Mingyang Gu

Knowledge-Intensive Conversational Case-Based Reasoning in Software Component Retrieval

Doctoral Thesis
for the degree of Philosophiae Doctor (PhD)

Trondheim, August 2006

Department of Computer and Information Science
Norwegian University of Science and Technology

NTNU
Innovation and Creativity

Doctoral Thesis
for the degree of Philosophiae Doctor (PhD)

Department of Computer and Information Science
Faculty of Information Technology, Mathematics and Electrical Engineering
Norwegian University of Science and Technology
Trondheim, Norway

ISBN 82-471-8002-2 (printed version)
ISBN 82-471-8001-4 (electronic version)
ISSN 1503-8181

Doctoral Thesis at NTNU, 2006:121

printed by NTNU-trykk, October 15, 2006

Contents

List of Figures

Abstract

Case based reasoning (CBR) is a problem solving method that reuses the previous problem solving experiences (represented as cases) to solve the new problem. As a type of interactive CBR, conversational CBR has been proposed to help users construct their problem descriptions incrementally through a mixed-initiative question-answering sequence. In software component retrieval, users meet the difficulty in well-defining their component queries. As a solution to release this difficulty, we proposed and implemented a conversational component retrieval model (CCRM) using a knowledge-intensive conversational CBR method in the thesis.

The research activities and contributions followed two directions: theoretical research on conversational CBR to provide an efficient and natural conversation process, and applying and adapting conversational CBR to support software component retrieval.

In the theoretical research direction, we first provided a framework to classify the similarity methods in CBR from the perspective of what features were taken into account during the similarity calculation, and further analyzed and illustrated that the query-biased similarity methods (only considering the features appearing in the query during the similarity calculation) were more suitable for conversational CBR applications. A knowledge-intensive conversational CBR method was designed that was able to utilize the general domain knowledge to improve the efficiency and naturalness of the conversation process. Four knowledge-intensive question selection tasks, including the feature inferencing, the integrated question ranking, the consistent question clustering, and the coherent question sequencing, were identified and handled in this method. We also proposed a lazy dialog learning mechanism that could continuously improve the performance of conversational CBR.

Following the conversational component retrieval application direction, we reviewed and analyzed the current software component retrieval methods and proposed a conversational component retrieval model. In order to represent both the software components and the component queries as cases, it is necessary for a case to have multiple values on some features (generalized cases). In the research, we analyzed the feasibilities and discussed the methods to extend conversational CBR to support generalized cases from three aspects: the case representation, the similarity calculation metric, and the question selection method. At the end, a knowledge-intensive conversational software component retrieval system (TrollCCRM), enhanced by the above research findings, was implemented and evaluated on the image processing software component retrieval application.

The evaluation results so far gave us positive results that the TrollCCRM system provided an efficient and natural conversation process guiding users to find their desired software components.

Preface

I started my PhD process in the software engineering group and the research topic was 'Creativities and Constraints in Software Development'. At that time, I investigated open source software (OSS) for new ideas. When I explored the OSS project collection website 'http://sourceforge.net/', I found it was very difficult to locate a desired software project out of tens of thousands of available projects. In most of the cases, I simply did not know how to use the website-allowed language to specify my searching requirements. I felt that this was a big problem with the rapidly increasing number of software components. Later, I moved to the artificial intelligence group and changed my research topic to intelligent software component retrieval. The present thesis reports our research efforts about how to use an AI method, i.e., conversational CBR, to help users handling the difficulty in well defining their component queries.

This thesis is submitted to the Norwegian University of Science and Technology in partial fulfillment of the requirements for the doctoral degree. The work reported in the thesis was carried out and funded by the Department of Computer and Information Science, under the supervision of Professor Agnar Aamodt. The thesis is organized as a collection of papers. Eight published papers are included in the second part of this thesis, and the first part, called 'Research Overview and Summary', presents the research context and describes the papers from the perspective of an integrated research structure.

Acknowledgments

First and foremost, I would like to thank my supervisor, Professor Agnar Aamodt, for his patient guidance, precious advice, and continuous support. My gratitude also goes to my co-advisors, Associate Professor Ketil Bø, and Dr. Torgeir Dingsøyr for their time and discussion.

I would like to thank Frode Sørmo and all the other people who developed the TrollCreek system on which my research implementation was based. Thanks to Jörg Cassens, Jingyue Li, and Jianyun Zhou for their kind help at the stage of the system evaluation.

I owe a thank you to Professor Maria Letizia Jaccheri, Dr. Tore Dybå, and Professor Torbjørn Skramstad for their guidance when I was in the software engineering group. My thanks go to Professor Reidar Conradi and Professor Eric Monteiro for their help during the period I changed my research topic.

I would like to thank all my colleagues for their inspiring discussions and kind help, in particular Pinar Öztürk, Anders Kofod-Petersen, Peep Küngas, Waclaw Kusnierczyk, and Rune Sætre. They made the four-year PhD study such a memorial period.

My sincere thanks go to all my family members back in China for always being there. My deepest thanks go to my wife, Xin, for her great love and support in many ways. Finally, I would like to thank our daughter, Wenqi, for her lovely smiles and encouragement using her own magic words.

Mingyang Gu
October 15, 2006

Part I

Research Overview and Summary

Chapter 1

Introduction

Case based reasoning (CBR) (Aamodt and Plaza 1994; Kolodner 1993) is a problem solving and machine learning method. The main idea is to reuse previous related experiences, called cases, to solve a new problem.

In the traditional CBR process, it is assumed that the new problem is well-defined before it is used to retrieve similar cases. However, this assumption is not always realistic. The knowledge gap between the current user and the case provider restricts the user from describing her problem in a way that is consistent with the representation of similar cases (Bergmann et al. 2001). The high acquisition cost of a complete problem description is another factor that makes this assumption unrealistic (Cunningham and Smyth 1994). Conversational CBR has been proposed to solve the problem of the problem description (query) construction (Aha, Breslow, and Muñoz-Avila 2001; Aha, McSherry, and Yang 2006). Conversational CBR provides a mixed-initiative dialog to guide users in constructing their queries incrementally through a question-answering process.

One of the main research topics in conversational CBR is question selection, i.e., how to select the most discriminative questions and how to display them in a natural way during a query-acquiring dialogue to alleviate users' cognitive load (Schmitt and Bergmann 2001; Shimazu 2002). Recently, there has been plenty of research concentrating on this topic in the conversational CBR community. The research presented in this thesis also followed this line. Far from the final word on the subject, this thesis is the author's attempt to identify the possible aspects influencing the efficiency and naturalness of the conversation process in conversational CBR and to design corresponding methods to handle them.

This thesis takes the form of 'paper collection' and is divided into two parts. The first part, called 'Research Overview and Summary', provides the contextual description for the collected papers and puts them in the perspective of an integrated research structure. The second part includes the eight papers leading to the present thesis.

The first part is organized into four chapters. The current chapter gives the background knowledge about my research, including CBR and conversational

CBR, my research activities and the contribution list. Chapter 2 summarises the papers from the perspective of an integrated research structure. Chapter 3 provides mistake corrections and supplements for each collected paper. The last chapter includes the research conclusions and the author's vision for the further development.

1.1 Case-Based Reasoning

CBR is a type of analogical problem solving and lazy machine learning method. Previous problem solving experiences are organized into episodic cases (i.e., stored cases) and stored in the case base. A new problem, captured as a target case or a new case, is solved by applying the solution adapted from that of the most similar stored case(s) (correct case(s)).

The content of a case includes the following three parts (Kolodner 1993):

- Problem description: the state of the world at the time of the case, and, if appropriate, what problem needed to be solved at that time.
- Solution description: the stated or derived solution to the problem specified in the problem description.
- Outcome: the resulting state of the world after the solution was carried out.

As summarized in (Aamodt and Plaza 1994), the problem solving process in CBR contains four steps: RETRIEVE, REUSE, REVISE and RETAIN (Fig. 1.1). Given a new problem description, a CBR system transforms it into a new case (only containing the problem description part compared to the content of a stored case). This new case is used to **retrieve** the correct case/cases from the case base through calculating the similarities between the new case and each stored case. The problem solving information or knowledge contained in the correct case(s) (i.e., the solution description part and the outcome part) is **reused** to devise a new solution to the current new problem. The devised solution is evaluated in a simulated or real environment and the evaluation result is used to **revise** the devised solution. The newly gained problem solving experience likely to be useful in the future is captured and **retained** in the case base as a new stored case.

1.2 Conversational Case-Based Reasoning

In traditional CBR, it is assumed that a user can well define her problem, and a CBR system can use it to find the correct case from the case base. However, in most of the cases, there is a knowledge gap between the user and the case author. That is, the user normally has less expertise than the author and she

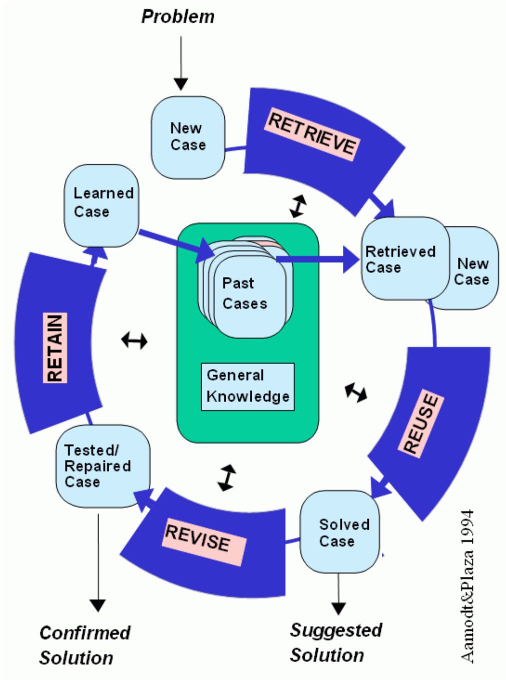

Figure 1.1: The case-based reasoning cycle (from (Aamodt and Plaza 1994))

does not know how to describe the problem properly, such as what terms to describe the problem and to what degree to describe the problem is enough for a CBR system to retrieve the correct case. As an example of the knowledge gap, the case base author may describe stored cases with technical features, while the user may describe her query by desired functionalities (Vollrath 1998). The high acquisition cost of a complete problem description is another reason why a complete problem description is not always available. Normally the user needs to execute a set of tests to find the properties of the problem, and these tests cost a lot of resources. Instead of forcing the user to guess the problem description or to complete all the property identification tests, conversational CBR has been proposed to provide an efficient dialog process to alleviate users' difficulties in constructing problem descriptions.

Conversational CBR is a type of interactive CBR that provides a mix-initiative dialog guiding a user to construct her problem description incrementally through a question-answering sequence. As illustrated in Fig. 1.2, a user's initial problem description is transformed into an initial new case (only containing one or a few features). This initial new case is used to retrieve a set of most similar stored cases from the case base. A group of discriminative questions are identified based on the returned cases, and ranked according to their capabilities to discriminate the stored cases. Both the returned cases sorted according to their similarity values and the ranked questions are displayed to the user. The user either finds

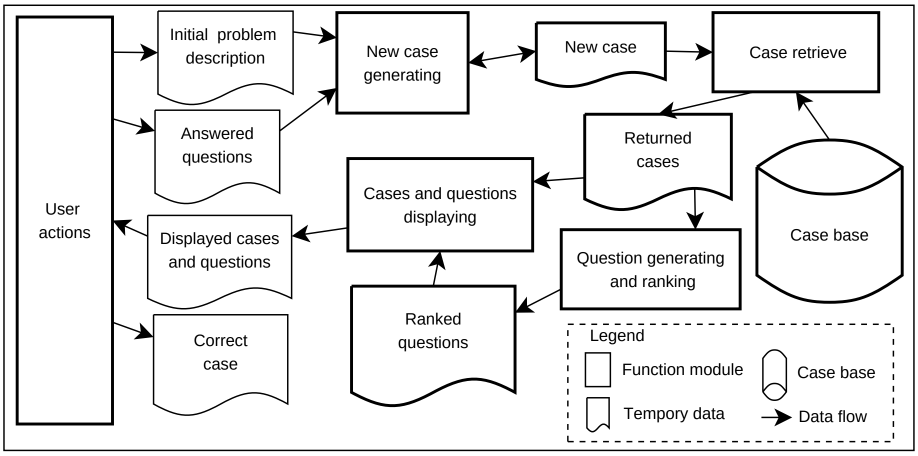

Figure 1.2: The conversational case-based reasoning process

a satisfactory stored case, which then terminates the case retrieval phase, or chooses one or more questions to answer. The newly gained answers and the current new case are combined to construct an updated new case. A new round of retrieval and question answering is started, and the iteration continues until the user finds a satisfactory stored case or there are no discriminative questions left to choose.

As summarized in (Aha, Breslow, and Muñoz-Avila 2001), the problem description of a case in conversational CBR is represented using a set of <question, answer> pairs. In the thesis, we use <feature, value> pairs to represent the problem description of a case. During a dialogue process, a feature can be transformed into a user-readable question, and the corresponding feature value is the answer.

One of the major research topics in conversational CBR is how to alleviate users' cognitive load in the question-answering process. The efforts to handle this task are divided into two research lines: designing question-ranking metrics to ensure that the more discriminative questions are asked at earlier stages so as to find the correct case by using the shortest conversation, and designing question-displaying strategies to provide a natural question-answering process.

Following the first research line, the proposed question-ranking metrics in the conversational CBR research community include:

- Information Gain Metric

  Information gain (Quinlan 1986) is most commonly used in the fields of information theory and machine learning. For example, in a decision tree algorithm, information gain is used to choose the most valuable feature for 'splitting'. Based on the entropy, information gain measures how well the feature separates the cases according to their solutions. If the feature is very discriminative, the expected entropy after using this feature will be close to 0 and the information gain will be close to 1. The feature with a higher information gain value gets a higher priority to be transformed into a discriminative question and displayed to users. Information gain metrics have been adopted to rank discriminative questions in several conversational CBR systems, including (Shimazu 2002; Cunningham and Smyth 1994; Carrick et al. 1999; Göker and Thompson 2000).

In (McSherry 2001b), the author further showed that if the solution description part of each case in a case base is unique (so-called irreducible case base), the information gain of a feature is calculated by the following equation:

$$-\sum_i p_i log p_i \tag{1.1}$$

where $p_1, p_2, ..., p_r$ are the propositions of this feature's values in the counted cases.

Unlike decision tree learning, where information gains are calculated statically at the tree construction stage, conversational CBR calculates information gains dynamically in each question-ranking session at runtime. In decision tree learning, only the feature with the largest information gain value is selected at the point of a 'decision making' node. In conversational CBR, a group of most informative features are selected, transformed into questions for users to inspect instead of forcing users to follow a static question-answering sequence. A dialogue process by applying the decision tree algorithm might have problems, as pointed out in (Schmitt 2002), 'if a static decision tree process is adopted in the question selection process, it can cause the effect that a dialog might be terminated too early without the user having found the product she was looking for (it basically uses an exact matching process instead of a partial matching)'. Conversational CBR is such a method using partial matching to help users find the most appropriate cases.

- Occurrence Frequency Metric

  In NaCoDAE (Navy Conversational Decision Aids Environment) (Aha, Breslow, and Muñoz-Avila 2001), the identified discriminative questions are ranked according to their frequencies of appearance in the problem description part of the returned cases. The larger the number of the returned cases in which one feature is assigned a value, the higher ranking priority that feature gets. In this metric, it is assumed that the cases are highly heterogeneous, that is, a feature appearing in some cases may not appear in other cases.

- Importance Weight Metric

One problem in CBR is the curse of the dimensionality (Mitchell 1997), that is, not all the available features are relevant or equally important in the similarity calculation process. Feature-weighting methods can alleviate this problem by assigning features with different importance weights according to their contributions, such as EACH (Salzberg 1991) and Relief (Kira and Rendell 1992). The importance weight-based question-ranking metric ranks the discriminative questions according to the weight value of the corresponding feature. The idea behind this metric is that the most relevant or important features can provide more information than other features to discriminate cases from one another. This method has been used in (McSherry 2003; Gu, Tong, and Aamodt 2005; Gu and Aamodt 2006a).

There are different knowledge resources in a CBR system (Richter 1995): the case base, the similarity measure, the adaptation knowledge, and the vocabulary. If the importance weights of features are learned through a similarity calculation process, the conversational CBR system using the importance weight metric implicitly takes the knowledge encoded in the similarity measures into account during the question-ranking process.

- Similarity Variance Metric

  In (Schmitt 2002), a similarity-influenced information measure (SimVar) was proposed. The author concentrated on the fact that each case retrieval method is based on the similarity values between stored cases and a new case and the goal state of the retrieval is to identify the correct case which has enough similarity variance from the rest of the stored cases. In this method, a feature that can provide a higher similarity variance in the candidate cases gets a higher priority to be selected to form a question. A term called discriminative power is introduced to describe how strong a feature can be used to discriminate the candidate cases using the SimVar measure.

Each question-ranking metric has its own particular application situation; e.g., the information gain metric is suitable for applications where there are not so much missing data, while the occurrence frequency metric is applicable to situations with plenty of missing data.

The following research efforts are classified into the second research line, i.e., providing a natural question-answering process in conversational CBR.

- Explaining the Question Relevance

  Inspired by the evidence-gathering strategies used by medical doctors, McSherry proposed a question selection method called Strategist (McSherry 2001a). This method provides four feature-selection strategies, CONFIRM, ELIMINATE, VALIDATE, and OPPOSE, listed in order of priority. A feature supports the CONFIRM strategy, if it has a value that occurs only in the returned cases with the target solution class. A feature supports the ELIMINATE strategy if one of its values occurs in the returned cases with the target solution class but not in those with the likeliest alternative solution class. A feature supports the VALIDATE strategy if one of its values is more likely in the cases with the target solution class than those with any other class. And a feature supports the OPPOSE strategy if one of its values is less likely in the cases with the likeliest alternative solution class than those with any other class. The relevance of a question proposed by a conversational CBR system can be explained by the strategic terms (McSherry 1999), as described above.

- Pruning the Trivial or Repeated Question

  If a candidate question can be answered by exploiting common sense knowledge or the knowledge users have already provided, this question should be answered automatically instead of posing it to users. In (Aha, Maney, and Breslow 1998), the authors proposed a model-based dialogue inferencing method. In their method, the general domain knowledge is represented in a model library (including object models and question models) taking the form of a semantic network. At run time, a set of rules are extracted from the model library using an implication rule generator, and the generated rules and the existing problem description are inputted into a query retrieval system (PARKA-DB) to infer the implicit knowledge.

  Carrick et al. designed an architecture where the features of a new case were considered as questions or information acquisition tasks to be executed on multiple information sources, including databases, the internet, and users themselves. The trivial and the repeated questions that can be answered automatically through accessing other information sources would be eliminated from users (Carrick et al. 1999).

  With the same intention, user modeling technologies are used to filter out the candidate questions that can be answered by pre-collected user profiles. In the Adaptive Place Advisor system (Göker and Thompson 2000), user preference information is used to construct the initial query, and the questions whose answers can be found in this initial query will not be asked any more. Furthermore, in the WebSell project (Cunningham et al. 2001; Bergmann et al. 2002), a user's profile is used to identify the community this user belongs to, and the displayed cases are drawn from the profiles of the entire members of the community.

- Question Answering Cost

  The difficulties or costs to answer different questions are various, and people normally hope to answer questions following a sequence from easy to difficult. In (Schmitt 2002), the answering cost of a question is detected by the user's possible reaction to the corresponding question (Easily answered, Answered with difficulty, Don't care, Don't understand), and the expected utility of this question is calculated through combining its discriminative power and its possible answering cost. A method based on such expected utilities is proposed to guide the question selection process, which can adapt to users' capabilities or possible costs to answer the questions.

