Mapping Goals and Kinds of Explanations to the Knowledge Containers of Case-Based Reasoning Systems

Thomas R. Roth-Berghofer[1,2] and Jörg Cassens[3]

[1] Knowledge-Based Systems Group, Department of Computer Science, University of Kaiserslautern, P.O. Box 3049, 67653 Kaiserslautern
[2] Knowledge Management Department, German Research Center for Artificial Intelligence DFKI GmbH, Erwin-Schrödinger-Straße 57, 67663 Kaiserslautern, Germany
thomas.roth-berghofer@dfki.uni-kl.de
[3] Norwegian University of Science and Technology (NTNU), Department of Computer and Information Science (IDI), 7491 Trondheim, Norway
jorg.cassens@idi.ntnu.no

Abstract. Research on explanation in Case-Based Reasoning (CBR) is a topic that gains momentum. In this context, fundamental issues on what are and to which end do we use explanations have to be reconsidered. This article presents a prelimenary outline of the combination of two recently proposed classifications of explanations based on the type of the explanation itself and user goals which should be fulfilled. Further on, the contribution of the different knowledge containers for modeling the necessary knowledge is examined.

1 Why Bother to Explain?

In everyday human-human interactions explanations are an important vehicle to convey information in order to understand one another. Explanations enhance the knowledge of the communication partners in such a way that they accept certain statements. They understand more, allowing them to make informed decisions. According to Schank [1] explanations are the most common method used by humans to support their decision making.

This is supported by Spieker's investigation into natural language explanations in expert systems [2]. We identify some typical reactions of humans as soon as we cannot follow a conversation:

- we ask our conversation partner about concepts that we did not understand,
- we request justifications for some fact or we ask for the cause of an event,
- we want to know about functions of concepts,
- we want to know about purposes of concepts, and
- we ask questions about his or her behavior and how he or she reached a conclusion.

H. Muñoz-Avila and F. Ricci (Eds.): ICCBR 2005, LNCS 3620, pp. 451–464, 2005.

All those questions and answers are used to understand what has been said and meant during a simple conversation. An important effect of explanations is that the process of explaining certainly has some effect on one's trust in the competence of a person or machine: We keep our trust, we increase or decrease it. At least, providing explanations makes decisions more transparent, and motivates the use to further use the system.

The need for explanations provided by knowledge-based systems is well-known and was addressed by such fields as expert systems. For knowledge-based systems, explanations and knowledge acquisition are the only two communications channels with which they interact with their environment.

The adequacy of explanations as well as of justifications is dependent on pragmatically given background knowledge. What counts as a good explanation in a certain situation is determined by context-dependent criteria [3,4].

The more complex knowledge-based systems get, the more explanation capabilities the users expect when using such systems. This requirement was recognized early on in expert systems research and development [5,6,7]. Considerable results were produced, but research activity decreased together with the general decline of expert systems research in the 1990s. The major problems in connection with classical expert systems seemed to be solved.

At the same time there was an increasing interest on this topic in Case-Based Reasoning (CBR) [8,9]. At the turn of the century, we find the issue discussed again in the context of knowledge-based systems [10,11]. Recently, we can see a renewed focus in CBR on this track of research. ECCBR 2004 featured, for example, a workshop on Explanation in Case-Based Reasoning as well as a couple of papers on explanation at the main conference [12,13].

Research on explanation is of interest today because it can be argued that the whole scenario on research on knowledge-based systems has changed [14]: knowledge-based systems are no longer considered as boxes that provide a full solution to a problem. Problem solving is seen as an interactive process (a socio-technical process). Problem description as well as the special input can be incomplete and changing. As a consequence, there has to be communication between human and software agents. Communication requires mutual understanding that can be essentially supported by explanations. Such explanations can improve the problem solving process to a large degree.

It is important to note here that the term explanation can be interpreted in two different ways. One interpretation deals with explanations as part of the reasoning process itself. The other interpretation deals with usage aspects: making the reasoning process, its results, or the usage of the result transparent to the user. In this paper, we will focus on the second interpretation.

The remainder of this paper is organized as follows: In the next section, we describe the setting for explanation-aware CBR systems as being a component of socio-technical systems. In section 3, we present two perspectives on explanation that can help understand and organize what to explain and when. The subsequent section focusses on knowledge containers and their contribution to the explanation capabilities of CBR systems. In Section 5, we propose a system design process achitecture. We explore further on the relations of explanation goals, explanation kinds, and knowledge containers in a simplified example. We conclude our paper with an outlook on further research.

