20th ISPE International Conference on Concurrent Engineering
C. Bil et al. (Eds.)

doi:10.3233/978-1-61499-302-5-324

A Case Study on Implementing Design Automation: Identified Issues and Solution for Documentation

Morteza POORKIANY[a,1], Joel JOHANSSON[b] and Fredrik ELGH[c]
[a] *PhD Candidate, Jönköping University, Sweden*
[b] *Assistance Professor, Jönköping University, Sweden*
[c] *Associate Professor, Jönköping University, Sweden*

Abstract. Computer supported engineering design systems are used as support for designers by automating some tasks/activities of design process. From industrial aspect, implementation of a developed prototype system is a critical task. User acceptance is of high importance and strongly related to the access and understanding of the knowledge which requires a high level of system transparency. In addition, integration of the system in the environment or its compatibility with other systems/tools should be considered. Our experiences in industry show that two major issues are usually raised up during implementing a design automation system which are: documentation and organization. Documentation concerns the way of capturing, storing and distributing the information in systems, and organization concerns alignment of the system with other systems or tools as well as communication and collaboration among system participants and users. The focus of this paper is on documentation and the importance of reuse, design rationale and traceability is discussed. In order to align closely with industry practices, the thoughts are presented along with an on-going case study, where the development and analysis of roof racks for cars are being automated, and a number of challenges have been discussed.

Keywords. Computer supported engineering design systems, Documentation, Design Rationale and Traceability

Introduction

Many companies put much effort and investment in order to develop computer supported engineering design systems automating a variety of engineering design activities throughout the development process and production preparation. For example, Sellgren developed a framework for simulation-driven design [1], in which simulation models were extracted based on the CAD-model relationships. Also, Chapman and Pinfold described how to use KBE and FEA for the design automation of a car body [2], and a system was presented by Hernández and Arjona that automatically designs distribution transformers and that also uses FEM automatically [3]. The design process of different jet engine components has also been the subject for design automation using KBE (or KEE) integrated with FEA [4, 5]. Stolt developed methods to automatically develop FEM-models for die-cast components [6], and so on.

[1] PhD Candidate, department of Mechanical Engineering, Jönköping University
P.O.Box 1026, SE-551 11, Jönköping, Sweden. Phone: +46 (0)36 101571, Fax: +46 (0)36 125331,
Email: morteza.poorkiany@jth.hj.se.

The mentioned system architectures and solving methods have been tested through developed prototype systems. From industrial perspective the implementation of an idealized prototype is a critical process and of high importance for the actual use and consequently the benefits achieved and future return on the investment. There is a need to ensure the compatibility of the system with dependent methods and tools as well as the company's IT infrastructure. User acceptance is of high importance and strongly related to the access and understanding of the underlying knowledge which requires a high level of system transparency. Experiences show that when a system is being used in a company, extracting and utilizing the information and knowledge is an important task. The effective utilization and application of this information and knowledge assist the decision making process [7]. Moreover, the system should be expanded and updated over time by adding the information and knowledge of new tasks/activities or adding new knowledge sources in order to keep usefulness of the system [8].

Our experience in industry shows that in order to successfully utilize and keep the usefulness of the system, usually two major issues should be focused significantly during implementing the system. One issue is documentation which concerns the way of capturing, structuring, storing, and distributing the required information within the system. During documenting the knowledge, one should identify what knowledge should be captured and in which level.

The second issue is organization, which concerns the integration of the system in environment or alignment of the system with other systems or application software. At organization aspect, processes, methods and tools have to be systematically addressed and handled to ensure proper and effective usage. In addition, according to Turban and Aronson [9] in modern organizations groups make major decisions and therefor, communication and collaboration among system users is important especially when the users are in different locations.

Each of these issues, on its own, provide sufficient material for an entire research paper, therefore, the focus of this paper will be only on documentation. The paper provides a framework for documenting engineering knowledge discussing the main challenges. Then a pilot system with an information model is developed for a case study presenting the way of modelling the knowledge considering type and level of knowledge.

1. Documentation

An effective way of capturing and storing knowledge is using computer supported systems. By this, collective mind of individuals are transferred into a computerized system. Then a major aspect that should always be considered is that these two sources, collective mind of individuals and collective knowledge captured in computers, also require updates over time (see Figure 1).

