In: *Wirtschaftsinformatik* 38, 1/96, special issue on case-based decision support (edited by D. Ehrenberg), 6-14, 1996

Decision Support for Case-Based Applications

Klaus-Dieter Althoff[1], Brigitte Bartsch-Spörl[2]

[1]Centre for Learning Systems and Applications (LSA)
Department of Computer Science
University of Kaiserslautern
67653 Kaiserslautern
Email: althoff@informatik.uni-kl.de

[2]BSR Consulting GmbH
Wirtstraße 38
81539 München
Email: brigitte@bsr-consulting.de

Keywords: case-based reasoning, business applications, decision criteria, analytic tasks, synthetic tasks, development methodology

Abstract: This paper aims at helping to decide whether case-based reasoning (CBR) is an appropriate problem-solving approach for particular business application needs.

It starts with a short introduction to case-based reasoning. In its main part, the paper deals with criteria that have to be fulfilled by application domains that are appropriate for being tackled by a CBR approach. For this purpose, we present a structured list of decisive questions and interpret the consequences of possible answers to these questions. Finally, we give an outlook to further work in the direction of making CBR approaches even more user-friendly for non-expert users.

1 Introduction

Case-based reasoning has recently become a raising star in knowledge-based decision support technology [EhSc96]. It is especially appealing to those professionals who solve problems by recalling what they did in similar situations that happened in the past.

The first generation of commercial tools for making the development of CBR applications easier appeared on the market in the early 90's. Meanwhile, these tools had to undergo a selection process and those that have survived now develop further releases that have typical second generation features like ease of integration with other software tools like data bases, spreadsheets, graphical user interface tools etc. and also with other applications via standardised ways of calling remote procedures and exchanging objects.

Moreover, CBR has been used to create numerous applications in a wide range of domains, including financial analysis, risk assessment, technical maintenance, process control, quality control, medical diagnosis, software support systems, forecasting, planning, design, classification of objects, photo-interpretation, and real estate appraisal.

This paper mainly concentrates on the topic of how to determine the suitability of the CBR approach for business applications. We will neither lay much stress on nor ignore the tool selection and tool usage topic completely. At those points where we speak about tools we try to concentrate on their generic features and not so much on specific implementations that are going to be replaced by a different version rather quickly.

Due to the restricted length of this paper, we are not able to give a state-of-the-art of CBR (for this purpose, we recommend [AaPl94]) and we even have to neglect approaches that directly concern issues mentioned in this paper. But nevertheless we hope that the paper will be worthwhile to read for practitioners without too much scientific background in CBR and in decision theory.

2 Case-Based Reasoning

Case-based reasoning can be regarded both as a cognitively sound modelling approach for explaining human problem solving in domains where experience plays an important role [StJa90] and as a software engineering approach for how to model and implement decision support systems that are able to use past experiences for suggesting solutions or making predictions [BaBa95].

Currently these are the most predominant views on the matter - but there are even more. Case-based reasoning has brought up interesting mathematical and decision-theoretic questions [Ri92] and in particular the learning issues stimulate significant investigations in the learning prerequisites and capabilities of different kinds of case-based approaches [AaAl93].

For this paper we will adopt the view that case-based reasoning is an approach to problem solving that is by no means new and widely used by human experts in experience-intensive application domains. Since about ten years [KoSi85, Ba87], the artificial intelligence community has begun to recognise that this approach can make their life easier in case they want to build, deliver and maintain systems that are of practical use. Additionally, some early success stories of CBR systems that bring impressive amounts of monetary payback to their users have raised a lot of interest in rather different application fields and have motivated and continue to motivate quite a number of system development efforts.

So the time has come to generalise the experiences from a steadily increasing number of successful CBR projects and to make this experience reusable for others. It is primarily this goal that brought together the authors of this paper who stand for a major part of the experiences coming out of two big and particularly innovative CBR research and application development projects currently underway in Europe, namely INRECA [AuWe95] and FABEL [BA95].

2.1 A PROCESS MODEL FOR CASE-BASED REASONING

The most popular process model for CBR is shown in figure 1 and used in the following section for going a little bit more into the details of what case-based reasoning essentially does.

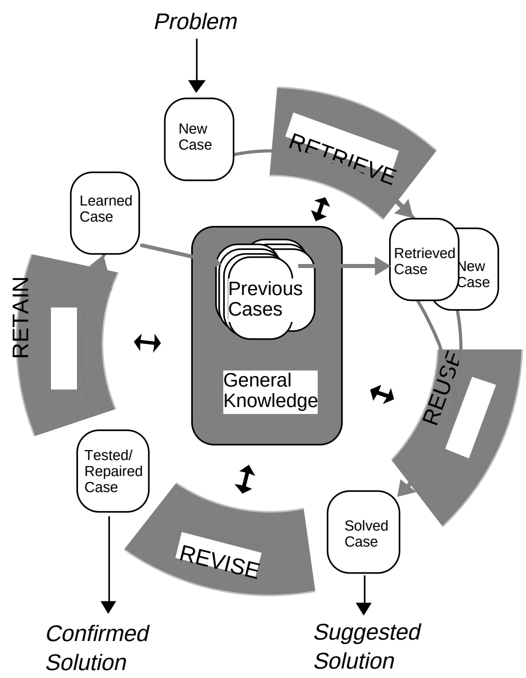

Fig. 1. The case-based reasoning cycle [AaPl94]

The event that triggers the whole process of problem-solving is the appearance of a new problem, e.g. via a call from a customer on the telephone. The first sub task is now to **RETRIEVE** one or more similar cases from the case library where former experiences are stored in the form of previous cases. A case consists at minimum of a problem description and a solution description. For this first retrieval step it is necessary to know what similarity of cases with respect to the purpose of the system means and to have an effective procedure that goes through the case library and brings back a set of similar cases.