In (Carrick et al. 1999), the authors also took the question-answering costs into account when selecting a task (question) to execute instead of only counting its information quality metric value. In this method, an execution plan for each question is formulated using a hierarchical task network (HTN). The estimated cost for each question is calculated through propagating cost values upward from leaves to the root by using the mini-max algorithm.

- Taxonomic Conversational CBR

  In (Gupta 2001), the author proposed a taxonomic conversational CBR approach to tackle the problems caused by the abstraction relations among features. In this approach, a case is described using several features. For each feature, an independent subsumption taxonomy of its values is created by a library designer in advance. If more than one value in a feature taxonomy occurs in one case, only the most specific value is selected to describe the case. The similarity of a feature between a new case and a stored case is calculated based on the relative positions of their feature values in the taxonomy structure. The question generated from a higher-level feature value in a feature taxonomy is constrained to be asked before those that come from the lower-level feature values.

- Causal Conversational CBR

  The authors in (Aha and Gupta 2002; Gupta, Aha, and Sandhu 2002) identified the dependency relations among features and proposed a method to elaborate the causal relations in conversational CBR. In their method, if both the post-condition feature and the pre-condition feature in a dependency relation appear in the case description, the post-condition feature will be excluded from the case description. In the question-ranking process, the question generated from a post-condition feature has a higher priority to be asked than the question formalized using the pre-condition feature.

Other research activities in the conversational CBR community that cannot be clearly classified into the 'efficiency' category or the 'naturalness' category include:

- Conversational Case Base Maintenance

  In CLIRE (Aha and Breslow 1997), an algorithm was proposed to refine the conversational case base. In this algorithm, a decision tree is constructed to index all the cases, in which the occurrence frequency metric is used to select the branching features, and the cases with the vacant value on one branching feature are classified as a new branch under the corresponding feature node. For each case, the features that have not been used in the path from the root node to the leaf node containing the case itself are pruned.

- Termination Criterion of the Conversation

  In (McSherry 2003), the author discussed the balance between the conversation efficiency and the solution quality in conversational CBR. He argued that the normal dialog termination criteria, e.g., the pre-defined similarity threshold, did not guarantee that the recommended solution was not sub-optimal. He further proposed a dominance-based dialog termination criterion that provided necessary and sufficient conditions for a dialog to be terminated with a global-optimal solution. In his research, one case, $C1$, is said to be dominated by another case, $C2$, with respect to an incomplete query, Q, if $Sim(C2, Q) < Sim(C1, Q)$, and $Sim(C2, Q^+) < Sim(C1, Q^+)$ for all possible completions, Q^+ of Q. The safe dialogue termination condition is defined as 'if and only if the current query Q is such that: (a) all the returned cases have the same values on all the features that are not specified in the query and (b) all the non-returned cases are dominated by all the returned cases'.

- Coverage of the Retrieved Cases

  The similarity-based retrieval criteria cause the problem that all the displayed cases tend to be similar to each other; therefore users are offered very limited alternatives to choose. In other words, the displayed cases may not be able to provide sufficient representatives of the compromises that the user may be willing to accept. In (Smyth and McClave 2001), the authors discussed the problem of similarity and diversity of the displayed cases and proposed several methods to improve the diversity. Later, McSherry (McSherry 2004a) proposed the CORE (Coverage-Optimized Retrieval) approach, ensuring that 'for any case that is acceptable to the user, one of the recommended cases is at least as good in an objective sense and so also likely to be acceptable'. The coverage-optimized retrieval algorithm in the CORE approach is illustrated in Table 1.1, in which Q is the target query and $Cases$ is a list of candidate stored cases, sorted in order of non-increasing similarity to the target query.

Besides the theoretic research in conversational CBR, there are plenty of conversational CBR applications. For example, the product or service recommendation systems, including ExpertClerk (Shimazu 2002), iCARE (Doody et al. 2006), WEBSELL (Cunningham et al. 2001), FindMe (Burke, Hammond, and Yound 1997), and Adaptive Place Advisor (Göker and Thompson 2000), the

Table 1.1: The coverage-optimized retrieval algorithm in the CORE approach.

algorithm $CORE(Query, Cases)$
begin
  $RetrievalSet \longleftarrow \emptyset$
  while $|Cases| > 0$
  begin
    $C_1 \longleftarrow first(Cases)$
    $RetrievalSet \longleftarrow \{C_1\} \cup RetrievalSet$
    $covers(C_1) \longleftarrow \{C_1\}$
    for all $C_2 \in rest(Cases)$ do
    if C_1 is at least as good as C_2 with respect to
      one particular preference criterion
    then $covers(C_1) \longleftarrow covers(C_1) \cup \{C2\}$
    $Cases \longleftarrow Cases - covers(C_1)$
  end
  return $RetrievalSet$
end

fault diagnosis systems (Gupta 1998; Cunningham and Smyth 1994), the planning systems (Muñoz-Avila et al. 1999), and the workflow management systems (Weber et al. 2005).

Our research concentrated on the efficient and natural discriminative question selection and the research work was validated with a conversational software component retrieval system, called TrollCCRM.

1.3 Research Activities and Contributions

In this section, we first describe our research activities leading to the present thesis. Then we summarise our contributions to the research community. The references of the 8 papers included in the thesis are listed at the end.

1.3.1 Research Activities

In the spring of 2004, we completed our research proposal titled 'Component Retrieval Using Conversational Case-Based Reasoning' (CCRM application, Gu&Aamodt&Tong, ICIIP2004[1])). In the proposal, we reviewed and analyzed the

[1] The detailed references for the papers included in the thesis can be in the 'Publication List' subsection.

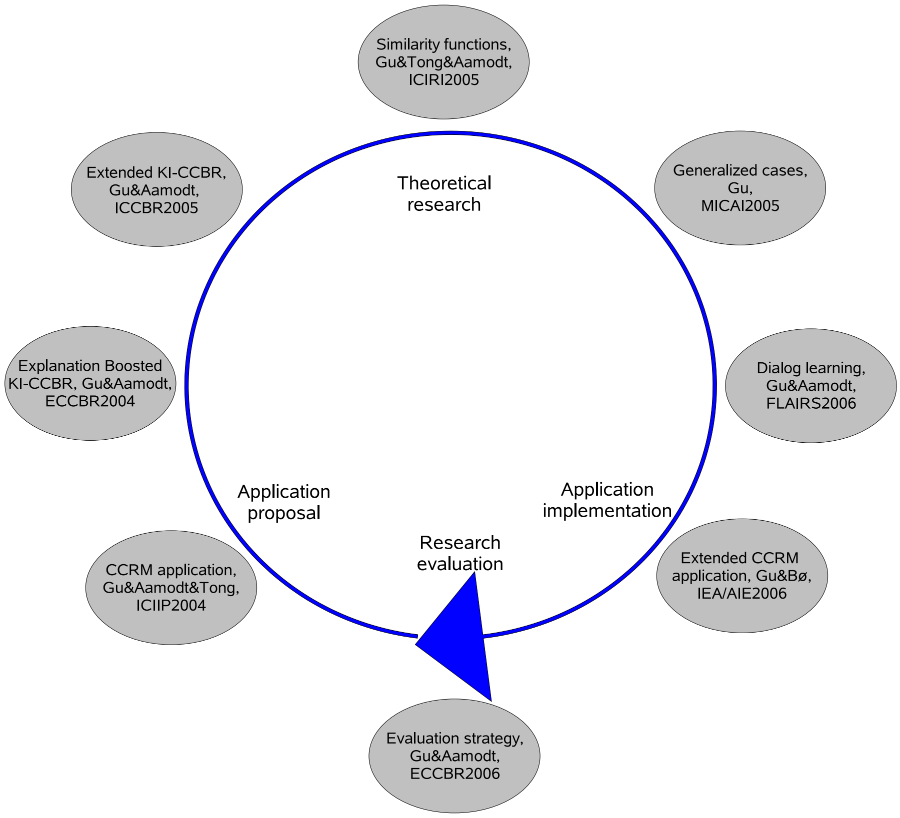

Figure 1.3: The research activities and corresponding publications

software component retrieval methods, and proposed a conversational software component retrieval model (CCRM) using the conversational CBR method. This model tries to release the user's difficulty in constructing the component query.

After the research proposal, we concentrated on the theoretic research part of conversational CBR. We noticed that there were several research aspects, including the feature inferencing, and the discriminative question ranking, which the knowledge-intensive method developed in our group could help the handling of. One paper was completed to propose using the knowledge-intensive method to support conversational CBR (Explanation boosted KI-CCBR, Gu&Aamodt, ECCBR2004). Based on the collected comments and our further research findings, the proposed knowledge-intensive conversational CBR method was extended to support the consistent question clustering and the coherent question sequencing, and the knowledge-intensive conversational CBR method was implemented in TrollCreek (Extended KI-CCBR, Gu&Aamodt, ICCBR2005).

We did some research about dynamic feature weighting (Tong, Öztürk, and Gu 2004; Tong and Gu 2005), in which a user's query was considered as her attention focus and a source of problem solving context, and a set of feature-weighting methods that took the query information into account were proposed and evaluated. Inspired by these research results, we proposed a framework to classify the similarity methods in CBR into three categories from the perspective of what features were taken into account during the similarity calculation. We also analyzed and illustrated that the query-biased similarity methods (only considering the features appearing in a query during similarity calculation) were more suitable for conversational CBR applications from the perspective of using shorter conversation length to find the correct case. The performances of the conversational CBR processes using different similarity calculation methods were compared within the Weka evaluation framework (Witten and Frank 2005) using the data sets from the UCI repository of machine learning databases (Newman et al. 1998) (Similarity functions, Gu&Tong&Aamodt, ICIRI2005).

In the conversational software component retrieval model, in order to represent both software components and the component queries as cases, a case needs to have more than one value on some features (generalized cases). We discussed the feasibilities and adapting methods to support generalized cases in conversational CBR from three aspects: the case representation, the similarity calculation, and the discriminative question ranking (Generalized cases, Gu, MICAI2005).

The knowledge-intensive software component retrieval method was implemented as the TrollCCRM system and evaluated in an image processing software component knowledge base including the image processing general domain knowledge and the software components come from the DynamicImager system (DynamicImager 2006) (Extended CCRM application, Gu&Bø, IEA/AIE2006).

After that, we designed a lazy dialog learning mechanism in an attempt to improve the dialog efficiency in conversational CBR through capturing and reusing the previous conversational case retrieval experiences. This dialog learner was implemented within Weka and tested on a group of UCI data sets (Dialog learning, Gu&Aamodt, FLAIRS2006). How to incorporate this dialog learning mechanism into the TrollCCRM system is among our future research.

Our research was evaluated using two types of data: the similarity calculation methods and the dialog learning mechanism were evaluated using the data sets from the UCI data repository, and the knowledge-intensive conversational software component retrieval method was evaluated on the image processing software component knowledge base. This evaluation process using different data sources to satisfy various evaluation goals was generalized as a common evaluation strategy for CBR systems (Evaluation strategy, Gu&Aamodt, ECCBR2006).

As shown in Fig. 1.3, our research was motivated by the software component retrieval application, concentrated on the theoretical research aspects in conversational CBR, and concluded with the evaluation of the theoretical research findings on the software component retrieval application.

1.3.2 Contributions

My main contributions to the research community include:

1. Providing a knowledge-intensive method supporting an efficient and natural conversation process in conversational CBR, in which we identified five question selection tasks, the feature inferencing, the knowledge-intensive similarity calculation, the knowledge-intensive question ranking, the consistent question clustering and the coherent question sequencing, and used the proposed explanation boosted method to solve them.

2. Designing and implementing a conversational software component retrieval model using the knowledge-intensive conversational CBR method, in which we compared the similarity methods in conversational CBR and adopted the query-biased similarity method, we also extended the conversational CBR process to support generalized cases.

3. Improving the dialog efficiency in conversational CBR through introducing a lazy dialog learner.

4. Providing a general evaluation strategy to CBR systems that uses different data sources to satisfy various evaluation goals, and the evaluation process to our research, including the statistical evaluation, the expert validation, the characteristic analysis and the ablation study, was acted as a case study to this evaluation strategy.

Fig. 1.4 shows what contribution items or sub-items were contained and discussed in which included publictation.

1.3.3 Publication List

- **Explanation-Boosted KI-CCBR, Gu&Aamodt, ECCBR2004**, Mingyang Gu and Agnar Aamodt: 'Explanation-Boosted Question Selection in Conversational CBR', In Pablo Gervás and Kalyan Moy Gupta (eds.), Proceedings of the ECCBR 2004 Workshops, the 7th European Conference on Case-Based Reasoning, Madrid, Spain, 30th August - 2nd September, 2004, Technical report, Vol. 142, No. 04, pp. 105 - 114, Universidad Complutense de Madrid, Departemento de Sistemas Informaticos y Programacion. (Paper A)

- **CCRM application, Gu&Aamodt&Tong, ICIIP2004**, Mingyang Gu, Agnar Aamodt, and Xin Tong: 'Component Retrieval Using Conversational Case-Based Reasoning', In Zhongzhi Shi and Qing He (eds.), Intelligent Information Processing II - International Conference on Intelligent Information Processing (IIP2004), Beijing, China, 2004, IFIP International Federation for Information Processing, Vol. 163, pp. 259 - 271, Springer Science + Business Media Inc. (Paper B)

- **Similarity functions, Gu&Tong&Aamodt, ICIRI2005**, Mingyang Gu, Xin Tong, and Agnar Aamodt: 'Comparing Similarity Calculation Methods in Conversational CBR', In Du Zhang, Taghi M. Khoshgoftaar and Mei-Ling Shyu (eds.), Proceedings of the 2005 IEEE International Conference on Information Reuse and Integration, Hilton, Las Vegas, Nevada, USA, August, 2005, pp. 427 - 432, IEEE Press. (Paper C)

- **Extended KI-CCBR, Gu&Aamodt, ICCBR2005**, Mingyang Gu and Agnar Aamodt: 'A Knowledge-Intensive Method for Conversational CBR', In Héctor Muñoz-Avila and Francesco Ricci (eds.), Case-Based Reasoning Research and Development, Proceedings of the 6th International Conference on Case-Based Reasoning, Chicago, Illinois, August, 2005, Lecture Notes in Artificial Intelligence, Vol. 3620, pp. 296 - 311, Springer Verlag. (Paper D)

- **Generalized cases, Gu, MICAI2005**, Mingyang Gu: 'Supporting generalized Cases in Conversational CBR', In Alexander Gelbukh, Alvaro de Albornoz and Hugo Teras-hima-Marin (eds.), Advances in Artificial Intelligence, Proceedings of the Fourth Mexican International Conference on Artificial Intelligence (MICAI2005), Monterrey, Mexico, November, 2005, Lecture Notes in Artificial Intelligence, Vol. 3789, pp. 544 - 553, Springer Verlag. (Paper E)

- **Dialog learning, Gu&Aamodt, FLAIRS2006**, Mingyang Gu and Agnar Aamodt: 'Dialog Learning in Conversational CBR', In Geoff C. J. Sutcliffe and Randy G. Goebel (eds.), Proceedings of the 19th International FLAIRS Conference (Florida Artificial Intelligence Research Society), pp. 358 - 363, Melbourne Beach, Florida, May 11 - 13, 2006, AAAI Press. (Paper F)

- **Extended CCRM application, Gu&Bø, IEA/AIE2006**, Mingyang Gu and Ketil Bø: 'Component Retrieval Using Knowledge-Intensive Conversational CBR', In M. Ali and R. Dapoigny (eds.), Proceedings of the 19th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (IEA/AIE2006), Annecy, France, 27 - 30, June, 2006, Lecture Notes in Artificial Intelligence, Vol. 4031, pp. 554 - 563, Springer Verlag. (Paper G)

- **Evaluation strategy, Gu&Aamodt, ECCBR2006**, Mingyang Gu and Agnar Aamodt: 'Evaluating CBR Systems Using Different Data Sources: A Case Study', In T.R. Roth-Berghofer et al. (eds.), Proceedings of the 8th European Conference on Case-Based Reasoning (ECCBR2006), Fethiye, Turkey, 4 - 7, September, 2006, Lecture Notes in Artificial Intelligence, Vol. 4106, pp. 121 - 135, Springer Verlag. (Paper H)

Papers	*KI-CCBR*					*CCRM application*			*Dialog learning*	*Evaluation*				
	KI-Similarity Calculation	Feature inferencing	KI-question clustering	Consistent question clustering	Coherent question sequencing	Similarity method	Generalized cases	Implem-entation		Evaluation strategy	Statistical evaluation	Expert validation	Charact. analysis	Ablation study
CCRM application, Gu&Aamodt&Tong, ICIIP2004	*							*						
Explanation boosted KI-CCBR, Gu&Aamodt, ECCBR2004		*	*											
Extended KI-CCBR, Gu&Aamodt, ICCBR2005		*	*	*	*			*						
Similarity functions, Gu&Tong&Aamodt, ICIRI2005						*					*			
Generalized cases, Gu, MICAI2005							*							
Dialog learning, Gu&Aamodt, FLAIRS2006									*		*			
Extended CCRM application, Gu&Bø, IEA/AIE2006	*	*	*	*	*		*	*				*		
Evaluation strategy, Gu&Aamodt, ECCBR2006		*	*	*	*	*	*	*	*	*	*	*	*	*

Figure 1.4: The contributions and the included publications containing them

Chapter 2

Research Description

Our research started with proposing a CCRM using the conversational CBR technology. The main research efforts concentrated on the theoretical research aspects of conversational CBR towards providing an efficient and natural dialogue process. At the end, we implemented this model as the TrollCCRM system and evaluated our research findings in it.

In this chapter, we describe the architecture of the CCRM model in Section 1; then we give an introduction to the CREEK architecture (Case-based Reasoning through Extensive and Explicit general Knowledge) and the TrollCreek system on which our research and implementation are based in Section 2; in Section 3, we describe the research findings that can be used to handle the tasks or improve the existing methods in CCRM; in Sections 4 and 5, we describe the system implementation and the evaluation.

2.1 Overview of the Conversational Component Retrieval Model

Software component retrieval, which concerns how to locate and identify the most appropriate components for users, is one of the major problems in software component reuse (Fernández-Chamizo et al. 1996; Iribarne, Troya, and Vallecillo 2002; Mili, Mili, and Mittermeir 1998). The current component retrieval methods include the free-text-based retrieval methods (Frakes and Nejmeh 1987; Helm and Maarek 1991; Klein and Bernstein 2001; Magnini 1999), the pre-enumerated vocabulary methods (Prieto-Daz 1991), the signature matching methods (Zaremski and Wing 1995), the behavior-based retrieval methods (Park 2000), and the faceted selection methods (Prieto-Daz 1991). All these methods have an assumption that users are able to well define their component queries by themselves. However, before users know the components available for them to choose, they often lack clear ideas about what they need, and usually cannot define their queries properly.

In order to release this unrealistic assumption, we proposed a conversational

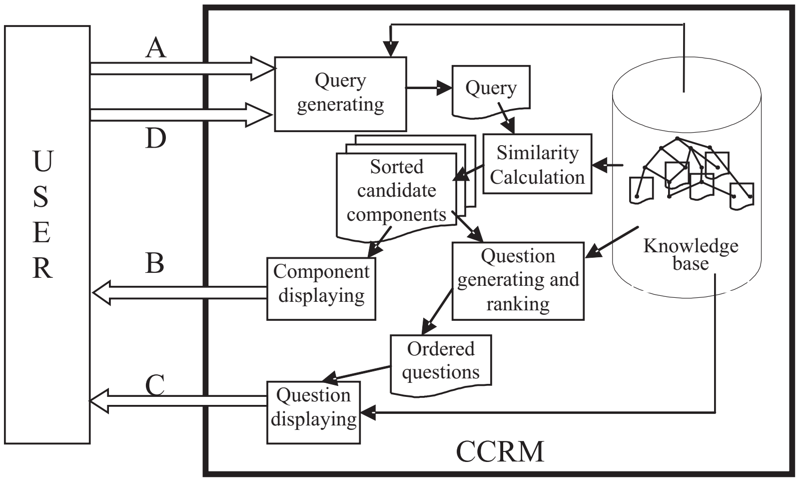

Figure 2.1: The architecture of the conversational component retrieval model (CCRM)

component retrieval model (CCRM) using the conversational CBR technology. In this model, each software component is represented as a stored case and a component query is represented as a new case. A conversational CBR process is used to guide users to construct their component queries incrementally and find the correct components. As illustrated in Fig. 2.1, CCRM includes six parts: a knowledge base, a query-generating module, a similarity calculation module, a question-generating and -ranking module, a component-displaying module, and a question-displaying module.