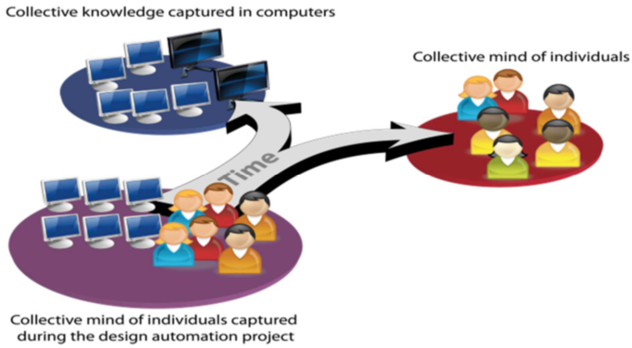

Figure 1. Changing in individuals sources and computer sources over time.

Based on the type and objective of the system many sources and types of information and knowledge could be used or produced during utilizing the system such as; product information, process information, the required knowledge describing assemblies and parts, records of previous activities, as well as catalogued information, CAD files, features, rules, bill of material and so on. The system user collects, verifies, and stores this knowledge in different repositories and formats, also associates it to different processes and knowledge sources. The main challenge regarding system use and longevity is the access to information and knowledge generated or utilized during using the system. As Baxter et al. [10] note, around 20% of the designer's time is spent searching for information and only 40% of design information requirements are met by documentation sources. This implies that design information and knowledge is not represented in a simply accessible knowledge base.

Because of the diversity of this knowledge capturing, structuring, storing, and representing the knowledge need significant effort. Stokes [11] described a methodology for the development of knowledge based engineering applications called MOKA. He states, capturing engineering knowledge consists of some steps: 1) prepare for collection, 2) collect required knowledge, 3) structure raw knowledge, 4) check for fitness of purpose, and 5) annotate and file models in knowledge repository. In addition, structuring the knowledge is the way to represent the knowledge in a form that make it easy for reusing existing knowledge in the future activities. Reuse of the knowledge is perceived to significantly increase the efficiency and is a means to reduce the product development lead time. In order to easily reuse the knowledge, availability and relevance of the knowledge should be considered during storing the knowledge.

1.1 Design Rationale and Traceability

During developing a product some decisions are made or some rules are created which access to design rationale gives insight into the reasons of making those decisions or rules that will support engineers for reuse or revision of the product in the future [12]. Falessi et al. [13] define rationale as,"… *not only the reasons behind a decision but also the justification for it, the other alternatives considered, the tradeoffs evaluated and the argumentation that led to the decision*". The access to design rationale can support development of new artifact, modification of existing artifacts (design changes) or the reuse of an existing solution in a new context.

Generally, it is hard to obtain design rationale from design specifications because there is no systematic practice to capture them. Tang et al. [14] mention even when some design rationale are captured, they are not structured in such a way that they can be retrieved and tracked easily. The realization of design rationale system includes methods and tools to capture, structure, manage and share information across organizations, processes, systems and products.

Since the knowledge is stored in different levels of sources and repositories, traceability is the key for supporting the ability to follow the origin of knowledge component and pursuing effected objects when changes occur in design. The information is traceable if one can detect (adapted from Kirkman [15]):

- the source of the information
- the reason why the information exists
- what other information are related to it or how the information is related to other knowledge.

1.2 User Guides

Information and knowledge should be captured and stored in such a way to make it flexible for reusing and revising. To be able to reuse a solution access to the knowledge that once was used is required. Finding the desired knowledge is easier by having a structured documentation. The documentation includes explanation about the product, parts, assemblies and parameters, and the relation between different parts and rules. Capturing the knowledge should be done in a way that knowledge bases are kept current, relevant and commitment as well as updated and modified.

The requirements concerning the scope and the granularity of design rationale to be captured depend on future needs of the knowledge. In many companies when studying the documentation of automated systems, it is directed towards describing the final results of different activities (answer to question What?) rather than describing the reason and the origin behind that activity (answer to questions Why, When, How?…). The former description is the definition of design (design definition) and the latest one is the rationale of design (design rationale). The documentation should be constituted by design definition and completed by design rationale, considering traceability for detecting the source and origin of information.

1.3 Developer Guides

To build and run a system one matter is extracting the design knowledge from designers in order to execute it. According to Sunnersjö [16] transforming the product knowledge from individual’s minds to executable codes is often extensive and many gaps or weaknesses in existing knowledge are often revealed. He states that design knowledge is a mixture of company policies and rules, experience of designer, design rules and so on, and for system developer will be soon clear that extracting design knowledge will not be easily done by just interviewing the designers, because usually a part of design knowledge and rules are based on the experience of the designers and does not exist in any documentation.