The main hypothesis behind case-based reasoning is quite simply "similar problems have similar solutions" - or put the other way round - "you can reuse the solution of a similar problem in order to solve your actual problem". Based on this hypothesis the reuse of the solution of the most similar problem is a very efficient and straightforward way to come to a suggested solution for the actual problem. This step is called **REUSE**.

But in reality it is not always so easy to come to an appropriate and reliable solution. Therefore, it has proven to be a recommendable approach to evaluate the suggested solution from the reuse step, e.g. for its compatibility with a certain amount of general knowledge that is available and to **REVISE** it just as much as necessary. After having this adapted or repaired solution applied to the actual problem, a second step of revision may be in place before the user arrives at a confirmed solution.

The last step stores valuable confirmed solutions for further reuse. This step is called **RETAIN** and completes the experience feedback loop that is a necessary prerequisite for enabling a system to learn from experience.

2.2 KEY ISSUES IN BUILDING A CBR SYSTEM

On the basis of the process model described above it is now rather easy to identify what are the key issues in building a useful and usable CBR system:

First of all, it is necessary to have or to be able to acquire enough cases to achieve a representative coverage of the application domain.

Next, it is essential to represent the cases in a way that is storable in a computer memory, accessible by a software programme and capable to capture the true meaning of a case.

In order to do efficient case retrieval, it is important to have or to develop an indexing procedure that works automatically and brings back a certain number of similar cases in a rather short time.

The programme that assesses the similarity between the actual and the stored cases problem description has to be both precise and robust enough to enable the taking over of solutions without too much risk.

If the domain is of a kind where it makes sense to adapt suggested solutions to the actual situation then another key issue is the ability of the CBR system to carry out the necessary adaptation steps.

Last but not least, CBR systems usually come together with organisational changes. Therefore, it is of major importance to achieve a proper integration of the system into mostly new or changed business processes and to get the CBR system accepted and properly used by all persons in an organisation that are expected to benefit from its use.

3 Decision Support Criteria for Developing CBR Applications

The issue of system development is tightly connected with the notion of evaluation.

Any fielded decision support system needs to be evaluated with respect to a list of evaluation criteria in order to make sure that it meets at least the most important requirements of the application task at hand.

If the system is to be based on a commercial tool then it is worthwhile to make an evaluation of several candidate tools before the right decision for one of these tools - or in the extreme case against the intended tool usage - can be made [DrMo92].

Finally, if there is a choice among different development methodologies then it may be necessary to make another decision for the most appropriate among these methodologies [Aa94].

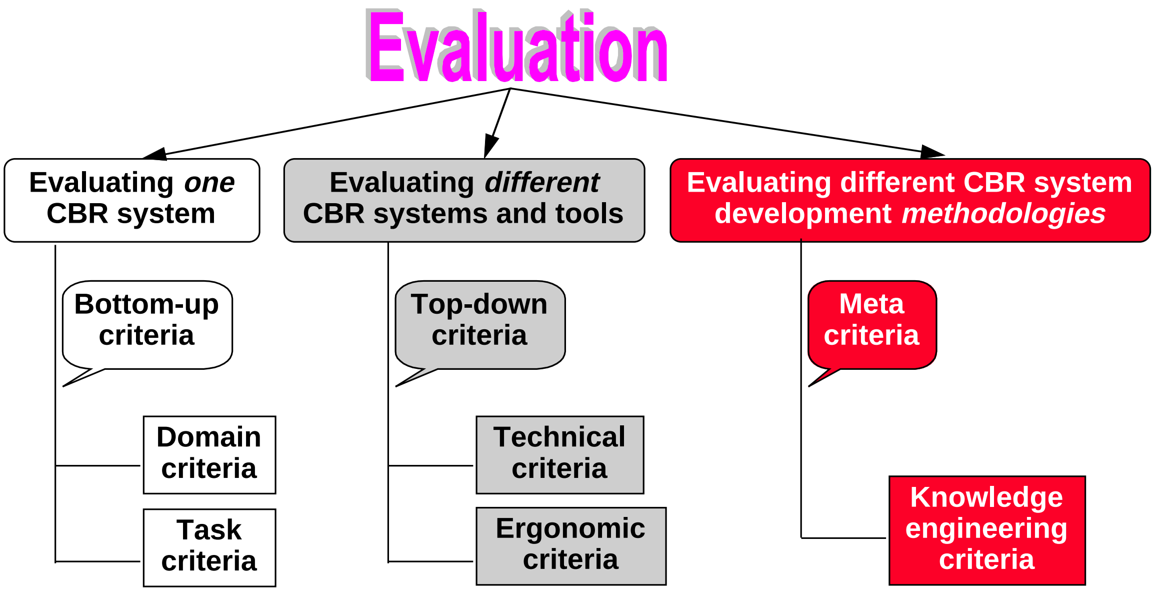

Fig. 2. Different views on evaluating CBR systems

Evaluation is an important issue for every scientific field and a necessity for an emerging software technology like CBR [Co89, Ah94, Al95]. In this paper, we adopt an integrative view on CBR system evaluation that comprises several important aspects from currently known views and thus covers all relevant aspects (cf. fig. 2):

- Domain and application task oriented criteria (e.g., size, theory strength, openness)

  These criteria are important in order to describe the requirements that arise from the application domain and the application task as well. They have to be satisfied whenever it is necessary to ensure that an application problem is solvable by CBR [Be93].
- Technically and ergonomically oriented criteria (e.g., case and knowledge representation, similarity assessment, user acceptance)

  These criteria are used to describe both the potential and the actual capabilities of CBR systems and allow the comparison of different systems.
- Knowledge engineering criteria (e.g., ease of use of the methodology, development phases, available tools)

  These criteria are used to describe the basic characteristics of different development methodologies and help to select an appropriate one.

