INTERNATIONAL DESIGN CONFERENCE - DESIGN 2006
Dubrovnik - Croatia, May 15 - 18, 2006.

DESIGN 2006

TOWARDS A STRATEGY FOR MAPPING OF DESIGN PROBLEMS TO SUITABLE SOLUTIONS – A CASE OF DESIGN AUTOMATION USING CBR

M. Cederfeldt

Keywords: design automation, solution strategy, case based reasoning (CBR), variant design

1. Introduction

In order to make the designing of product variants more efficient and effective there often exists the possibility of automating the process, or at least implementing some form of computer support to aid the designers. Designing though, is not a simple and single task. Instead, it often consists of several interlinked sub-tasks that have to be performed either in some previously known order, iteratively, or perhaps even by inference. Furthermore, the levels of knowledge formalisation, task formalisation, and process maturity may vary from known and clearly documented tasks (explicit), to known but undocumented tasks (implicit), or even unclear and unstructured tasks (ad-hoc) [Cederfeldt and Elgh 2005]. To address this there is a need to break down and analyse the design process that is intended to be automated or supported [Cederfeldt 2004]. In doing so, a clearer picture of the actual design process will emerge. From this a problem definition and a preliminary system specification can be outlined [Cederfeldt 2005]. This will in turn give rise to new questions that needs to be answered in parallel with the setting up of a final system specification. Some of these questions address the choice of solution approach related to the design process and its inherent knowledge.

This paper presents one such attempt at breaking down a design problem, defining its process character and capturing its inherent domain knowledge. This is then mapped to suitable tools, and computer implementations. Also, one of the tools chosen in this work, Cased Based Reasoning (CBR), will be addressed further together with some implementation issues of CBR as well as the advantages of a variant design approach to setting up of CBR indexing templates.

2. Mapping of problem definition to solution strategy

Product design is according to Roozenburg and Eekels [1995] “the process of devising and laying down the plans that are needed for the manufacturing of a product”. Andreasen [1991] states that the design process encompasses all aspects from product and market planning to the solving of individual tasks. This work is focused on finding computer support for mature and repetitive design tasks (and processes) of “general problem solving” character [Andreasen 1991] where the aim is to free the designers by automating these processes. In this work the terms Computer Support (CS) and Design Automation (DA) [Cederfeldt and Elgh 2005] will be used synonymously and refers to:

“Engineering IT-support by implementation of information and knowledge in solutions, tools, or systems that are pre-planned for reuse and support the progress of the design process. The scope of the definition encompasses computerised automation of tasks that directly or indirectly are related to the design process in the range of individual components to complete products.”

In order to create automated design systems one must first categorise the process, design task/s, and problem/s for which the system is intended. Then an appropriate computer implementation can be selected. The process of mapping a problem definition to a suitable solution strategy (related to design automation and computer support) is divided into four interlinked sub-domains of design automation (Figure 1) [Cederfeldt 2005].

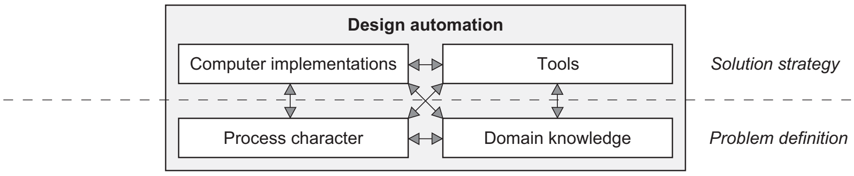


Figure 1. The sub-domains of design automation. In Cederfeldt [2005] adapted from conference presentation of Cederfeldt and Elgh [2005]

Addressing these sub-domains should, ideally, start by breaking down the design process and identifying the domain knowledge linked to it. This is done with the purpose of formulating a problem definition. Examples of approaches suited for this purpose are the use of Dependency Structure Matrices which is described by, among others, Browning [1998], and the use of principals for formal documenting and structuring of knowledge according to, for example, the MOKA consortium [MOKA 2001]. When the process, its knowledge, and the tasks to be performed are known, the appropriate tools have to be chosen. Following this is the identification and selection of ways of computer implementations.