Another matter is the rationale behind the design and development of the system. The system developer uses a bunch of information and knowledge and makes some decisions during developing the system. Knowing the reason and origin of these decisions will be important in the future for modifying or maintaining the system corresponding to new changes (changes might be in product, process or technology) or updating the software or hardware of the system. This becomes more critical when the system developer is not a member of organization and he/she might not be available in the future for supporting the system maintenance.

In order to explore the discussed issues a design automation system was selected as a case study. The selected system is a part of a project running in a company with effort on reusing the knowledge in a new context or a modified solution.

2. Case Study: Thule Rack System (TRackS)

As a research case, an ongoing design automation project was selected. The project running at Thule Group company aims to automate the development process of car roof rack. The automation specifically targets roof racks that are mounted directly on the car's roof, i.e. there are no rails on the car. Consequently, the roof rack product has to be adapted to every car model it is supporting. The adaption is done by changing two components, the footpad and the bracket (see Figure 2). The footpad is a rubber pad on which the rack is standing on the roof, and the bracket is used to fix the rack by keeping around the roof end where the doors are.

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. When the roof geometry is collected, for a particular car model (A in Figure 2) a footpad (B) is retrieved or developed in the design automation system. The rack is subsequently placed on the footpad in the virtual model (C) and finally a bracket is retrieved or developed in the design automation system (D).

Since the number of developed brackets and footpads are increasing by entering new cars to the market, searching among the existing brackets and footpads to be used for new cars (maybe with some minor changes) is an enterprise task. As an example reusing an existing bracket cuts the overall lead-time up to 40%. But a time-consuming step during the development process of a ski-rack is the manual search among existing brackets and footpads, taking up to several hours. Since manual search is a painstaking task with an ever-increasing list of products, the engineers tend to skip that step and draw new components instead.

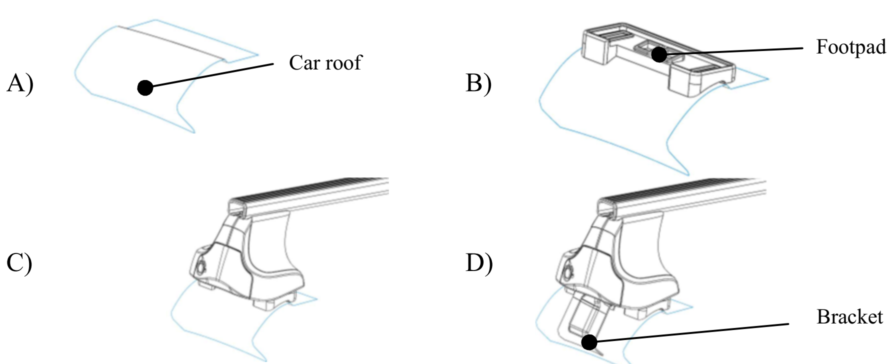

Figure 2. The roof rack product is adapted to new car-models by changing footpad and bracket components.

In order to search automatically among existing components, a computer-based system (TRackS) is developed [17, 18]. The system works as an add-ins to SolidWorks and uses the recorded design information. TRackS has the ability to search among the existing components and checks the applicability of previous solutions for a new car roof based on shape matching. TRackS utilizes a database which has been set up by the system developer for storing design information. Further, for running TRackS having access to roof data, drawings, CAD files, and bill of material is a necessity. To update TRackS, the files and knowledge should be updated and that would be possible when the rationale part of the knowledge is captured. Then it would be easier for the system user to reuse or update the documentation by knowing the origin and reason behind each activity.

On the other hand, documenting design information of the whole rack product is taking place traditionally by creating a folder on the company's database and saving all relevant documents and files there. The folder includes test reports, check lists, drawings, CAD files, features, BOM and so on. Although the engineers at the company try to describe and define the activities by writing reports but it seems that capturing design rationale is the missing mission in documenting the design knowledge.

Geometries and CAD files are the most type of information which are used by TRackS. This type of information mainly describe the results of the activities for a context. Such information might be enough if the context is to be used as it is, but if the context has to be modified and adapted to specific circumstances even more information is required to support the adaption.

3. Pilot System

A system founded on the presented framework of documentation for modeling design knowledge for TRackS was developed. A recently developed rack product was selected to be modeled in the system application (see Figure 3). The application which was used is based on wiki pages. CAD models, bill of material, features, rules, and roof data are the required information for TRackS and for each of them a wiki page is associated. The page contains all the required information for that context such as Excel, Word, figures and etc. These can be added to a page by uploading the specific files and then create links to them. The information and knowledge could be described by using text, figures, tables, and rules. The page also includes the principle and function of that item in the product, and the rules and their validity for the product family. Since in the current documentation of the company design rationale is not recorded, rationale behind every rule and knowledge can be discussed and documented during meetings and discussions with the designers.