2 Explanation in Socio-Technical Systems

Whenever one talks about a ‘system’ one has to clarify what is meant by that term. In decision- support scenarios, the human and the computer are the decision system. Such socio-technical systems can for example be modelled with the help of the Actor Network Theory, ANT ([15,16]). The basic idea here is fairly simple: whenever you do something, many influences on *how* you do it exist. For instance, if you visit a conference, it is likely that you stay at a hotel. How you behave at the hotel is influenced by your own previous experience with hotels, regulations for check-in and check-out, the capabilities the hotel offers you (breakfast room, elevators).

So, you are not performing from scratch, but are influenced by a wide range of factors. The aim of the ANT is to provide a unified view on these factors and your own acting. An actor network in this notion is *the act linked together with all of its influencing factors (which again are linked), producing a network* (see [16, p. 4]).

In this network, you find both technical and non-technical elements. In the ANT, technological artifacts can stand for human goals and praxis. Hotel keys, for example, are often not very handy, because the hotel owner has *inscribed* his intention (that the keys do not leave the hotel) into metal tags (which is why the guests *subscribe* to the owners intention: they do not want to carry this weight). A software system for workflow management is a representation of organizational standards in the company where it is used (and makes human users follow these standards).

One advantage of the ANT in the setting of intelligent systems is that it already comprises technical artifacts and humans in the same model. Humans and artifacts are to a certain degree exchangeable and can play the same role in the network. But in contrast to traditional artifacts, which are merely passive (black boxes in which human interests are subscribed) or which active role is restricted to translating intentions of the designer into changes of the praxis of the user, AI systems play a more active role. It has also been argued that intelligent systems have to show certain capabilities usually ascribed to humans in order to interact with the user in a meaningful way [17], and we would include the ability to give good explanations.

Moreover, the issue of ‘trust’ is generally important for socio-technical systems. ‘Trust’ can be defined in different ways, for the purpose of this paper it is sufficient to describe the problem as to whether and to which degree a human is willing to accept proposals from technical components, and to which degree he is willing to give up control. For a detailed survey on different definitions of trust in the context of automation systems, see e.g. [18]. In the context of expert systems, it has been shown that explanation capabilities have a large effect on the user’s acceptance of advices given by the system [19].

To summarize, the ability of an IT system to give good explanations is important for the functioning of a socio-technical system. Good explanations depend on the context, it would therefore be helpful to be able to include an analysis into the system design process.

3 Views on Explanations

In this section, we outline two perspectives on explanation: The *Explanation Goals* focus on user needs and expectations towards explanations and help to understand *what* the system has to be able to explain and *when* to explain something. The *Kinds of Explanations* focus on different *types* of explanations, their *usefulness* for the user, and how they can be represented in the different *knowledge-containers* [20].

Any kind of interactivity implies that one has some kind of user model that provides answers based on what the user knows and what he or she does not know [21]. The user (probably) knows about the used vocabulary, about general strategies, policies, or procedures to follow, and about (most of) the standard situations in the given problem domain. But he or she may not know all the details and data, about rare cases and exceptions, and about consequences of combinatorial number of interactions of different alternatives. Then, a basic approach to explanation would be to not comment on routine measures (without being asked), to emphasize on exceptional cases (e.g., exceptions from defaults and standards, exceptions from plausible hypotheses), and to allow for further questions.

It is hard to anticipate user needs due to two main reasons [21]: First, not all of the needs must be met, but those important to the user. Second, all deficits and their estimated importance depend on the specific user. Thus, personalization is a basic requirement, not only some added value.

3.1 Explanation Goals

Sørmo et al. [22,23] suggest several explanation goals for Case-Based Reasoning systems (which are valid for knowledge-based systems, in general). They also argue that those goals are indeed reachable because case-based reasoners are mostly made to perform limited tasks for a limited audience, thus allowing to make reasonable assumptions about the user's goals and the explanation context. The identified explanation goals are:

Transparency: Explain how the system reached the answer
"I had the same problem with my car yesterday, and charging the battery fixed it."

The goal of an explanation of this kind is to impart an understanding of how the system found an answer. This allows the users to check the system by examining the way it reasons and allows them to look for explanations for why the system has reached a surprising or anomalous result. If transparency is the primary goal, the system should not try to oversell a conclusion it is uncertain of. In other words, fidelity is the primary criterion, even though such explanations may place a heavy cognitive load on the user. The original *how* and *why* explanations of the MYCIN system [24] would be good examples.

