



Title:

Capturing and Sharing Design Rationale for Customized Design Variants in an Integrated Design Environment

Authors:

Morteza Poorkiany, morteza.poorkiany@ju.se, Jönköping University

Joel Johansson, joel.johansson@ju.se, Jönköping University

Fredrik ELgh, fredrik.elgh@ju.se, Jönköping University

Keywords:

Design Rationale, Customized Design Variants, Design Space, Design Automation System, Domain Specific Language

DOI: 10.14733/cadconfP.2021.339-343

Introduction:

Some companies represent their product family design as a space of design alternatives. The design space is a mathematical space that is governed by applying a set of design rules to design parameters. A key principle in establishing a design space instead of developing a single solution is to put minimum requirements for design optimization in the early stages of development. Developing a new design variant requires understanding the base design and adapting the design according to the new requirements. Use of design rationale can provide such understanding. With increasing the number of variants, access to design rationale becomes more important. Understanding why a design variant was accepted and if there have been any other alternatives in the beginning but were rejected (due to functionality or manufacturing limitations) can help the designers to avoid developing new design variants that have previously been proven wrong. Or it can support the designers to propose design variants that can be accepted under specific conditions for unique customers. The repetitive and time-consuming design activities can be assigned to design automation systems in order to increase efficiency and shorten the lead time. The use of design automation system is more vital when the number of design variants drastically increase. Use of Domain Specific Language (DSL) can amplify the applicability of the design automation system by targeting specific domains. A DSL is a computer language that is developed focusing on a particular application domain [3] (some DSLs are widespread such as SQL which is used for database querying, but many are not widespread).

In this paper, a method is presented for capturing and sharing design rationale for each design variant in an integrated design environment. An argumentation-based approach is used for capturing design rationale. An industrial partner which has a long tradition in automating its design process has been involved in the research, and the results of the research have been tested and evaluated in the company. In total, 11 workshops, 15 meetings and more than 20 semi-structured interviews with engineers from different departments in the company were conducted. A method and prototype system were developed to enable capture and share design rationale. The prototype system was demonstrated for a group of engineers on-site and a group of designers across the globe as part of the development. DSL is used initially by the company to develop the automation system. Our proposed design rationale system was developed as an extension to the automation system.

Case Company:

The industrial partner in this research follows an engineer-to-order business process. During the development process, each product family gets a number of defined standard variants like any other

product on the market, but the process also includes establishment of a “design space” for the product family. Design space is the valid range of product variants and is governed by a set of parametric design modules and the way these modules are combined. A set of variants to represent the entire design are developed. Based on the representative variants, geometrical rules and models are defined by the designers so that any variant of the product within the design space can be represented. A rule-based system (i.e., the design automation system) is developed to execute the design rules and generate new design variants. The system at run time selects, modifies, and combines design modules. The input to the system is specifications for a new design and the output is product related information such as 3D models, drawings, selected production unit, NC codes, cost and measurement instruction. DSL is used in the company for developing the design automation system.

Product Maintenance engineers and Special Design engineers are the main consumers of the information generated by the system. Product maintenance engineers are responsible to improve the design based on new technology, new knowledge or complaint from customers. Special design engineers are responsible to design a new variant that is highly specified according to the customers’ specific needs and fall outside of the design space.

To proceed with the research, first, engineers from Product Maintenance and Special Design departments were interviewed to investigate what information and knowledge they needed. Understanding how the design space is established was the major concern for almost all the engineers. In other words, they would like to know about those decisions that contribute to defining the design space. This mainly targets the evaluation of design alternatives. We simplify this into the following question:

What are the boundaries of the design space and what constraints contribute to establishing the design space?

To answer to the question mentioned above, it is required to capture the decisions that accept or eliminate a design alternative or a set of alternatives. For example, for a milling product, when the designer decides to set the rule:

If diameter=23 mm and if length= 25 mm \iff 1 < number of inserts = < 7

he/she should state why 7 inserts is acceptable for this design and what will happen if we would like to go for 8 inserts instead (beyond design space). To make this article simple to understand we tried to minimize the number of parameters in the above rule and we only mentioned the critical parameters.