The four sub-domains are, together with some examples, described in more detail as:

- Process character – The design process and its handling of the domain knowledge and design information. A design process is, for example, based on optimisation, packing, configuration, choice, or reasoning. It can also be, for example, time demanding, iterative, or ad-hoc.
- Domain knowledge – The type of knowledge that is to be handled in the design process. The knowledge, information, or data is, for example, explicit, implicit, structured, unstructured, delimited, or aggregated.
- Tools – Suitable tools (methods) that support the handling of domain knowledge and information for the intended solution principals. Examples of tools include Design Structure Matrix, Function Means Tree, inferencing, Cased Based Reasoning, Neural Network, modularity, parametrics, and different computational approaches.
- Computer implementations – Suitable computer implementations supporting the identified process character, domain knowledge, and tools. The implementations of the tools suited for an identified process character and its domain knowledge can be done, for example, as total or part solutions, in different execution paradigms (sequential or declarative), with Knowledge Based Systems or CAD macros, commercial off the shelf (COTS) application software, or specialised (in-house developed) application software.

The purpose of addressing these sub-domains of design automation is to find the best way to combine process character and domain knowledge with appropriate tools and computer implementations through the mapping of problem (and task definitions) to solution strategies. “The best way to combine” implies finding combinations and implementations that meet the requirements and prerequisites of the implementer, and preferably doing so in the most cost-beneficial way.

3. Case of application

The case of application used in this paper for the purpose of mapping a problem definition to a suitable solution strategy is the process of designing (dimensioning) components for roof-mounted car rack systems. The mounting system for these racks consists of a rail, a locking system, a main housing, a modular rubber mounting foot, and a car model specific fixturing bracket (Figure 2).

This work focuses on the development of a design automation system for variant design of the fixturing brackets. The case DA system is defined and delimited by the following problem description problem definition and its mapping to a suitable solution approach.

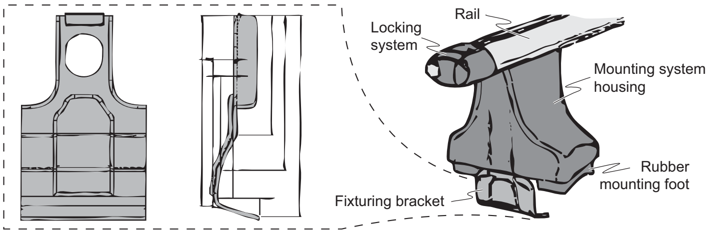


Figure 2. Car rack mounting system (right) and a fixturing bracket variant (left)

3.1 Problem description

For every new car model a new mounting system has to be designed. Unless a contract for a car model specific rack system exists between the company and the car manufacturer, the mounting system is designed as an after-market system. This involves choosing the best suited mounting foot from a set of variants and designing (dimensioning) a fixturing bracket variant suited for the car roof profile. The design of the bracket is based on measurements made on a physical car made available to the designers, and not on a CAD-model. Based on the measured roof profile the designer can sketch a profile for the bracket. The different bracket variants include both topological and dimensional variations. If the sketched profile is found to be similar to any prior bracket solution, that already existing bracket can be used for testing and on-site modification for prototyping purposes.

The main problem is finding suitable prior solutions among up to 800 variants, documented only in drawings. This task has to be performed by the designer based on his/her personal recollection of earlier designs. Furthermore, finding a prior suitable solution is no guarantee for that solution being the most suitable one, as several other good solutions might exist. Not being able to find an existing prior suitable solution (or not finding the right one) may result in designing of duplicates. These unintentional duplicates result in a growing number of variants which are increasingly difficult to manage. There is also an addition in workloads as testing and verification of new bracket designs by simulations and physical tests are necessary. In addition, new manufacturing tools are required with unnecessary added product cost as a result.

4. Mapping of case of application problem definition to solution strategy

According to the process of mapping problem definition to solution strategy, as described in paragraph 2, the case of application, i.e. designing of product variants of fixturing brackets, is in the following paragraphs addressed in more detail. As a base for decision making, implementation criteria for design automation systems [Cederfeldt 2005] were used.

4.1 Process character

As in most cases the main design objective is to optimise the solution. This however is not always the character of the operative tasks in the design process. The process has to be further broken down in order to specify the type and character of the process as well as its design tasks. In this case, as the problem description outlines, the main problem is to design bracket variants with minimal effort. This involves, if possible, *selection of prior suitable solutions* in order to simplify designing and prototyping, and also to eliminate the risks of duplicate designs. In summary, the objective is to find prior cases and to select the most suitable prior designs for prototyping.