The representation of knowledge, including design rationale for TRackS is based on the information model depicted in Figure 4. The model is implemented in Microsoft Visual Studio and uses the concept of classes for describing the items (rule, BOM, and …). Of central importance is the Rationale class that connects to all other classes within the design process. Basically the type of rationale would be different for each stated class in figure 4. The required information for bracket, footpad and roof data are inherited from the CAD-model class which are not shown in this picture. The model shows the relation between different classes and also the required classes to create an item. The item would be the final assembly model of a product variant.

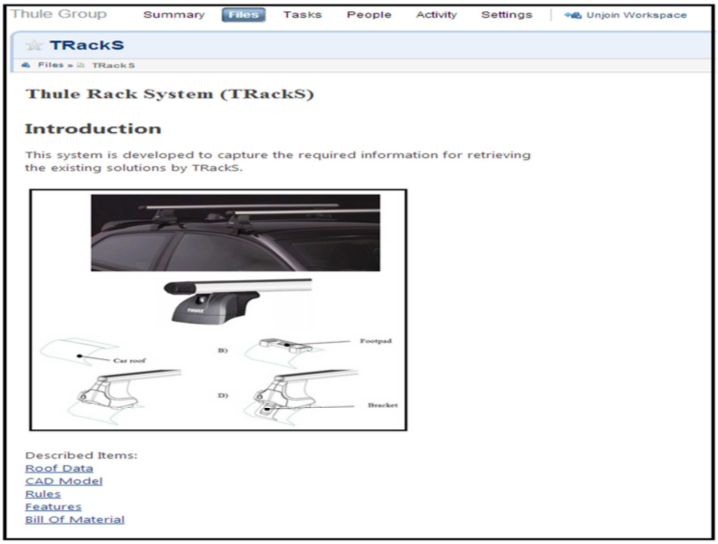

Figure 3. Main page of developed system's knowledge repository.

4. Conclusion

Documentation and organization are the two major issues which are identified during implementing engineering design automation systems. The objective of the paper was to illustrate the challenges that exist during documenting the design knowledge in systems. The research was expressed in a better way by implementing a case study. As a general overview, the engineers at the company are satisfied by the results of TRackS. 50% reduction of the costs for making new tools and 40% reduction of lead time for developing the product are the two great benefits of using TRackS at the company.

In most design automation systems, reusing product knowledge for modifying an existing product or developing a new product variant is a significant task. Therefore, structuring and organizing documentation including design rationale and considering traceability is a necessity with the objective to reuse, and maintain the generic product family objects embedded in design automation systems. Design knowledge can be modeled in a system application in order to support capturing of design definition and design rationale, facilitate high quality documentation, link models, items and supported documents together, and update the documentation easily.

The system developer of TRackS confirms the applicability of the proposed framework for documenting and modeling the knowledge in general but further researches are required to fully validate and evaluate the approach and supporting tools for feeding the system with the right level and right quality of information. Also more study is required regarding organization, the integration and dependency of the system, as well as communication and collaboration among system users.