This goal is most important with knowledge engineers seeking to debug the system and possibly domain experts seeking to verify the reasoning process [10]. It is also reasonable to think that in domains with a high cost of failure it can be expected that the user wishes to examine the reasoning process more thoroughly.

Justification: Explain why the answer is a good answer
"You should eat more fish - your heart needs it!"
"My predictions have been 80% correct up until now."

This is the goal of increasing the confidence in the advice or solution offered by the system by giving some kind of support for the conclusion suggested by the system. This goal allows for a simplification of the explanation compared to the actual process the system goes through to find a solution. Potentially, this kind of explanation can be completely decoupled from the reasoning process, but it may also be achieved by using additional background knowledge or reformulation and simplification of knowledge that is used in the reasoning process.

Empirical research suggests that this goal is most prevalent in systems with novice users [25], in domains where the cost of failure is relatively low, and in domains where the system represents a party that has an interest in the user accepting the solution.

Relevance: Explain why a question asked is relevant
"I ask about the more common failures first, and many users do forget to connect the power cable."

An explanation of this type would have to justify the strategy pursued by the system. This is in contrast to the previous two goals that focus on the solution. The reasoning trace type of explanations may display the strategy of the system implicitly, but it does not argue why it is a good strategy. In conversational systems, the user may wish to know why a question asked by the system is relevant to the task at hand. It can also be relevant in other kinds of systems where a user would like to verify that the approach used by the system is valid. In expert systems, this kind of explanations was introduced by NEOMYCIN [24].

Conceptualization: Clarify the Meaning of Concepts
"By 'conceptualization' we mean the process of forming concepts and relations between concepts."

One of the lessons learned after the first wave of expert systems had been analyzed was that the users did not always understand the terms used by a system. This may be because the user is a novice in the domain, but also because different people can use terms differently or organize the knowledge in different ways. It may not be clear, even to an expert, what the system means when using a specific term, and he may want to get an explanation of what the system means when using it. This requirement for providing explanations for the vocabulary was first identified by Swartout and Smoliar ([7]).

Learning: Teach the user about the domain
"When the headlights won't work, the battery may be flat as it is supposed to deliver power to the lights."

All the previous explanation goals involve learning – about the problem domain, about the system, about the reasoning process or the vocabulary of the system. Educational systems, however, have learning as the primary goal of the whole system. In these systems, we cannot assume that the user will understand even definitions of terms, and may need to provide explanations at different levels of expertise. The goal of the system is typically not only to find a good solution to a problem, but to explain the solution process to the user in a way that will increase his understanding of the domain. The goal can be to teach more general domain theory or to train the user in solving problems similar to those solved by the system. In other words, the explanation is often more important than the answer itself. Systems that fulfill the relevance and transparency goals may have some capabilities in this area, but a true tutoring system must take into account how humans solve problems. It cannot attempt to teach the user a problem solving strategy that works well in a computer but that is very hard to reproduce for people.

For the remainder of this paper we will not focus on the learning goal since it is specifically targeted towards educational systems.

3.2 Kinds of Explanations

Roth-Berghofer [26] looks at explanations from a knowledge-container perspective. He addresses the issue of what can naturally be explained by the four containers (see Section 4).

One starting point is the work of Spieker [2] on the usefulness of explanations. According to Spieker, there are five useful kinds of explanations he discusses in the context of expert systems:

Conceptual Explanations: They are of the form 'What is ...?' or 'What is the meaning of ...?'. The goal of conceptual explanations is to build links between unknown and known concepts. Conceptual explanations can take different forms:

- Definition: "What is a bicycle?" "A bicycle is a land vehicle with two wheels in line. Pedal cycles are powered by a seated human rider. A bicycle is a form of human powered vehicle."
- Theoretical proposition: "What is force?" "Force is Mass times Acceleration."
- Prototypical example: "What is a bicycle?" "The thing, the man there crashed with."
- Functional description: "What is a bicycle?" "A bicycle serves as a means of transport."

Conceptual explanations are answers to extensional or descriptional questions.