4.2 Domain knowledge

The knowledge needed to select the most suitable prior solution is documented in *archived drawings* (*explicit knowledge*). The task of finding these drawings is based on the individual designers' "*expert knowledge*" and recollection of prior solutions. Decisions on which drawing (and design) that is the most suitable one is also based on the individual designers' "expert knowledge" as well as on *heuristics* ("rules of thumb", which can be seen as *implicit knowledge*).

4.3 Tools

Based on the process character and domain knowledge, *Case Based Reasoning* (CBR) is identified as a tool suitable for finding prior cases when there exists a vast amount of structured and searchable data. The case data in this case of application existed in archived drawing printouts (not immediately searchable without digital database archiving). For this reason the brackets were *parameterised* in order to obtain structured and searchable data. For selection of best suitable prior solution based on heuristics and rules of thumb, CBR has its limitations. For this further selection, reasoning based on heuristic knowledge is needed. For example, the best suitable prior solution might not be the one which has the highest similarity based on geometry, but instead the one that will need the least, or from a mechanical point of view the most beneficial, adaptation for prototyping purposes. For the selection between the most suitable prior solutions based on heuristics, some form of *Expert System* (ES) or *Decision Support System* (DSS) can be recommended.

4.4 Computer implementations

To implement the identified suitable tools, a system architecture was proposed according to paragraph 6. In the first stage (CBR screening), a pilot system was set up for evaluation of the CBR-approach. This was done using an open source CBR software, FreeCBR [Johansson, L. 2005], together with an Excel sheet as a temporary database in which the existing solutions' searchable data was entered. A student project then began development of an operational system with a more graphical oriented user interface. This system incorporates CBR functionality linked to a database containing all desired data about bracket designs. In the not yet implemented second stage (ES/DSS selection), heuristics and rules of thumb will be captured at the company. In parallel to this, the selection of suitable ES/DSS software for implementation of a second stage pilot system (see paragraph 6 and Figure 4) is performed.

5. A variant design approach to storing and retrieving design solutions

CBR was chosen for its obvious advantages in quick identification of similarities between vast amounts of data. Summaries, reviews, and examples of CBR in design can be found in, among others, Maher and Pu [1997] and issues concerning difficulty in information retrieval and indexing are presented in Li et al [2004]. In this case, the problem definition singles out the process of screening close to 800 already existing variants of fixturing brackets in order to find the most similar and best suited previous bracket designs for prototyping purposes. To be able to perform the CBR screening the existing solutions were digitally documented and stored in a database. The first task in this process was to decide on a proper approach of indexing. In many CBR cases, searchable parameters have to be selected from a vast amount of data. When searching for similarities between design solutions based on two-dimensional drawings it is, for example, possible to search by information in the title block, number of view perspectives, parts material, and main dimensions/parameters [Johansson, P. 2005], just to name a few. However, in this case of application there was no explicit standard for dimensioning of bracket designs. This makes searching for similar designs somewhat more problematic although geometrical similarities could be identified through, for example, picture analysis. To solve this, a less complex and more straightforward approach was adopted as the bracket design was decided to be parameterised. After cataloguing of the different topological variants, a pattern of an implicit dimensioning standard could be found. Once this implicit dimensioning standard was defined and turned explicit, parametric bracket variants could be created. Design parameters were decided upon for these brackets, enabling searchable data (and CBR indexing) for all future designs. However, for prior solutions to conform to the new parametric bracket designs the old designs had to be re-documented using the defined dimension standard and parametric design variables.

5.1 Parameterisation of the bracket

The parameterisation of the fixturing brackets were based on a variant design approach divided between five topological main variants (based on the designers' description on different bracket functionality such as number of bends and grip angle direction). From these topological variants an arbitrary number of dimensional variants can be created. Figure 3 exemplifies the parameterisation of one of the topological variants, showing coordinates for bending of the bracket as well as additional performance parameters such as grip length and angle (derived and made searchable from the bending coordinates). Dividing the bracket designs into several topological sub-variants simplifies the process of parameterisation. It also renders the CBR indexing process fairly easy compared to other more complex designs.