Methodologies are especially useful in order to increase the efficiency, reliability, maintainability, and modifiability of systems. Examples for knowledge-based system building methodologies are e.g. Components of Expertise [St90], CommonKads [WiVe93] or Generic Tasks [ChJo93].

A methodology depends to a high degree on the above mentioned domain/task and technical/ergonomic criteria. This leads to the insight that it is favourable to develop CBR systems based on explicitly known patterns of relationships between domain/task and technical/ergonomic criteria. In practice, such knowledge has been used implicitly for a long time. More recent work tries to make it more explicit and directly use it within CBR system development [AlAa95].

We would not like to call the combination of criteria, as presented in this paper, a methodology but it is a set of *decision support criteria* for CBR application development (cf. fig. 3).

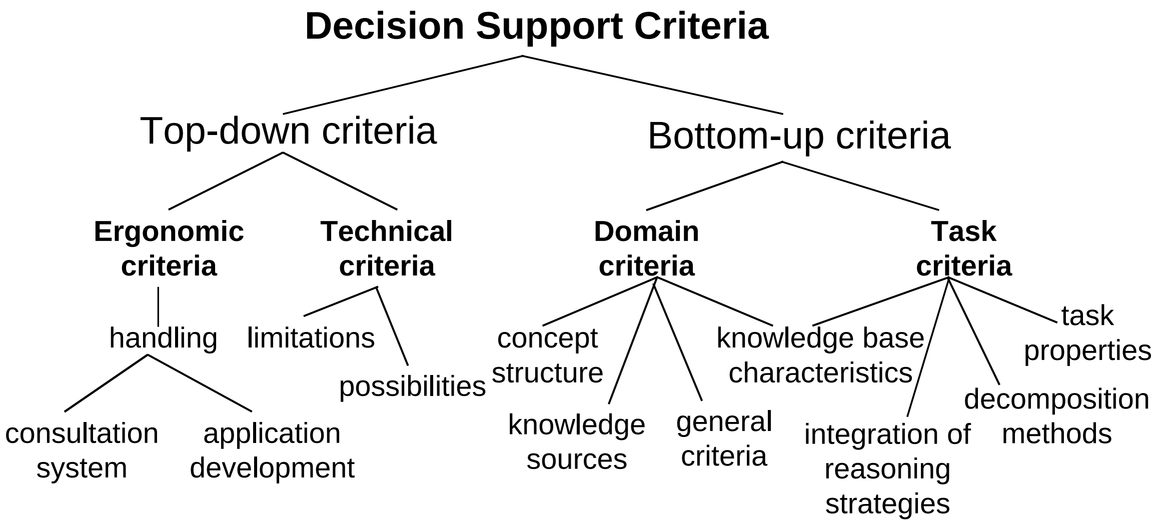

Fig. 3. Decision support criteria for CBR system development

4 On the Appropriateness of Case-Based Reasoning for Business Applications

4.1 CASES

The most basic notion in CBR is, of course, the notion of a case. CBR technology can only be applied if cases are available. If there exists a natural notion of a case in the domain under consideration then this is a valuable hint that CBR may be applicable.

Another important characteristic is whether the domain experts directly interpret their cases - in contrast to a situation where they mostly reason on the basis of general knowledge.

For using CBR, the interpretation of cases needs an operational notion of similarity. Does there any natural notion of similarity exist in the domain, or does at least a notion of neighbourhood exist? Otherwise, it might be difficult to implement an effective similarity measure.

In general, cases are interesting and important sources of knowledge. Johnston [Jo94] pointed out that they should be viewed as a part of an „institutional memory“, i.e. collective experience within an organisation that needs to be preserved and updated. While in medicine and law this is already current practise at least up to a certain degree, in science, technology, and business cases are often considered to be less important. Especially in new and not yet orderly structured domains, cases are not seldom the primary source of knowledge to solve a certain task. They can also be used as exceptions. Then general knowledge is used for „straight-forward“ problem solving whereas the cases are used to cover all known exceptions.

Independent from their role as a knowledge source, cases can be seen and modelled either as flat feature vectors or as complex structured objects. They can include symbolic descriptions as well as multimedia information. Cases can be represented and processed in many different ways, mostly depending on the complexity of the case content and the application task.

The simplest way for doing case retrieval is sequential search. Clearly, this cannot be recommended for larger amounts of cases. Therefore, more complex problems need one or more index structures to be built and cases are then searched by traversing the respective index structures.

For every candidate case found in the case library, its similarity with the actual case must be determined. This can be either simply computed from a similarity function or evaluated from the „closeness“ within a graph structure or be driven by more complicated processes involving similarity functions, graph structures, and additional general knowledge.

4.2 CASE-BASED REASONING FOR ANALYTIC TASKS

The major part of the business problems belongs to the so-called analytic tasks. This term is used for problems that consist of analysing a situation and trying to map this situation to a classification, a diagnosis or a prefabricated suggestion for a solution.

This class of problems comprises all tasks dealing with object classification and identification, monitoring, technical and medical diagnosis, therapy selection and repair, data analysis and prediction, advice giving, help desk, service support, information retrieval etc. According to their underlying complexity we subdivide analytic tasks into the following three groups of problems (cf. fig. 4).