Why-explanations: Why-explanations provide causes or justifications for facts or the occurrence of events. Whereas the first concept is causal in nature and not symmetrical, the latter only provides evidence for what has been asked for. For example:

- Justification: "Why is it believed that the universe expands?" "Because we can observe a red shift of the light emitted by other galaxies."
- Cause: "Why is it believed that the universe expands?" "Because, according to the Big Bang theory, the whole matter was concentrated at one point of the universe and the whole matter moves away from each other."

Why-explanations explain single events or general laws and can consist of single causes/justifications (among others) or a complete list of causes/justifications.

How-explanations: How-explanations are a special case of why-explanations, describing processes that lead to an event by providing a causal chain. They are similar to action explanations (see below) that answer how-questions. How-questions ask for an explanation of the function of a device, for example:

- "How does a combustion engine work?" "A combustion engine is an engine that operates by burning its fuel."

Purpose-explanations: The goal of *Purpose-explanations* is to describe the purpose of a fact or object. Typical questions are of the form 'What is ... for?' or 'What is the purpose of ... ?', for example:

- "What is a valve for?" "The valve is used to seal the intake and exhaust ports."

Cognitive Explanations: Cognitive Explanations explain or predict the behavior of 'intelligent systems' on the basis of known goals, beliefs, constraints, and rationality assumptions. There are action and negative explanations:

- Action explanation: "Why was this seat post selected?" "For the given price, only one other seat post for this bicycle is currently available. But that seat post is too short."
- Negative explanation: "Why was no carrier chosen?" "A carrier is only available for touring bikes. The user did not choose a touring bike."

4 Knowledge Containers

Knowledge containers, according to Richter [27,28], contain and structure the knowledge of a knowledge-based system. A knowledge container is a collection of knowledge that is relevant to many tasks. For rule-based systems, for instance, one can easily identify facts and rules as important knowledge containers. For CBR systems, Richter describes four knowledge containers: *vocabulary*, *similarity measures*, *adaptation knowledge*, and *case base*. They are depicted in Fig. 1.

The *vocabulary* defines attributes, predicates, and the structure of the domain schema. Thus the vocabulary forms the basis for all of the other three containers. Hierarchies, if available, can be used to order domain concepts.In object-oriented

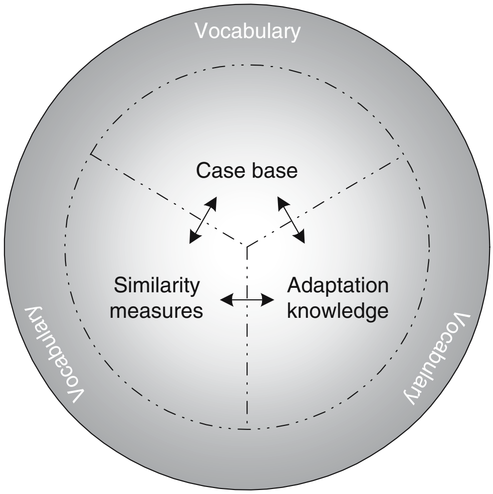

Fig. 1. The four knowledge containers of a CBR system

models, inheritance (*is-a*) and decomposition (*part-of*) induce hierarchical orderings quite naturally. Additional ontological relations can further add hierarchical information. Those hierarchies can be exploited for conceptual and (partly) for purpose explanations (because the ordering often is inferred from specialization/generalization). Other easily available information is information on the kind of attribute. *Input attributes* may be used to infer information for *retrieval attributes* as well as for filling *output attributes* of a query or a case. For example, imagine a CBR system for PC configuration in an electronic commerce scenario. The request for a multimedia PC triggers completion rules for filling such retrieval attributes as `processor` and `graphic card` accordingly. Not specified attributes of the query automatically become output attributes. The CBR system now could use the information for cognitive explanations based on why it filled the retrieval attributes etc.

The knowledge that determines how the most useful case is retrieved and by what means the similarity is calculated, is held by the *similarity measures* container, which can be further divided into the sub-containers for local similarity measures and amalgamation functions. Each local measure compares values of one attribute of a case. It contains domain knowledge, e.g., about different processor speeds or graphic cards. Amalgamation functions are task oriented and contain utility knowledge (relevances for the task, e.g., the importance of the graphic card vs. the importance of the processor speed when selecting a multimedia PC). The already mentioned completion rules provide knowledge about dependencies between attributes.

