Defining the initial case-base for a CBR operator support system in digital finishing

A methodological knowledge acquisition approach

Leendert W.M. Wienhofen[1] and Bjørn Magnus Mathisen[1,2]

[1] SINTEF ICT, PO Box 4760, Sluppen, NO-7465, Trondheim, Norway
{leendert.wienhofen, bjornmagnus.mathisen} @sintef.no
[2] Department of Computer and Information Science, Norwegian University of Science and Technology, NO-7491 Trondheim, Norway
bjornmm@idi.ntnu.no

Abstract. Case-based reasoning (CBR) literature defines the process of defining a case-base as a hard and time-demanding task though the same literature does not report in detail on how to build your initial case base. The main contribution of this paper is the description of the methods that we used in order to build the initial case-base including the steps taken in order to make sure that the quality of the initial case set is appropriate. We first present the domain and argue why CBR is an appropriate solution for our application. Then we detail how we created the case base and show how the cases are validated.

1 Introduction

Case-based reasoning (CBR) literature defines the process of defining a case-base as a hard and time-demanding task though do not report in detail on how to actually build your initial case base. Öztürk and Tideman say in their 2014 review paper [17]: "Initial population of a case base is a daunting task in classical CBR because it is manually crafted by knowledge engineers who make use of domain experts or written material to extract the case content. ... We believe case grounding problem is the reason why CBR has not seen wide-spread adoption in the industry - because manual extraction of cases from reports and records is costly and time consuming". In this context, we present a knowledge acquisition process that was applied to create an initial set of cases while constructing a CBR system in an industrial setting. We explain the domain in which we applied CBR and argue why it is an appropriate solution for our application. This is followed by a description of a methodological approach for building an initial case base. Revision and validation of the case base and the similarity features are presented in the discussion section.

The main contribution of this paper is the description of the methods that we used in order to build an initial case-base for our CBR system in an industrial domain, including the steps taken in order to make sure that the quality of the initial case set is appropriate.

1.1 Background

The case-based decision support system described in this paper is part of a project that is trying to increase the speed of digital conversion. Digital conversion is the process of cutting or milling various types of materials into shapes, based on a digital design. The speed is to be increased concerning the actual cutting speed as well as the time to shift between different jobs.

This paper will focus on the latter and the main objective, as set forward in the project proposal, is to decrease the time an operator uses between jobs by 80%.

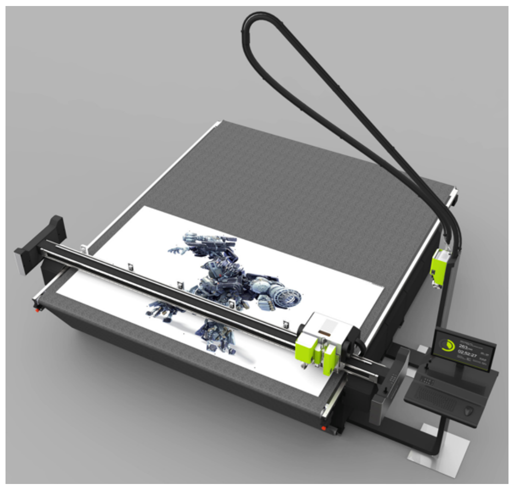


Fig. 1: A digital conversion table from Esko Graphics (Copyright Esko Graphics)

Digital conversion machines (such as shown in Fig. 1), also referred to as cutting tables, offer a plethora of different settings and the intervention is suggested to be an intelligent operator user interface to the conversion machine, based on case-based (CBR) and rule-based reasoning (RBR). By automating parts of the process relating to load shifts, the job for the operator will be easier and faster, with a lower margin for errors compared to the current situation.

The finished system should facilitate and automate learning from past experiences (meaning cutting/milling jobs with settings specific for a design and material) within a specific company. Future work will enable the system to share data between deployments of the system, so that even competing companies can share their experiences without sharing their competitive advantage.