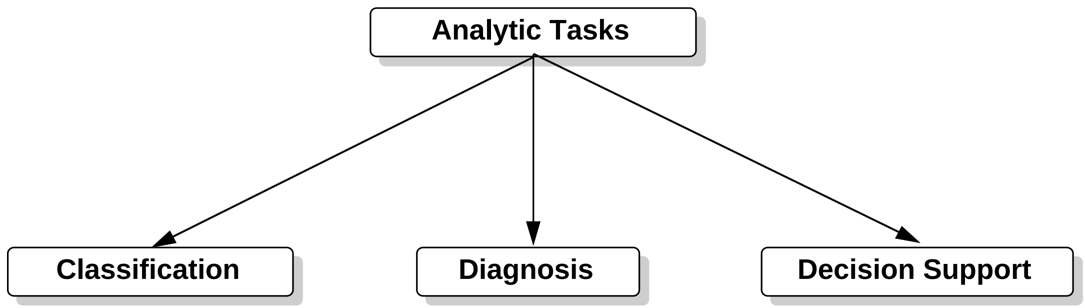

Fig. 4. Structuring analytic tasks (adapted from [We95])

4.2.1 Classification

The basic classification process maps objects or situations to a given set of classes. For this mapping, all necessary information has to be available when the problem solving process starts (cf. fig. 5). Examples of classification tasks are risk assessment of credit cards or loans or the analysis of marketing data or client data in order to determine what is worth how much and what kind of effort.

For basic classification tasks, CBR stands in competition with many other approaches. While inductive machine learning approaches normally reason with generalised knowledge and, by contrast, data bases store all the case data, CBR is more in flexible in the sense that it distributes its competence on both the cases and the similarity measure [WeGl94], (cf. fig. 11). Depending on the specific requirements of an application, a simple or a well-informed measure can be used. The better the similarity measure is suited for a given task, the less cases must be stored. This means that the CBR approach offers the flexibility to decide about this distribution at a very late stage in system development and that whenever this kind of flexibility is of use then CBR is a good choice.

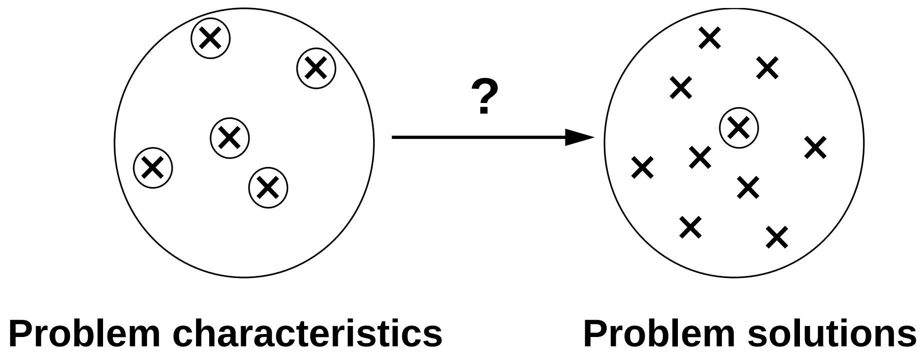

Fig. 5. The classification sub task

4.2.2 Diagnosis

Diagnostic processes differentiate from the above mentioned basic classification processes by the problem of incomplete information, i.e. before carrying out the classification process normally (much) more information has to be acquired (cf. fig. 6). Thus, the diagnosis task includes, in addition to the classification task, the task of selecting the best test(s) to acquire the missing information as cheap, fast and secure as possible. Examples are fault diagnosis of technical equipments like cars, CNC machines, robots or aircraft engines.

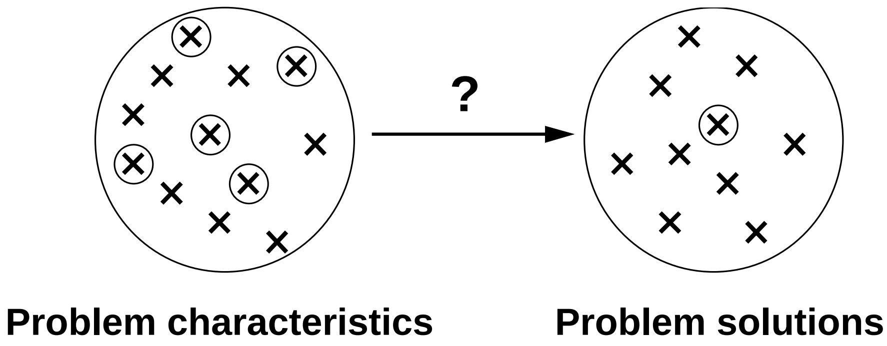

Fig. 6. The diagnosis sub task (adapted from [Pu90])

This additional test selection sub task makes the diagnosis task a very specific one. Often general domain knowledge is required to reasonably guide the selection process [Al93]. For some application tasks, the degree of incomplete information can be very high (>90%). Therefore, approaches based on induction have serious difficulties in coping with such problems in an efficient way. CBR can also offer a solution to the test selection problem itself by using an additional kind of cases.

4.2.3 Decision Support

Decision support processes differentiate from the classification and diagnosis processes by the necessity of representing more general knowledge and allowing user-computer interaction to a very high degree. While the classification goal is clearly defined for the classification and the diagnosis processes (static target definition), decision support processes have to cope with (more explorative) problems where the classification goal is defined or at least refined during the problem solving process (dynamic target definition; cf. fig. 7). Examples are looking for the right sort of financial investment, a house to buy, a last-minute trip without clearly defining the location and the kind of travel or the search for new equipment like trucks or manufacturing machines.