The *adaptation knowledge* container covers the knowledge for translating a prior solution to fit a given query and the *case base* stores the experience of the CBR system, i.e., the cases. Knowledge about the types of cases used by

Table 1. Knowledge containers and their contribution to explanations [26]

Knowledge container	contributes to
Vocabulary	conceptual explanations, why-explanations, how-explanations, and purpose explanations
Similarity measures	why-explanations, how-explanations, purpose explanations, and cognitive explanations
Adaptation knowledge	why-explanations, how-explanations, and cognitive explanations
Case base	why-explanations, how-explanations, and context

the case-based reasoner, such as *homogeneous* vs. *heterogeneous* and *episodic* vs. *prototypical* cases [29] as well as cases of *rule* vs. *constraint* type [30], structures this knowledge container further.

Table 1 shows an overview of which knowledge container contributes to which kind of explanation (see [26] for details).

5 Exploring the Relations of Goals and Kinds

As we have outlined before, there is a need to take the context of explanations as well as different goals with and types of explanation into account. A methodology for the development of explanation-aware CBR systems should therefore comprise components for the workplace analysis (like ANT described in section 2 or activity theory [31]) as well as methods to translate the analytical findings into system synthesis. Further on, this process has to be integrated with methods for the continuous maintenance of the CBR system [32]. We propose therefore a overall process architecture as depicted in figure 2.

During the remainder of this article, we will propose a 3-step process to identify which explanations a CBR system should be able to give and to understand how to make the necessary knowledge accessible in the different knowledge containers (see the grey box in figure 2):

1. Use the *Explanation Goals* perspective to identify user needs for explanations from a user model and system view which takes the usage situation into account.
2. Use the *Explanation Kinds* view to find useful prototypical explanations and assess the requirements for contents that have to be modeled into the system.
3. Use the different *Knowledge Containers* to store the necessary knowledge to support the different kinds of explanation identified.

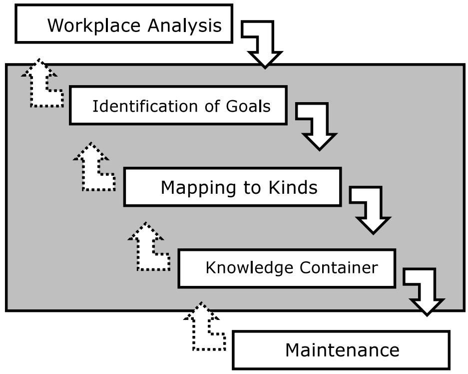

Fig. 2. The overall process architecture

The mapping of goals to kinds and kinds to containers, respectively, is not necessarily a one to one relation which can be followed mechanically. The mapping proposed in this paper gives rather hints for the modeling task by focusing the work of the system designer on probable solutions.

As a simplified example, we look at a case-based diagnostic system for engine failures. We have a mixed initiative dialogue system where the system can ask questions about the engine status and the user can voluntarily provide information he deems important.[1] The system can give detailed explanations on possible causes for the problems as well as advice on how to avoid future occurrences. It is supportive, e.g., the user should be enabled to understand similar situations in the future without having to rely on the system.

There is no adaptation of cases since we are purely interested in the possible cause of a failure and not a solution to solve this problem. Further on, we assume the system to be capable of generating plausible and justified explanations itself without going into details about the underlying mechanism.

Conceptualization goal fullfilled by a conceptual explanation (definition): During the symptom assessment, the system asks the user to fill in the specific gravity of the fuel. The user is not familiar with the term specific gravity so he asks the system to explain this. The system gives this explanation in the form of a *conceptual explanation*, in our example as a *definition*:
User: "What is the specific gravity?"
System: "It is a measure of the density of a liquid relative to the density of water, with water having a specific gravity of 1.0."

Following our argument on the contribution of different Knowledge Containers to explanation kinds, the necessary knowledge for this kind of answers would come from the *vocabulary* of the system. Undoubtedly, conceptual explanations

[1] Please, keep in mind that there are other possible human computer interfaces than natural language dialogue systems as depicted here for exemplary reasons. For example, one possibility is to provide the user with pre-formulated queries from which he or she can choose the most appropriate one.

need much more knowledge than provided in most CBR systems, even in knowledge rich systems such as CREEK [33,34], where general (domain-dependent) knowledge is represented as semantic network.[2]

Relevance goal fullfilled by a causal why-explanation: The user can further on ask to know why the system asked this specific question. The system should give explanations which set the question asked into context with the symptoms already known. This can be described in terms of a *causal why-explanation:*
User: "Why is the density of the fuel interesting?"
System: "Air/Fuel ratio is based on pounds of AIR to pounds of FUEL. Changing the specific gravity of your fuel can have a profound affect on the A/F ratio of your engine since carburetors are metered by volume."