A cutting table is a further development of a flatbed pen plotter, where the pen can be substituted by knives and millings bits, and the drawing paper by other types of material. Operations on the X and Y-axis (given a certain depth and pressure on the Z-axis) vary per material type. The optimal speed and acceleration for a given actuator depend not only on the material type, but also on the vendor (as quality can vary from vendor to vendor), the wear and tear of the actuator as well as the complexity of the design to be cut/milled out from the material, just to name a few. Therefore, these cutting tables have a myriad of settings and require an experienced operator in order to get the best results. Most of the knowledge required for configuring the machine correctly is currently implicit, and knowledge transfer is typically done on a face-to-face basis between operators. Generally speaking, we can say that inexperienced operators do not dare to use the full potential of the table in fear of damaging the materials on which the cutting or milling operation is to be carried out on.

By making domain knowledge explicit in the form of a domain model with instances, an inexperienced user can find similar cases and re-use the settings. In our approach, we take this one step further by applying case-based reasoning, which automatically selects the most applicable case and related setting for the user so that the full potential of the machine can be used.

1.2 Case-based reasoning as an enabler for experience transfer

Based on interviews and observations at companies using digital conversion tables, we conclude that experience is typically not stored in a structured manner and knowledge transfer happens in an informal way between co-workers. Operators of these machines typically learn by doing, and because of this the full potential of a machine is not always reached, especially when operated by inexperienced users. Users report that they are afraid of breaking something when they apply parameters they are unsure of.

In some cases a note with settings is taped on the operator console, though these contain proven "safe settings" for a typical material and tool combination. Another company uses a whiteboard for settings, though it is rarely updated and personnel indicate that they actually do not use the settings that are noted there and rather trust their own feelings concerning the settings. There is no structured means of storing experiences among the companies that have been observed during the case study.

As the working situation is based on a known desired outcome, case based reasoning is an appropriate manner of addressing the problem at hand.

We intend to create a knowledge base where the digital finishing machine retains the settings, material type and other relevant parameters.

1.3 Distributed case-based reasoning

In the digital finishing industry companies use many different material types, some on a more regular basis than others. As this is a very experience-based process, chances are that the proper expertise is not available in all companies. By providing access to case bases created in other (competing) companies, one can draw from the experience.

1.4 Related work

It has been shown [12] that CBR is well suited as a means of decision support for operators in a manufacturing setting. CBR is a form of AI where the decision making support is based on a known outcome. It takes a case (which is the product to be made) as input and tries to find the most similar case in a case base. This means that cases with a similar profile are suggested.

Competing companies can help each other increase their efficacy by sharing case bases can be achieved via distributed case-based reasoning. Distributed CBR has been around for a while and is well described in among others [14,16,18]. However, it seems to be limited to non-competing companies, making knowledge sharing a clear-cut benefit. In order to avoid potential problem with patented designs, we decouple the geometry information from the design, reducing it to an indication of the complexity of the design based on a float where 0.0 is the least complex and 1.0 the most complex. From the technical point of view there is no real difference in the implementation.

Our CBR system will implement explanatory features enabling the operator to choose to either apply the suggested settings or retain the self-chosen setting based on the suggested settings and the corresponding explanation. Based on the interviews, we can state that it is important that the CBR system does not actually make a decision, rather suggests a decision based on the most similar case. This way the system supports the operator in his decision. The explanation helps the operator to understand why a certain proposal has been made by the system and therewith enables to operator to make an informed decision. The fundamental issues of explanations in CBR are described well in [21].

Aamodt and Plaza [2] have formalized Case-based reasoning for purposes of computer reasoning as a four-step process: Retrieve, reuse, revise and retain.

In order to retrieve a case, one needs to identify features, collect descriptors, interpret problem and infer descriptors. Prior to being able to do that, one needs to have a case base. It is of importance that the right features are extracted from a case as it will be the fundament for further reasoning.Case acquisition is often manually intensive. According to [10], a manually intensive approach for storing experiences of individuals has been widely used in many CBR applications. The general approach –as case bases are very domain specific- is to talk to a domain expert and extract which parameters are of most value and use that as a starting point. Getting the full picture, however, requires talking to more than one expert and an iterative approach in order to make sure that the right parameters are used for the case base. In the following sections we present such an approach.

The case quality needs to be safeguarded as the case base must contain a representative set of problem solution pairs from the domain at the initial stage of the CBR system. At the same time we need to ensure that the case-base yields high quality results. Little attention has been given to case-quality in the available literature, and therefore the CBR expands without inspecting itself [29]. We want to address the quality problem by making sure that both the initial case information as well as the cases to be learned will be initiated and checked by humans.