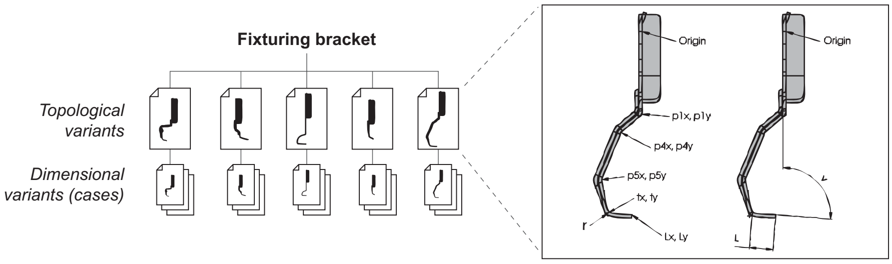


Figure 3. The parameterisation of one of the five topological variants of the fixturing bracket

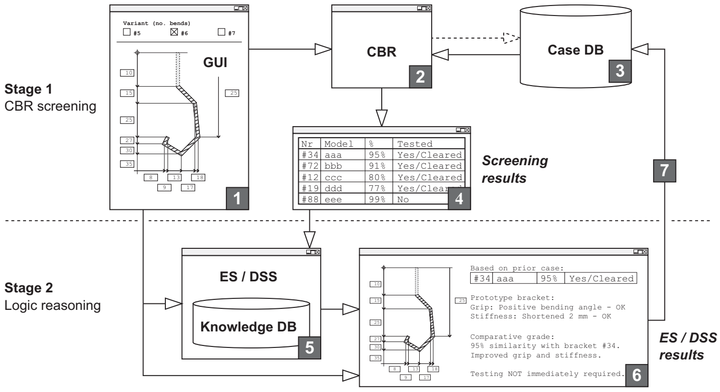
5.2 Advantages of using a variant design approach

A commonly difficult task in CBR is to capture the knowledge needed for retrieval of prior cases. An even more difficult task is to decide on an indexing template for storing the needed knowledge in a searchable format. In most CBR implementation cases a knowledge engineer (or someone with similar function) will try to capture the right knowledge and store this in an indexing template suited for CBR. This may result in a case description that is not in accordance with the designers' way of describing the product and/or case. Therefore it is important that the designers participate in the case indexing process in order to gain an understanding and familiarity with the process of CBR. Li et al [2004] highlights the difficulty in using CAD/PDM/PLM systems for retrieving prior cases as they often are documented based on some company naming convention, pointing out that retrieval should instead be based on function, intended use, or context information. This design case however, is fairly straightforward when broken down as it involves designing (dimensioning) of variants that only changes their topology (and function) in a clearly defined number of ways. Further, by using a parametric design, case retrieval will be made based on an explicitly defined geometry built up of a number of two-dimensional parameters.

The advantage of using a variant design approach to generating searchable documentation comes from focusing on parameterisation of the design. This is because search oriented parameters are an inherent part of variant design when driven by the actual design parameters. By focusing on the designers' way of describing the product [Cederfeldt 2004] and carefully selecting design parameters for parameterisation, the usability of the CBR approach is enhanced as an indexing template suited for CBR is automatically obtained. Furthermore, this indexing template can later be used to automatically generate variant designs from adapted prior cases. Also, new variant designs (cases) automatically conform to the CBR indexing template. This type of approach also has the potential of increasing the level of transparency and understandability of the system and its documented knowledge [Cederfeldt and Elgh 2005]. The only data missing from the geometrical parameterisation of the design is information on designer, case and drawing number, date, and information about performed physical testing (and links to test documentations). These are added to the indexing template and stored with final designs in the case database.

6. Pilot system

The pilot system's architecture was according to the mapping of case of application problem definition to solution strategy (paragraph 4) divided into two stages. One consisting of a screening process using CBR, and the other (not yet implemented) consisting of an ES or a DSS for further logical reasoning. This reasoning is to be based on the relation between the prior solutions, found by CBR similarity, and the proposed new bracket design as well as on designers' captured and documented knowledge turned into explicit rules. Figure 4 depicts the system information flow.



Nr	Model	%	Tested
#34	aaa	95%	Yes/Cleared
#72	bbb	91%	Yes/Cleared
#12	ccc	80%	Yes/Cleared
#19	ddd	77%	Yes/Clear
#88	eee	99%	No

Based on prior case:

#34	aaa	95%	Yes/Cleared

Prototype bracket:
Grip: Positive bending angle - OK
Stiffness: Shortened 2 mm - OK

Comparative grade:
95% similarity with bracket #34.
Improved grip and stiffness.

Testing NOT immediately required.

Figure 4. Architecture of the pilot system (where stage 1 is implemented)

The general principle of the system, and its information flow as depicted in Figure 4, is described as the following steps. The designer enters the desired parameters based on a measured roof profile in a graphically oriented user interface (1). The parametric data is then compared to prior cases in a screening process by CBR (2), where the prior cases are retrieved from a case database (3). A list of results from the CBR screening is then presented showing the level of similarity with prior cases (4). This gives the designer valuable input for choice of prior bracket design for prototyping.