Again, the necessary knowledge to answer this question can be found in the *vocabulary*. This causal why-explanation will typically rely on the domain knowledge modelled in the vocabulary of the system.

Justification goal fullfilled by a theoretical conceptual explanation: The system finally presents a solution to the user which requires him to change the spark lead time. In this situation, the user might want to know why this is a solution. This goal can be satisfied by presenting a *theoretical conceptual explanation*:
User: "How is this adjustment related to the problem of too much CO_2 in the exhaust gases?"
System: "Different fuels and engine setups require different spark lead times. When using a wrong timing, the combustion process is suboptimal, and this will lead to high CO_2 values."

Also here, the necessary knowledge can best be stored in the *vocabulary* container. In order to answer this question we need some explicitly modeled domain knowledge. A knowledge poor CBR system could only point to cases and its own similarity measure and leave it to the user to find the explanation himself.

Transparency goal fullfilled by a cognitive explanation: The user might not be fully satisfied and tries to understand how the system came up with this answer. The system can explain the reasoning process using a *how explanation*:
User: "How did you come up with this answer?"
System: "Looking at my case base we can see that the most relevant cases had problems with the spark lead time as well, with only the exception of those cases where the problem can be explained by the wrong type of fuel."

The explanatory knowledge for this question can be found in the *similarity measure* of the system. The system needs to be able to explain why it delivered a certain case in terms of its similarity assessment. The *case base* container provides the context for the explanation by restricting the problem space to the

[2] In CREEK, explanations are generated to explain reasoning steps or to justify conclusions to the user, but mainly for the internal use of the reasoner.

available cases. Please note that a knowledge rich CBR system might be able to explain the absence of certain features in the solution case by referring to its domain knowledge, stored in the *vocabulary*.

6 Conclusions and Future Research Directions

We have outlined a unified view on explanations in Case-Based Reasoning, which takes both the goals of the user and the type of an explanation into account. Both perspectives are to a certain degree independent from each other.

The next step in our fellow work is to integrate an explanation goals view with methods for the analysis of workplace situations like ANT and activity theory (as proposed, e.g., by Cassens [31]) and integrate the explanation kind perspective with existing design and maintenance methodologies (such as Inreca [35] and Siam [32]).

We want to develop further our structural view on explanations and supporting knowledge available in CBR systems, with the ultimate goal of providing a methodology on how to develop explanation-aware CBR systems in the future.

References

1. Schank, R.C.: Explanation Patterns: Understanding Mechanically and Creatively. Lawrence Erlbaum Associates, Hillsdale, NJ (1986)
2. Spieker, P.: Natürlichsprachliche Erklärungen in technischen Expertensystemen. Dissertation, University of Kaiserslautern (1991)
3. Cohnitz, D.: Explanations are like salted peanuts. In Beckermann, A., Nimtz, C., eds.: Proceedings of the Fourth International Congress of the Society for Analytic Philosophy. (2000) `http://www.gap-im-netz.de/gap4Konf/` Proceedings4/titel.htm [Last access: 2004-08-11].
4. Leake, D.B.: Goal-Based Explanation Evaluation. In: Goal-Driven Learning. MIT Press, Cambridge (1995) 251–285
5. Swartout, W.: What Kind of Expert Should a System be? XPLAIN: A System for Creating and Explaining Expert Consulting Programs. Artificial Intelligence **21** (1983) 285–325
6. Buchanan, B.G., Shortliffe, E.H.: Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project. Addison Wesley, Reading (1984)
7. Swartout, W., Smoliar, S.: On Making Expert Systems More Like Experts. Expert Systems **4** (1987) 196–207
8. Leake, D.B., ed.: Case-Based Reasoning: Experiences, Lessons, & Future Directions. AAAI Press/MIT Press, Menlo Park (1996)
9. Schank, R.C., Kass, A., Riesbeck, C.K., eds.: Inside Case-Based Explanation. Lawrence Erlbaum Associates, Hillsdale, New Jersey (1994)
10. Gregor, S., Benbasat, I.: Explanations From Intelligent Systems: Theoretical Foundations and Implications for Practice. MIS Quarterly **23** (1999) 497–530
11. Swartout, W.R., Moore, J.D.: Explanation in second generation expert systems. In David, J., Krivine, J., Simmons, R., eds.: Second Generation Expert Systems. Springer Verlag, Berlin (1993) 543–585

