A Categorization of Explanation Questions for Task Processing Systems

Deborah L. McGuinness[1], Alyssa Glass[1,2], Michael Wolverton[2], Paulo Pinheiro da Silva[3]

Stanford University[1]
Stanford, CA
[dlm, glass] @ ksl.stanford.edu

SRI International[2]
Menlo Park, CA
mjw@ai.sri.com

University of Texas, El Paso [3]
El Paso, TX
paulo@utep.edu

Abstract

A critical aspect of any explanation module is the set of user questions the system will be able to address. However, there has been relatively little work on listing and organizing the various categories of questions helpful to explanation. In this paper we address this problem by proposing a categorization of question types relevant to explaining task processing. For each question type, we also propose alternative *explanation strategies* for answering them. This categorization has helped to drive our work on an integrated cognitive explanation environment that has been used to explain the behavior of CALO, a software cognitive assistant that learns and organizes.

Introduction

As complex adaptive agents become increasingly autonomous and handle more complicated workloads on behalf of their users, building trust between user and system becomes more important. A critical step towards building this trust lies in the agent's ability to explain its reasoning, to provide transparency into both what the system is doing and what knowledge provenance was used to justify those actions. We envision a system in which explanation capabilities built around a question-and-answer dialogue between the user and the system is used to provide this transparency. This paper focuses on the explanations for task-oriented processing within such a system. We describe a set of general categories of task-oriented questions and answers. The questions and answers are instances of more general patterns, which are also described. Using a sample scenario representative of task processing, we provide specific example questions and answers. We also include alternative answers illustrating a spectrum of possible answers.

Using this categorization, we have created the Integrated Cognitive Explanation Environment (ICEE) which provides dialogue capabilities for explaining task processing in the Cognitive Assistant that Learns and Organizes [CALO, 2007] system. CALO uses the SPARK [Morley & Myers, 2004] belief-desire-intention (BDI) execution environment as the basis for its task processing, but our categorization is designed to apply to any cognitive task environment [McGuinness et. al, 2007]. In ICEE, we have implemented many but not all of the explanation strategies detailed in this paper. We note, however, that the example questions and answers categorized below are idealizations of the explanations provided in ICEE. Our system uses pre-generated templates based on the strategies categorized here and is integrated into the CALO user interface component. Full generalization of natural language is beyond our current implementation and beyond the scope of this paper.

The example questions in the following sections are based on a simple scenario concerning the purchase of a new laptop computer. The first subtask of buying a laptop is "Get Quotes," and it has no preconditions. This subtask terminates when it has obtained quotes from three different sources. The second subtask is "Get Approval." Before this task can be started, the system needs to know that three quotes have been obtained. This task is completed when a form is signed by an organization representative. The final subtask is "Send Order to Purchasing." Before this task can start, the system needs to know that the correct approval form was signed.

Task-Oriented Questions and Answers

We provide a categorization of questions and answers. Note that these categories are not hard and fast, and many questions could be placed in multiple categories. This categorization provides one view into the breadth of explanation possibilities for task-oriented systems and provides structure to help suggest explanation strategies.

Our intent in creating this categorization is to provide the creators of intelligent agents with guidance for the scope of explanation capabilities that should be provided to explain task processing. The list of questions and strategies within each category is not exhaustive. However, in building ICEE and in surveying the users of the CALO intelligent agent, we have found the included questions and strategies to be sufficient to cover explanations requirements. Additionally, this set has remained remarkably stable even as the task capabilities of CALO, the learning components, and the representations of the underlying task processor, have grown and matured. The stability that we have

observed in these question classes has given us confidence that other agent designers would similarly benefit from the sample questions we include here in each category.

Motivations for Tasks

These questions focus on the purpose for which the system is performing a task, and generally take the form of "Why?" questions. There are many ways the following question could be interpreted, depending on the context in which it is asked. The answer strategies below address different interpretations.

Question: Why are you doing <task>?

Answer Strategy 1: Identify the requestor.
Example Question: Why are you trying to buy a laptop?
Example Answer: Sally told me to.
Note that this strategy may be most useful when the task is a root task and thus not in service of other tasks.

Answer Strategy 2: Provide an abstraction of the task plan. Specifically, state that <subtask> is required for accomplishing the current goal, <supertask>. (Here, the above question is interpreted as "What goal or goals is <task> in service of?")
Variation: mention all subtasks at the same level as the current subtask required to accomplish the supertask.
Example Answer: I am trying to buy a laptop, and get approval is one subtask in the process.