The knowledge base stores both the component-specific knowledge (cases) and the general domain knowledge. After a user provides her initial requirement specification (arrow A), the query-generating module uses it to construct an initial query. Given the query, the similarity calculation module calculates the similarities between the query and each stored component and returns a set of components whose similarities surpass a threshold (the threshold is predefined and can be adjusted during the execution). In the question-generating and -ranking module, discriminative questions are identified from the returned components and ranked. The component-displaying module displays the returned components, ordered by their similarities, to the user (arrow B). The question-displaying module displays the ranked questions (arrow C). If the user finds her desired component in the displayed components, she can select it and terminate the retrieval process. Otherwise, she chooses one or several questions and provides the answers to the system (arrow D). Then the query-generating module combines the current query and the newly gained answers to construct an updated new query, and a new round of retrieving and question answering is started until the user finds her desired component (success) or there are no questions left for her to choose (failure).

In (CCRM application, Gu&Aamodt&Tong, ICIIP2004), we provided a review about the current software component retrieval methods and described the architecture of CCRM. In (Extended CCRM application, Gu&Bø, IEA/AIE2006), we further proposed using the knowledge-intensive conversational CBR method in CCRM.

In order to realize this model, there is a list of open research topics. These topics include:

- Knowledge Representation: Research questions inside this topic include how both the general domain knowledge and the case-specific knowledge are represented in the knowledge base, how to describe a software component and a component query, and how the stored knowledge can be organized and utilized to complete the identified tasks.
- Case Retrieval: The research questions, such as whether the entire similarity methods used in CBR are also suitable to be used in the conversational CBR applications, what type of similarity metric is suitable for CCRM, and how the general domain knowledge can be used to improve the similarity matching process, are included in this research topic.
- Discriminative Question Selection: This is one of the main research topics in the conversational CBR research community. The active research questions contained in this topic include how to generate the candidate discriminative questions, how to arrange the question-answering sequence to provide users an efficient and natural conversation process, and whether the general domain knowledge can enhance the question selection process and how.

The research in the thesis is based on the achievements of our group on the knowledge-intensive CBR research. In the following section, we will introduce the knowledge-intensive CBR architecture, CREEK, and its implementation, TrollCreek, developed in our group.

2.2 Introduction to CREEK and TrollCreek

Our group has been doing knowledge-intensive CBR research for more than 16 years, and the most important achievements are the knowledge-intensive CBR architecture, CREEK, and its corresponding implementation, TrollCreek system.

CREEK (Case-based Reasoning through Extensive Explicit Knowledge) is a knowledge-intensive case-based problem solving and sustained-learning architecture (Aamodt 1991; Aamodt 2001; Aamodt 2004). On a high level, CREEK includes three parts: a comprehensive knowledge representation model, a knowledge-intensive case-based problem solving process, and a sustained- learning process.

The comprehensive knowledge representation model, called CreekL (Aamodt 1994b), is a frame-based system in which each meaningful term in a Problem solving domain is explicitly defined as a concept. Each concept is defined using a set of relationships. A relationship is a <relation, value> pair where the 'value' is another concept in the knowledge base. The general domain knowledge is represented as a semantic network with a group of concepts connected by the various relations. Cases are submerged into this semantic network, since all the features used to define them are concepts in the network. We instantiated this representation in the car fault detection application in (Explanation boosted KI-CCBR, Gu&Aamodt, ECCBR2004) and (Extended KI-CCBR, Gu&Aamodt, ICCBR2005), and in the image processing software component retrieval application in (CCRM application, Gu&Aamodt&Tong, ICIIP2004) and (Extended CCRM application, Gu&Bø, IEA/AIE2006).

The knowledge-intensive case-based problem solving process utilizes both the CBR and the model-based reasoning (MBR) methods to complete the problem solving tasks. The CBR method acts as the first principle to tackle the problem solving tasks. The model-based reasoning method is used to enhance the reasoning process through applying the general domain knowledge to improve each step in the case-based reasoning process (Aamodt 1994a). A three-step general reasoning process was designed to combine these two types reasoning methods together to complete each task in the case-based problem solving process. The first step, ACTIVATE, determines what knowledge (including the case-specific knowledge and the general domain knowledge) is involved in one particular task. EXPLAIN, the second step, builds up explanation paths to explore possible knowledge-intensive solutions for that task. And the last step, FOCUS, evaluates the generated explanation paths and identifies the best one(s) to accomplish that particular task. The paper (Explanation boosted KI-CCBR, Gu&Aamodt, ECCBR2004) gave an example about how this three-step general reasoning process was initiated to support the feature inferencing and knowledge-intensive question ranking in the car fault detection application. Another example of using this reasoning process to support the knowledge-intensive similarity calculation in the software component retrieval application was given in (CCRM application, Gu&Aamodt&Tong, ICIIP2004).

The sustained-learning process is supported by the case-based retain process. The general domain knowledge is used to guide the crucial steps in the learning process, e.g., how to create the reminding (index) for a retained case.

With the aim to provide an experimentation environment to explore the knowledge-intensive CBR research and applications in our group, the CREEK framework was first implemented in a lisp-based version by Agnar Aamodt and later transplanted into a java-based version by Frode Sørmo. The java-based version is called TrollCreek (TrollCreek 2005). TrollCreek supports the comprehensive knowledge representation mechanism and the knowledge-intensive case-based problem solving process, and provides a friendly graphic interface for the knowledge base definition and exploration, and the reasoning operation and explanation.

Based on TrollCreek, our group has done a set of theoretical research in CBR, e.g., incorporating the plausible inheritance (Aamodt 1991; Sørmo 2000) and the abductive inference (Öztürk 2000). A list of CBR applications have been fielded in variant domains, for instance, context modeling or awareness (Öztürk and Aamodt 1997; Kofod-Petersen and Aamodt 2003), intelligent tutoring system (Sørmo and Aamodt 2002), disease explanation at the level of functional genomics (Kusnierczyk, Aamodt, and Lægreid 2004; Kusnierczyk, Aamodt, and Lægreid 2005), medical image understanding (Grimnes and Aamodt 1996), and fault prediction in the oil drilling domain (Skalle and Aamodt 2004).

2.3 Research Issues in CCRM

In this section, the research work is described from four issues: the knowledge representation, the case retrieval method, the discriminative question selection method, and the continuous dialog learning mechanism.

2.3.1 Knowledge Representation

In CCRM, CreekL is used to represent both the case-specific knowledge (software components and component queries) and the general domain knowledge in the knowledge base, which forms an object-level knowledge model.

In order to support an efficient and natural dialog process in CCRM, a list of question selection tasks[1] are identified. A meta-level knowledge representation model is abstracted from the object-level knowledge model. The meta-level model is used to organize the semantic relations according to their contributions to different question selection tasks. If a type of relation can be used to support a question selection task, it will be linked to the task by a 'subclass of' relation. The relations that are subclasses of a specific task are involved in the knowledge-intensive problem solving process for the corresponding task.

In addition, the introduction of the meta-level knowledge representation model provides CCRM the flexibility and the extendibility by separating the content of the knowledge base and the implementation of the question selection tasks from each other. A knowledge engineer can construct the knowledge base from the knowledge engineering perspective, and a software engineer can implement the problem solving logic from the software engineering point of view.

In (Explanation boosted KI-CCBR, Gu&Aamodt, ECCBR2004), we used the proposed knowledge-intensive conversational CBR method to handle two question selection tasks, the feature inferencing and the knowledge intensive question ranking, and in (Extended KI-CCBR, Gu&Aamodt, ICCBR2005) it was illustrated how easy it is to extend that method to support the other two question

[1]The identified question selection tasks can be found in (Extended KI-CCBR, Gu&Aamodt, ICCBR2005).

selection tasks, the consistent question clustering and the coherent question sequencing. If a new semantic relation is identified to be useful for completing a question selection task, it can be put into use in the reasoning process through simply linking this relation into the meta-level knowledge model.

In CCRM, both software components and component queries need to be represented as generalized cases. In the CBR community, cases that may have multiple values on some features are nominated as generalized cases (Mougouie and Bergmann 2002; Maximini, Maximini, and Bergmann 2003; Tartakovski et al. 2004), and cases that only have at most one single value on each feature are named point cases. For a software component, to have multiple values on one feature in the corresponding stored case means this component can function in variant environments specified by the different values on that feature. For a component query, the different values on one feature in the corresponding new case express users multiple requirements on that feature to be satisfied.

The research work of extending conversational CBR to support generalized cases was described in (Generalized cases, Gu, MICAI2005) and the implementation of supporting generalized cases in TrollCCRM was reported in (Extended CCRM application, Gu&Bø, IEA/AIE2006).

2.3.2 Case Retrieval

In CBR, to retrieve the most similar case is to find the case with the largest similarity value with the new case. There are plenty of similarity calculation methods proposed for the CBR process. Whether the similarity methods suitable for CBR are also effective for conversational CBR is the question we met when designing the case retrieval method for CCRM.

We use the weighted Euclidean distance to measure the similarity between a query and a stored case, which is illustrated in Equation 2.1:

$$distance(q, c) = \sqrt{\frac{\sum_{f \in FS} w_f dif^2(q_f, c_f)}{\sum_{f \in FS} w_f}} \tag{2.1}$$

where q, c, f, FS, and w_f denote a query, a stored case, a particular feature, a selected feature set, and the weight for the feature f, respectively. $dif(q_f, c_f)$ is a function to compute the difference between q and c on feature f.

From the perspective of what features are taken into account during the similarity calculation (the different value-assigning strategies of FS in Equation 2.1), we provided a framework that classified the similarity calculation methods in CBR into three categories:

- Case-Biased Similarity Methods: When calculating the similarity value between a stored case and a new case, only the features appearing in the stored case are taken into account.

- Query-Biased Similarity Methods: When calculating the similarity value, only the features appearing in the new case (query) are counted.
- Equally-Biased Similarity Methods: All the features appearing in both the stored case and the new case are considered during the similarity calculation.

We noticed the special characteristic of a new case in conversational CBR that the new case was partially specified during the conversation process. We argued that the query-biased similarity calculation methods were more suitable for conversational CBR applications.

The framework to classify the similarity calculation methods was described and the reason why query-biased similarity calculation methods were more suitable for conversational CBR was explained in (Similarity functions, Gu&Tong&Aamodt, ICIRI2005). We also designed an experiment to compare the performances of the conversational CBR processes using these three different similarity methods on 36 data sets within Weka. The experiment results, reported in (Similarity functions, Gu&Tong&Aamodt, ICIRI2005), gave positive support for the query-biased similarity methods.

The knowledge-intensive similarity calculation method proposed in CREEK is adopted in CCRM. (CCRM application, Gu&Aamodt&Tong, ICIIP2004) described how the knowledge-intensive process was used to improve the similarity calculation quality in CCRM.

As discussed in the 'Knowledge Representation' topic, CCRM uses generalized cases to represent both software components and component queries. In (Generalized cases, Gu, MICAI2005), we proposed a method to calculate the similarities for generalized cases in a knowledge-poor context. The knowledge-intensive similarity calculation process described in (CCRM application, Gu&Aamodt&Tong, ICIIP2004) can be directly used to support generalized cases through applying it to the entire $< feature, value >$ pairs specified in a new case (including the $< feature, value >$ pairs that share the same feature type).

2.3.3 Discriminative Question Selection

In the conversational CBR community, besides the statistical metric based question selection methods, there is some research that tries to use the general domain knowledge to help the question selection process, including dialogue inferencing (Aha, Maney, and Breslow 1998), causal conversational CBR (Aha and Gupta 2002), and taxonomic conversational CBR (Gupta 2001; Gupta, Aha, and Sandhu 2002). We proposed an integrated question selection method that utilized both the statistical question-ranking metric and the general domain knowledge to provide an efficient and natural question selection process.

In the research, four question selection tasks were identified from the perspective of 'efficiency' and 'naturalness' of a dialog process that general domain knowledge can help handling. These tasks include:

- **Feature Inferencing**[2]: The features that can be inferred from the current query description are added into the query description automatically. They will not be transformed as questions and posed to users any more.

- **Knowledge-Intensive Question Ranking**: The semantic relations among discriminative questions should be taken into account during question ranking. For instance, if one answer of question A can be inferred by one answer of question B, question B should be asked before question A.

- **Consistent Question Clustering**: The questions that are connected by certain semantic relations, e.g., a causal relation or a subclass relation, should be grouped and displayed together.

- **Coherent Question Sequencing**: If a question in a subsumption taxonomy is asked in the current question-answering cycle, the question one level lower should get higher priority to be asked in the next cycle than other unrelated questions.

The meta-level knowledge representation model introduced in the Knowledge Representation sub-section is used to represent what semantic relations in the knowledge base can be used to support each question selection task.

The three-step explanation-boosted reasoning process is instantiated to explore the general domain knowledge to complete each question selection task. The statistical question-ranking metric, occurrence frequency metric, is applied to sort the questions that cannot be discriminated using the knowledge-intensive method. The reason to select the occurrence frequency metric is due to the characteristic of the constructed software component knowledge base (i.e., high-level heterogeneity).

In (Explanation boosted KI-CCBR, Gu&Aamodt, ECCBR2004), the first two knowledge-intensive question selection tasks, the feature inferencing and the knowledge-intensive question ranking, were identified and handled, and in (Extended KI-CCBR, Gu&Aamodt, ICCBR2005), the other two tasks, the consistent question clustering and the coherent question sequence, were further identified and handled.

In (Generalized cases, Gu, MICAI2005), we reviewed the currently existing statistical question-ranking metrics in conversational CBR and discussed the feasibilities and adaptation methods to apply them to support generalized cases.

2.3.4 Dialog Learning

Conversational CBR is a type of problem solving model that constructs the new case through a question-answering dialog process and finds the correct case at the same time. To improve the performance of this problem solving process,

[2]Referred as dialogue inferencing in (Aha, Maney, and Breslow 1998)

we proposed a lazy dialog learning method for conversational CBR to retrieve, reuse, and retain successful dialog experiences.

A dialog case base is maintained for the dialog learner where the successful conversational case retrieval experiences are captured and stored as dialog cases. The problem description part of a dialog case contains all the information that happened in the conversational case retrieval process, including the initial new case and the later question-answering process. The solution description of a dialog case points to the previous accepted application case by users in that dialog.

The main steps in this proposed dialog learner include:

- Dialog Case Retrieve

  The most similar dialog case is retrieved for a new dialog based on the similarity comparison. In the similarity calculation, not only the $< feature, value >$ pairs appearing in the stored dialog case and the current new dialog case, but also the positions of each $< feature, value >$ pair are taken into account.

- Dialog Case Reuse

  The most similar dialog case retrieved in each conversation session is used in two ways: first, the solution description part (i.e., the chosen application case in the dialog) of the most similar dialog case is displayed to the user; second, if a discriminative question generated by the current conversational session also appears in the most similar dialog case, it will be assigned a higher priority to be displayed to the user.

- Dialog Case Retain

  The dialog case retain strategy is to only store the most general dialog cases. One dialog case is more general than the other means that the two dialog cases share the same solution, and the problem description part of the first dialog case is a subset of that of the second one.

(Dialog learning, Gu&Aamodt, FLAIRS2006) described the architecture of the lazy dialog learner, its implementation and evaluation within Weka. Due to the time constrains, we did not implement this learning process into TrollCCRM, and the challenging research topics to integrate it into TrollCCRM are discussed in Chapter 4.

2.4 System Implementation

CCRM has been implemented into a system, called TrollCCRM, based on Troll-Creek system. We only gave a short introduction to this implementation in

Papers (CCRM application, Gu&Aamodt&Tong, ICIIP2004), (Extended KI-CCBR, Gu&Aamodt, ICCBR2005), (Extended CCRM application, Gu&Bø, IE-A/AIE2006), and (Evaluation strategy, Gu&Aamodt, ECCBR2006), so we will give more information about the implementation in this section.

2.4.1 Conversational CBR Modules in TrollCCRM

In order to support the conversational component retrieval, particularly the knowledge-intensive method proposed in (Explanation-Boosted KI-CCBR, Gu&-Aamodt, ECCBR2004) and (Extended KI-CCBR, Gu&Aamodt, ICCBR20-05), we extended TrollCreek with a query-generating module, a feature inferencing module, a question identification module, an integrated question-ranking module, a consistent question clustering module, a coherent question sequencing module, and a graphical user-machine interaction interface. The case-biased similarity calculation method is used in TrollCreek. In TrollCCRM, we changed to using the query-biased similarity calculation method.

After a user provides her initial requirement, the query-generating module transforms it into a query including a set of $< feature, value >$ pairs. Given the query, the feature-inferencing module extends it with inferred $< feature,\ value >$ pairs through exploring the general domain knowledge using the explanation-boosted reasoning mechanism (Explanation-Boosted KI-CCBR, Gu&Aamodt, ECCBR2004) and (Extended KI-CCBR, Gu&Aamodt, ICCBR20-05). The extended query is taken as the input to the knowledge-intensive case retrieval module (CCRM application, Gu&Aamodt&Tong, ICIIP2-004) and a set of most similar components are returned. The user-machine interaction interface module displays these returned components to the user. The features that appear in these returned components, but not in the extended query, are identified and transformed into the candidate discriminative questions by the question identification module (the features that are inferred by the feature-inferencing module will not be identified as discriminative questions since they are currently in the extended query).

The integrated question-ranking module first explores the general domain knowledge and finds the semantic question-ranking explanation paths between the questions. And the 'focused' explanation paths are used to classify all the questions into two groups: free questions and constrained questions (constrained by some explanation paths to be asked after other questions). The questions in the first group are further ranked using the knowledge-poor statistical question-ranking method, i.e., the occurrence frequency metric. The questions in the latter group are ranked according to how strong they are constrained to be asked after other questions (the explanation strength of the strongest explanation path constraining one question to be asked after other questions is selected as the ranking key of that question).

The order of these questions is further adjusted by the coherent question-sequencing module if there is one or several questions answered in the last question-answering cycle. The coherent question sequencing module first identifies the questions from the sorted questions in the present question-answering cycle that are demanded to be asked immediately after the questions answered in the past question-answering cycle. These identified questions are then put at the beginning of the entire sorted questions. The inside order of these questions are decided by how strong they are demanded to be asked immediately after the answered questions (the explanation strength of the strongest question sequencing explanation path attached on one question is selected as its sorting key).

Finally, the sorted questions are displayed to users in the user-machine interaction interface. When a user selects a question to inspect, the alternative answers to the question are displayed to the user and the related questions identified by the consistent question-clustering module are also shown to the user. The user may terminate the conversation process by choosing a satisfactory component, or she selects one or more questions and submits the answers to the system. The query-generating module transforms the answered question(s) into $< feature, value >$ pair(s) and adds it(them) to the current query to generate an updated query. Then a new round of feature inferencing, retrieving, and question-answering process starts. The process also terminates when there are no discriminative questions left for users to select.

From the perspective of improving users' trust in the reasoning results, it is important for an intelligent system to provide more explanations about the system inside reasoning process (Sørmo, Cassens, and Aamodt 2005; Reilly et al. 2005; McSherry 2005; McSherry 2001a). In TrollCreek, the reasoning process about how a returned case is matched to a new case is explained (as illustrated in Fig. 2.4). In TrollCCRM, we extended the explanation mechanism through providing the explanations for the following questions:

- Why is a $< feature, value >$ pair added into the new case by the feature-inferencing module?
- Which feature is a question generated from in the question-identification module?
- Why is a question given higher priority to be asked in the current conversation session by the coherent question-sequencing module?
- Why is a question identified as a related question by the consistent question-clustering module?