12. Gervás, P., Gupta, K.M., eds.: Proceedings of the ECCBR 2004 Workshops. Number 142-04 in Technical Report, Madrid, Departamento de Sistemas Informáticos y Programación, Universidad Complutense Madrid (2004)
13. Funk, P., Calero, P.A.G., eds.: Advances in Case-Based Reasoning: Proceedings ECCBR 2004. Number 3155 in LNAI, Berlin, Springer (2004)
14. Richter, M.M.: Remarks on current explanation research in artificial intelligence (2005) Personal notes.
15. Latour, B.: Technology is Society made Durable. In Law, J., ed.: A Sociology of Monsters. Routledge (1991) 103–131
16. Monteiro, E.: Actor-Network Theory. In Ciborra, C., ed.: From Control to Drift. Oxford University Press (2000) 71–83
17. Pieters, W.: Free Will and Intelligent Machines. Project Report, NTNU Trondheim (2001)
18. Lee, J.D., See, K.A.: Trust in Automation: Designing for Appropriate Reliance. Human Factors **46** (2004) 50–80
19. Ye, L.R., Johnson, P.E.: The impact of explanation facilities on user acceptance of expert systems advice. MIS Q. **19** (1995) 157–172
20. Richter, M.M.: The knowledge contained in similarity measures. Invited Talk at the First International Conference on Case-Based Reasoning, ICCBR'95, Sesimbra, Portugal (1995)
21. Richter, M.M.: Prinzipien der Künstlichen Intelligenz. 2. edn. B. G. Teubner, Stuttgart (1992)
22. Sørmo, F., Cassens, J.: Explanation goals in case-based reasoning. [12] 165–174
23. Sørmo, F., Cassens, J., Aamodt, A.: Explanation in Case-Based Reasoning – Perspectives and Goals. To be publisehd (2005)
24. Clancey, W.J.: The epistemology of a rule-based expert system: A framework for explanation. Artificial Intelligence **20** (1983) 215–251
25. Mao, J.Y., Benbasat, I.: The Use of Explanations in Knowledge-Based System: Cognitive Perspectives and a Process-Tracing Analysis. Journal of Managment Information Systems **17** (2000) 153–179
26. Roth-Berghofer, T.R.: Explanations and case-based reasoning: Foundational issues. In Funk, P., Calero, P.A.G., eds.: Advances in Case-Based Reasoning, Springer-Verlag (2004) 389–403
27. Richter, M.M.: The knowledge contained in similarity measures. Invited Talk at the First International Conference on Case-Based Reasoning, ICCBR'95, Sesimbra, Portugal (1995) `http://wwwagr.informatik.` uni-kl.de/˜ lsa/CBR/Richtericcbr95remarks.html [Last access: 2002-10-18].
28. Lenz, M., Bartsch-Spörl, B., Burkhard, H.D., Wess, S., eds.: Case-Based Reasoning Technology: From Foundations to Applications. Volume LNAI 1400 of Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin (1998)
29. Watson, I.: Survey of CBR application areas (1999) Invited Talk at the 3rd International Conference on Case-Based Reasoning ICCBR.
30. Richter, M.M.: Generalized planning and information retrieval. Technical report, University of Kaiserslautern, Artificial Intelligence – Knowledge-based Systems Group (1997)
31. Cassens, J.: Knowing what to explain and when. [12] 97–104
32. Roth-Berghofer, T.R.: Knowledge Maintenance of Case-Based Reasoning Systems – The SIAM Methodology. Volume 262 of Dissertationen zur Künstlichen Intelligenz. Akademische Verlagsgesellschaft Aka GmbH / IOS Press, Berlin, Germany (2003)

33. Aamodt, A.: Explanation-driven case-based reasoning. In Stefan Wess, K.D.A., Richter, M., eds.: Topics in Case-Based Reasoning, Berlin, Springer-Verlag (1994)
34. Aamodt, A.: Knowledge-Intensive Case-Based Reasoning in CREEK. [13] 1–15
35. Bergmann, R., Althoff, K.D., Breen, S., Göker, M., Manago, M., Traphöner, R., Wess, S.: Developing Industrial Case-Based Resoning Applications: The INRECA Methodology. Second edn. LNAI 1612. Springer-Verlag, Berlin (2003)