Answer Strategy 3: Expose preconditions. Specifically, the reason the system is doing a task is that its preconditions have already been met, so identify those conditions. (Here, the above question is interpreted as "What conditions exist that allow you to do <task>?")
Example Answer: I am trying to buy a laptop, and I have completed the previous requirement to get quotes, so I am now working on get approval.

Answer Strategy 4: Identify termination conditions not yet met. Specifically, bind any termination requirements not yet satisfied and reveal them to the user. (Here, the above question is interpreted as containing an implied "still.")
Example Question: Why are you [still] doing "Get Approval?"
Example Answer: I am trying to buy a laptop computer and I can only do it if I have a form signed by the contracts office, and I do not have it yet.

Variations on Task Motivation Questions

The questions and answers in this section focus on explanations for task motivation that are not easily understood as "Why?" questions.

Question: What goal or goals is this in service of?
Approach: Abstract up the task hierarchy
Example Question: Why are you getting a quote for a laptop?
Example Answer: I am trying to purchase a laptop.
Answer Strategy 1: Keep abstracting up the hierarchy one step at a time.
Answer Strategy 2: Move up the hierarchy to previously annotated tasks that are known to be typically important, or important to users in a particular context.

Question: What conditions exist that allow you to do some task?
Approach: List preconditions and state that they are met.
Answer Strategy 1: Expose meta information on preconditions.
Answer Strategy 2: Expose dependencies between preconditions.

Question: Why do the conditions not exist to terminate this task? (Or alternatively: why are you still doing this task?)
Answer Strategy: Identify the termination conditions that are not met yet.

Task Status

These questions focus on the agent's current execution state. While some of the information needed for these questions is straightforward status information, other questions rely on being able to gather and consolidate execution information from disparate reasoners in a single agent, and then to abstract away the status that is too detailed for the current user and context.

Question: What are you (the system) doing?
Answer Strategy1: Name the current task
Example Answer: I am getting laptop quotes

Additional strategies can interpret this question as a rewording of "Why are you doing <task>?" and employ one of the strategies listed above.

Answer Strategy 2: State top level task highlighting any stalled goals (or any goals taking longer than expected)
Example Answer: I am trying to buy a laptop and get laptop quotes is stalled.

Question: What is the status of task <x>?
Answer Strategy 1: Return an answer from a predefined set of status values.
Example Question: What is the status of "Get Quotes?"
Example Answer: In process.

Answer Strategy 2: Provide details relative to progress toward meeting termination conditions.
Example Answer: I have quotes from Toshiba and IBM and have asked for a quote from Sony.

Question: Are you having trouble completing task <x>?
Answer Strategy: Check duration on current task with expected duration, and if it exceeds anticipation, report it.
Example Question: Are you having trouble completing "Get Quotes?"
Example Answer: I asked for a quote from IBM which usually takes 10 minutes and I have been waiting 2 hours.

Question: Are you having trouble completing any tasks?
Answer Strategy: Check all tasks in process and compare current duration with expected duration. Report tasks that are not yet completed that are over the expected duration.
Example Answer: On the laptop purchase for Paulo, I am waiting longer than anticipated for a quote from Dell. On the laptop purchase for Michael, I am waiting longer than anticipated for approval from Ray.

Task History

These questions focus on task execution that has already occurred. Like with the questions about task status, the ability to consolidate and then abstract detailed information is important both to provide useful answers and to avoid overwhelming the user.

Question: What have you done (recently)?
Answer Strategy 1: List recent top level completed tasks.
Example Question: What have you done recently?
Example Answer: I purchased a laptop for Paulo.

Answer Strategy 2: List current top level task and completed subtasks within that task.
Example Answer: In trying to purchase a laptop for Paulo, I got quotes and have not finished getting approval.

Answer Strategy 3: Summarize top level tasks completed.
Example Answer: I purchased 3 laptops (for Paulo, Michael, and Deborah) and I scheduled 2 meetings (for Deborah and Michael).

Answer Strategy 4: Provide a report of completed tasks.
Example Answer: <customizable table of main tasks>

Question: What have you started recently?
Answer Strategy 1: List top level tasks started.
Example Answer: I am trying to buy 2 laptops (for Paulo and Michael).

