whys and design alternatives for the Footpad via the user interface (stated “Footpad Height Alternatives” in Excel in Fig. 4). The FEM engineer captures information explaining material specification for the Footpad (stated “Footpad Simulation” in Word in Fig. 4). In addition, two wiki pages; one describing the Footpad, its design process of the Footpad and supported documents, and the other one explaining the simulation workflow and tasks are presented in the bottom of Excel and Word.

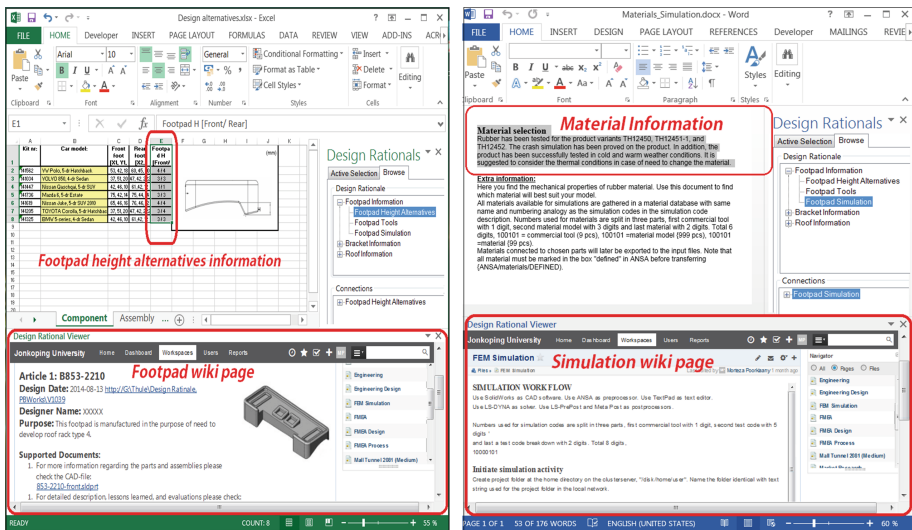


Fig. 4. Design rationale targets, descriptions, and connections in Excel and Word.

4.2 Design Rationale Structure

Figure 5 shows the content classification of wiki pages created in the case company. The engineering knowledge is classified into two groups: process knowledge and product knowledge. Process knowledge contains knowledge about select or development of required tooling, intended manufacturing process and such. Product knowledge contains knowledge about different phases of the development process such as product design and FEA. When there is a relation, for instance, between product design and

tooling design the wiki pages can be connected together by inserting links. New wiki pages can be created within current or new categories. Figure 6 shows the wiki page with the same user interface as the other software, containing the simulation information (this wiki was presented in Word in Fig. 4).

Additionally, the prototype system allows classifying the design rationale into folders. Each folder can be used for a specific context. For example, the design rationale targets in Fig. 4 are stored in three folders named: Bracket Information (contains targets for the Bracket component), Footpad Information (contains targets for the Footpad component), and Roof Information (contains targets for the cars' roofs). The system allows the engineers to create folders and classify the design rationale based on their context in the respective folder.

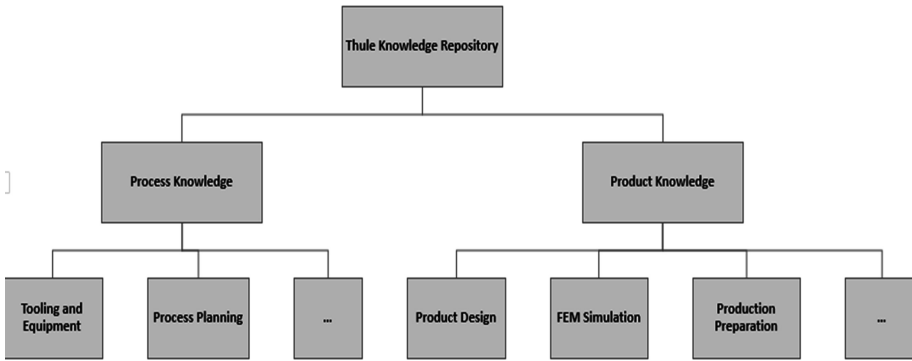


Fig. 5. Structuring knowledge in wiki pages.

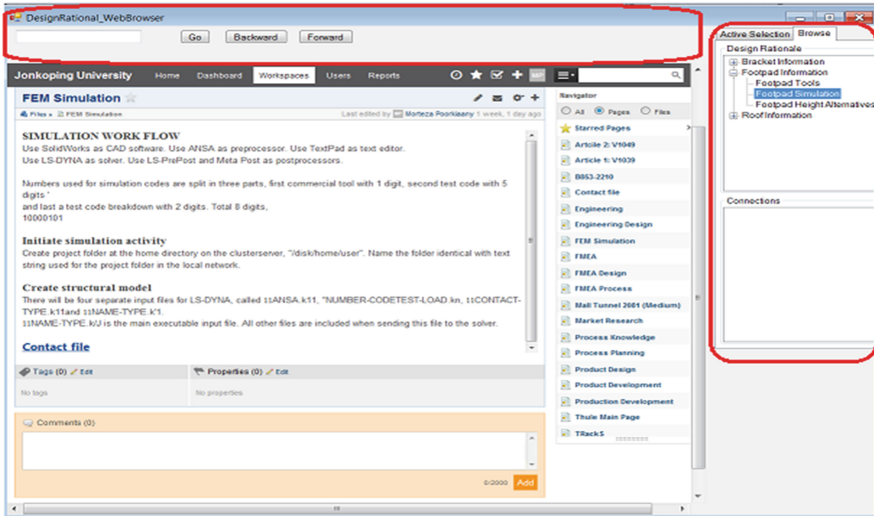


Fig. 6. The wiki page containing information regarding simulation process and the provided user interface.