During meetings with the engineers, we investigated design decisions. We realized that in many cases design variants were evaluated against three main criteria sources. These criteria include marketing, engineering design and manufacturing. Marketing concerns the market requirements and customer needs which is about business goals, specifications, and cost of the product. Engineering design concerns the feasibility of the design and that if the design meets the functional requirements. Manufacturing concerns the possibility of manufacturing the intended design. Accepting or rejecting a proposed design variant depends on if it can meet each of these criteria or not. We suggest keeping the number of criteria at a minimum to make it feasible to capture the rationale using this method

Using argumentation-based approach is a common way for capturing design rationale. In argumentation-based approach, semi-formal graphical nodes for laying out the structure of arguments are used [4]. The representation is based on node-and-link. Nodes represent the design concepts and links represent the relationships between the nodes. Question Option Criteria (QOC) [2] is one kind of such approach that is used for analyzing and representing the design space around the product. *Questions* are the identified key design issues, *Options* are the possible answers to the *Questions*, and *Criteria* are used to assess and compare the *Options*. We choose QOC model for the base of our proposed design rational approach. We used marketing limitations, engineering design limitations and manufacturing limitations as criteria in the QOC model to evaluate design variants.

In Figure 1, an information model is provided by integrating the QOC model. The QOC model is adjusted (Question is removed) to meet our expectations. The information model is to be used as the backbone for developing the design rationale system. The information model focuses on the evaluation of the design alternative according to the criteria and the decisions that are made based on

the evaluation for each design alternative. The argument class provides a description of the result from the evaluations and can be provided for each criterion. The argument class corresponds to the arguments and evidence to support the decisions. It was inspired from a study that aimed to clarify root cause analysis and accordingly support effective redesign [1].

The criterion class has three super classes: manufacturing limitations, marketing limitations and engineering design limitations. This allows the addition of more sources for criteria if needed. The properties of the decision class are: “decision maker” which corresponds to the person who is responsible for making the decision, the “date” of making the decision, and the “action” explaining whether the design alternative is rejected or accepted. Since the 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 and Part in CAD-models. This is represented in the model by the associated relation between the design alternative class and the part and assembly classes. There is an aggregation between part and feature and parameter and entity, which means that those classes are “part of “ the Part class.

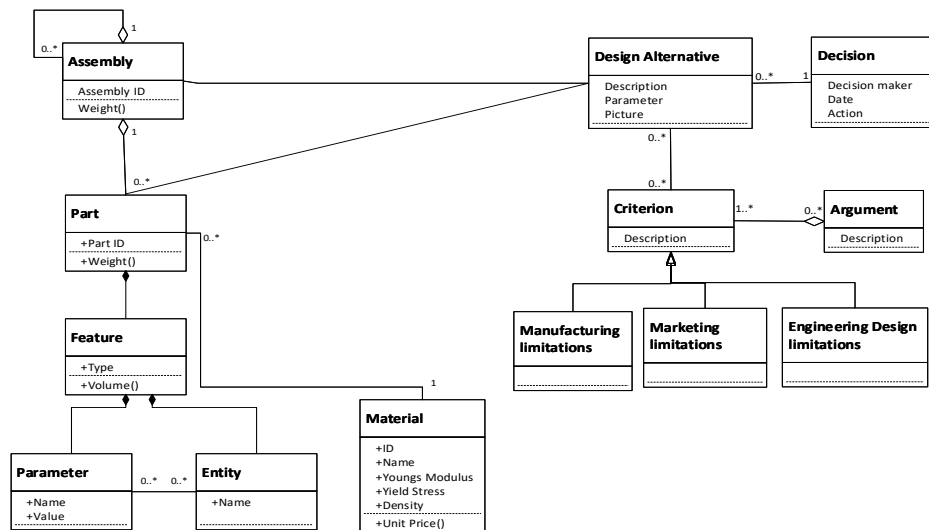


Fig 1: The proposed information model for developing the design rationale system.