Answer Strategy 2: List top level tasks and subtasks.
Example Answer: I am trying to buy a laptop for Paulo and have obtained quotes and have not received approval, and I am trying to buy a laptop for Deborah and have not obtained three quotes yet.

Question: Why did you do <x>?
Comment: Often the implicit translation of this question is "Why did you do <x> instead of the (more commonly done) <y>?" The system needs to infer when the user is likely not asking for a description of the goal and precondition dependencies.
Answer Strategy 1: Check a statement for anything that is not typical and report differences.
Example Question: Why did you get approval from David [where David is not the normal approver]?
Bad Answer: I am trying to buy a laptop, and approval is required, and the approval process follows the quotes process, and the quotes process was complete.
Example Answer: The typical approver (Ray) was unavailable and David was a substitute approver.

Answer Strategy 2: Report failure modes and possible workarounds.
Example Answer: I tried to get approval from Ray and failed. I succeeded in getting approval from David.

Question: Why didn't you do <x> [in context of task <y>]?
Answer Strategy: Trace back to the condition that would have allowed <x>, and explain why it wasn't met.
Example Question: Why didn't you get Ray's approval?
Example Answer: He failed to respond within my time limit of two days.

Question: How did you do <x>?
Answer Strategy: State the sequence of subtasks performed to achieve <x>.
Example Question: How did you do "Get Quotes?"
Example Answer: I sent requests to the vendor websites for: Toshiba, Sony, Dell, and Gateway. I received responses from all of vendors. I was able to extract a quote from all but Gateway.

Question: How did you avoid the typical constraint <p>?
Answer Strategy: State the conditions that must be met to avoid <p>, and the sequence of subtasks taken to establish those conditions.
Example Question: How did you avoid the constraint that laptop cost should be under $2500?
Example Answer: I obtained a signed budget approval form (for the IBM TL1300 cost of $3200).

Question: Which task could you not complete on <past time point or period>?
Answer Strategy: List subtasks not completed at end of time period or time point.
Example Question: Which task could you not complete yesterday?
Example Answer: I was trying to complete the following tasks: (1) Get a second quote for laptop Sony Vaio; (2) Get a first quote for laptop IBM ThinkPad.

Question: Why could you not complete <task> on <past time point or period>?
Answer Strategy: Expose a failed precondition
Example Question: Why could you not "Get Quotes" for Paulo's laptop purchase yesterday?
Example Answer: I could only buy a laptop if I had a form signed by the contractor's office and I did not have it.

Task Plans

While the previous sections handled questions about current and past execution, these questions focus on projection, requiring planning about what the agent will do in the future under (possibly hypothetical) future scenarios.

Question: What are you doing next?
Answer Strategy 1: Identify tasks supporting the same higher level task type that have the current task as a precondition.
Example Question: What are you doing next?
Example Answer: [Assume the system is currently getting quotes.] I will be getting approval to purchase the laptop.

Question: When will you start task <x>?
Answer Strategy 1: Provide a relative answer identifying the precondition(s) needed to complete prior to task <x>
Example Question: When will you send the laptop order to purchasing?
Example Answer: After I have approval to purchase. (I have already obtained three quotes).

Answer Strategy 2: Provide a time estimate based on meta information or statistical information about time required to complete unsatisfied preconditions.
Example Answer: I still have to obtain approval to purchase which typically takes 2 days.

Question: Why do you expect to do task <x> next?
Answer Strategy 1: Assumes the system is using a classical AI planning approach. The answer traces through expected effects and preconditions in the next portion of the plan.
Example Question: Why are you getting approval next?
Example Answer: At the end of my current task, I expect to have 3 quotes for laptops. Once we have the quotes, the only remaining requirement before sending the purchase order is getting approval from management.

Answer Strategy 2: Assumes the system has no explicit planning capability aside from a procedural task execution system. The answer is based on determining the usual course of action from situations similar to the current one.
Example Answer: I'm currently finishing obtaining quotes. In purchases of laptops less than $2500, the typical next step is to obtain approval from management.

Question: How will you do task <x>?
Answer Strategy: Regardless of whether the system is using a planner or whether we construct an expected "plan" by analyzing past examples, the approach here is to describe the subtasks in the plan, including the particular conditions that lead to that expansion.
Example Question: How will you get approval?
Example Answer: Because this laptop is being purchased with project funds, and because it is needed in a rush, approval from management is not required. I will request approval from the project PI, Michael.

Task Ordering

These questions focus on the explicit ordering of tasks and subtasks. Like the task planning section, this section can also provide the user with the ability to ask hypothetical questions about why tasks are being done in specific ways.