4.3 Design Rationale Access

One way to support retrieval of existing design rationale is to allow search based on project, domain, date, name, etc. which is a simple task for system developers to provide such functionality. For instance, a user can search among the design rationales looking for a specific notation or vocabulary included in their contents. In order to expose information retrieval, in this study a search function is provided based on the names of the design rationale targets. Figure 7 shows an editor that is developed for the prototype system. The editor could have a number of functionalities which the most important ones are edit, manage, and search for the information. It is possible to monitor all the design rationale targets by selecting “Show all” in the “select” combo box (see Fig. 7 left side), or search for specific notation in the targets’ namespaces by selecting “start with” or “contains”. Figure 7 right side shows the result of a search for keyword “footpad” in the combo box. For each monitored design rationale target in the list view, the attached wikis (description) and connections are displayed in the property grid (see Fig. 7 right side).

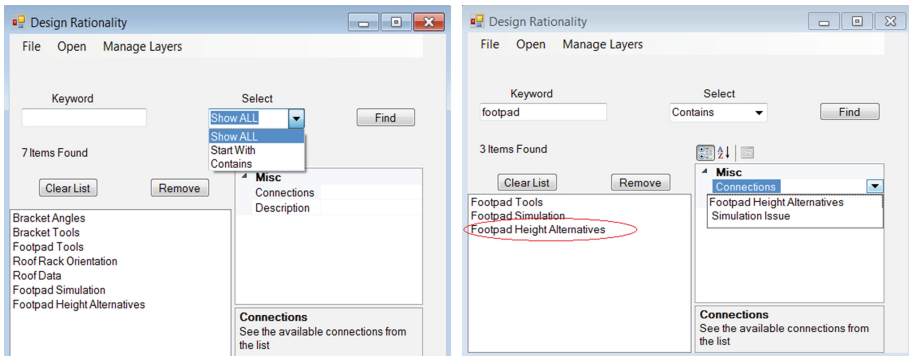


Fig. 7. Search function in the prototype system.

5 Conclusion

This study examined the importance of retrieving design rationale to support the use and maintenance of design automation systems. A method addressing capturing, structuring, and accessing to design rationale was presented. An example for capturing the design rationale in component level was provided. However, more studies are required regarding details of the information to be captured in both component level and assembly level while avoiding knowledge duplications. Retrieval of the design rationale based on the targets’ namespaces was investigated to show the high potential of search engines in finding the design rationales, more specifically, when particular keywords are predefined.

A prototype system for managing design rationale by embedding the common design software was evaluated in this study. A major benefit of such integration is to allow the organizations to model the design rationale system according to their needs

and strategic goals. However, traditional issues regarding aligning the system with design environment, platforms, and IT-infrastructure always occur during implementing the system which is one reason for failing the systems in practice. Currently, both the design rationale and the design automation systems are in prototype phase so it is not possible to quantify the benefits of the proposed method at this time. There is a need to investigate implementing and integrating the system across complex platforms on a larger scale in the company.

Acknowledgement. This study is a part of the Impact project financially supported by Knowledge Foundation (KK-stiftelsen) in Sweden. The authors are grateful to Thule Group for providing the technical facilities for the research.

References

1. Bergsjö, D.: Product Lifecycle Management—Architectural and Organisational Perspectives. Chalmers University of Technology (2009)
2. Sunnersjo, S., et al.: A transparent design system for iterative product development. *J. Comput. Inf. Sci. Eng.* **6**(3), 300–307 (2006)
3. Stokes, M.: Managing engineering knowledge: MOKA: methodology for knowledge based engineering applications. Vol. 3. Professional Engineering Publishing London (2001)
4. Tang, A., et al.: A survey of architecture design rationale. *J. Syst. Softw.* **79**(12), 1792–1804 (2006)
5. Lee, J.: Design rationale systems: understanding the issues. *IEEE Intell. Syst.* **12**(3), 78–85 (1997)
6. Dutoit, A.H., et al.: Rationale management in software engineering: concepts and techniques. In: Dutoit, A.H., McCall, R., Mistrík, I., Paech, B. (eds.) *Rationale Management in Software Engineering*, pp. 1–48. Springer, Heidelberg (2006)
7. Johansson, J., Poorkiany, M., Elgh, F.: Design rationale management—a proposed cloud solution. In: *Moving Integrated Product Development to Service Clouds in the Global Economy* (2014)
8. Poorkiany, M., Johansson, J., Elgh, F.: Supporting tooling design of customized products by instant access to design rationale. In: *The 6th International Swedish Production Symposium 2014* (2014)
9. Johansson, J.: A feature and script based integration of CAD and FEA to support design of variant rich products. *Comput.-Aided Des. Appl.* **11**(5), 552–559 (2014)