In the second stage (not yet implemented) the designer's choice of prior bracket design will be further supported by logic reasoning through an Expert System or a Decision Support System in which the designers' heuristic knowledge has been captured and implemented (5). This will result in a report on the selected bracket, how it compares to prior designs, and in what way the designer should proceed with the new bracket design (6). When the design is accepted the new case is entered into the case database for future use (7).

The main reasons for this two stage approach combining CBR and an ES or a DSS is added documentation of process decisions steps. This documentation is needed (or even required) for quality control and follow up of different designs. An ES or a DSS ensures that choices are based on the same rules each time, thus eliminating some level of uncertainty and also adding some quality assurance.

6.1 Evaluation based on users' experiences of implemented pilot system

Stage 1 of the pilot system has been implemented and extensively tested and evaluated by the company designers, who also are the intended users of the system. Their main conclusion is that the system solution approach is the right one and that the CBR implementation finds suitable bracket designs among the almost 800 prior cases (each with about 15-25 search parameters). It has even found near duplicate designs which might have been unnecessary and avoided if such a system had been used in the past. With the addition of the more graphically oriented user interface of the operational system, they also feel that the system will serve its intended purpose for the foreseeable future. On system maintenance and future development the designers stated that adding system functionality (added searchable parameters, links to documentations, etc.) is seen as a much easier task than the already performed parameterisation of the brackets, and therefore perceived as highly achievable.

6.2 Pilot system performance

The time savings of using the CBR system are clearly noticeable. There is an estimated time saving of up to 50% in finding suitable candidates for prototyping, and this is without the added user friendliness of the intended operational system's user interface. No apparent positive effects on solution quality could be seen, mainly because a final "manual" decision on which previous design that should be used still is needed. The one possible source of error that was identified is the data that has been manually entered in the database. If there are any parameters that have been incorrectly entered for any of the previous brackets, then those brackets will show up with a lower level of similarity and thus not qualify for prototyping. However, this problem should not arise for any new designs since they are entered into the database using the design's parameterisation for indexing, i.e. the designs are entered into the database based on their actual (physical) geometry.

7. Operational system

The operational system (under evaluation) is based on a graphically oriented interface incorporating CBR functionality linked to a database containing all desired data about bracket designs (Figure 5) [Artursson and Petersson 2006]. The designer is thus presented with a graphical representation of the proposed bracket as he/she is entering the design parameters. The system incorporates editing functionality where the designer can retrieve and edit old cases (if so necessary), propose new designs, perform CBR screening, and add new designs to the database. The system also allows the designer to save images of the proposed bracket design together with similar prior brackets for superimposing on drawings or sketches of the car roof profile for further manual control of the design.

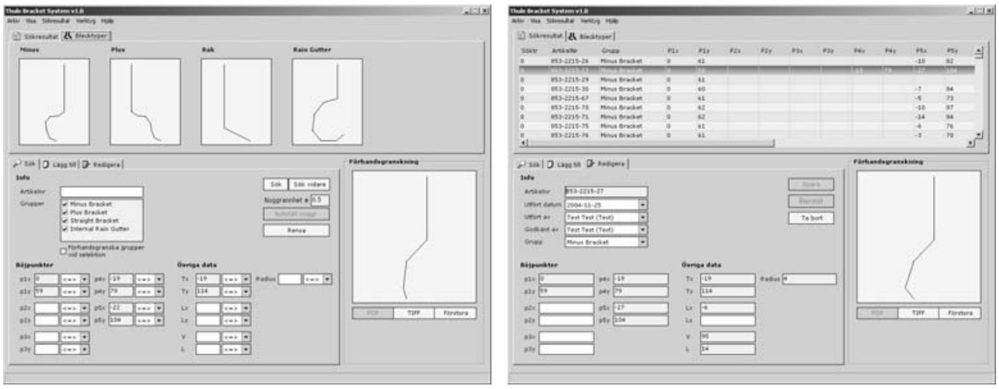


Figure 5. Two user views in the operational system: Topology chooser and CBR screening (left) and presented screening results and database editor (right)

8. Conclusions

By carefully breaking down a process and mapping it and its inherent knowledge to suitable solution strategies, a system specification can be presented. Although it in this case exists several plausible solutions that could meet the problem definition of designing brackets with minimal effort, the choice of CBR is based on the identified main problem of screening prior solutions. An automated variant design system, for example a CAD system incorporating bracket design rules, could just as well have been implemented, but the risk of duplicate designs would then still exist. The development of a pilot system for CBR showed that the chosen approach fulfils the problem definition as well as the company's requirements. Furthermore, the pilot system development and the work of capturing knowledge needed for CBR showed that a variant design approach is a powerful way of adding CBR functionality to common variant design. It also showed that by creating a design parameterisation (intended for variant design), CBR indexing templates that conform to the way the designer describes the designs are acquired automatically, thus eliminating some CBR indexing complications.