Question: Why are you doing task <x> before task <y>?
Answer Strategy 1: Identify if there are direct precondition relationships between the tasks and if so, note them.
Example Question: Why are you doing "Get Quotes" before "Get Approval?"
Example Answer: Getting laptop approval requires three quotes to be obtained before the process can start.

Answer Strategy 2: Identify any transitive precondition relationships between the tasks and if so note them.
Example Question: Why are you doing "Get Quotes" before sending the order to purchasing?
Example Answer: Sending a request to purchasing requires obtaining approval, which requires getting quotes.

Question: Why have you not started task <x> yet?
Answer Strategy: Identify unsatisfied preconditions.
Example Question: Why have you not sent Deborah's laptop request to purchasing?
Example Answer: I am waiting for approval. (Quotes have been obtained and the request for approval was sent yesterday.)

Question: What remains to be done to complete task <y>?
Answer Strategy 1: Report unsatisfied direct preconditions.
Example Question: What do you still need to do before sending the laptop order over to purchasing?
Example Answer: I need to get approval to order it.

Answer Strategy 2: Report unsatisfied direct and transitive preconditions.
Example Answer: I need to get approval to order it, and prior to that I need to obtain three quotes.

Explicit Time Questions

These questions focus on providing the user with the ability to probe the agent's explicit reasoning about time,

with both qualitative (ordering) queries and quantitative (time points and time spans) queries.

Question: When will task <x> begin?
Answer Strategy 1: Provide a relative answer that is after the preconditions are completed:
Example Question: When will "Get Approval" begin?
Example Answer: After I have obtained three quotes where one costs under $2500 or I have approval to spend more from a manager of level 3 or above.

Answer Strategy 2: Provide a time estimate based on meta information or statistical information about task duration.
Example Answer: Getting laptop quotes typically takes 2 days. However, if this process requires special approval for a laptop costing more than $2500, it may take 4 days.

Question: When will task <x> end?
Answer Strategy 1: Provide a time estimate based on meta information or statistical information about task duration.
Example Question: When will "Get Quotes" end?
Example Answer: Typically one can obtain quotes in two days, but I have had trouble in the past with obtaining quotes from IBM, so it may take longer.

Answer Strategy 2: If termination condition is non-obvious and unsatisfied, expose the termination condition.
Example Answer: I will not be able to finish until I can access the Dell website.

Question: When did task <x> happen?
Answer Strategy 1: Report the location of <x> on timeline.
Example Question: When did you send the laptop request for Deborah's laptop purchase to purchasing?
Example Answer: On October 31, 2004 at 10am.

Answer Strategy 2: Provide a relative answer identifying the preconditions that were completed before this task.
Example Answer: After I obtained three quotes (on October 29), and received approval (on October 30), I sent the laptop request to purchasing (on October 31).

Question: How long did it take you to complete task <x>?
Answer strategy 1: Report the difference between the time point of the initial request for task x and the time point of the last termination condition for task x.
Example Question: How long did it take you to order Deborah's laptop?
Example Answer: 4 days.

Answer strategy 2: Same as above, but expose time points.
Example Answer: 4 days. I received the request on Monday and completed the purchase on Friday.

Question: Why did task <x> take so long?
Answer Strategy: Find previous subtasks that took longer than expected (based on meta-information or statistical information), and describe the discrepancy.
Example Question: Why did "Get Approval" take so long?
Example Answer: To get approval, I need to fill out the approval form and obtain management signature. Filling out the approval form is normally completed within 4 hours of the request, but in this case, it took two days.

Question: Why are you doing task <x> already?
Answer strategy: This is the complement of the previous strategy. Find past subtasks that finished faster than expected and describe the discrepancy.
Example Question: Why are you sending the purchase order already?
Example Answer: "Get Approval" took much less time than expected. Normally, it takes 24 hours; with this purchase, however, it took only 10 minutes.

Discussion and Related Work

Our work focuses on a categorization of explanation types and answer strategies for environments that include a task execution component. Others have worked on explanation descriptions with varying goals. For example, [Leake, 1992] focuses on evaluating explanations, but in that context he provides perspective on types of explanations, focused on anomalies. While we are also interested in anomalies and evaluation, the primary focus of our categorization task was to provide a foundation on which to describe questions, thus providing an organization and structure for explanation strategies. In later work, Leake and colleagues [Leake, et al, 1999] examined how to provide relevant task information in a proactive manner from cases in an automotive parts stamping advisor. This work, however, focused on anticipatory (rather than reactive) explanations in a case-based approach, as opposed to a general BDI framework as in CALO and similar agent systems. Initial results from a trust study we conducted among CALO users shows that users require distinct types of explanations when actively asking questions rather than being proactively provided with information.