In (Generalized cases, Gu, MICAI2005), we analyzed the necessity of supporting both software components and component queries using generalized cases. In TrollCCRM, generalized cases are supported from the following three aspects: the case representation, the similarity calculation, and the question-ranking process. In the case representation, it is permitted to have more than one relationship that share the same relation type in one case. The similarity measure in TrollCCRM is based on how many and to what degree the relationships in the new case (including those that share the same relation type) are satisfied by each stored case. In the knowledge-intensive question-ranking process, the relations between value concepts are transferred to the corresponding feature concepts (each value concept can only act as the value of one specific feature concept in TrollCCRM), and the explanation-boosted reasoning process allows two features (questions) to have more than one relation between them. In this way, the knowledge-intensive question-ranking method supports both generalized cases and point cases. The occurrence frequency metric, selected as the statistical question-ranking metric in TrollCCRM, supports generalized cases directly, since it only considers whether or not one feature appears in a case and does not care how many values that feature has in the case.

2.4.2 Conversational Component Retrieval Process in TrollCCRM

In TrollCCRM, a conversational component retrieval process contains one or several conversation sessions (the number of sessions depends on when the user finds her satisfactory component and whether there are still discriminative questions left).

As an example to illustrate the conversational component retrieval process, an initial component query (as illustrated in Fig. 2.2) is used to start a retrieval process.

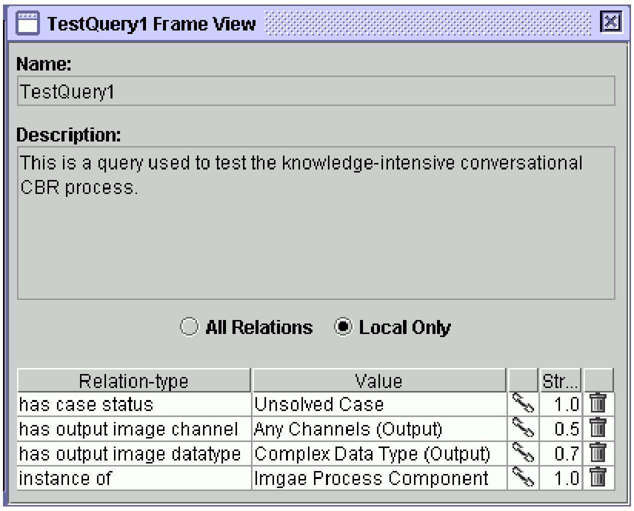

Figure 2.2: The content of a component query to start a conversational component retrieval process in TrollCCRM.

In the first question-answering session, assume the user didn't find her desired component and selected the question 'What type of unitary transform operation do you want?' to answer. The answer 'Fourier Transform' was provided to the

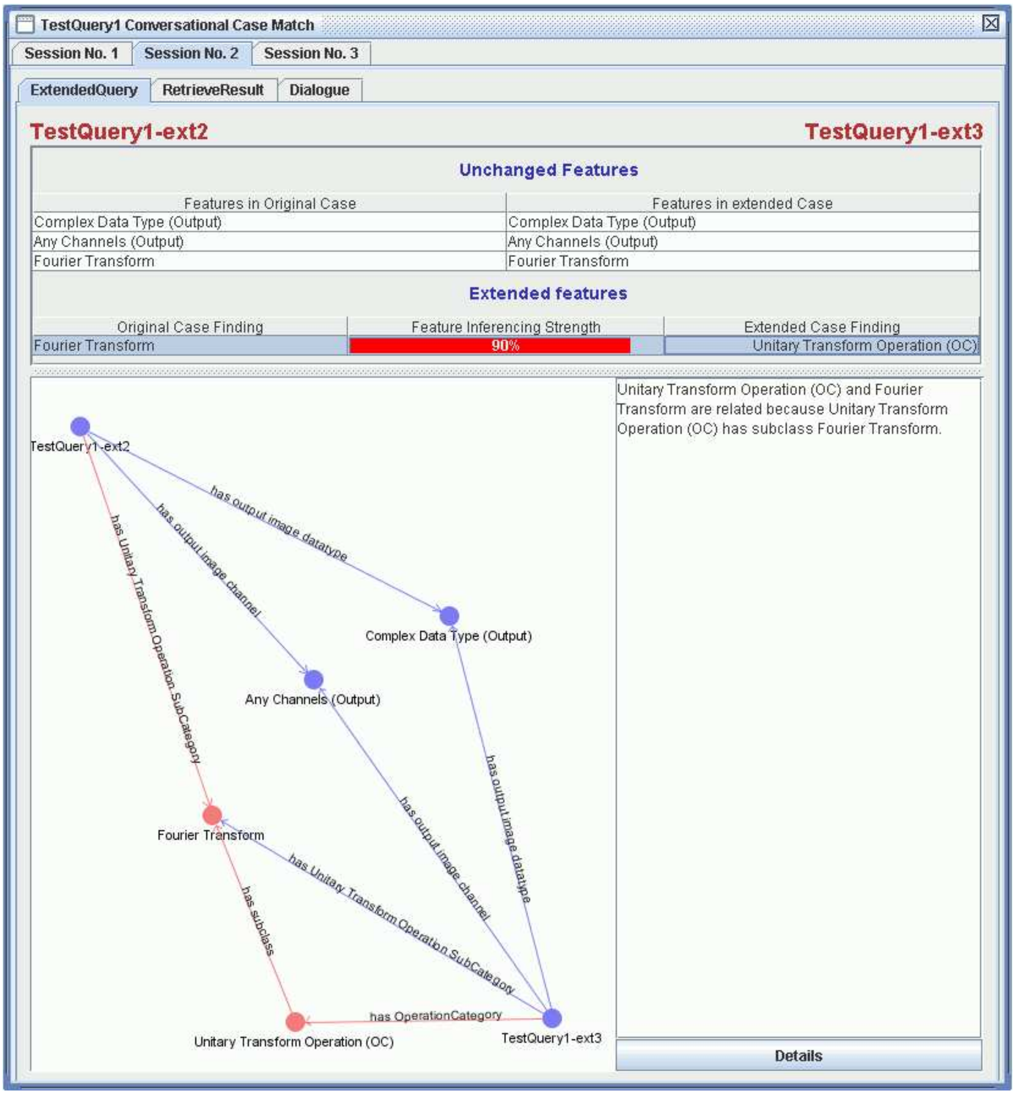

Figure 2.3: The 'ExtendedQuery' pane - The interface used to show how a component query is extended by the feature-inferencing module.

system to start the second conversation session. The following paragraphs show the graphical interfaces and operations for the second conversation session.

There are three windowpanes in the computer interface to move between for each conversation session.

The first is called the ExtendedQuery pane (shown in Fig. 2.3), which is used to display the query and the extended query extended by the feature-inferencing process. The explanation about why a $< feature, value >$ pair is inferred and added into the query is also shown on this pane. In this example, a feature 'has OperationCategory' with its value 'Unitary Transform Operation (OC)' is

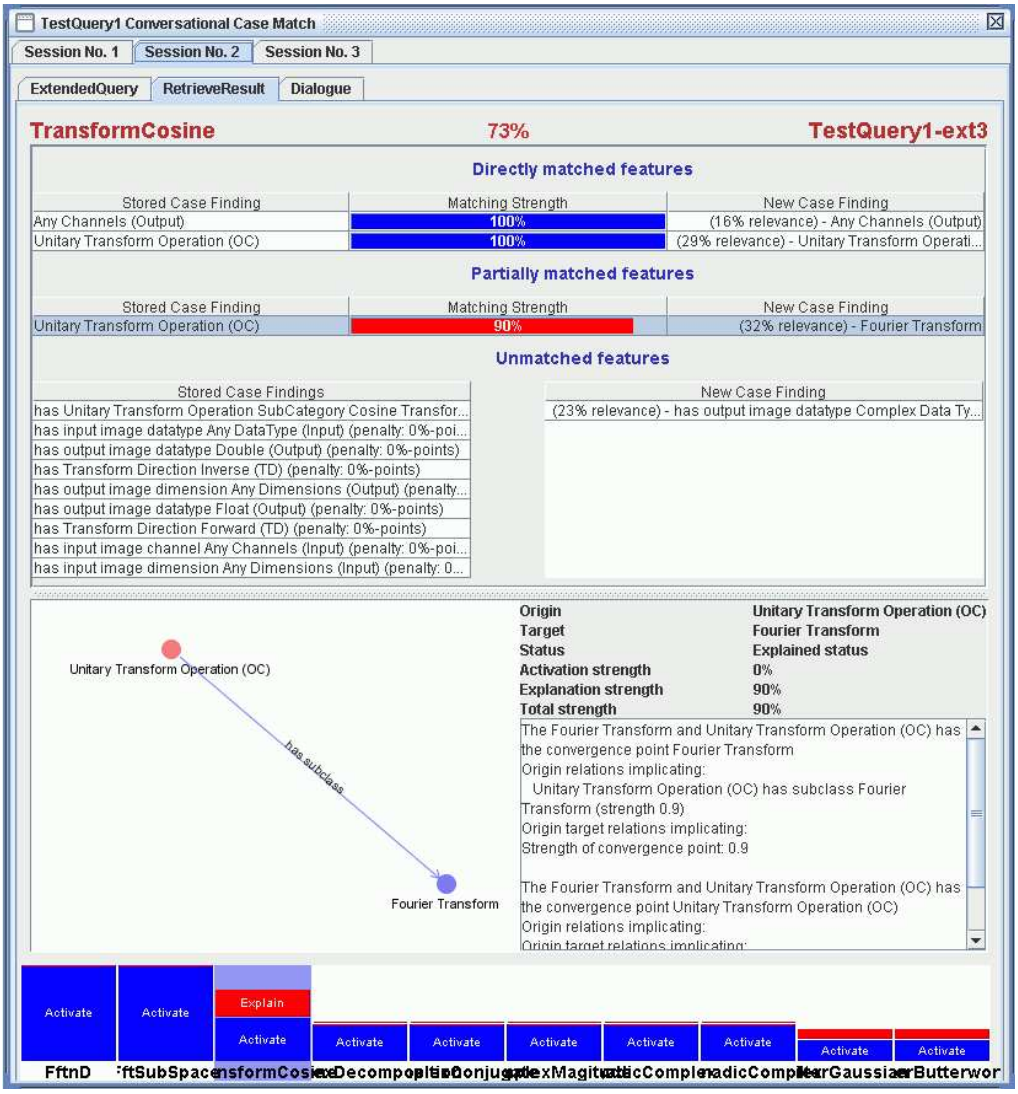

Figure 2.4: The 'RetrievalResult' pane (after clicking the 'Details' button) - The interface used to show the returned components and the matching details.

inferred and added into the new case with the explanation: 'Unitary Transform Operation (OC)' can be inferred by 'Fourier Transform' because 'Fourier Transform' is a subclass of 'Unitary Transform Operation (OC)'.

Based on the extended query, the knowledge-intensive case retrieval module returns a set of components and displays them in the RetrieveResult pane (as illustrated in Fig. 2.4). In this pane the user can inspect the matching details between each retrieved component and the extended query. In this example, as shown in Fig. 2.4, there are 10 similar components returned and displayed at the bottom from left to right according to their similarity values. If the user

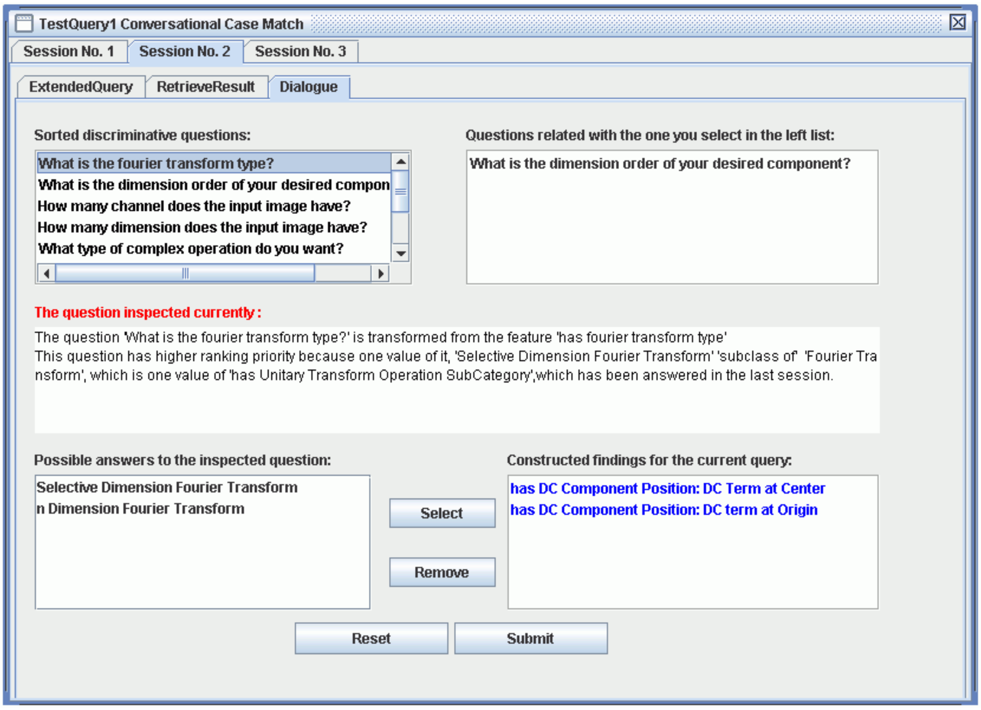

Figure 2.5: The 'Dialogue' pane - The interface used to show the discriminative questions and support the question-answering dialog. This figure gives the explanation about why a question is given higher priority by the coherent question-sequencing module.

clicks the third retrieved component 'TransformCosine', the matching details between this component and the extended query is displayed. In Fig. 2.4, there are two directly matched features and one partially matched feature for the extended query. If the user selects the partially matched feature by clicking it, the text area will explain how the feature 'has Unitary Transform Operation SubCategory' of the extended query is satisfied (with 90%) by the feature 'has OperationCategory' of the 'TransformCosine' component. The explanation is: the component 'TransformCosine' has the value 'Unitary Transform Operation (OC)' on the feature 'has OperationCategory' and the extended query has the value 'Fourier Transform' on the feature 'has Unitary Transform Operation SubCategory', and 'Fourier Transform' is the subclass of 'Unitary Transform Operation (OC)'. The concrete explanation about how the explanation is generated through exploiting the general domain knowledge is shown in the same text area after clicking the 'Details' button, as seen in Fig. 2.4.

If the user is not satisfied with the components, she can go to the dialogue pane to inspect and answer the discriminative questions, as shown in Figs. 2.5 and 2.6. In the dialogue pane, the identified discriminative questions ranked by the integrated question-ranking module and adjusted by the coherent question-

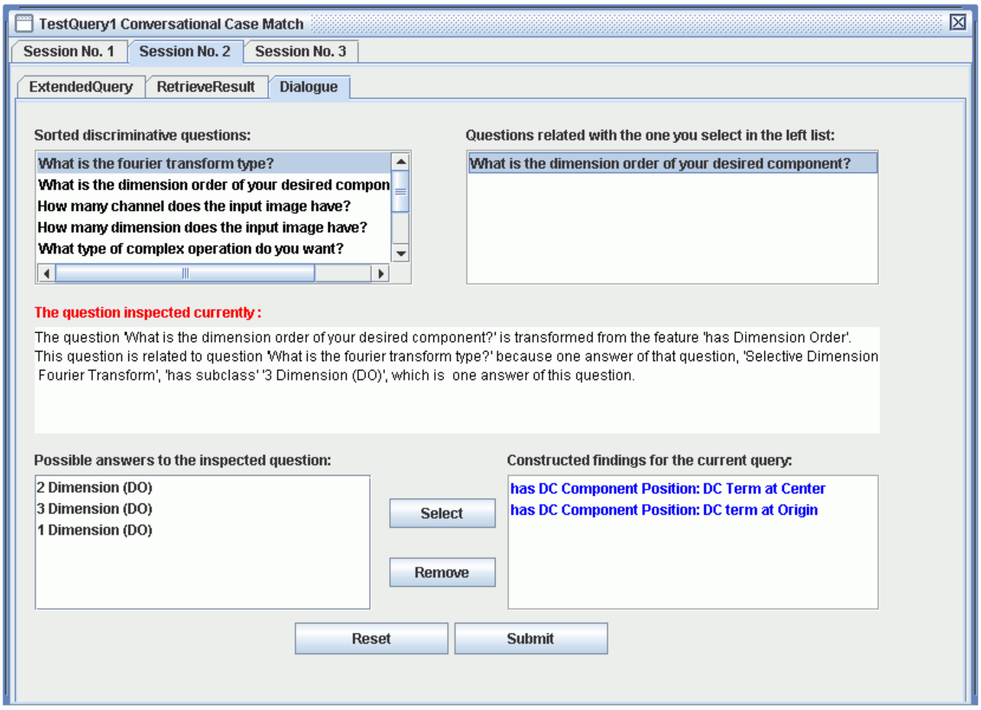

Figure 2.6: The 'Dialogue' pane - The interface used to show the discriminative questions and support the question-answering dialog. This figure gives the explanation about why a question is clustered together with another question by the consistent question-clustering module.

sequencing module are displayed in the upper-left list. When the user selects a question in this list, its alternative answers will be displayed in the bottom-left list and the explanation to this question is shown in the middle text area. In this example, as shown in Fig. 2.5, when selecting the question, 'What is the fourier transform type?', its alternative answers, 'Selective Dimension Fourier Transform' and 'n-Dimension Fourier Transform', are listed. The explanation to this question is: the question 'What is the fourier transform type?' is transformed from the feature 'has fourier transform type' and this question has higher ranking priority because one of its alternative values, 'Selective Dimension Fourier Transform', is the subclass of the answer, 'Fourier Transform', which is inputted into the system in the last question-answering cycle. If there are related questions to the current selected question, they will appear in the upper-right list. When the user selects one related question, its alternative answers and the explanation are displayed on this pane (shown in Fig. 2.6).

When one or several questions are selected and answered, a new conversation session can be started through clicking the 'Submit' button on the Dialogue pane. A user also has the option to go back to one of the conversational session pages to re-choose the questions or adjust their answers and re-submit them.

2.5 Research Evaluation

The research reported in the thesis was evaluated through a combination of different evaluation methods from various perspectives. From the evaluation subject point of view, we used the cross validation methods to simulate the human-computer interaction and also invited the domain experts to evaluate the system concerning the cognitive or psychological aspects.

- The leave-one-out cross validation method was used to evaluate the similarity calculation methods in conversational CBR and the lazy dialog learning mechanism. The leave-one-in cross validation method was used to complete the ablation evaluation of the individual contributions of each knowledge-intensive module in TrollCCRM. The performance criterion for these evaluations is the average number of conversation sessions to find the correct case.

- Four experts from two different domains were invited to execute the TrollCCRM system in an attempt to evaluate whether the knowledge-intensive conversational CBR method can alleviate users' cognitive load during the conversational component retrieval process.

From the used data source perspective, we used both the simple data sets from the UCI data repository and a complex image processing software component knowledge base.

- The UCI data sets were used to evaluate the similarity calculation methods in conversational CBR and the lazy dialog learning mechanism. These two evaluation aspects are domain-independent, and the statistical evaluation results on multiple data sets are able to provide solid evidence supporting their generalities.

- We constructed an image processing software component knowledge base to illustrate whether the TrollCCRM could really utilize the general domain knowledge to improve the conversation efficiency and naturalness. To complement the weakness of the limited number of available knowledge bases, we used multiple evaluation methods on the same knowledge base to see whether they could output consistent results. The multiple evaluation methods include the system characteristic analysis, the domain expert validation, and the simulated ablation evaluation.