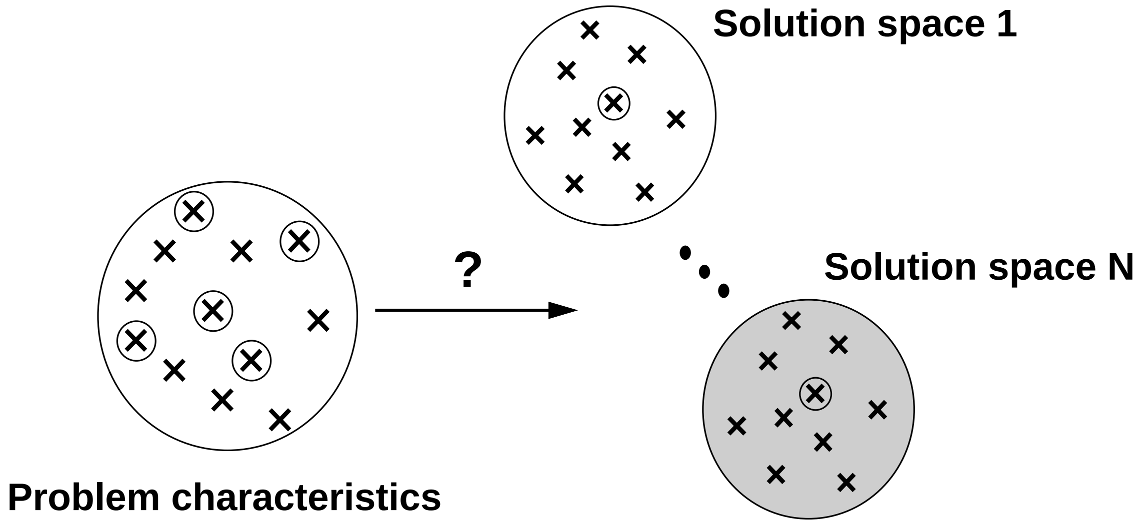

Fig. 7. The decision support sub task

In general, recognising the differences between diagnosis and decision support problems is not always easy. E.g., while PATDEX [We93] is a well-suited CBR system for fault diagnosis of engineering systems, INRECA is a more general decision support system and thus had to include many additional features. In PATDEX, an attribute-value-based case representation was sufficient whereas the scope of INRECA required an object-oriented representation. One main restriction of PATDEX is its inability to efficiently support local similarity measures. In INRECA, an extension of the k-d(imensional) tree retrieval structure has been introduced such that local similarity measures can be efficiently used [AuWe95]. These and other changes resulted in a much more flexible INRECA system which is, of course, also much more complex.

Beyond case representation and case retrieval, the use of general knowledge is usually required for decision support tasks. In INRECA, constraints, completion rules, and adaptation rules have been introduced in order to cope with such tasks. For decision support tasks, an additional structuring of the domain is necessary that otherwise is more naturally given e.g. by intermediate and final diagnoses.

4.2.4 A Case Study on Comparing CBR Tools for Analytic Tasks

In [AlAu95] the following technical criteria have been used to compare different commercial CBR tools (fig. 8):

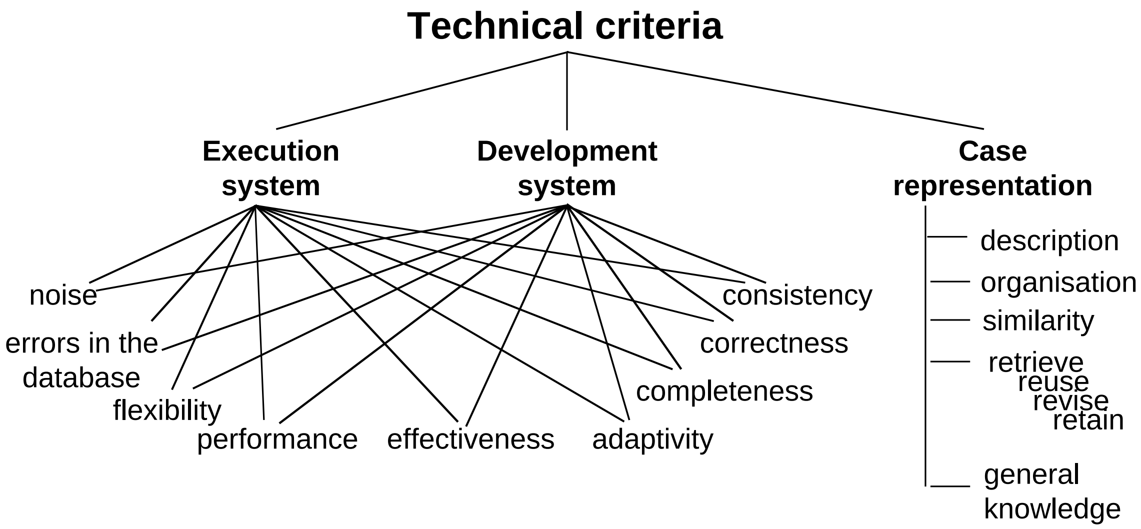

Fig. 8. Analysing technical issues for CBR systems

The results of this comparison have been used to guide the design and development of the final integrated INRECA system.

4.3 Case-Based Reasoning for Synthetic Tasks

Most but not all business problems belong to the class of analytic tasks. There is also a significant amount of problems that have to generate complex plans or constructions as their result. This class of problems is called synthetic tasks. The term synthetic stands for the need to build up a solution from parts - mostly obeying to a set of domain specific construction rules.

The class of synthetic problems comprises all tasks dealing with solutions for transportation logistics or production planning problems, with the configuration of technical devices and other artefacts to be constructed from prefabricated parts and last not least with the reuse of CAD plans and software modules.

Decisive for belonging to this class of synthetic tasks is the fact that the result of the problem solving process is a complex artefact and not just a class name or the selection of a predefined object. From this definition it is clear that synthetic tasks are usually more complex than analytic tasks - not only but also when tackled with CBR approaches.

Particularly characteristic features for CBR approaches in synthetic domains are the following ones:

It is very unlikely that a former solution can be taken over without changes. From this arises the necessity to provide at least a certain level of adaptation capabilities.

Due to different aspects involved and the overall complexity of the cases, it is less easy to rank solutions of similar problems according to their utility for the problem at hand. Therefore, additional criteria like "minimal adaptation effort" or "minimal risk of adaptation failures" become more interesting that pure similarity.

General knowledge usually plays an important role because it is an necessary prerequisite for the adaptation of former solutions and for assessing the quality of several candidate solutions.