If the case template is wrong, the result will be wrong. There is a need to understand how the case template is defined, in practice. Next we will see who has addressed this central issue and what they can tell us about how to do address it.

Öztürk and Tideman's [17] statement "We believe case grounding problem is the reason why CBR has not seen wide-spread adoption in the industry - because manual extraction of cases from reports and records is costly and time consuming" is one of the main reasons why we report our approach related to knowledge acquisition. We do agree that it is a time consuming effort, though, when consulting existing literature for knowledge acquisition for CBR to learn how to extract and categorize the relevant information, we did not find any clear guidelines or methodological descriptions for case grounding. This might be an additional reason to why CBR has not seen a widespread adoption in the industry.

The recent trend is to (semi)-automate the case acquisition process [10,15,23, 24,29]. The approach sketched by [10] is based on initiating the case base with random values, though still based on a formalized data-sheet template for case representation. However, there is no mentioning on how the template was established (the assumption is that domain experts have been asked). They state: "Case engineering is among the most complicated and costly tasks in implementing a case-based reasoning system".

The cases that are part of the case-base are supposed to yield solutions to the problems with minimal adaptation or human input. This is desirable as otherwise the major usefulness of a CBR system to reuse existing knowledge would be substantially harmed [8]. This implies that the case base must support this type of knowledge.

Richter [19] describes knowledge containers as keepers of case information. The first requirement is that the case base should only contain cases (p, s) where the utility of s is maximal or at least very good for the problem p. This is knowledge contained in the individual cases.

The case acquisition process itself, meaning the initiation of a case base, is not described though 4 different sources are mentioned in Richter's invited talk at ICCBR in 1995 [27]: domain knowledge, cases, similarity knowledge and adaptation knowledge.

The domain knowledge is what fills the template which can be used for matching cases. According to, template retrieval is similar to SQL queries in databases, where all cases fitting a template of parameters are retrieved. The main merit of using of template retrieval is that the faster retrieval and high currency by prevents irrelevant case from being considered in similarity matching.

Aamodt [1] described a framework for modeling the knowledge contents of CBR systems based on Richters knowledge containers. The model suggests decomposition in three perspectives. The power of using three perspectives (tasks, methods, and models) for knowledge level modeling lies in the interaction between the perspectives, and the constraints they impose on each other. However, there is no description on how to initiate the case base.

Cordier et al [6] state that when there is a lack of domain knowledge, the system may infer a solution that is correct with respect to the knowledge base but not with the real world: making the results invalid in the real world. The FRAKAS system [7] is an approach for interactive domain knowledge acquisition. Learning takes place during the use of the system and aims at acquiring domain or adaptation knowledge. The evaluation of the adapted solution may highlight that it does not meet the requirements of the target problem. In this situation, a reasoning failure occurs and is processed by a learning process. The expert is involved in the process of identifying inconsistent parts of the solution which helps to augment the knowledge base. The expert is involved in a simple manner to point out faulty knowledge and he/she may provide a textual explanation of the identified error to support complementary off-line knowledge acquisition. The approach defined here is interesting with respect to further population of a knowledge base, and a similar approach can be used both to fill the knowledge base once a basic case set has been established as well as a part of the regular learning curve (one of the 4 R's).

As in the CBR literature little is mentioned on how to populate the initial case-bases, we turn to the cognitive science domain where the fundamental concept is that "thinking can best be understood in terms of representational structures in the mind and computational procedures that operate on those structures."[3] Cognitive science in turn is related to the knowledge management and knowledge engineering field where extracting information from experts in order to create the foundation for among others expert systems has matured over the past decades. Watson [28] does describe how to apply knowledge management for CBR, however, it lacks detail on the establishing of the case base. Cognitive science is also mentioned in [20] and regarding representation of knowledge they state the following: "more generic issues of knowledge representation are seldom addressed". Followed by "The case base plays a special role because the cases can be entered without understanding them. The main point is that knowledge can be shifted between containers (their content is not invariant), which can be modeled using a learning process. In addition, the shifting can be done manually without the support of a learning method".

Our guiding motivating hypothesis is that an operator support system based on case-based reasoning can help speed up the cutting/milling process while maintaining satisfactory quality results.