In order to compare the performances of different similarity methods in conversational CBR, we implemented three variants of the conversational CBR processes within Weka, each of which used a particular similarity calculation method (query-biased, case-biased and equally-biased). On 31 out of the entire 36 data sets from the UCI data repository, the conversational CBR process using the query-biased similarity method achieved higher performance than the processes using the case-biased or the equally-biased method.

The lazy dialog learner was implemented and evaluated within Weka using 32 UCI data sets. The results showed us that the introduction of the lazy dialog learner continuously improved the retrieval efficiency on 29 out of the entire 32 data sets. And, following the argument by the author in (Bareiss 1989) that it was important to inspect the characteristics of the case base in a case-based learning system in its evaluation phase, we studied the growth rate of the dialog case base during the learning process in the evaluation. The results showed us that the dialog case base was maintainable, that is, with the dialog learning process going on, fewer and fewer dialog cases were added into the dialog case base.

In (McSherry 2001a), seven characteristics were identified for a sequential diagnosis system, and the characteristic analysis results showed us that the TrollCCRM supports the entire seven characteristics. The domain expert assessment showed that TrollCCRM did provide a natural conversation process. The simulated ablation evaluation results told us that the TrollCCRM used 16% fewer questions to find the correct component compared with the knowledge-poor conversational CBR-based component retrieval process, and the coherent question sequencing module contributed to the performance improvement most (76%).

The evaluations to the similarity calculation methods and the dialog learner using UCI data sets were reported in (Similarity functions, Gu&Tong&A-amodt, ICIRI2005) and (Dialog learning, Gu&Aamodt, FLAIRS2006), respectively. The evaluation of TrollCCRM on the image processing software component knowledge base was described in (Evaluation strategy, Gu&Aamodt, ECCBR2006).

We did not give much information about the expert evaluation in (Evaluation strategy, Gu&Aamodt, ECCBR2006). The hypothesis list and the feedback form used in the expert evaluation are included in the current thesis as Appendixes A and B. We assigned three different utility values to each answer of a question. The utility value assignment guideline was that the highest utility value, 2, was assigned to the answer that indicated TrollCCRM had more advantages than the comparing method (one-shot CBR-based retrieval method); the middle utility value, 1, went to the answer that indicated TrollCCRM had modest advantages; and the lowest utility value, 0, was given to the answer that made no difference between TrollCCRM and the comparing method. The concrete value assignment can be found in Table 2.1. Fig. 2.7 illustrates the summarized results of the collected feedback forms, which gives an impression that the experts give the positive evaluation to TrollCCRM for all the entire identified questions. The expert evaluation conclusions in (Evaluation strategy, Gu&Aamodt, ECCBR2006) were not only based on the collected feedback forms, but also based on the informal interviews with the experts.

Because of the complexity of CBR systems (Santamaria and Ram 1994), it is difficult to construct or adapt multiple complex case bases to evaluate CBR

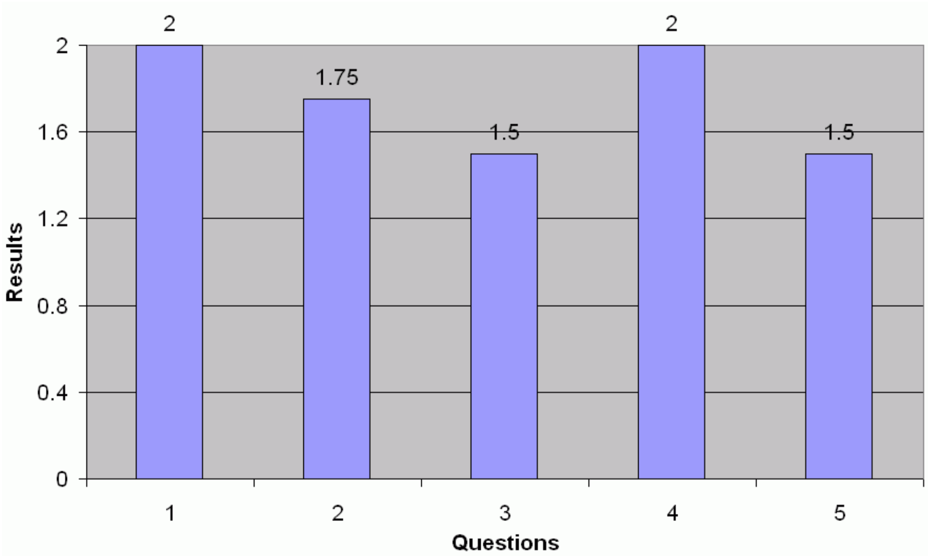

Figure 2.7: The direct expert evaluation results based on the collected feedback forms.

Table 2.1: The utility value assignment for each question based on its answer

Questions	2	1	0
Question 1	Answer B	Answer C	Answer A
Question 2	Answer A	Answer B	Answer C
Question 3	Answer A	Answer B	Answer C
Question 4	Answer A	Answer B	Answer C
Question 5	Answer A	Answer B	Answer C

systems (Díaz-Agudo and González-Calero 2000). We designed an evaluation strategy for CBR systems from the perspective of using different data sources to satisfy various evaluation goals. The proposed evaluation strategy was described in (Evaluation strategy, Gu&Aamodt, ECCBR2006), and the evaluation process to our research was included in that paper as a case study.

Chapter 3

Errata and Supplements

This chapter contains the error corrections and supplements for each included paper. The eight papers included in the present thesis are selected from the 13 total publications during my PhD research period. Papers (CCRM application, Gu&Aamodt&Tong, ICIIP2004) and (Dialog learning, Gu&Aamodt, FLAIRS2006) are re-formatted from two-column style into one column style in order to avoid the tiny fonts. The other six papers are in their original publication format.

Concept Clarification

Different names were used to refer to the same type of cases in the included papers, so we define them here and refer to these definitions in each paper interweaving these names.

- Stored Case (Previous Case)

  Stored cases refer to the cases used to represent the episodic problem solving experiences and stored in the case base. A stored case includes all the three parts of the case representation: the problem description, the solution description, and the outcome.

- New Case (Target Case)

  A new case is used to represent the current unsolved problem, which might be fully specified, as in the traditional CBR process, or partially specified, as in the conversational CBR process. A new case only contains the problem description part compared with a stored case.

- Correct Case (Right Case)

  The correct case refers to the most similar or appropriate stored case reused to help in solving the unsolved problem or the most similar case returned, based on the fully specified new case. At the middle stage of a conversational CBR process, the most similar case to the current partially specified new case is called 'the most similar case to the current new case' instead of a right case or correct case.

- Base Case

  In the evaluation part of the research, we use cross-validation[1] to simulate the human-computer conversation process in conversational CBR. In the simulation process, the problem description of a stored case in the case base is taken as a new problem description. At the beginning, a small part of $< feature, value >$ pairs in this stored case is used to construct the initial new case. One or several further $< feature, value >$ pairs in this stored case are put into the new case in each dialog session to simulate the question-answering process in conversational CBR. In this cross validation process, we call a stored case used to incrementally construct a new case a base case.

3.1 Paper A

In this paper, two question selection tasks, the feature inferencing and the integrated question ranking, were identified and handled. In (Extended KI-CCBR, Gu&Aamodt, ICCBR2005), the entire four question selection tasks were analyzed and handled. The reason why this paper is still included in the thesis is because it provides an integrated example that helps readers to understand the explanation-boosted question selection process.

The terms 'desired case' and 'case query' used in this paper should be 'correct case (right case)' and 'new case (target case)', respectively, according to the definitions at the beginning of this chapter. The terms 'dialog inferencing' and 'Explanation Path Construction' used in this paper have the same meaning as the terms 'feature inferencing' and 'Transitive Explanation Construction' used in (Extended KI-CCBR, Gu&Aamodt, ICCBR2005), respectively. And the 'follows' relation used in this paper was renamed as the 'appears after' relation in (Extended KI-CCBR, Gu&Aamodt, ICCBR2005) in order to avoid the confusion between 'follows' and 'succeeds', which was used in the 'coherent question sequencing' task in (Extended KI-CCBR, Gu&Aamodt, ICCBR2005).

In this paper and (Extended KI-CCBR, Gu&Aamodt, ICCBR2005), we defined the "Dependency relations" as "the appearance of one concept depends on the existence of the other", which is confused with the definition of the "necessary condition" in the classical logics. The dependency relation in our research should be defined as the same as in (Gupta, Aha, and Sandhu 2002): the dependency relation is about what needs to be true before it is worthwhile to check out the value of another proposition.

[1]The detailed cross validation processes can be found in Papers (Similarity functions, Gu&Tong&Aamodt, ICIRI2005), (Dialog learning, Gu&Aamodt, FLAIRS2006) and (Evaluation strategy, Gu&Aamodt, ECCBR2006).

3.2 Paper B

In this paper, each concept in a knowledge base is described by a list of slots. The term 'slot' has the same meaning as the term 'relationship' used in other papers to describe the frame structure of a concept.

The term 'explanation driven' used in this paper has the same meaning as the term 'explanation boosted' used in other papers and the present thesis. The CCRM model reported in this paper only displays the most discriminative question on the screen, and if the user cannot or will not answer it, the system will prompt the next most discriminative question. According to the research findings in conversational CBR (McSherry 2001a), providing a group of candidate questions for users to choose from is a type of 'mixed-initiative', which is one of the important characteristics for sequential diagnosis systems. So, instead of displaying discriminative question one by one, in other papers included in the present thesis we prompted a group of sorted questions (5-9 questions (Miller 1956)) for users to inspect.

In this paper, we proposed using the information metric as the syntactical question-ranking metric for CCRM. This proposal was based on the analysis that the software component base was irreducible and of a low heterogeneous level. But later, when we really completed the component knowledge base, we found its heterogeneous level was rather high. So, instead of the information metric, we selected the occurrence frequency metric in the TrollCCRM system.

3.3 Paper C

The terms 'base case' and 'target case' used in this paper should be 'correct case (right case)' and 'base case', respectively, according to the definitions at the beginning of this chapter.

In line 8 on Page 6 of this paper, the sentence, '... why do dot you show me ...', should be '... why do not you show me...'.

In line 11 on Table 4 of this paper, the sentence, '... and $X_q \neq$ null' should be '... and X has more features left'.

The third hypothesis in Page 12 should be 'H3: there does not exist a performance difference between the conversational CBR system using the case-biased similarity calculation method and the one using the equally-biased method (that is, these two systems use almost the same number of conversation sessions to find the correct case)'.

When we designed the experiment for the research concerning the dialog learning (Dialog learning, Gu&Aamodt, FLAIRS2006), we reused some of the code originally designed for the experiment for the similarity calculation method. When we inspected the program code, we found we made a mistake in that Equation 2 was not consistent with the code we actually used in the experiment. The correct equation should be as in Equation 3.1.

Table 3.1: The updated hypothesis test results for (Similarity functions, Gu&Tong&Aamodt, ICIRI2005)

Null hypothesis	Tailed type	Critical value	t-value	Result
$H0\_1 : V_{Equally} = V_{Query}$	one-tailed	2.44	3.81	refuse H0_1
$H0\_2 : V_{Case} = V_{Query}$	one-tailed	2.44	4.05	refuse H0_2
$H0\_3 : V_{Case} = V_{Equally}$	two-tailed	2.724	1.68	can not refuse H0_3

$$dif(q_f, c_f) = \begin{cases} |q_f - c_f| & f \text{ is a numerical feature (normalized)} \\ max\{q_f, 1 - q_f\} & f \text{ is a numerical feature (normalized)} \\ & \quad \text{and } c_f \text{ is missing} \\ max\{c_f, 1 - c_f\} & f \text{ is a numerical feature (normalized)} \\ & \quad \text{and } q_f \text{ is missing} \\ 0 & f \text{ is a nominal feature, and } q_f = c_f \\ 1 & f \text{ is a nominal feature, and } q_f \neq c_f \\ 1 & f \text{ is a nominal feature, and } q_f \ or \ c_f \\ & \quad \text{is missing} \end{cases} \tag{3.1}$$

In this paper, we used the one-sample t-test in the significance tests for the three hypotheses. The test parameters for each hypothesis were calculated through subtracting one series of experiment results from another. For example, the values in the column 'Equally-Query' were chosen as the test parameter for the first hypothesis, which were computed by subtracting the values in column 'Query Biased' from those in column 'Equally Biased'. Compared with the one-sample t-test, the two-related-samples t-test should be more suitable for these hypothesis tests[2] (Cooper and Schindler 2003). The two-related-samples t-test was executed for the three hypotheses, in which the values in column 'Equally Biased' and those in column 'Query Biased' were selected as the test parameter for the first hypothesis, 'Case Biased' and 'Query Biased' for the second hypothesis, and 'Case Biased' and 'Equally Biased' for the third hypothesis. With the degree of freedom of 35 and the significance level of 0.01, the significance test results were illustrated in Table 3.1. Since the hypothesis test results using the two-related-samples t-test are the same as those using the one-sample t-test, the conclusions made in this paper are still valid.

As a related work, the authors in (Bogaerts and Leake 2004) also concentrated on the problem of how to calculate the similarities between partial specified problem descriptions. Their solution focuses on assigning or calculating a

[2]We used the two-related-samples t-test in (Dialog learning, Gu&Aamodt, FLAIRS2006).

difference value for a feature on which either a new case or a stored case has a missing value. Their difference value assignment strategies on value-missing features include:

- Default Difference (x)

  This strategy assigns a fixed default difference, x, whenever the feature in either the new case or the stored case has a missing value. An optimistic approach normally assigns a minimal default difference value, e.g., ’0’, while a pessimistic approach likely assigns a maximal difference value, e.g., ’1’.

- Full Mean

  In this strategy, if the value of a feature in one of the counted two cases is missing, the mean value of that feature for all the available cases is assigned as its estimated value and used to calculate the difference value on that feature. If a feature misses its value on both the new case and the stored case, the Default Difference (x) strategy is applied.

- NN Mean (Nearest Neighbor Mean)

  Compared with the Full Mean strategy, the NN Mean strategy computes the estimated value for a value-missing feature through a CBR-based process. Instead of the mean value for all the available cases, the mean value for the similar cases returned by a CBR process is assigned as the estimated value for the value-missing feature. The shortcoming of this strategy is its high computation cost of executing the NN algorithm online to find the similar cases.

- Region Mean

  The Region Mean strategy tries to maintain NN Mean’s advantage and avoid its high execution cost. This strategy clusters all cases offline and constructs a prototype case for each cluster with the mean values in that cluster as its feature values. Online execution time for this strategy is reduced compared with the NN Mean strategy through executing the CBR process in the small prototype case base instead of the large original case base.

- Composite Strategies

  Feature inter-dependencies are taken into account in Composite Strategies. The meaning of ’composite’ comes from using one strategy for independent features and another for dependent features.

Our similarity calculation framework focuses on the scope of the features that are considered in the similarity calculation. Their method takes all the features into account, which corresponds to the equally-biased similarity methods in our framework. The difference value assignment strategy on value-missing features used in our framework is a type of pessimistic strategy. However, unlike the Default Difference (x) strategy, when assigning a difference value on a feature, the context information is also taken into account. For example, for a numeric feature, if its value is missing in one of the counted case, the difference value is calculated using the formula: $max\{y, 1-y\}$, where y is the known value of the numeric feature in the other counted case[3]. In addition, we actually assign the difference value 0 to the features that are not considered during the similarity calculation in the case-biased and query-biased similarity methods, which is one type of pessimistic difference value assignment strategy.

3.4 Paper D

The terms 'desired case' and 'case query' used in this paper should be 'correct case (right case)' and 'new case (target case)', respectively, according to the definitions at the beginning of this chapter. The caption of Fig. 3 in this paper should be 'A part of the knowledge base in the car fault detection application in TrollCreek'.

3.5 Paper E

The term 'base case' used in this paper should be 'correct case (right case)' according to the definitions at the beginning of this chapter.

As discussed in the supplements to (CCRM application, Gu&Aamodt&Tong, ICIIP2004), the occurrence frequency metric should be selected as the syntactic question-ranking method instead of the information metric in this paper.

The proposed Equation 2 in this paper should be changed to Equation 3.2. The modification is mainly concerning how to assign the difference value for a numeric feature if it has the missing value in the corresponding stored case. Instead of the blinded pessimistic value, 1, a possible maximal pessimistic value is calculated.

[3]Referring to the corrected Equation 3.1

$$
dif(fv, c) =
\begin{cases}
1 & f \text{ is a nominal feature, and } c \text{ has not the } fv \text{ pair in its problem description} \\
0 & f \text{ is a nominal feature, and } c \text{ has the same } fv \text{ pair in its problem description} \\
max\{v, 1-v\} & f \text{ is a normalized numeric feature, } v \text{ is a single number and } c \text{ has missing value on } f \\
max\{bottom(v), 1-top(v)\} & f \text{ is a normalized numeric feature, } v \text{ is a interval value and } c \text{ has missing value on } f \\
|v - near(f, v, c)| & f \text{ is a normalized numeric feature, and } v \text{ is a single number} \\
1 - (\frac{len(cov(f,v,c))}{len(v)}) & f \text{ is a normalized numeric features, and v is an interval value}
\end{cases}
\tag{3.2}
$$

where fv is a $< feature, value >$ pair specified in the new case, c is a stored case, $bottom(v)$ and $top(v)$ return the bottom value and top value of interval v, respectively, $near(f, v, c)$ is a function to find out the nearest value of v in c on feature f, $cov(f, v, c)$ is a function to return the subintervals of v covered by the values in c on feature f, and $len()$ is a function to compute the length of its parameter (interval).

3.6 Paper F

The term 'target case' used in the experiment section of this paper should be replaced with 'base case' according to the definitions at the beginning of this chapter.

3.7 Paper G

For this paper, there are no corrections or supplements.

3.8 Paper H

The terms 'satisfactory stored case' and 'base case' used in this paper should both be 'correct case (right case)', and the terms 'target case' and 'test case' should both be 'base case' according to the definitions at the beginning of this chapter.

Chapter 4

Conclusions and Discussions

The research presented in the thesis is trying to build a conversational component retrieval system to alleviate users' difficulty in well-defining the component queries and help users to find their desired components. The contributions come from two directions: theoretical research on conversational CBR to provide an efficient and natural conversation process, and the application of conversational CBR to support component retrieval. In Section 1 we summarise our research work and list the future work. In Section 2, from the problem acquisition and the dialog management points of view, we discuss other research directions beyond the incremental problem description construction method in conversational CBR.

4.1 Conclusions and Future Work

As illustrated in Fig. 4.1, we try to provide an efficient and natural conversation process in CCBR, which is further formalized into five question selection tasks: the feature inferencing, the knowledge-intensive similarity calculation, the knowledge-intensive question ranking, the consistent question clustering and the coherent question sequencing. Based on the achievements of our group in knowledge-intensive CBR research, we proposed a knowledge-intensive conversational CBR method, in which we used a common explanation-boosted process to solve all these five tasks (Explanation-Boosted KI-CCBR, Gu&Aamodt, ECCBR2004) and (Extended KI-CCBR, Gu&Aamodt, ICCBR2005).