There has also been work on explaining knowledge-based systems, for instance, in explaining the HALO system [Barker, et. al, 2004]. One of the strengths of this work is its support for user-generated "explanation frames" associated with facts requiring explanations. This approach tends to produce natural explanations but may be more specific to domains and labor intensive to generate. Our approach of identifying general question types and general explanation strategies based on the structure of the question and representation of the general task processing environment could be used in combination with such template-based knowledge engineering-based approaches.

A number of researchers have categorized explanation types, goals, and/or criteria; [Sørmo et al., 2005] provides a useful survey of these efforts (and of other analyses of explanations). Many of the previous categorizations have focused on the high level, partitioning the general concept of explanation into a handful of high-level explanation types. Our work, which focuses on lower-level question types for a particular area of explanation, can be thought of as an extension of many of these previous taxonomies. For example, most of questions and question types described in this paper could be considered *ratification* explanations as defined by [Wick & Thompson, 1992], or *goal-based* explanations as defined by [Keil & Wilson, 2000], or *cognitive explanations* as defined by [Spieker, 1991] (via the English summarization in [Roth-Berghofer, 2004]). A key contribution of our categorization is that it focuses specifically on explanations for task processing. This focus means that the explanation categories explored here deal with task-related factors largely uncovered by previous categorizations—especially temporal relations, the distinction between goal decomposition and gating conditions, and the distinction between initiation and termination conditions.

A different approach to categorizing questions and answers appears in [Belnap & Steel, 1976], which focuses on developing a formal logical notation for representing questions, based on a categorization of the expected answers. In addition to being mainly a notational, rather than operational, categorization, this work also focuses exclusively on questions that have "direct answers," as would be asked through queries to a database system. Our focus on questions designed to build trust in users of more complex and adaptive systems requires the inclusion of more ambiguously asked questions, for which the questioner himself may not know the form of the answer that is needed. Our illustration of multiple strategies for the same question type, and the encouragement of a dialogue of follow-up questions, as provided in ICEE, addresses this ambiguity problem and enables us to categorize and address a larger class of questions, including the important category of "Why?" questions that is not effectively addressed in the previous categorization.

[Ram 1994] proposes perhaps the most detailed taxonomy of explanation questions developed to date. The work provides a number of categories, subcategories, and specific questions useful for story understanding. While some of the types Ram describes provide interesting insight for all types of explanations, many of them deal with the subtleties of human behavior reflected in stories—e.g., interpersonal interactions, social control, interference from other actors, etc. The computerized assistant being explained in our work typically won't exhibit these same subtleties in its behavior.

Some issues that we are addressing in our work on answer strategies include considerations related to level of granularity. As with the early days of explanation generation, our work needs to consider not just presenting an entire line of reasoning used to generate an answer (even along any single particular strategy), but also issues of abstraction and pruning. Our system must choose what information to include and how much detail to present. For example, when considering why a process has not completed, the system needs to identify which failed termination conditions should be presented. Similarly, when answering questions about why a task is being executed, the system must choose which satisfied preconditions should be included. Another topic of active research is context. One current issue we are considering is the degree of user context (potentially in combination with learned or volunteered preference information) to exploit in selecting answer strategies, choosing explanation content, and suggesting follow-up questions.

The setting in which we have implemented the explanation approach outlined in this paper is described in [McGuinness, et. al, 2007]. Many of the strategies categorized in this paper have been included in the ICEE system, which uses the Inference Web explanation framework [McGuinness, et. al, 2004] for generating, manipulating, and presenting explanations. ICEE is used inside the CALO adaptive cognitive agent to explain the task processing of an executive office assistant [Myers, et. al., 2007]. This BDI-based task execution system, populated with processes both hand-written and learned through a wide variety of machine learning techniques, has provided us with an excellent test bed to evaluate our explanation categories and strategies.