As the intention is to create an operator support system using CBR, we need a formal representation of the cases. By creating a domain model, we separate domain knowledge from the operational knowledge, enable the reuse of domain knowledge and make domain assumptions explicit. Once the domain model is in place, we can also populate the case base with relevant cases. Finding out what a relevant case is and what needs to be represented in the domain model go hand in hand. Our second hypothesis is that a user-centered iterative approach is a

[3] Thagard, Paul, Cognitive Science, The Stanford Encyclopedia of Philosophy (Fall 2008 Edition), Edward N. Zalta (ed.).

good method to create a good formal representation as a basis for the operator support system.

2 Method

While many publications (i.e. [3–5, 9, 26]) do describe the knowledge acquisition approach for their domain, most do it on a relatively technical level. We have applied several methods for knowledge acquisition and the focus has been on a user-centered iterative process. In the subsections below we give a brief explanation of these methods and highlight our experience with these forms of knowledge acquisition.

2.1 Research method

To systematically guide our research in this project we used the design science research method is used according to [11], as depicted in Fig. 2. The research environment consists of machine supplier experts, as well as machine operators. The research is driven by the need to use the machines in an optimal manner, with the assumed outcome a more optimal operation and therewith cost reductions. The knowledge base is based on the existing literature on CBR and knowledge acquisition as well our own findings.

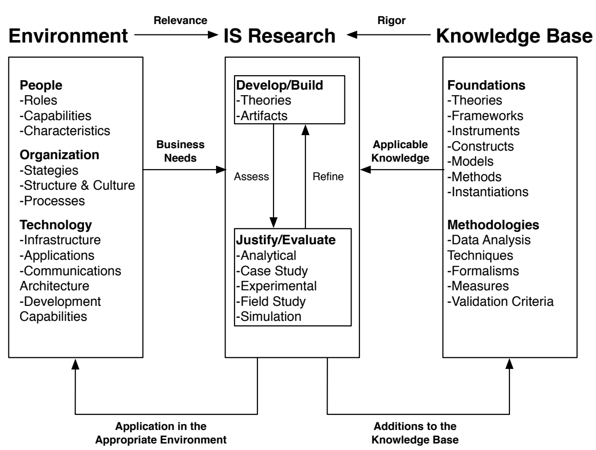


Fig. 2: Conceptual framework for IS research [11]

RQ1: What is the effect of introducing an expert system based on CBR on the effectiveness of operators?

RQ2: What is the effect of introducing a distributed expert system based on CBR?

User-centered design is conducted prior to the development of complex systems to ensure deep understanding of user and stakeholder roles. The aim is to ensure that system designed support the daily work of end users and the role of stakeholders [13, 22]

We have applied user-centered design in all activities in the iterative process of assessing and refining our artifacts, adding it to our knowledge base. The activities are carried out in close cooperation with real stakeholders by means of various methods for data collection, as described in the sections below.

2.2 Data collection

In our study we focus on a single manufacturer of digital conversion tables. The study is based on design science research and evaluation research and has been implemented at 3 different locations that represent a typical customer of this manufacturer.

The intervention is the introduction of a distributed case-based decision support system to support operators to make the right decisions quicker and therewith both reduce the number of errors and speed up the full process of job shifting.

The artifact to be created for this intervention is a research challenge itself as populating a knowledge base is a non-trivial task. A step wise approach for populating a CBR knowledge base will be developed and the effect will be tested.

Some of the needed information can be retrieved from logs, though this is a non-validated information source.

The study is divided in two parts: data collection participants and intervention participants. The data collection methods are described in the sub-sections below.

In both cases the population is recruited the manufacturer of digital conversion tables- and the inclusion criterion is that the participant is currently a customer operating digital conversion machines. We focus on the data collection part in this paper.

For the data collection, we focus around the following questions

DC1: Which information to extract from the operators?
DC2: What is/are the bottleneck(s) in the load shift?
DC3: Which factors impact the time used?
DC4: What is the mean time?
DC5: Does the knowledge of an operator impact the operation? And in what way?
DC6: How much information are companies willing to share with competitors?
DC7: How and when to present suggestions from the expert system to operators?

In the sub-sections below we first present which methods we have used for the data collection and in section 3 we provide the results of the activities.

Observation The first data gathering activity was based on observations. The intention was to form a structure for later interviews and the first subject was asked to explain (while preparing and operating the cutting table) what he was doing and why he was doing it this way. The observer did not interfere with the process.