The prototype system:

Our proposed design rationale system was built as an extension (add-in) to the design automation system that has been developed by the company to generate design variants. The provided extension allows to type in text in form of comments or insert pictures in the code. This functionality was used to capture design rationale. To share the design rationale with stakeholders, two aspects of the life cycle of a product family must be considered: sharing during the development process and sharing after releasing the product to the market. During the development process engineering teams are intensively working with the design and share knowledge and rationale through personal contact. The personal contact allows the engineers to develop a shared understanding of the product family. After the product release, this shared understanding is quickly eroded and should, of course, at that stage already be captured and secured. The only way to ensure that is to make the sharing process itself be the capturing process, i.e. make the engineers communicate through the capturing system. Two types of systems that do so exist today: your mailbox and social networks. One main drawback with these two types of systems is that they do not capture the rationale in relation with the concepts of the product knowledge. It is, in other words, hard to apply different perspectives to mailboxes (which, by the way, are private) and to wiki pages or other social media. This together with the growing trend of tailoring programming languages to the specific domain of the products, leads to the idea of making the design environment become the communication platform during the product family development

project. The second part of the life cycle includes Product Maintenance and Special Design. Special Design usually occurs a long time after the production release and is often under heavy time pressure. Special Design engineers need precise information regarding why decisions were made during the development project. Since the Special Design department is not necessarily within the same organization as the Product Development department, a minimum of information should be provided. The information regarding the entire product family must not all be shared with the Special designers. They would drown in information and, also, the combined information, including test reports and patents, is sensitive and might be considered confidential. However, information regarding derived instances from the product family should be shared.

All the captured design rationale for a specific design variant can be generated through the execution of the design automation system. This leads to the idea of compiling electronic publications (e.g., ePub or Pdfs) during runtime based on the design rationale stored within the semantic models in the DSL, see figure 2.

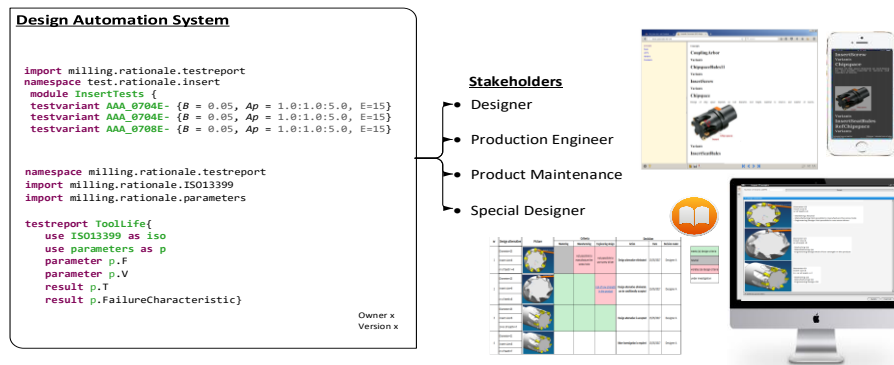


Fig 2: Capturing design rationale in form of comments in the code, sharing design rationale through the CAD software or in form of ePub and pdf.

Sharing design rationale in CAD environment as well as in form of e-book and Pdf:

The design automation system is a rule-based system that, based on input parameters, selects, modifies and combines design modules in CAD software. We realized that the best way to share design rationale to the engineers is in run time through the CAD environment when input parameters are entered. This provides better support for engineers and less intrusion to design.

The developed extension into the system enables sharing information in CAD environment. It extracts comments from the code for each executed code block when running the system. These comments may contain hyperlinks to other code blocks and to media content. The developed extension provides an "i" button (a symbol for more information) for the critical parameters in the CAD software during the "select" phase. If the user clicks on the i-button for a parameter in the CAD software, the design rationale that was typed in before into the system is shown in a pop-up window in the CAD software. Figure 3 shows the accepted and rejected design alternatives and the rationale behind them for a milling product. The comments from the code can also be compiled into electronic books (ePub and Pdf which can be viewed on most devices) containing all the information for that specific design alternative in an interlinked way. Thus, when viewing a page describing a class definition, for instance, it is possible to click on links to pages describing its members and other classes where it is used.