Noticing the special characteristic of the new case in conversational CBR, i.e., the new case is partially specified or incompletely defined, we proposed to use query-biased similarity calculation methods for conversational CBR applications. Through returning the cases that most satisfy users' current attention focus or interests (reflected in the specified features in the new case), the query-biased similarity method can benefit conversational CBR applications by using fewer conversation sessions to find the correct case than other similarity methods (i.e., the case-biased and the equally-biased methods) (Similarity functions,

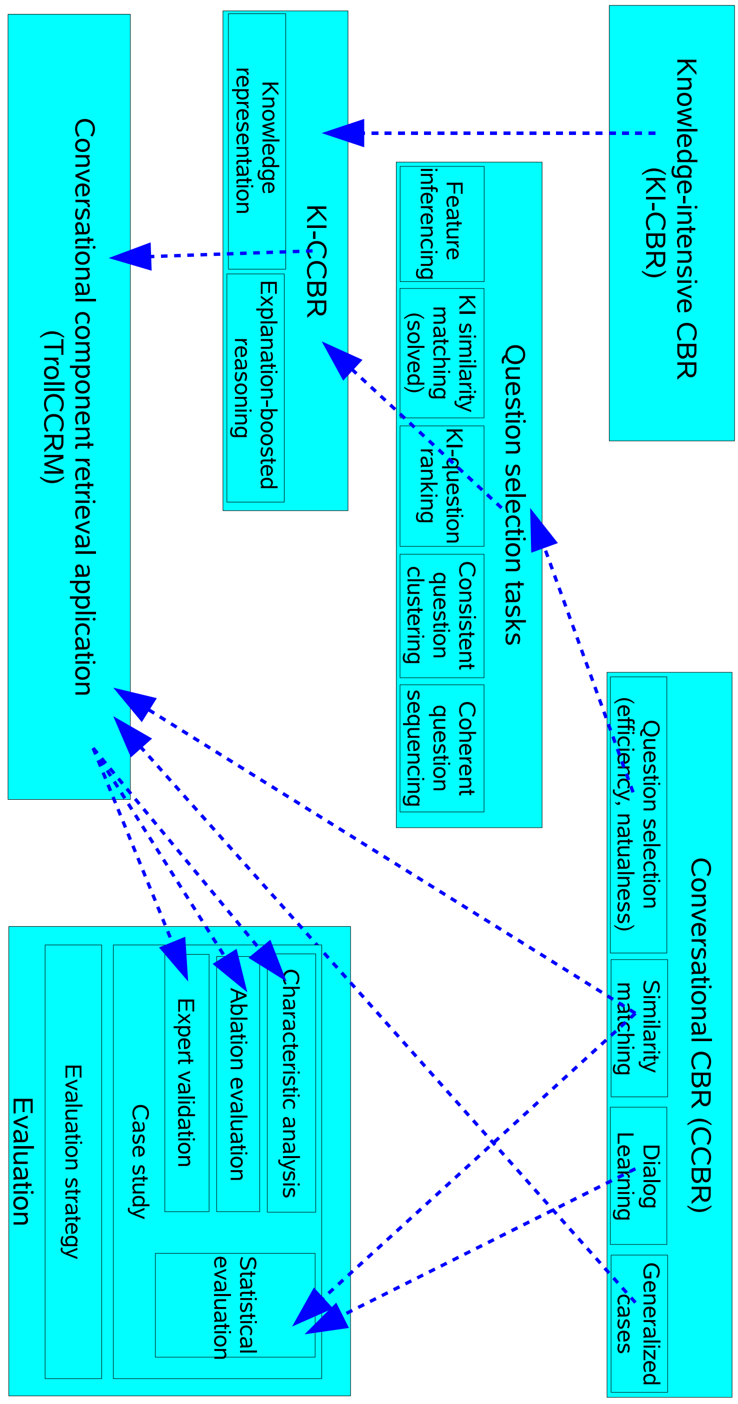

Figure 4.1: The structure of our research

Gu&Tong&Aamodt, ICIRI2005).

In order to provide a continuously improved conversational process in conversational CBR, we proposed a dialog learning method. This method is able to capture and organize the successful conversational case retrieval experiences as dialog cases, which can be retrieved and reused in the later conversational case retrieval process to improve the retrieval performance (Dialog learning, Gu&Aamodt, FLAIRS2006).

We analyzed the current software component retrieval methods and attempted to release the unrealistic assumption inside them, i.e., users can well define their component queries. Our solution is to use our proposed knowledge-intensive conversational CBR method (except the dialog learning mechanism) to extract users' component requirements incrementally through a question-answering dialog process (CCRM application, Gu&Aamodt&Tong, ICIIP2004) and (Extended CCRM application, Gu&Bø, IEA/AIE2006).

In order to apply conversational CBR in software component retrieval, it is necessary to extend conversational CBR to support generalized cases, which are used to represent both software components and component queries. In (Generalized cases, Gu, MICAI2005), we discussed the feasibilities and possible methods to support generalized cases in conversational CBR from three aspects: the case representation, the similarity calculation method, and the question-ranking method.

Our proposed conversational component retrieval model (CCRM) was further implemented as the TrollCCRM system. This system was evaluated on an image processing software component knowledge base through applying multiple evaluation methods, the characteristic analysis, the expert validation and the ablation study. The evaluation results so far gave us positive evidence that this model provided an efficient and natural conversation process for software component retrieval (Papers (Extended CCRM application, Gu&Bø, IEA/AIE2006) and (Evaluation strategy, Gu&Aamodt, ECCBR2006)).

Besides the evaluation of the TrollCCRM system with the image processing component knowledge base, we also assessed the similarity calculation methods in CCBR and the dialog learning mechanism using the statistical evaluation method on many simple case bases. The combined evaluation process has been generalized as an evaluation strategy for CBR applications in general (Evaluation strategy, Gu&Aamodt, ECCBR2006).

One limitation of our research is the assumption of the pre-existence of the knowledge base, including both the software components and the corresponding general domain knowledge. The knowledge base construction puts a considerable workload on knowledge engineers. However, recent achievements in the area of knowledge acquisition and modeling provide systematic methods to help reduce this problem (Aamodt 2001).

In the knowledge base, the general domain knowledge is stored, which explicitly expresses the concepts and their relations. This knowledge can conflict with the knowledge provided by the user, i.e., the initial problem description and the later answers to discriminative questions. For example, if the relationship '*A causes B*' is in the knowledge base, and the user-provided knowledge includes '*A*' and '*not*(*B*)', '*B*' and '*not*(*B*)' conflict with each other. As discussed in (Evaluation strategy, Gu&Aamodt, ECCBR2006), feature inferencing partially handles the knowledge inconsistency problem by inferring the answers to some questions automatically, instead of asking users for their answers. Further, an automatic knowledge inconsistency detecting mechanism can be incorporated to reduce this problem. Whenever a user inputs information into the system, the inconsistency detecting process is activated to check whether or not the new information is consistent with the knowledge already known by the system. If there is an inconsistency, the user is warned to revise her information or a knowledge base inconsistency message is sent to the knowledge base maintainer to modify the predefined mistakes in the knowledge base. Designing and implementing this knowledge inconsistency detecting mechanism is among our further work.

Other future work is to integrate the dialog learning mechanism into the TrollCCRM system in an attempt to improve its performance. The dialog learning mechanism introduced in (Dialog learning, Gu&Aamodt, FLAIRS2006) is basically knowledge-poor. How to integrate this learning mechanism with a knowledge-intensive conversational CBR process is a new research topic. The challenging questions include how to make use of the feature sequencing information during the knowledge-intensive similarity measurement and how to reuse the similar dialog case in the knowledge-intensive question selection process.

Our future work also includes incorporating TrollCCRM into the DynamicImager system where the software components of the knowledge base originally come from. The initial idea is adding the conversational component retrieval process as an alternative to the current manual component selection method. Our long-term goal is to provide a CBR- and conversational CBR-supported software configuration system. The idea is inspired by the IBROW project (IBROW-project 2005), where both software configurations and software components are represented as cases and only the CBR technology is applied. In our vision, the conversational CBR method can be used to identify the correct configuration or component when there are a large number of configurations and components satisfying the initial requirements.

4.2 Discussions

Besides the conversational CBR-based method that incrementally constructs the under-specified new case, there is some research that tries to release the over-specified new case through a constraint release process. In (McSherry 2004b), encountering an over-specified query (failed query), which means no stored cases can fully satisfy this query, the author proposed a method to provide a mixed-initiative dialog process to guide users selecting and eliminating constraints (requirements) step by step. On each constraint release session, a list of constraints (specified features) are sorted and displayed for users to select to eliminate. The guideline to sorting the constraints is eliminating the fewest constraints to get a successful query. In (Mcsherry 2005), instead of a multi-cycle constraint release sequence, the author further proposed to display all the maximal successful subqueries of the failed query for users to choose. In (Ricci et al. 2002), the authors designed a constraint release strategy that displays the number of the satisfied cases as constraint release heuristics if each specific constraint is eliminated. But these methods assume the pre-existence of the over-specified queries defined using the system-understandable terms or concepts. This assumption may not be practical when users lack the domain expertise or feature-acquisition tests are expensive.

Another type of research in CBR, the critique-based case navigation, can be helpful for the situation where users are not satisfied with the returned case. In (Burke 2002), the author proposed an approach that can guide the user to navigate in the case space based on her critiques to the current inspected case. In this approach, an entry point (case) is initially identified through a named case or a case-based search process and a user's unsatisfied preferences are captured as the critiques to the entry case. The new case is modified based on the identified critiques and a new stored case is returned based on the updated new case. So, a new set of critiques are further identified and used to guide the user to navigate in the case space. This method can be integrated together with conversational CBR through taking the user-selected case in conversational CBR as the entry case. This combination can give a user an option to find the best compromise in the case base if the cases returned by conversational CBR cannot satisfy her totally.

Conversational CBR includes a human-computer dialog process, so dialog management methods have the potential of helping control the conversation process in conversational CBR. As summarized in (Branting, Lester, and Mott 2004; Rudnicky and Xu 1999), dialog management methods are classified into three categories: graph based, frame based, and plan based.

Graph-based dialog management methods (also referred to as call-flow-based methods) guide the dialog interactions following pre-defined finite state machines. In these methods, questions are formalized as states and each anticipated alternative answer to a question leads to a new state (question). The static decision tree based question-answering method belongs to this category. The advantage of these methods is that they provide a well-structured dialog process. The limitations of them include the rigid conversation process, only supporting machine-initiated questions, and incapability to handle unexpected conversational events.

Frame-based dialog management methods (also referred to as slot-filling methods) manage the dialog with the aim to extract all or enough information for a specific task (filling in the items of an empty form or finding the values for a vacant feature vector). Each item in a form or each feature in a feature vector corresponds to a question. Frame-based dialog management methods support a broader range of conversations based on various question ordering methods. Conversational CBR methods basically adopt frame-based dialog management methods, in which questions or features are ranked dynamically and displayed in an efficient and natural way in the conversation process.

One limitation of both graph-based and frame-based dialog management methods is their only supporting the single-goal problem solving. In other words, the target problem for these two types of methods only contains one pre-defined goal and this goal is unchangeable during the conversation process. Unexpected events in the conversation process, e.g., human- or system-initiated information clarification questions, are normally not supported by these methods. Plan-based dialog management methods provide the possibility to support the dynamic or emerging tasks (goals) during the dialog process. Given a task (pre-defined or newly emerged), a plan-based dialog manager will create a domain-specific plan, and incorporate a plan executor or a set of task handlers to realize the devised plan.

In order to provide a natural dialog process in conversational CBR, it is necessary to handle the unexpected events that happen in the dialog process. As a new research direction, the plan-based dialog management mechanism should be introduced into the conversational CBR applications. There have already been several studies targeting this specific topic.

In the RealDialog system (Branting, Lester, and Mott 2004), the authors provide a dialog-management framework, Discourse Goal Stack Model (DGSM). DGSM treats all the dialog tasks, including the initial case identification task, the later information clarification tasks initiated by the system or the user, and the topic-jumping tasks, as discourse goals and pushes them into a discourse goal stack following their appearance sequence. A forest of augmented transition networks (ATNS), in which nodes are discourse goals and arcs are speech acts by the user or the system, are stored beforehand and used to handle each discourse goal popped out from the stack.

In (Gómez-Gauchía et al. 2005), the authors provided a method to deal with users' mood changes during the conversation process. Users' mood information is collected by some simple questions prompted during the conversation process. Users' answers to these questions depict their mood states. Users' current mood states decide whether to continue the current causal loop or change to another causal loop, which further decides the conversation actions taken by the system.

Part II

Papers

Paper A

Explanation-Boosted Question Selection in Conversational CBR

```
@INPROCEEDINGS{Gu2004b,
  author = {Gu, Mingyang and Aamodt, Agnar},
  title = {Explanation-Boosted Question Selection
          in Conversational CBR},
  booktitle = {Proceedings of the ECCBR 2004 Workshops,
      the 7th European Conference on Case-Based Reasoning},
  year = {2004},
  editor = {Pablo Gerv{\'a}s and Kalyan Moy Gupta},
  volume = {142},
  number = {04},
  series = {Technical report},
  pages = {105-114},
  address = {Madrid, Spain},
  month = {30th August - 2nd September},
  organization = {Universidad Complutense de Madrid,
      Departemento de Sistemas Informaticos y Programacion},
}
```

Explanation-Boosted Question Selection in Conversational CBR

Mingyang Gu and Agnar Aamodt

Department of Computer and Information Science, Norwegian University of Science and Technology, Sem Sælands vei 7-9, N-7491, Trondheim, Norway
{Mingyang.Gu, Agnar.Aamodt}@idi.ntnu.no

Abstract. A core research concern in conversational case-based reasoning (CCBR) is how to select the most discriminative and natural questions to ask the user in the conversational process. There are two ways to realize this task: one is to remove the questions whose answers can be inferred from the information a user has provided, which is called dialogue inferencing; the other is to rank the questions to guarantee the most informative questions are asked first, which is referred to as question ranking. In this paper, we present a common explanation-boosted CCBR approach, which utilizes both general domain knowledge and case-specific knowledge to realize dialogue inferencing and question ranking. This approach provides a flexible meta-level knowledge representation model to be able to incorporate richer semantic relations. An application of this approach is illustrated in a car fault detection domain.

1 Introduction

Conversational case-based reasoning (CCBR) [1] is an interactive form of case-based reasoning (CBR)[2]. It uses a mixed-initiative dialog to guide users through a question-answer sequence to refine their problem description incrementally. CCBR applications have been successfully probed in the troubleshooting domain [3], and in the selection of products or services in E-Commerce [4].

As illustrated in Fig. 1, conversational CBR adds user-system interactions to the standard CBR cycle [2]. A user's initial textual problem description is formalized into a structured case query (represented as <question, answer> pairs or <attribute, value> pairs). A CBR retrieve process is executed based on the case query and the knowledge base and a set of retrieved cases, sorted decreasingly by their similarities to the case query, are returned. Unknown questions are identified from the retrieved cases and ranked in a certain way. Both the sorted cases and ranked questions are displayed to the user. The user either find his desired cases, which means the CCBR process is completed, or

```
CaseQuery := Case-Query-formalize(InitialProblemDescription);
Repeat:
    CaseQuery:=Dialogue-Inference(CaseQuery);
    RankedRetrievedCases := CBR-Retrieve(CaseQuery);
    UnknownQuestions := Question-Identify(RankedRetrievedCases, CaseQuery);
    RankedUnknownQuestions := Question-Rank(UnknownQuestions);
    Display(RankedUnknownQuestions, RankedRetrievedCases);
    If (users find their desired cases or have no question to answer) then
        Exit loop;
    Else
        UserAnswer := User-Select-and-Answer-Question();
        CaseQuery :=Case-Query-Update(CaseQuery, UserAnswer);
```

Fig. 1. The conversational case retrieval process in CCBR

select a question to answer, which is followed by a new round of retrieve and conversation based on the updated case query.

A core concern in conversational CBR is how to minimize the cognitive load required by users to retrieve their desired cases [4], which requires to select the most discriminative [1, 5, 6] and natural [7] questions in the dialogue process.

Up to now, several methods, such as the information gain [4, 8, 9], the occurrence frequency metric [1], and the information quality [5], have been proposed to realize question selection. However, all the methods mentioned above are basically knowledge-poor, that is, they only take the superficial statistical information into account. However, general domain knowledge also has a potential to play a positive role in selecting questions. For example, in a car fault detection system, if users have said that the fuel pipe is broken, the system should be able to infer that the fuel transmission system has a problem instead of still asking users "what is the status of the fuel transmission system". Another example is that if the answer of question A is easier or cheaper to obtain than question B's, or if the answer of question B can be inferred from that of question A, question A should be prompted to users before question B. The first example is referred to as dialogue inferencing [6] (see the underlined line in Fig. 1), which concerns inferring the potential knowledge from the current known knowledge, so the questions that can be answered implicitly by the current known knowledge would not be prompted to users. The second one is referred to as a knowledge-intensive question ranking [5, 10-13] (corresponding to the line in bold in Fig. 1), which ranks the candidate questions based on their semantic relations besides their statistical metrics.

In this paper, we present an explanation-"boosted" reasoning approach to support knowledge-intensive question selection, in which general domain knowledge is captured and integrated as explanatory machinery to support dialogue inferencing and knowledge-intensive question ranking in the CCBR process. Here, what we mean by explanation-boosted reasoning is a particular method for constructing explanation paths that explore general domain knowledge for question selection tasks. These explanation paths can also be displayed to users to justify the involved intelligent actions.

The rest of the paper is organized as follows. In Section 2, we identify several semantic relations relevant to question selection. In Section 3, our explanation-boosted question selection approach is introduced. In Section 4, an application of our approach is illustrated in a car fault detection domain. Related research is summarized in Section 5. Discussions and future work are given at the end (Section 6).

2 Semantic Relations Related to Question Selection

As we discussed in Section 1, general domain knowledge is useful for question selection. In this paper, we identify the following relations among concepts, which influence dialogue inferencing and knowledge-intensive question ranking:

Concept abstraction One factor of a case can be described using concepts at different abstraction levels. The lower level a concept belongs to, the more specifically it can describe this factor. The appearance of a lower level concept can be used to infer the existence of its higher concepts. For example, the concept of "fuel transmission mistake" is a lower level concept than that of "fuel system mistake". Here, we define a relation "subclass of" to express the relation of "concept abstraction". "A is a subclass of B" means A is a lower level concept than B. When it comes to question selection, this relation can be used in two ways. In dialogue inferencing, if A is a subclass of B and we have A, then we can infer B (i.e. we need not ask the question about B). In question ranking, if A is a subclass of B then a question about A should be asked after the question about B [11, 12].

Dependency relations We say there is a dependency relation between two concepts if the appearance of one concept depends on the existence of the other. For instance, the assertion that the fuel pump can pump fuel depends on that the car has fuel in its fuel tank. Here, we define a relation "depends on" to describe dependency relations. "A depends on B" means B is the necessary condition for A. This relation can also be used in question selection. In dialogue inferencing, if A depends on B and we have A, then we can infer B. In question ranking, if A depends on B, then a question about A should be asked after the question about B [10, 11].

Causality relations The causality relation means one concept can cause the occurrence of another concept. For example, an electricity system mistake in a car can cause its engine not to start. Here, we define a relation "causes" to express the causality relation. "A causes B" means B is the result of A. We can make use of this relation in question selection. In dialogue inferencing, if A causes B and we have A then we can infer B. In question ranking, if A is caused by ("caused by" is the inverse relation of "causes") B then a question about A should be asked after the question about B since if we get B from the question about B, we need not ask users the question about A.

Correlation relations A particular relation, "correlates", is defined to express the relationship between two concepts that they always happen together, even though we can not tell which one causes the other. This "correlates" relation can only be used in dialogue inferencing (from each of these two concepts, we can infer the other), but not in question ranking.

Practical costs The costs to obtain answers to different questions are various. For instance, to test whether a switch has a mistake is more difficult than to test whether the battery has electricity. The relation "more costly than" is defined to represent that to obtain the answer to one question is more difficult than to obtain the answer to another question. This "more costly than" relation can be used in question ranking: if A is more costly than B, then A should be asked after B [5].

3 An Explanation-Boosted Question Selection Approach

In this section, we introduce our explanation-boosted question selection approach from three perspectives: knowledge representation, explanation construction, and explanation-boosted reasoning.

3.1 Knowledge Representation

Knowledge is represented at two levels in our approach: the first one is the object-level, in which case-specific knowledge and general domain knowledge are represented within a single representation framework; the other is the meta-level, which is used to express the inter-relations of the semantic relations introduced in Section 2.