Semi-structured interviews We have conducted interviews at digital finishing companies in Norway, Belgium and The Netherlands. The interview subjects were mainly cutting table operators, though also managers/owners. As the companies were relatively small, the latter category also in all cases were table operators, yet not on a daily basis. The interview questions were based on the results from the observation session and have been expanded based on finding between the interviews. We used a set with main questions and expanded while commencing the interview.

Questionnaire We have developed a questionnaire in order to map the time operators use when operating the machines. It was sent to 100 digital finishing companies throughout the world.

Workshops The technology provider catered for a workshop with employees with a computer science background. During this workshop technical boundaries were explored and details regarding the integration of the operator support system discussed.

Re-use of available data We have gained access to a product guide describing which tools can be used for which materials, and for some of these also a set of settings for certain material types. However, the settings are relatively conservative as they pertain to a material family. Specific materials use material specific settings which can be much faster than the material family setting. For the most used specific materials, specific settings are available. Also an operator manuals of the current Esko machines with i-cut software has been used as an information source.

3 Results

3.1 Case study: As is situation

Input for the study uses the data gathering methods described above, in addition, one of the researchers took a table operator course to get a real hands-on feel of using the system.

In Fig.3 you see the repetitive and cyclic process of enhancing the input, which can be mapped to the IS research part of Fig. 2; both Develop/build and justify/evaluate to ensure both relevance and rigor.

The methods have been applied to digital finishing companies in Norway, Belgium and The Netherlands.

Unfortunately, the response rate for the questionnaire was so low that we were unable to use the results as a pinpoint for the average type of operator and other information regarding machine use.

From the observation and interview activities, we learned that machine operation to a large degree is completely experience based and that the experience transfer is sub-optimal. Some factors that influence the choices are the quality of the material that is used, the wear and tear of the used actuator, the desired output quality (not all customers demand a high quality finish) as well as the time available between jobs. An ideal situation according to one of the shop managers is that the machine is in use continuously. We did a test using optimal speed settings with new actuators and high quality material vs the regular settings with a new actuator and high quality material. We found that the cutting speed in this specific case was 13 minutes vs 22 minutes. This supported the assumption that the operators do not use the optimal settings and that an operator support system indeed can be useful. For this specific case, relating to RQ1, we can state that there is a good effect in using the operator support system recommendations.

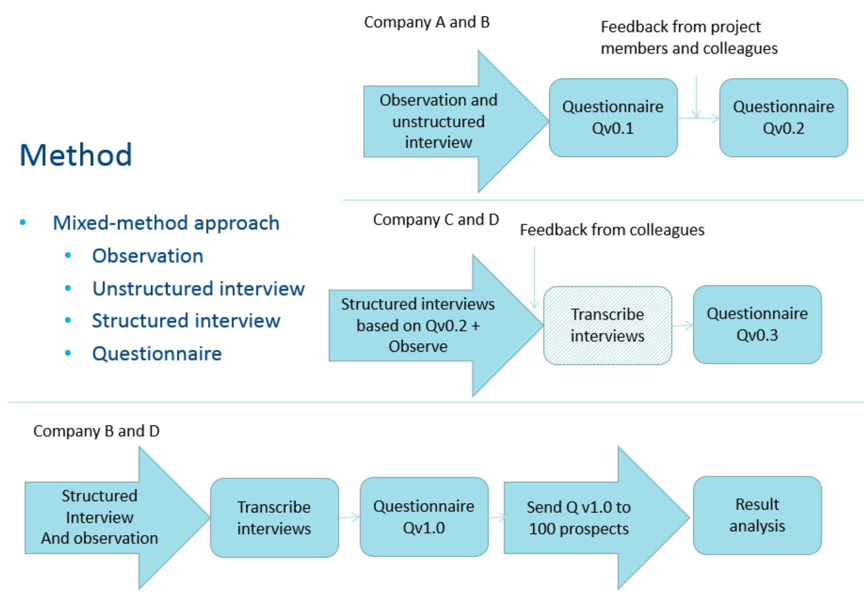


Fig. 3: Knowledge acquisition

Knowing the type of information the operators wish to use and how they wish to use it, we discussed the technical boundaries with the table and cutting table software provider. We gained access to subsets of the required information required to create an operator support system. All of the gathered information has been structured into a domain model. See the next subsection for more details.