The results of the research were examined by the engineers in the company through three workshops. The workshops were scheduled with a presentation of the methods and tools followed by questionnaires. The engineers expressed that this type of information and support will help them when performing their tasks. At this early stage, with a proof of concept, the evaluation is not complete. When a special order of the selected product family is requested, and the special designers use this support, then it can be evaluated whether the benefits outweigh the time and resources required for capturing the rationale.

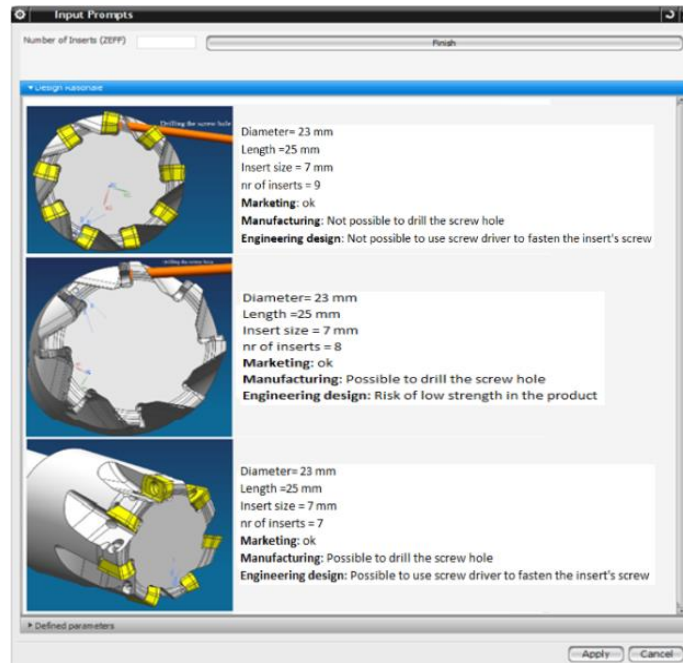


Fig 3: Sharing design rationale through CAD environment.

Conclusion:

As a domain specific language targets a specific problem domain, it is very beneficial to use it in development of customized design variants by reusing large amount of the programming code. What makes DSLs interesting to knowledge management of product families is the extensive use of semantic models which are lacking in Knowledge-Based Engineering systems. In this research, the design rationale was embedded in the semantic model. Capturing design rationale for a design space requires understanding the valid ranges for the design. The provided information model supports understanding the evaluation of design alternatives against three criteria. The evaluation can be made in part or assembly level. Sharing vs capturing – two sides of one coin, yet two fundamentally different functions that must be coherently aligned. Capturing the design rationale without sharing it is useless. Sharing the design rationale for stakeholders, facilitates reuse and supports the designers in solving similar design problems. There is always a trade-off between allocating resources for capturing the design rationale and how much the company gains from using the design rationale. It is not possible to analyse it in the short term. More research is required on that.

References:

- [1] Aurisicchio, M.; Bracewell; R.; Hoey, B. L.: Rationale mapping and functional modelling enhanced root cause analysis, *Safety Science*, 85, 2016, 241-257. <https://doi.org/10.1016/j.ssci.2015.12.022>
- [2] MacLean, A.; et al.: Questions, options, and criteria: Elements of design space analysis, *Human-computer Interaction*, 6(3-4), 1991, 201-250. <https://doi.org/10.1080/07370024.1991.9667168>
- [3] Kosar, T.; Bohra; S.; Mernik, M.: Domain-specific languages: a systematic mapping study, *Information and Software Technology*, 71, 2016, 77-91. <https://doi.org/10.1016/j.infsof.2015.11.001>
- [4] Shum, S. B; Hammond, N.: Argumentation-based design rationale: what use at what cost? *International Journal of Human-Computer Studies*, 40(4), 1994, 603-652. <https://doi.org/10.1006/ijhc.1994.1029>