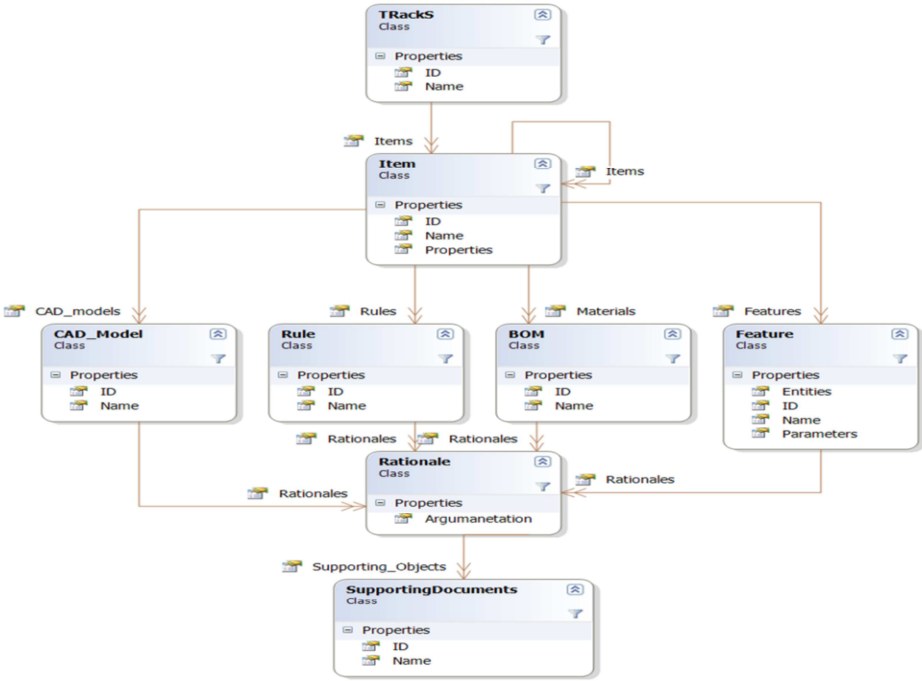

Figure 4. Information model for TRackS

Acknowledgments

The authors express their gratitude to Thule Group Corporation and the Knowledge Foundation (KK-stifitelsen) in Sweden for financing the research project. Also special thank is given to the engineers and managers at Thule company for technical support.

References

1. Sellgren, U., *Simulation driven design-A functional view of the design process.* Licentiate Thesis, Department of Machine Design, Royal Insitute of Technology, Stockholm, 1995.
2. Chapman, C.B. and M. Pinfold, *The application of a knowledge based engineering approach to the rapid design and analysis of an automotive structure.* Advances in Engineering Software, 2001. **32**(12): p. 903-912.
3. Hernández, C. and M. Arjona, *Design of distribution transformers based on a knowledge-based system and 2D finite elements.* Finite elements in analysis and design, 2007. **43**(8): p. 659-665.
4. Sandberg, M., *Design for manufacturing: methods and applications using knowledge engineering.* 2007.

5. Boart, P., *The enabling of product information in the conceptual phase*, 2007, Citeseer.
6. Stolt, R., *CAD-model Parsing for Automated Design and Design Evaluation*, in *Institutionen för produkt- och produktionsutveckling, produktutveckling*2008, Chalmers tekniska högskola. p. 162.
7. Hicks, B., et al., *A framework for the requirements of capturing, storing and reusing information and knowledge in engineering design.* International journal of information management, 2002. **22**(4): p. 263-280.
8. Elgh, F. and M. Cederfeldt. *A Design Automation System Supporting Design for Cost-Underlying method, system applicability and user experiences*. in *CE2005: ISPE International Conference on Concurrent Engineering*. 2005. Fort Forth, Texas, United States.
9. Turban, E., J. Aronson, and T.-P. Liang, *Decision Support Systems and Intelligent Systems 7"'Edition*. 2005: Pearson Prentice Hall.
10. Baxter, D., et al., *An engineering design knowledge reuse methodology using process modelling.* Research in engineering design, 2007. **18**(1): p. 37-48.
11. Stokes, M., *Managing engineering knowledge: MOKA: methodology for knowledge based engineering applications*. Vol. 3. 2001: Professional Engineering Publishing London.
12. Sung, R., et al., *Automated design knowledge capture and representation in single-user CAD environments.* Journal of Engineering Design, 2011. **22**(7): p. 487-503.
13. Falessi, D., G. Cantone, and M. Becker. *Documenting design decision rationale to improve individual and team design decision making: an experimental evaluation*. in *Proceedings of the 2006 ACM/IEEE international symposium on Empirical software engineering*. 2006. ACM.
14. Tang, A., Y. Jin, and J. Han, *A rationale-based architecture model for design traceability and reasoning.* Journal of Systems and Software, 2007. **80**(6): p. 918-934.
15. Kirkman, D.P., *Requirement decomposition and traceability.* Requirements Engineering, 1998. **3**(2): p. 107-114.
16. Sunnersjö, S. *PLANNING DESIGN AUTOMATION SYSTEMS FOR PRODUCT FAMILIES-A COHERENT, TOP DOWN APPROACH*. in *Proceedings of the 12th International Design Conference DESIGN 2012*. 2012.
17. Johansson, J. and M. Cederfeldt. *INTERACTIVE CASE BASED REASONING THROUGH VISUAL REPRESENTATION-SUPPORTING THE REUSE OF COMPONENTS IN VARIANT-RICH PRODUCTS*. in *Proceedings of the 12th International Design Conference DESIGN 2012*. 2012.
18. Johansson, J., *Combining Case Based Reasoning and Shape Matching Based on Clearance Analyzes to Support the Reuse of Components.* 2012.