3.2 Domain model

In order to model the domain, we need to map domain knowledge (for an impression, see Fig. 4 a). The main parameters that need to be contained can be summarized as such:

A cutting/milling ***job*** *is performed by an* ***operator*** *on a* ***cutting table*** *which uses a* ***tool set*** *with different* ***actuators*** *on a* ***material type*** *following* ***patterns*** *stemming from a* ***design****. Considerations regarding the* ***speed*** *and* ***quality*** *of the job are done by the* ***operator*** *based on* previous experiences *and* ***customer demands****.*

The previous sentences describe what the domain model needs to include on an overall level. In short, it needs to include all relevant information for an operator to be able to do a job in the fastest possible manner or with the highest quality possible. These two are not always mutually exclusive, though high speeds can sometimes lead to a lower end-product quality. In some cases, the lower quality is still within the quality assurance threshold.

- Some questions that the operator support system needs to be able to help answer are: Which settings should I avoid to use?
- What is the most optimal setting for this particular job with regards to either quality or speed?
- What is the maximum speed I can use?
- Will these settings break stuff?
- Which settings should I change?
- Will this actuator (bit/blade) work with this material?
- What are the limitations of this tool applied on this material?

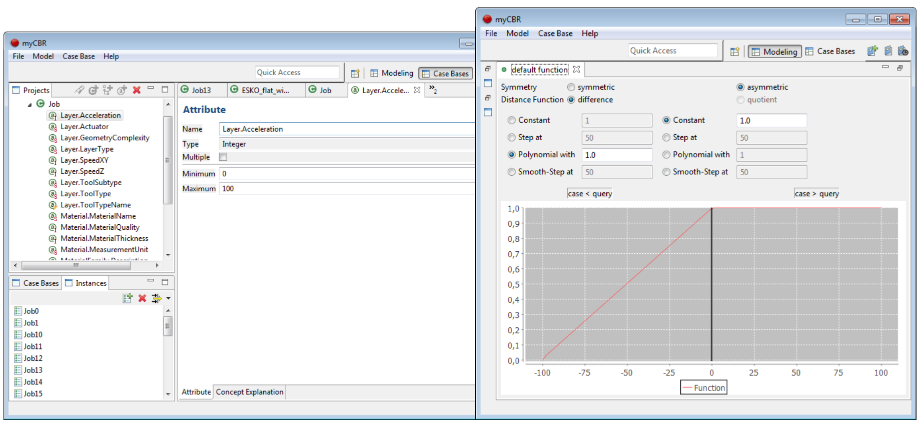
These questions imply that we need to know about the properties of the materials, design, tools and table. During the domain knowledge gathering process, we have identified the relevant terms to include in the domain model. Due to space restrictions, we do not include the domain model in this paper, though some of it can be seen in the screenshots from MyCBR.

One of the results from the interviews shows that operators are more likely to trust a recommendation if an explanation is given. If the settings are presented following a pattern such as “in a similar case we have successfully applied the following settings with a satisfactory quality” followed by a question if the operator wishes to use these settings instead, the operators responded positively. However, without such explanation, the operators would not simply accept new settings.

3.3 CBR

We have applied the domain model and created a CBR system prototype using MyCBR [25].

The initial case base has been made in close cooperation with experts from the company. Instances with proven cases in different levels of aptness have been entered. It is important to note that these cases are based on material family and not a specific instance of the material itself. As properties are supplier specific, different settings should be used. These settings will during the course of the use of the system be formed as cases.



(a) (b)

Fig. 4: A screenshot of the domain model (a) and an example of a similarity measure (b).

The similarity features (Fig. 4 b) are based on conversations with a tool and material experts. Each specific material combined with specific actuators have specific settings, also pertaining to the complexity of the output to be generated.