Additionally, the more complex ones among the synthetic tasks are so-called open world problems [BaBa94]. This means that a software system will never be able to contain a complete model of the domain or a complete description of problem situation. This leads to the consequence that in open world domains, CBR systems can only provide suggestions for solutions checked for compatibility with the background knowledge that is available in the software system. The task of checking the compatibility with the rest of the relevant knowledge remains to be done by the user.

Rather often, classical AI approaches cannot be used in such complex, open and real world domains. In these situations, CBR may be the only chance to provide effective support - maybe not for all problems and not without risk - but this is still an advantage compared to having no support at all.

Similar to the analytic tasks, the synthetic tasks can also be subdivided into the following three groups (cf. fig. 9).

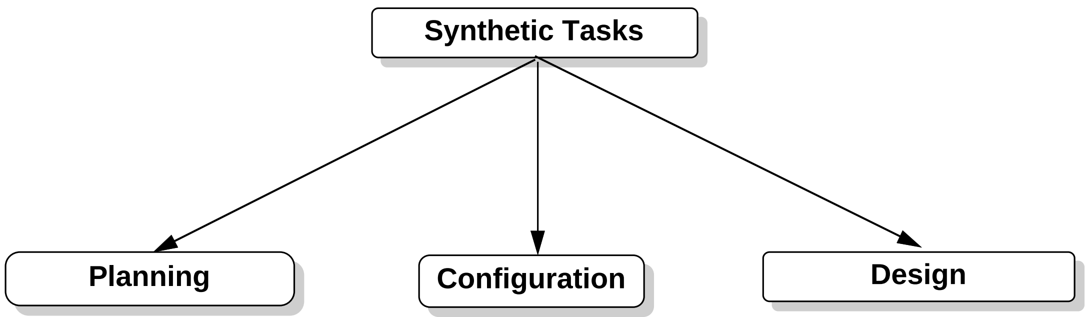

Fig. 9. Structuring synthetic tasks

4.3.1 Planning

Decisive for planning tasks is that they solve problems relative to given time constraints. This means that time is a predominant attribute in the case representation and for judging the suitability and adaptability of solutions.

Examples for CBR systems in planning applications are the planning of transportation logistics or of CNC machine manufacturing tasks or the assignment of resources like e.g. personnel or rooms with special equipment according to given schedule and time constraints.

4.3.2 Configuration

Characteristic for configuration tasks is that the resulting artefacts can be constructed from a reservoir of prefabricated parts with known functional behaviour and compatibility constraints.

Examples for CBR configuration systems are the configuration of computer equipment, x-ray and image processing devices, new materials, chemical substances, office rooms, kitchens, bathrooms etc.

4.3.3 Design

In contrast to configuration, design problems contain at least subparts where no straightforward configuration procedure which builds up solutions from prefabricated parts is applicable. Therefore, most design problems are open world problems [Ba95]. Here, both functional and geometrical requirements are predominant in the case representation and the problem specification can only be transferred to suggestions for solutions because of a usually existing lack of general knowledge.

Examples for CBR in design are support systems for the construction of industrial buildings or bridges, the layout for production lines or the construction of not too complicated mechanical devices.

4.3.4 Classifying Synthetic Tasks

Similar to the situation in the class of analytic tasks, it is not always easy to see at first sight where a particular application problem belongs to. Therefore, figure 10 contains an overview on the most predominant features of the different sorts of synthetic tasks that is meant to be helpful for classifying synthetic tasks.

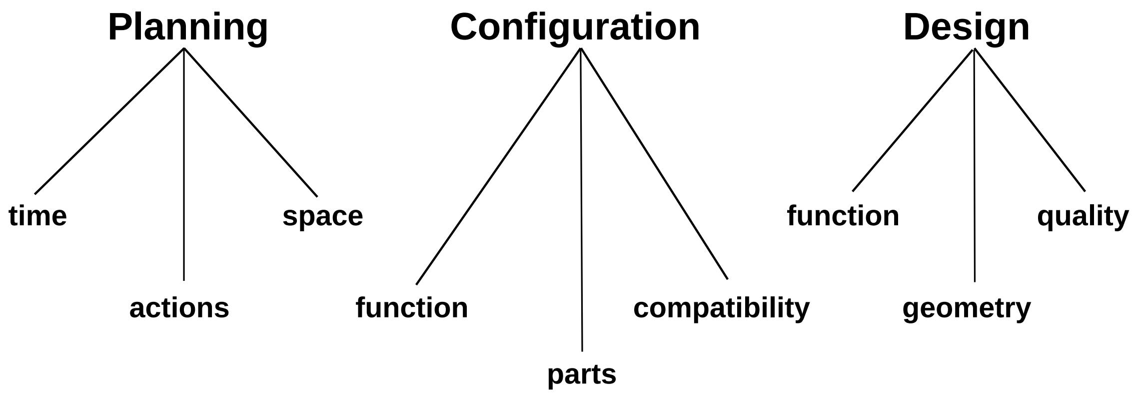

Fig. 10. Predominant features of synthetic tasks

5 Some Further Topics Concerning CBR System Development and Maintenance

Domains with only a small amount of cases can often be handled with a simple sequential search mechanism. Such a mechanism can also handle complex-structured cases rather easily because no extra index structure is needed.

For domains with larger amounts of cases, a non-trivial index structure has to be built. If the number of cases to be stored is steadily increasing and becoming really huge, the integration of a database system may turn out to be necessary.

If cases are only used for browsing then the underlying case representation can be simpler than for applications that require to carry out case adaptation.

While a lot of technical domains like fault diagnosis of CNC machines can be viewed and treated as closed world domains, others like oil well drilling or most medical diagnosis applications must - at least to a certain degree - be treated as open world domains. For the latter, the resulting CBR systems need a more encompassing case representation and should include more general knowledge.