9. Future work

The model for mapping process character and domain knowledge to tools and implementations will be further developed and linked to the criteria for design automation development [Cederfeldt 2005]. The next phase in the case project is linking an ES or DSS to the CBR system for added quality of decisions on fixturing brackets for prototyping (briefly described in paragraphs 4.3 and 6). In order to do this, the designers' heuristic knowledge has to be captured and documented. Also a second pilot system will be implemented.

Acknowledgement

This work was conducted within the Vinnova program for "Generative CAE systems," and financial support is gratefully acknowledged. The author would also like to express his gratitude to Thule Sweden AB for information and knowledge about the case of application as well as for helpful discussions. The development of an operational system has been conducted by Andreas Artursson and Erik Petersson (students at the School of Engineering's bachelor program for Information Technology) and their contribution is greatly acknowledged.

References

Andreasen, M.M., "Design Methodology", Journal of Engineering Design, Vol.2, No.4, 1991, pp 321-335.

Artursson, A., Petersson, E., "Thule Bracket System", bachelor thesis to be published at: School of Engineering, Jönköping University, Jönköping, Sweden, 2006.

Browning, T.R., "Use of Dependency Structure Matrices for Product Development Cycle Time Reduction", Proceedings of the Fifth ISPE International Conference on Concurrent Engineering: Research and Applications, Tokyo, Japan, July 15-17, 1998.

Cederfeldt, M., "Planning Design Automation Systems – Criteria and Guidelines", Licentiate thesis, Chalmers University of Technology, Gothenburg, Sweden, 2005.

Cederfeldt, M., Elgh, F., "Design Automation in SMEs – Current State, Potential, Need and Requirements", Proceedings of ICED 05: International Conference on Engineering Design, Melbourne, Australia, August 15-18, 2005.

Cederfeldt, M., "Variant Design Automation - Strategy and Procedure for Storing Design Process Information", Proceedings of DETC2004: ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Salt Lake City, Utah, USA, September 28 - October 2, 2004.

Cederfeldt, M. and Sunnersjö, S., "Solid Modelling with Dimensional and Topological Variability", Proceedings of ICED 03: International Conference on Engineering Design, Stockholm, Sweden, August 19-21, 2003.

Johansson, P., "Knowledge Reuse In The Design Of Steel Connections Using 2d-CAD Drawings", submitted to ITcon, [http://www.itcon.org/], December, 2005.

Johansson, L., "FreeBCR", Public domain license software at: [http://freecbr.sourceforge.net], acc. November 08, 2005.

Li, Z., Liu, M., Ramani, K., "Review of Product Information Retrieval: Representation and Indexing", Proceedings of DETC2004: ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Salt Lake City, Utah, USA, September 28 - October 2, 2004.

Maher M.L., Pu P., "Introduction to the Issues and Applications of Case-Based Reasoning in Design", Issues and Applications of Case-Based Reasoning in Design, Maher M.L., Pu P. (eds), Lawrence Erlbaum Associates, Mahwah, NJ, pp. 1-10, 1997.

MOKA, "Managing Engineering Knowledge", MOKA - Methodology for Knowledge Based Engineering Application, Stokes, M. (ed), Professional Engineering and Publishing Ltd., London, Great Britain, 2001.

Roozenburg, N.F.M. and Eekels, J., "Product Design: Fundamentals and Methods", John Wiley and Sons Ltd., West Sussex, Great Britain, 1995.

Mikael Cederfeldt
Ph.D. Candidate, Lic.Eng., M.Sc.
School of Engineering, Jönköping University, Dep. of Mechanical Engineering
P.O. BOX 1026, 551 11, Jönköping, Sweden
Tel.: +46 36 10 16 71 (telephone)
Email: mikael.cederfeldt@ing.hj.se
URL: http://www.cederfeldt.com/mikael/csed.html