3.1.1 An Object-Level Knowledge Representation Model

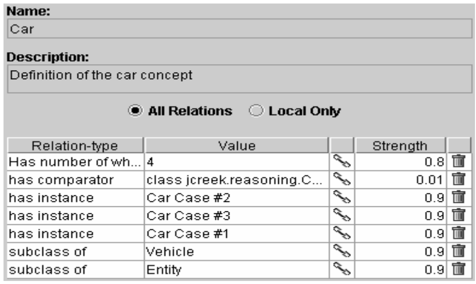

Fig. 2. The frame structure for the concept of car in CREEK

In our approach, a frame-based knowledge representation model, which is a part of the CREEK system [14-16], is adopted to represent the object-level knowledge. In CREEK, both case-specific knowledge and general domain knowledge is represented as concepts, and a concept takes the form of a frame-based structure, which consists of a list of relationships. A relationship is described using an ordered triple $<C_f, T, C_v>$, in which C_f is the concept described by this relationship, C_v is another concept acting as the value of this relationship (value concept), and T designates the relationship type. The equation $T=C_v$ can also be used to describe a relationship when C_f is default. Viewed as a semantic network, a concept corresponds to a node, and a relationship corresponds to a link between two nodes.

Fig. 2 illustrates, in a frame view, how the car concept in the car fault detection domain is represented in CREEK. Fig. 3 shows, in a network view, a part of the knowledge base for that domain. As seen, the semantic relations identified in Section 2 are represented as relationships connecting different concepts. Cases are integrated into the general domain model, since all case features are defined as concepts in it.

The concepts whose instance concepts appear in the retrieved cases, but do not appear in the case query can be converted into discriminative questions. For example, the concept “fuel pumping status” has an instance concept “can pump fuel” appearing in the retrieved cases, but has no instance concepts appearing in the case query, so a discriminative question “what is the fuel pumping status of your car” is added to the discriminative questions list.

In this paper, we define a function, Q: concepts set → questions set, to complete the operation of transforming from a concept into a question. On this function, we further define the following properties:

- The question transformed from one concept is the same as the questions formed by its instance concepts. For example, Q(“fuel pumping status”)=Q(“can pump fuel”)=Q(“can not pump fuel”)= “what is the fuel pumping status of your car”.
- A set of concepts that share the same transformed question are referred to as a SQCS (Same Question Concepts Set). We only predefine one question for each SQCS, which is connected with the super-concept within the SQCS.
- The semantic relations that exist between two SQCSs are transferred to the two questions generated from these two SQCSs, for instance, the “depends on” relation that “can pump fuel” depends on “has fuel” is transferred to Q(“can pump fuel”) depends on Q(“has fuel”), that is, the question “what is the fuel pumping status of your car” depends on the question “is there any fuel in you fuel tank”.

3.1.2 A Flexible Meta-Level Representation Model and Its Reflective Reasoning Method

We define a basic relation “infers” for dialogue inferencing. The “infers” relation means that if A infers B, we can get B from the existence of A. This relation has the

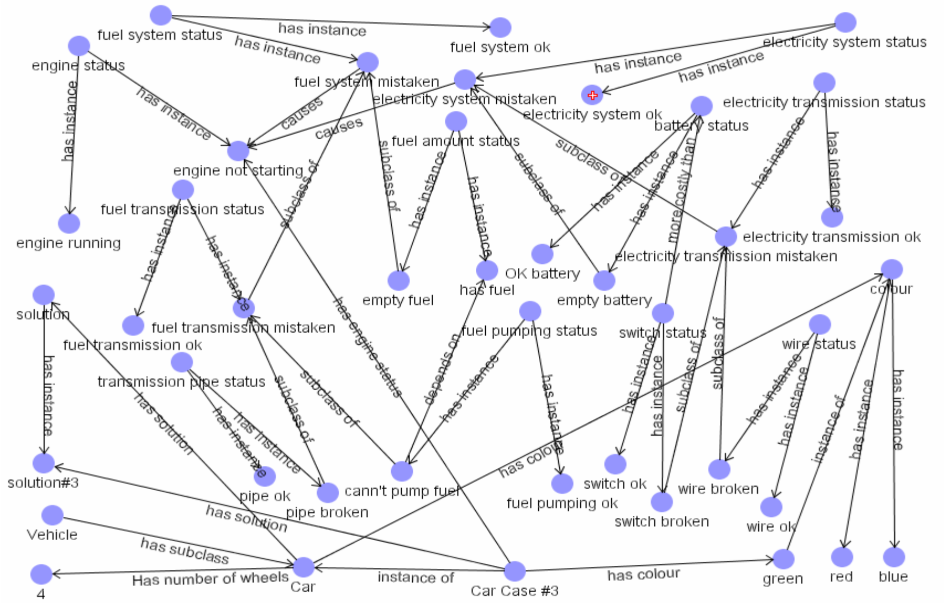

Fig. 3. A part of the knowledge base in the car fault detection application in CREEK

property of transitivity that if A infers B and B infers C then A infers C. We define several semantic relations identified in Section 2, “subclass of”, “depends on”, “causes” and “correlates” as the subclasses of “infers” since all these relations can be used to infer the existence of a post-condition based on the appearance of the pre-condition.

The other fundamental relation “follows” is defined for question ranking. So, “A follows B” means that Q(A) should be asked after Q(B). This relation also has the property of transitivity that if A follows B and B follows C, then A follows C. We define several relations identified in Section 2, “subclass of”, “depends on”, “caused by” and “more costly than” as the subclasses of “follows” because all these relations can rank the pre-condition question to be asked after the post-condition question. Fig. 4 illustrates the meta-level structure for semantic relations described above.

One type of reflective reasoning operation, basic subclass inheritance, is made explicit in this meta-level knowledge representation model. Subclass inheritance makes subclass relations inherit the properties and reasoning operations (i.e. the explanation construction introduced in next sub-section) defined on their parent relations. Thus we need only define the properties and reasoning operations once on the parent relations (“infers” and “follows”), and all their subclass relations, which express much richer domain-specific meanings, can inherit them automatically. The other benefit is that new semantic relations can be easily incorporated through defining them as the subclasses of “infers” or “follows”, and a

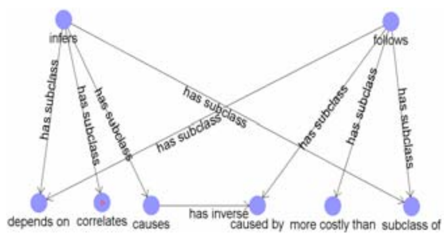

Fig. 4. The semantic relation hierarchy, for reflective reasoning, used to realize dialogue inferencing and question ranking

new application can be easily created through the same process.

3.2 Explanation Construction

Here, explanation construction is setting up a explanation path between two concepts and use it to explore solutions to question selection tasks.

We have defined two levels of explanation construction operations on the "infers" and "follows" relations. The first level is called "Direct Explanation Construction", which is suitable for the case that there is a direct relation between two concepts in the knowledge base. In dialogue inferencing, if concept A exists in a case query and there is a relation "A infers B" in the knowledge base, concept B can be inferred directly and can be integrated into the case query (so Q(B) will be removed from the potential discriminative questions list). In question ranking, if there are two questions Q(A) and Q(B) and there is a relation "A follows B", Q(A) is ranked after Q(B).

The second level is referred to as "Explanation Path Construction", which is suitable when there is no direct relation between two concepts in the knowledge base, but we can set up the "infers" or "follows" relation between them through exploring other relations in the knowledge base. In our group we have developed an abduction-based inference method referred to as plausible inheritance[14, 15], which is adopted to build up the explanation path.

Plausible inheritance is a general relation transitivity mechanism, based on which a relation on one concept can be transferred to another concept following not only the traditional "subclass of" and "instance of" relations, but also other relations, such as "is-part-of", "depends on", "causes" and so on.

In our approach, we define that both the "infers" relation and "follows" relation can be inherited (plausible inheritance) over themselves. So the transitivity property of "infers" relation and "follows" relation is realized. Through combining the subclass inheritance defined on the meta-level knowledge representation model and the plausible inheritance, the "infers" relation and its subclass relations can be transferred over each other. The transitivity property on the "follows" relation and its subclass relations is realized in the same way.

Fig. 5 illustrates an example of how to use plausible inheritance to build up an explanation path for question ranking. In Fig. 5, there are two relations: Q("battery status") is a subclass of Q("electricity system status"), and Q("switch status") is more costly than Q("battery status"), so following the "more costly than" relation, the first relation that Q("battery status") is a subclass of (follows) Q("electricity system status") is transferred to Q("switch status") that Q("switch status") follows Q("electricity system status"). Thus the question ranking explanation path from Q("switch status") to Q("electricity system status") is constructed. Thus if we have two questions Q("switch status") and Q("electricity system status"), we can rank them so that Q("switch status") should be asked after Q("electricity system status") through constructing the above explanation path using plausible inheritance.

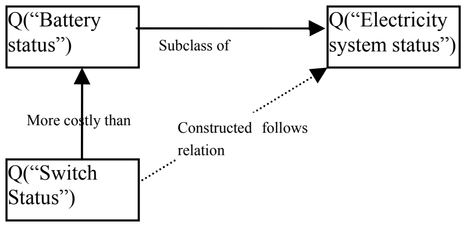

Fig. 5. A "follows" relation is transferred to the concept Q("switch status") using plausible inheritance

In our approach, each relation has a default explanation strength attached to it. The explanation strength of a constructed chain of linked relations, which constitute an explanation path, is calculated on the basis of these defaults (in the example shown in Section 4, we will simply use the product of these defaults to indicate the explanation strength of the constructed explanation path).

3.3 Explanation-Boosted Reasoning

The explanation-boosted reasoning process can be divided into three steps: ACTIVATE, EXPLAIN and FOCUS. ACTIVATE determines what knowledge (including case-specific knowledge and general domain knowledge) is involved in one particular task, EXPLAIN builds up different explanation paths to explore general domain knowledge related solutions for that task, and FOCUS is used to evaluate the generated explanation paths and identify the practical or best one/ones. A similar process is used in the retrieve phase in CREEK to explore the semantic similarities between a case query and stored cases. In this paper, this common explanation-boosted reasoning process is extended to support dialogue inferencing and knowledge-intensive question ranking tasks. The detailed reasoning steps will be illustrated in Section 4 using an example in a car fault detection application.

4 Exemplified Dialogue Inferencing and Question Ranking

In this section, the knowledge representation models, the explanation construction operations and the explanation-boosted reasoning process, introduced in Section 3, are combined together to illustrate how the dialogue inferencing and knowledge-intensive question ranking are completed in a car fault detection application (part of the domain knowledge used in this section can be found in Fig. 3).

4.1 Explanation-Boosted Dialogue Inferencing

In our approach, dialogue inferencing is tackled through using three steps ACTIVATE, EXPLAIN, and FOCUS.

In the ACTIVATE step, all the relevant knowledge with dialogue inferencing is activated, which includes the case query knowledge and semantic dialogue inferencing relations related to this case query knowledge from the knowledge base. For instance, we have the following case query knowledge (CQK):

- *CQK1: Has fuel transmission status = Fuel transmission mistaken*
- *CQK2: Has electricity system status = Electricity system mistaken*

The activated semantic dialogue inferencing relations (SDIR) include:

- *SDIR1: "Fuel transmission mistaken" is a "subclass of" "fuel system mistaken" (weight: 0.9)*
- *SDIR2: "Fuel system mistaken" "causes" "engine not starting" (weight: 0.8)*
- *SDIR3: "Electricity system mistaken" "causes" "engine not starting" (weight: 0.8)*

The EXPLAIN step uses the case query knowledge and activated semantic dialogue inferencing relations to reason or explain what knowledge can be inferred through using explanation construction operations. In this example, we get the following new case query knowledge (NCQK):

- *NCQK1: Has fuel system status = Fuel system mistaken (based on CQK1 and SDIR1, weight: 0.9)*
- *NCQK2: Has engine status = Engine not starting (based on SDIR1, SDIR2, and CQK1, weight: 0.72)*
- *NCQK3: Has engine status = Engine not starting (based on CQK2 and SDIR3, weight: 0.8)*

In the FOCUS step, all the inferred knowledge is evaluated, and the accepted knowledge is combined together with the current case query to form a new case query. In the evaluation process, only the knowledge whose weights surpass a particular threshold (say 0.8) is accepted, and the redundantly inferred knowledge is removed (the knowledge with the highest weight is kept). In our example, we get a new updated case query that includes CQK1, CQK2, NCQK1 and NCQK3.

4.2 Explanation-Boosted Question Ranking

Based on the updated case query, a retrieve process [16] is executed and the top ranked cases are returned. In this stage, if users can not find their desired cases, an explanation-boosted question ranking process is started, which is also divided into three steps.

In the ACTIVATE step, the unanswered questions are identified (see Section 3.1.1). All the semantic question ranking relations concerned with these identified questions are then activated from the knowledge base.

For instance, from the retrieved cases and the updated case query, we identify the following unanswered questions (UQ) (we assume that retrieved cases include all the value concepts appearing in Fig.3):

- *UQ1: Q("transmission pipe status")*
- *UQ2: Q("fuel amount status")*
- *UQ3: Q("fuel pumping status")*
- *UQ4: Q("switch status")*
- *UQ5: Q("battery status")*
- *UQ6: Q("electricity transmission status")*
- *UQ7: Q("wire status")*
- *UQ8: Q("colour")*

The activated semantic question ranking relations (SQRR) include:

- *SQRR1: Q("Fuel pumping status") "depends on" Q("fuel amount status") (weight: 0.8)*
- *SQRR2: Q("Wire status") is a "subclass of" Q("electricity transmission status") (weight: 0.9)*
- *SQRR3: Q("switch status") is "more costly than" testing Q("battery status") (weight: 0.75)*

The EXPLAIN step uses the identified unanswered questions and the activated semantic question ranking relations to reason or explain which questions should be asked before another one. For instance, we get the following question ranking knowledge (QRK) through using explanation construction operations:

- *QRK1: Q("fuel amount status") should be asked before Q("transmission pumping status") (based on UQ2, UQ3, and SQRR1, weight: 0.8)*
- *QRK2: Q("electricity transmission status") should be asked before Q("wire status") (based on UQ6, UQ7, and SQRR2, weight: 0.9)*
- *QRK3: Q("battery status") should be asked before Q("switch status") (based on UQ5, UQ4, and SQRR3, weight: 0.75)*

In the FOCUS step, the semantic ranking knowledge obtained in the EXPLAIN step is evaluated, and the questions are ranked combining the semantic question ranking knowledge and the superficial statistical metrics. In the evaluation process, only the ranking knowledge whose weights surpass one particular threshold (say 0.8) is accepted. In this case, the QRK3 is refused because its explanation strength is less than 0.8. In the question ranking process, all the questions are classified into two groups firstly: group one includes the questions whose ranking priorities are constrained by the question ranking knowledge (here, it has two questions: UQ3 (constrained by QRK1), and UQ7 (constrained by QRK2)); and group two contains all the remaining questions (UQ1, UQ2, UQ4, UQ5, UQ6 and UQ8). Secondly, the questions in group two are further ranked based on their superficial statistical metrics such as information gain or occurrence frequency. The questions in group one are sorted according to their biggest explanation strength selected from all the question ranking explanation strengths each question gets increasingly. In this example, the questions in group one are ordered as UQ3, UQ7. And the ranked questions in group two are prompted to users followed by the sorted questions in group one.

5 Related Research

In [6], Aha, Maney and Breslow propose a model-based dialogue inferencing method. In their method, the general domain knowledge is represented in a library model (including object models and question models) taking the form of a semantic network. At run time, a set of rules are extracted from the library model using an implication rule generator, and the generated rules and the existing problem description are input to a PARKA-DB to infer potential knowledge.

In [5], the authors try to eliminate the trivial and the repeated questions from users by accessing other information sources to answer them automatically. They take the cost factor into account when selecting a task (question) to execute instead of only the Information Quality metric. In this method, an execution plan is formulated for each question using a hierarchical task network (HTN). The estimated cost for each question is calculated through propagating cost values upward from leaves to the root using the mini-max algorithm.

In [12], Gupta proposes a taxonomic conversational CBR approach to tackle the problems caused by the abstraction relations among features. In his approach, cases are described using one or more factors. On each factor, an independent subsumption taxonomy is created by the library designer in advance, and only the most specific feature on each factor taxonomy is selected to describe a case. The similarity between one <question, answer> pair in a case query and one in a case is calculated based on their relative positions in the taxonomy. The question generated from a higher level feature in one factor taxonomy is constrained to be asked before those that come from the lower level features.

Aha, Gupta and Sandhu identify the dependency relation among features [10, 11]. In their method, dependency relations are only permitted to exist between the root nodes among various factor taxonomies, and the precedent node in one dependency relation is excluded from the case representation. In the question ranking step, the question generated from a precedent node in a dependency relation has higher priority to be asked than the question formalized by the dependent node.

Comparing with the above knowledge-intensive question selection methods, our model contributes to the conversational CBR research in three ways: it provides a common explanation-boosted reasoning process to support both dialogue inferencing and knowledge-intensive question ranking; it can rank discriminative questions through combining both their semantic question ranking relations and their superficial statistical metrics; by creating a meta-level knowledge representation model, our model has the capability to be easily extended to support richer inferring or ranking relations, and to be transformed to other application domains.

6 Discussion and Future Work

We will here address two potential limitations in our approach that need to be tackled in our future work. One is conflicting knowledge correction. We store the general domain knowledge in the knowledge base, which explicitly express the relations among concepts. However, the knowledge provided by users, expressed in case queries, can be conflicting with this stored general domain knowledge. In this case, there should be an automatic mechanism to detect the knowledge conflicts in order to warn users to revise their new cases or help knowledge base designers to update the predefined mistaken knowledge. Another problem is the preference cycle generated by a set of question ranking relations. For example, there are three questions, A, B, and C, and three question ranking relations that A should be asked before B, B should be asked before C, and C should be asked before A, so a preference cycle appears following A, B, C, and A. An automatic preference cycle detecting mechanism in the knowledge input phase will be helpful. Another possible solution is directly ignoring the ranking relation with the least explanation strength in any preference cycle.

References

1. D. W. Aha, L. A. Breslow, and H. Muňoz-Avila, "Conversational Case-Based Reasoning," Applied Intelligence, vol. 14, pp. 9-32, 2001.
2. A. Aamodt and E. Plaza, "Case-Based Reasoning: Foundational Issue, Methodological Variations, and System Approaches," AI Communications, vol. 7, pp. 39-59, 1994.
3. K. M. Gupta, "Knowledge-based system for troubleshooting complex equipment," International Journal of Information and Computing Science, vol. 1, pp. 29-41, 1998.
4. H. Shimazu, "ExpertClerk: A Conversational Case-Based Reasoning Tool for Developing Salesclerk Agents in E-Commerce Webshops," Artificial Intelligence Review, vol. 18, pp. 223 - 244, 2002.
5. C. Carrick, Q. Yang, I. Abi-Zeid, and L. Lamontagne, "Activating CBR Systems through Autonomous Information Gathering," International Conference on Case Based Reasoning, Germany, 1999.
6. D. W. Aha, T. Maney, and L. Breslow, "Supporting Dialogue Inferencing in Conversational Case-Based Reasoning," European Workshop on Case-Based Reasoning, Dublin, 1998.
7. M. Doyle and P. Cunningham, "A Dynamic Approach to Reducing Dialog in On-Line Decision Guides," European Workshop on Advances in Case-Based Reasoning, Trento, 2000.
8. M. H. Göker and C. A. Thompson, "Personalized Conversational Case-Based Recommendation," European Workshop on Case-Based Reasoning, Trento, Italy, 2000.
9. H. Shimazu, A. Shibata, and K. Nihei, "ExpertClerk: A Conversational Case-Based Reasoning Tool for Developing Salesclerk Agents in E-Commerce Webshops," Applied Intelligence, vol. 14, pp. 33-48, 2001.
10. D. W. Aha and K. M. Gupta, "Causal Query Elaboration in Conversational Case-Based Reasoning," International Florida Artificial Intelligence Research Society Conference, Pensacola Beach, Florida, USA, 2002.
11. K. M. Gupta, D. W. Aha, and N. Sandhu, "Exploiting Taxonomic and Causal Relations in Conversational Case Retrieval," European Conference on Case Based Reasoning, Aberdeen, Scotland, UK, 2002.
12. K. M. Gupta, "Taxonomic Conversational Case-Based Reasoning," International Conference on Case-Based Reasoning, Vancouver, BC, Canada, 2001.
13. D. Mcsherry, "Interactive Case-Based Reasoning in Sequential Diagnosis," Applied Intelligence, vol. 14, pp. 65-76, 2001.
14. F. Sørmo, "Plausible Inheritance: Semantic Network Inference for Case-Based Reasoning," Master thesis in Department of Computer and Information Science. Trondheim: Norwegian University of Science and Technology, pp. 102, 2000.
15. A. Aamodt, "A Knowledge Representation System for Integration of General and Case-Specific Knowledge," International Conference on Tools with Artificial Intelligence, New Orleans, 1994.
16. A. Aamodt, "Explanation-driven case-based reasoning," Topics in Case-based reasoning, Springer, pp 274-288, 1994.