The implementation of these strategies, however, is not sufficient to motivate the explanation framework. The ultimate goal of including an explanation framework as part of an agent architecture is the utility provided to the users of the agents. To examine this aspect of the framework, we initiated a user trust study of users of the CALO cognitive assistant, in an effort to identify what types of information they needed in explanation facilities in order to trust automated agents. We are currently analyzing the results, and modifying and augmenting our strategies as needed, to reflect input from this study. Our initial analysis confirms the importance of these explanation categories for providing the type of system transparency required by users to build trust in task processing systems and cognitive agents.

Conclusions

In this paper, we have presented a categorization of explanation questions for settings that include task execution environments. For each category of questions, we have identified answer strategies. These strategies, while not exhaustive, illustrate the breadth of the explanation possibilities within each of the explanation categories. We have implemented many of these strategies and embedded them in ICEE, our explanation environment. ICEE has been used to provide explanations of the CALO system, particularly oriented towards explaining the task execution system and several of its learning modules. Our initial user studies show that these explanations are an important step toward user trust and acceptance of cognitive agents.

Acknowledgements

We gratefully acknowledge funding support for this effort from the Defense Advanced Research Agency (DARPA) through contract #55-300000680 to-2 R2. We thank Cynthia Chang, Li Ding, Vasco Furtado, Karen Myers, Vinay Chaudhri, and the anonymous ExaCt reviewers for helpful collaborations and feedback on this work.

References

K. Barker, V. Chaudhri, S. Chaw, P. Clark, J. Fan, D. Israel, S. Mishra, B. Porter, P. Romero, D. Tecuci, P. Yeh. A Question-Answering System for AP Chemistry: Assessing KR&R Technologies. In the *Proceedings of the 9th International Conference on Knowledge Representation and Reasoning (KR'04)*, 2004.

N. D. Belnap and T. B. Steel. *The Logic of Questions and Answers.* Yale University Press, 1976.

CALO, 2007. http://www.ai.sri.com/project/CALO

F. C. Keil and R. A. Wilson (eds.). *Explanation and Cognition*, Bradford Books, Boston, MA, 2000.

D. B. Leake. (1992). *Evaluating Explanations: A Content Theory*. Hillsdale, NJ: Lawrence Erlbaum Associates.

D. Leake, L. Birnbaum, C. Marlow, and H. Yang. Task-Based Knowledge Mangement. *Proceedings of the AAAI-99 Workshop on Exploring Synergies of Knowledge Management and Case-Based Reasoning.* 1999.

D. L. McGuinness, A. Glass, M. Wolverton and P. Pinheiro da Silva. Explaining Task Processing in Cognitive Assistants That Learn. In the *Proceedings of the Twentieth International FLAIRS Conference*., Key West, Florida, 2007. ksl.stanford.edu/KSL_Abstracts/KSL-07-03.html

D. L. McGuinness and P. Pinheiro da Silva. Explaining Answers from the Semantic Web: The Inference Web Approach. *Web Semantics: Science, Services and Agents on the World Wide Web* 1(4), 2004. ksl.stanford.edu/KSL_Abstracts/KSL-04-03.html

D. Morley and K. Myers. The SPARK Agent Framework. In the *Proceedings of the 3rd Intl Joint Conference on Autonomous Agents and Multi Agent Systems* (AAMAS-04), New York, NY: 712-719, 2004. www.ai.sri.com/~spark/

K. Myers, P. Berry, J. Blythe, K. Conley, M. Gervasio, D. McGuinness, D. Morley, A. Pfeffer, M. Pollack, and M. Tambe, "An Intelligent Personal Assistant for Task and Time Management," to appear in *AI Magazine*, 2007.

A. Ram. AQUA: Questions that Drive the Explanation Process. *Inside Case-Based Explanation*, R.C. Schank, A. Kass, and C. K. Riesbeck (eds.), 207-261, Lawrence Erlbaum, 1994.

T. R. Roth-Berghofer. Explanations and Case-Based Reasoning: In the *Proceedings of the Foundational Issues Advances in Case-Based Reasoning. ECCBR-2004*. September 2004.

F. Sørmo, J. Cassens, and A. Aamodt. Explanation in Case-Based Reasoning - Perspectives and Goals. *Artificial Intelligence Review*, 24(2):109-143, October 2005.

P. Spieker, Natürlichsprachliche Erklärungen in technischen Expertensystemen, Dissertation, University of Kaiserslautern, 1991.

M. R. Wick and W. B. Thompson. Reconstructive Expert System Explanation, *Artificial Intelligence*, 54(1-2), pp. 33-70, 1992.