3.4 Validation

Testing has been done based on the different cases with each their rating. For material types two or more different cases have been entered in the initial case base, including an indication in the aptness. Similarity values and weights have been tuned in order to get the closest case to match. This was later tried with new cases and the results were satisfactory. A screenshot of matching results is shown in Fig. 5

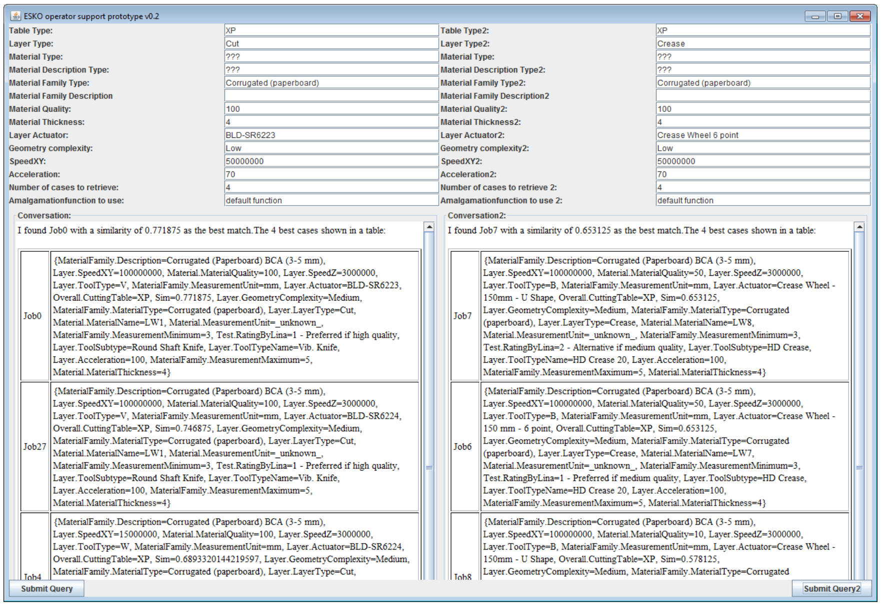


Fig. 5: Matching results screenshot

4 Discussion and lessons learned

The variety of knowledge elicitation methods we have used and the variety of companies visited may seem like an too rigorous information gathering, though we feel that in our case this was the right thing to do. It is time and resource demanding, though by presenting the various approaches, we hope to contribute to the knowledge gap that seems to exist concerning creating an initial case-base. Different situations cater for different methods of knowledge elicitation, and in many cases, a less rigorous approach might be sufficient. Creating a sound and valid foundation for the case template and case base is resources demanding. However creating a CBR system that is neither valid or useful is even more resource demanding. In general we can recommend to talk to the system owner and a variety systems users multiple times in order to best understand the problem at stake and validate that the researchers (CBR system builders) really understand the problem that the CBR system is to solve in a manner that is useful for the end-users.

5 Conclusion and further work

This study has presented a use case for how to create a CBR system with focus on building the initial case base, and the case template or domain model. To create grounded basis for our CBR system, case template and domain model we; observed the operators, performed interviews with the operators, organized interactive workshops with the operators, collected questionnaires and utilized available product data.

These data sources all went into the design of the case base, case template and domain model. An initial validation at one of the companies shows that operators recognize and understand the CBR system inputs and outputs. This serves as an example use case that works toward solving the problems highlighted by [17]. With regards to the main motivating hypothesis of this work initial tests also shows increase in the operation of the machine that is augmented by the CBR system. In the next part of this project the system will be tested more thoroughly in terms of performance increase in the target domain of the CBR system. In addition, we will develop a method for abstracting and extracting high level knowledge from cases to be sent into a distributed case base to ensure both knowledge sharing across competing stakeholders while not disclosing competitive advantages.

6 Acknowledgments

The authors gratefully acknowledge the Norwegian Research Council and the BIA program for financial support of the project (partially through grant 235427) as well as the participating case companies, which together enabled this study.