In general, the decision how much general knowledge is needed and whether it should be used to complete a CBR system or, vice versa, whether it should be used to build a knowledge-based system, depends on the available knowledge sources, their degree of availability as well as the respective reliability of the knowledge sources. General knowledge used within a CBR system can help to reduce the number of cases necessary for problem solving, to improve the reliability of potential solutions, to make the overall system more efficient in the handling of routine situations, to exploit *known* generalisations, and to immediately adapt to a changing environment. CBR offers a flexible intermediate position between „database-like“ and „generalisation-like“ approaches (cf. fig. 11). In the INRECA system, this flexibility is used for a learning process such that INRECA evolves from a pure CBR system to a system that is more and more based on generalised knowledge. Inductive learning is here used to improve the underlying similarity measure to avoid backtracking in INRECA's k-d tree retrieval structure with the final goal that the retrieval performance becomes as good as it is for consulting a decision tree.

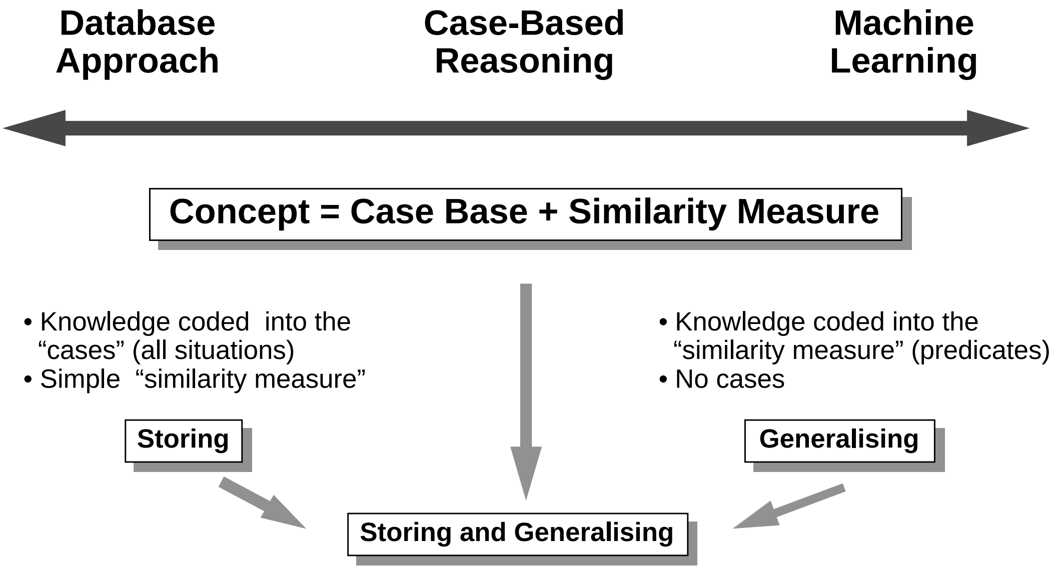

Fig. 11. General knowledge in case-based systems (adapted from [We95])

6 Summary and Outlook

CBR is a complementary approach not only to systems based on general knowledge, but also to statistical data analysis, information retrieval, neural networks, and database applications.

The CBR approach is especially suitable in domains where records of previously solved problems exist or where historical cases are viewed as an asset that ought to be preserved. If there is no case history, it should at least be clear that remembering previous experiences would be useful. Further hints that CBR is a suitable approach are that specialists talk about their domain by giving examples or that experience is at least as valuable as text-book knowledge, because CBR makes direct use of past experience.

Potential benefits of using CBR technology are discovering knowledge in data, delivering consistent decisions throughout an organisation, preserving the know-how of the most talented specialists by capturing their experience, transferring experience from the skilled specialist to the novice, and building a corporate memory by sharing individual experience.

The issues described in this paper imply that at least some knowledge about the technology and some skills in developing CBR systems or applying CBR tools are needed in order to build stable real-world applications.

Therefore, future work is needed to provide more guidance for non-CBR-experts in order to help "diagnose" the appropriate type of CBR application and to derive from this diagnosis e.g. what are the critical points, what are valuable hints, what tools have been used for similar applications and are there reusable components available, e.g. somewhere on the WWW.

The authors plan to initiate the collection of cases and the tool-based development of a CBR system that provides exactly such kinds of advice. This would mean to use CBR technology in order to ease and improve its own usage and handling.

7 Acknowledgement

Funding for this work has been provided within the INRECA project by the Commission of the European Union (Esprit Contract 6322) and within the FABEL project by the German Ministry of Research and Technology (BMBF contract 01IW104) to which the authors are greatly indebted. The partners of INRECA are AcknoSoft (Paris, coordinator), IMS (Dublin), tecInno (Kaiserslautern), and the University of Kaiserslautern. The partners of FABEL are BSR Consulting (München), GMD (Sankt Augustin, coordinator), HTWK Leipzig, Technical University of Dresden, University of Freiburg, and University of Karlsruhe.

The authors would like to thank Stefan Wess for the fruitful cooperation and many interesting discussions.

8 References

[Aa94] Aamodt, A.: Knowledge Acquisition and Learning by Experience – The Role of Case-Specific Knowledge. In [KoTe95], 197-245.

[AaAl93] Aamodt, A.; Althoff, K.-D.: Problem Solving and Sustained Learning from Experience: Analyzing Methods with respect to Domain Characteristics. In: M. van Someren (ed.), Proc. MLnet Workshop on „Learning and Problem Solving“, Blanes.

[AaPL94] Aamodt, A.; Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Communications, vol. 7, no. 1, 39-59.

[AaVe95] Aamodt, A.; Veloso, M. (eds.): Proc. 1st International Conference on Case-based Reasoning, Springer Verlag Berlin 1995, forthcoming.

[Ah94] Aha, D. W. (ed.): Proc. AAAI Workshop on „Case-Based Reasoning“ , Seattle 1994.

[Al93] Althoff, K.-D.: Lernverfahren in MOLTKE. In [PfRi93], 173-200.

[Al95] Althoff, K.-D. (1995a). Evaluating Case-Based Reasoning Systems. In: Proc. Workshop on Case-Based Reasoning: A New Force In Advanced Systems Development, 48-61, published by Unicom, Uxbridge, 1995.

[AlAa95] Althoff, K.-D.; Aamodt, A.: Relating Problem Solving and Learning Methods to Task and Domain Characteristics: An Analytical Framework, forthcoming.

[AlAu95] Althoff, K.-D.; Auriol, E.; Barletta, R.; Manago, M.: A Review of Industrial Case-Based Reasoning Tools. AI Intelligence, Oxford 1995.

[AuWe95] Auriol, E.; Wess, S.; Manago, M.; Althoff, K.-D.; Traphöner, R.: Integrating Induction and Case-Based Reasoning: Methodological Approach and First Evaluations. In [AaVe95].

[Ba87] Bartsch-Spörl, B.: Ansätze zur Behandlung von fallorientiertem Erfahrungswissen in Expertensystemen. KI, vol. 1, no. 4, 32-36.

[Ba95] Bartsch-Spörl, B.: Towards the Integration of Case-based, Schema-Based and Model-based Reasoning for Supporting Complex Design Tasks. In [AaVe95].

[BaBa94] Bakhtari, S.; Bartsch-Spörl, B.: Bridging the Gap between AI Technology and Design Requirements. In Gero, J.; Sudweeks, F. (eds): Artificial Intelligence in Design 94. Kluwer, Dordrecht 1994, 753-768.

[BaBa95] Bartsch-Spörl, B.; Bakhtari, S.: Wiederverwendung von Wissen: Case-Based Reasoning aus einer Engineering Perspektive, Proceedings ISKO'95, forthcoming.

[Be93] Benjamins, R.: Problem Solving Methods for Diagnosis. Ph.D. thesis, University of Amsterdam 1993.

[ChJo93] Chandrasekaran, B.; Johnson, T.: Generic Tasks and Task Structures: History, Critique and New Directions. In [DaKr93], 232-272.

[Co89] Cohen, P. R.: Evaluation and Case-based Reasoning. In [Ha89], 168-172.

[DaKr93] David, J.-M.; Krivine, J.-P.; Simmons, R. (eds.): Second Generation Expert Systems. Springer Verlag, Berlin 1993.

[DrMo92] Drenth, H.; Morris, A.: Prototyping expert solutions: an evaluation of Crystal, Leonardo, GURU and ART-IM. Expert Systems, vol. 9, no. 1 (1992), 35-45.

[EhSc96] Ehrenberg, D.; Schulz, R.: Wiederverwendung von Wissen in betrieblichen Informationssystemen, Wirtschaftsinformatik, vol. 38, no. 1 (1996).

[Ha89] Hammond, K. (ed.): Proc. of the 2nd DARPA Workshop on Case-Based Reasoning. Morgan Kaufmann, San Mateo 1989.

[Jo94] Johnston, R.: Case-based Reasoning and Institutional Memory. Irish Engineers, Journal, May 1994.

[KoTe95] Kodratoff, Y.; Tecuci, G. (eds.): Integration of Knowledge Acquisition and Machine Learning. Kluwer, Dordrecht 1995, forthcoming.

[KoSi85] Kolodner, J.; Simpson, R.; Sycara, K.: A Process Model of Case-Based Reasoning in Problem Solving. In: Proceedings IJCAI-85, Morgan Kaufmann, Los Angeles 1985, 284-290.

[PfRi93] Pfeifer, T.; Richter, M. M. (eds.): Diagnose von technischen Systemen: Grundlagen, Methoden und Perspektiven der Fehlerdiagnose. Deutscher Universitäts-Verlag 1993.

[Pu90] Puppe, F. (1990). Problemlösungsmethoden in Expertensystemen. Springer Verlag, Berlin 1990.

[PuGü93] Puppe, F.; Günter, A. (eds.): Expertensysteme 93 – Proc. of the 2nd German Conference on Expert Systems. Springer Verlag, Berlin 1993.

[Ri92] Richter, M.M.: Classification and Learning of Similarity Measures. In: Proc. of the 16th Annual Meeting of the German Society for Classification, Springer Verlag Berlin 1992.

[St90] Steels, L.: Components of Expertise. AI Magazine, vol. 11, no. 2 (1990), 29-49.

[StJa90] Strube, G.; Janetzko, D.: Episodisches Wissen und fallbasiertes Schließen: Aufgaben für die Wissensdiagnostik und die Wissenspsychologie. Schweizerische Zeitschrift für Psychologie, vol. 49, no.4 (1990), 211-221.

[We93] Wess, S.: PATDEX – ein Ansatz zur wissensbasierten und inkrementellen Verbesserung von Ähnlichkeitsbewertungen in der fallbasierten Diagnostik. In [PuGü95], 42-55.

[We95] Wess, S.: Fallbasiertes Schließen in wissensbasierten Systemen zur Entscheidungsunterstützung und Diagnose. Doctoral Dissertation, University of Kaiserslautern 1995.

[WeAl94] Wess, S.; Althoff, K.-D.; Richter, M. M. (eds.): Topics in Case-Based Reasoning. Springer Verlag, Berlin 1994.

[WeGl94] Wess, S.; Globig, C.: Case-Based and Symbolic Classification Algorithms - A Case Study Using Version Space. In [WeAl94], 77-91.

[WiVe93] Wielinga, B. J.; Van de Velde, W.; Schreiber, G.; Akkermans, H.: Towards a unification of knowledge modeling approaches. In [DaKr93], 299-335.