References

1. Aamodt, A.: Modeling the knowledge contents of CBR systems (2001)
2. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Communications 7(1), 39–59 (1994)
3. Bach, K.: Knowledge Acquisition for Case-Based Reasoning Systems. Verlag Dr Hut (2012)
4. Bergmann, Ralph, e.a.: Developing industrial case-based reasoning applications: The INRECA methodology. Springer (2003)
5. Bergmann, R.: Experience Management Foundations, Development Methodology, and Internet-Based Applications. Springer-Verlag Berlin Heidelberg (2002)
6. Cordier, A., Fuchs, B., Lieber, J., Mille, A.: Failure analysis for domain knowledge acquisition in a knowledge-intensive CBR system (2007)
7. Cordier, A.: Interactive and Opportunistic Knowledge Acquisition in Case-Based Reasoning. Thesis (2008), https://tel.archives-ouvertes.fr/tel-00364368
8. Cunningham, P.: CBR: Strengths and weaknesses, pp. 517–524. Lecture Notes in Computer Science, Springer Berlin Heidelberg (1998), http://link.springer.com/chapter/10.1007/3-540-64574-8_437
9. Díaz-Agudo, B., González-Calero, P.A., Recio-García, J.A., Sánchez-Ruiz-Granados, A.A.: Building CBR systems with jcolibri. Science of Computer Programming 69(1), 68–75 (2007)
10. Dufour-Lussier, V., Le Ber, F., Lieber, J., Nauer, E.: Automatic case acquisition from texts for process-oriented case-based reasoning. Information Systems 40, 153–167 (2014), http://www.sciencedirect.com/science/article/pii/S0306437912001573

11. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Quarterly 28(1) (2004)
12. Hinkle, D., Toomey, C.: Applying case-based reasoning to manufacturing. AI magazine 16(1), 65 (1995), http://www.aaai.org/ojs/index.php/aimagazine/article/viewArticle/1124
13. Kubie, J., Melkus, L.A., Johnson, R.C., Flanagan, G.a.: User-centred design. CRC Press, Inc., Boca Raton, FL, USA, 7th edn. (1999)
14. Leake, D.B., Sooriamurthi, R.: When two case bases are better than one: Exploiting multiple case bases (2001)
15. Manzoor, J., Asif, S., Masud, M., Khan, M.J.: Automatic case generation for case-based reasoning systems using genetic algorithms (2012)
16. Nagendra Prasad, M.V., Lander, S.E., Lesser, V.R.: On retrieval and reasoning in distributed case bases (1995)
17. Öztürk, P., Tidemann, A.: A review of case-based reasoning in cognition–action continuum: a step toward bridging symbolic and non-symbolic artificial intelligence. The Knowledge Engineering Review 29(01), 51–77 (2014), http://journals.cambridge.org/article_S0269888913000076
18. Plaza, E., McGinty, L.: Distributed case-based reasoning. The Knowledge Engineering Review 20(03), 261–265 (2005), http://journals.cambridge.org/article_S0269888906000683
19. Richter, M.M.: Knowledge containers. Readings in Case-Based Reasoning Morgan Kaufmann Publishers (2003)
20. Richter, M.M., Aamodt, A.: Case-based reasoning foundations. The Knowledge Engineering Review 20(03), 203–207 (2005), http://journals.cambridge.org/action/displayAbstract?fromPage=online&aid=435273&fileId=S0269888906000695
21. Roth-Berghofer, T.R.: Explanations and Case-Based Reasoning: Foundational issues, pp. 389–403. Springer-Verlag, September 2004
22. Shluzas, L.A., Steinert, M., Katila, R.: User-centered innovation for the design and development of complex products and systems. In: Design Thinking Research, pp. 135–149. Springer (2014)
23. Shokouhi, S.V., Aamodt, A., Skalle, P.: A semi-automatic method for case acquisition in CBR a study in oil well drilling (2010)
24. Shokouhi, S.V., Skalle, P., Aamodt, A.: An overview of case-based reasoning applications in drilling engineering. Artificial Intelligence Review 41(3), 317–329 (2014)
25. Stahl, A., Roth-Berghofer, T.: Rapid prototyping of CBR applications with the open source tool myCBR. Springer Verlag (2008)
26. Tautz, C.: Costumizing Software Engineering Experience Management Systems to Organizational Needs. Fraunhofer-IRB-Verlag (2000)
27. Veloso, M., Aamodt, A.: Case-Based Reasoning Research and Development: First International Conference, ICCBR-95, Sesimbra, Portugal, October 23-26, 1995. Proceeding, vol. 1010. Springer Science & Business Media (1995), http://www.springer.com/computer/ai/book/978-3-540-60598-0
28. Watson, I.: Applying Knowledge Management: Techniques for Building Corporate Memories. Morgan Kaufmann (2003)
29. Yang, C., Farley, B., Orchard, B.: Automated case creation and management for diagnostic CBR systems. Applied Intelligence 28(1), 17–28 (2008)