Paper B

Component Retrieval Using Conversational Case-Based Reasoning

```
@INPROCEEDINGS{Gu2004a,
  author = {Gu, Mingyang and Aamodt, Agnar and Tong, Xin},
  title = {Component Retrieval Using Conversational
          Case-Based Reasoning},
  booktitle = {Intelligent Information Processing II
          - International Conference on Intelligent
          Information Processing (IIP2004)},
  year = {2004},
  editor = {Shi, Zhongzhi and He, Qing},
  volume = {163},
  series = {IFIP - International Federation for
          Information Processing},
  pages = {259 - 271},
  address = {Beijing, China},
  publisher = {Springer Science + Business Media Inc},
}
```

COMPONENT RETRIEVAL USING CONVERSATIONAL CASE-BASED REASONING

Mingyang Gu[1], Agnar Aamodt[2] and Xin Tong[3]
[1,2,3]*Department of Computer and Information Science, Norwegian University of Science and Technology, Sem Sælands vei 7-9, N-7491, Trondheim, Norway +47 7359 7410 {[1]Mingyang, [2]Agnar, [3]Tongxin}@idi.ntnu.no*

Abstract: Component retrieval, about how to locate and identify appropriate components, is one of the major problems in component reuse. It becomes more critical as more reusable components come from component markets instead of from an in-house component library, and the number of available components is dramatically increasing. In this paper, we review the current component retrieval methods and propose our conversational component retrieval model (CCRM). In CCRM, components are represented as cases, a knowledge-intensive case-based reasoning (CBR) method is adopted to explore context-based semantic similarities between users' query and stored components, and a conversational case-based reasoning (CCBR) technology is selected to acquire users' requirements interactively and incrementally.

Key words: Software Component Retrieval, Conversational Case-Based Reasoning, Knowledge-Intensive Case-Based Reasoning, Semantic Similarity Calculation, Incrementally Query Acquisition

1. INTRODUCTION

One of the major problems associated with component reuse is component retrieval[1, 2, 3], which is concerned with how to locate and identify appropriate components to satisfy users' requirements. This problem becomes more critical as the emergence of several component architecture standards, such as, CORBA, COM, and EJB. These standards make software components interoperate more easily. Therefore component reuse surpasses the limitation of a single software company, that is, instead of getting components from an in-house component library, users search for desired components from component markets[4] (web-based software component collections provided by vendors or third parties), which separate component users and component vendors from each other.

A large and rapidly increasing number of reusable components put more strict demands on the retrieval efficiency[5]. If it is acceptable for users to look through tens of available components to identify the most appropriate ones, it is intolerable for them to look through hundreds, or thousands of candidate components, to select what they really need.

Several methods have been put forward to address the component retrieval problem. Most of them assume users can define their component query clearly and accurately, which puts too much impractical burden on component users. Based on the analysis of current retrieval methods, we propose a component retrieval model combining knowledge-intensive case-based reasoning technologies and conversational case-based reasoning methods.

Case-Based Reasoning (CBR) is a problem solving method[6]. The main idea underlying CBR is that when facing a new problem, we will search in our memory to find the most similar previous problem, and reuse the old solution to help solve the new problem.

A CBR process can be divided into four phases: retrieve, reuse, revise and retain, as described in[6]. Our research, as reported in this paper, focuses on the retrieve phase.

In the retrieve phase, a new case (new problem description) is compared to the stored cases, and the most similar one (ones) will be retrieved. Partial matching is adopted in the retrieve phase. Note that the CBR notion of partial matching, i.e. the matching of a group of features in order to return a best match, and where each feature typically has its own weight, distinguishes this technology from information retrieval and database access methods in general. Some CBR methods are 'knowledge-poor', which only consider superficial or syntactical similarities between a new case and stored cases, while other systems take both the syntactical similarity and the semantic similarity into account by combining case-specific knowledge and general domain knowledge. The latter approach is referred to as knowledge-intensive CBR[7].

Conversational case-based reasoning (CCBR) is an interactive form of case-based reasoning. It uses a mixed-initiative dialog to guide users to facilitate the case retrieval process through a question-answer sequence[8]. In the traditional CBR process, users are expected to provide a well-defined problem description (a new case), and based on such a description, the CBR system can find the most appropriate case. But usually users can not define their problem clearly and accurately. So instead of letting users guess how to describe their problem, CCBR calculates the most discriminative questions automatically and incrementally, and displays them to users to extract information to facilitate the retrieval process.

CCBR has been probed in several application domains, for instance, the customer support domain[9], and products or services selection in E-Commerce[10]. To our knowledge, current CCBR methods are to a large extent based on superficial feature properties, and there are so far no published results on CCBR applied to software component retrieval. In our research, we combine knowledge-intensive CBR and conversational CBR in an attempt to resolve the component retrieval problem.

The rest of this paper is organized as follows. In section 2, we review current existing retrieval methods, briefly discuss their advantages and disadvantages; in section 3, our conversational component retrieval model (CCRM) is proposed and some examples are illustrated; in section 4, we discuss the current status of using CBR technologies in the component retrieval field, and identify the advantages and limitations of our component retrieval model. In the end, we discuss our results so far, and point to future work (section 5).

2. CURRENT COMPONENT RETRIEVAL METHODS

A component retrieval method can be described from three aspects: component representation, component query (users' requirements) specification, and component retrieval process. A popular component retrieval method, named free-text-based retrieval method[11, 12], comes from the information retrieval community. In this method, components are represented as free-text-based documents, while a component query is described using keywords. The retrieval process is to look up the keywords in all component description documents. The components with most matched keywords will be selected. Vector space and indexing technology are used to facilitate documents organizing and matching. This method has low scores on both recall and precision[5]. Researchers and practitioners have proposed to use general thesaurus to extend keywords, by including their synonyms and antonyms, to get more relevant components[13]. In addition, general domain knowledge is also used to extend initial keywords to get more semantically relevant components[5]. However, both of these two improvements increase retrieval recall at the cost of retrieval precision.

Instead of free-text-based component and query descriptions, the following four types of retrieval methods represent components and specify queries using structural information from different perspectives. The pre-enumerated vocabulary method uses a set of pre-defined vocabularies to express both components and queries[14]. In this way, both recall and precision are increased at the cost of the flexibility to describe components and specify component queries. The signature matching method[15] describes both components and queries using signatures which specify the interfaces of components, for instance, the number and the type of input, and output variables. This approach is suitable for components implemented using strongly-typed programming languages. Its weakness is its lack of domain and searching context information. The behavior-based retrieval method[16] is based on the special characteristic of software components being executable. Components take the form of executable codes, and queries are represented by a set of input samples and their desired outputs. The retrieval proceeds by selecting samples, and executing components using the selected samples. The components that satisfy the desired output are retrieved. This method is designed for executable software components and has low efficiency because of long execution time.

The final method we want to mention in this category is faceted selection. This approach predefines a set of dimensions, called facets, which are used to classify components from different perspectives[14]. Users can find their desired components by searching down the stratified categories. This method is getting increasing attention because it takes domain knowledge into account when designing facets. But there exists a design embarrassment: If facets are designed too simple or few, there will be too many components in final categories, which will ask users to select further manually. On the other hand, if facets are designed too complex, it is hard for users to understand them and hard for designers to classify all components into different categories[17, 18]. In addition, the faceted selection method essentially uses the exact matching process. However, it is very hard to get the appropriate components through exact matching because of the universal differences between component requirements and components descriptions[19].

All the retrieval methods mentioned above have one common assumption, that is, users can well define their component queries, and the retrieval system can find one or a few appropriate components according to users' queries. However, this assumption is not always realistic. People often lack clear ideas about what they need while they begin searching for components and usually can not define their queries accurately. They need retrieval system to guide them refining their queries incrementally. Hence, an efficient component retrieval system should be able to support partial matching, select components based on both the syntactical similarity and the semantic similarity, and guide searchers to refine their component query incrementally. Conversational case-based reasoning, extended with knowledge-intensive CBR methods, provides a possibility for satisfying these requirements.

3. THE CONVERSATIONAL COMPONENT RETRIEVAL MODEL (CCRM)

3.1 CCRM Overview

As illustrated in Fig. 1, our conversational component retrieval model (CCRM) includes six parts: a knowledge base, a new case generating module, a knowledge-intensive CBR module, a component displaying module, a question generating and ranking module, and a question displaying module.

The knowledge base stores both component-specific knowledge (cases) and general domain knowledge (including a domain ontology). The new case generating module can set up a new case based on users' initial query and their later answers to discriminative questions. Given a new case, the knowledge-intensive CBR module calculates the similarities between the new case and stored component cases, and returns the components whose similarities surpass a threshold (the threshold is specified initially and can be adjusted following the execution of the system). The component displaying module displays the candidate components to users, ordered by their similarities. In the question generating and ranking module, possible unknown questions are identified, and an information gain algorithm[20] is used to rank the possible questions according to how much information it can provide if it has been answered. Then general knowledge is used to filter out those questions whose answers can be inferred from the initial query or previously answered questions. These ordered questions are further reordered according to some constraints inferred from general knowledge, for example, people normally prefer to answer the high level questions before answering low level ones. The question displaying module selects the most discriminative question, in order to optimize search towards a meaningful answer.

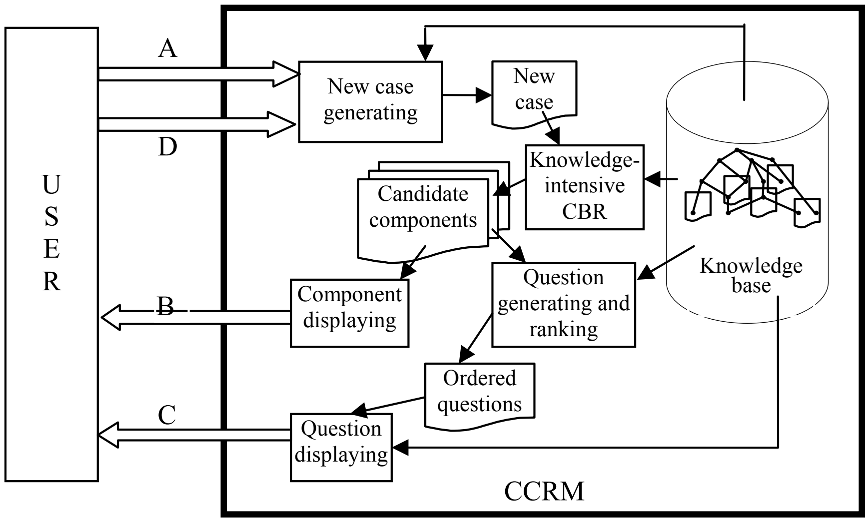

Figure 1. The architecture of conversational component retrieval model (CCRM).

Arrows: A, B, C and D, are interactive processes between users and CCRM. A: users input their initial query; B: the system provides users with top matched components; C: the system displays the most discriminative question to users; D: users select a displayed question and provide their answer to the system. Other processes are completed in the system automatically.

The retrieval process in the CCRM model can be described as the following steps:

1. Users provide their initial query, which takes the form of free-text-based terms.
2. The new case generating module transforms the initial query into a new case. In this step, a general thesaurus and a domain ontology are used to transform the free-text-based initial query into standard terms used in the internal system, and formalize them into a new case.
3. The knowledge-intensive CBR module calculates the similarities between the new case and stored cases through combining both component specific knowledge and general domain knowledge, and the components whose similarities surpass a threshold are returned.

4. If users find their desired component from the displayed candidate components, they can terminate the retrieval process. Otherwise, the conversational process is activated.
5. The question generating and ranking module identifies the unknown questions from the candidate components, and ranks them according to their information gains. Further, the ordered questions are filtered and reordered using general domain knowledge.
6. The question displaying module selects the most discriminative question, and displays it and its meaningful answers to users in a readable format.
7. Users provide the system with their answer to the displayed question. Otherwise, if users can not answer a displayed question, the question displaying module will display the next most discriminative question.
8. The new case generating module combines the previous new case and the newly gained answer to set up a new case.
9. The iterations from 3 to 8 continue until users find their desired component or there are no other discriminative questions left.

3.2 Component Representation in CREEK

In CCRM, we adopt a frame-based knowledge representation and reasoning system, CREEK[21], which can unify component-specific cases and general domain knowledge within a single representation system. In CREEK, all knowledge is represented as concepts, and a concept takes the form of a frame-based structure, which consists of a list of slots. A slot acts as a relation from the concept to a value related with another concept. Viewed as a semantic network, a concept (frame) corresponds to a node, and a relation (slot) corresponds to a link between nodes. Slot values have types or roles, referred to as facet. Typical facets include current value, default value, value class, and value constraint. So the knowledge in CREEK is represented in a 4-level structure, frame, slot, facet and value.

```
OutputComponent (partial)
subclass-of               value             Component
has-instance              value             Write BMP
has-instance              value             Write TIFF
has-instance              value             Write JPEG
has-error                 value             file-open-error
has-number-of-parameter   default           1
has-image-color-space     value-class       Color-space
has-image-dimension       value-class       Image-Dimension
has-image-file-type       value-class       Image-file-type
has-size-constraints      value-constraint  (and (> 0 Bytes) (< 100 MB))
…
```

Figure 2. The partial frame structure of an image OutputComponent concept.

Fig. 2 gives, in a frame view, an example to illustrate how a part of an image OutputComponent class is represented in CREEK. Fig. 3 shows, in a network view, a part of the knowledge base for components used in the image processing field. General domain knowledge can be represented as relations between different values. For example, the “extract to” relation from “3D” to “2D” means that 3 dimension images can be extracted to 2 dimension images. Similarly, the two relations “convert to" between "XYZ" and "RGB" mean that images described using "XYZ" color space and images described using "RGB" color space can be converted to each other without losing any information.

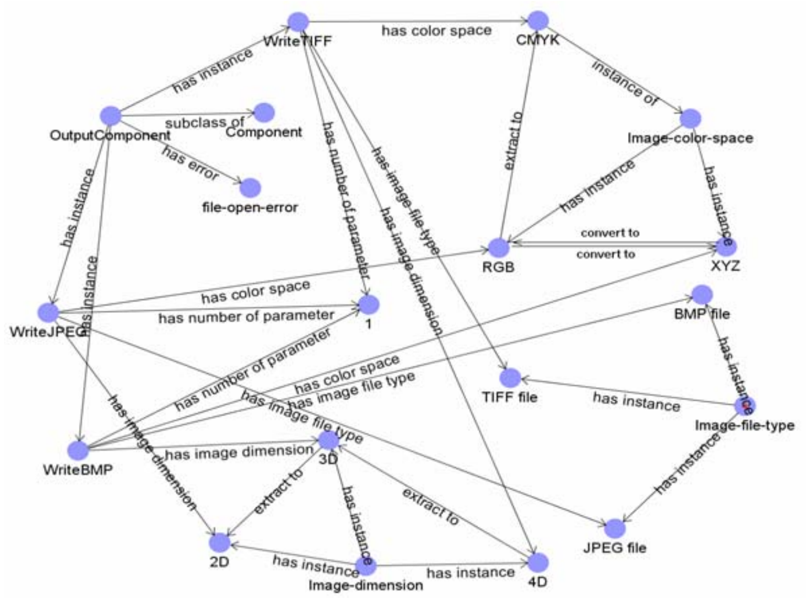

Figure 3. A part of the knowledge base (implemented in the CREEK system) for components in the image processing field.

```
Component query (partial)
has-number-of-parameter      1
has-image-color-apace        RGB
has-image-dimension          3D
has-image-file-type          BMP file
has-error                    file-open-error
has-file-size-constraints    5 megabyte
…
```

```
Write BMP component (partial)
has-number-of-parameter       1
has-image-color-space        XYZ
has-image-dimension          2D
has-image-file-type          BMP file
has-error                    file-open-error
has-file-size-constraints    (and (> 0 Bytes)
                              (< 100 Megabyte))
…
```

Figure 4. The partial frame contents of the component query and the stored component 'Write BMP'.

3.3 Knowledge-Intensive Similarity Calculation

In CCRM, we use an explanation-driven similarity calculating method[7], which can be divided into three steps, ACTIVATE, EXPLAIN and FOCUS. ACTIVATE determines what knowledge in the knowledge base is involved in the retrieval process, and calculates the similarities between the new case and activated stored cases based on a rather syntactical or superficial similarity measuring. The output of the ACTIVATE step is a set of components whose similarity values surpass a certain threshold. EXPLAIN is used to evaluate the similarities between the new case and stored cases, selected in the ACTIVATE step, based on general domain knowledge. The evaluation task concerns justifying that the well-matched slots are relevant to the problem goal, and "explaining away" the mismatched slots that are unimportant.

According to evaluation results, similarity values are adjusted. For instance, if one mismatched slot is evaluated as important for the problem goal, the total similarity value of the involved component is reduced. Otherwise, the similarity value is increased or keeps unchanged.

In the example shown in Fig. 4, there are two mismatched slots, "has-image-color-space" and "has-image-dimension", between the component query and the stored component. With the domain knowledge that "RGB convert to XYZ" and "XYZ convert to RGB", we can explain that "since the source image using RGB color space can be converted to an image using XYZ color space and vice versa, it is possible to use this stored component to realize the required task", and the similarity value generated in the ACTIVATE step can be kept unchanged or increased. On the contrary, there is not any explanation path from 2D to 3D, which means it is impossible for images with 2 dimensions to be converted to images with 3 dimensions, so the mismatch on the "has-image-dimension" slot can not be explain as unimportant and the similarity value of this stored component is reduced.

3.4 Question Selecting and Ranking

There are at least two requirements on the mixed-initiative question-answer interaction in conversational CBR. First, displayed questions should be easy to understand. Second, the selected question should be the most informative or discriminative one.

As to the first requirement, we predefine a question and its possible answers to each slot. For example, on the slot "has-image-file-type", we predefine a question that "what type of images do you want to deal with in this component?" and the possible answers, "BMP", "TIFF", "JPEG", or "Text". All the slots that appear in the candidate components, returned by the knowledge-intensive CBR module, but not in the new case are identified and transformed into unknown questions. Whether or not a possible answer is displayed to users in the conversational process depends on whether this answer appears in the candidate components.

As to the second requirement, "selecting the most informative question", we adopt the information gain metric[20] to quantitatively measure the information one slot (question) can provide (if we know the value of this slot).

The core concept in information gain is entropy. Given a collection S, its entropy value in state m can be calculated using the following formula:

$$Entropy\ (S_m) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

The number c means how many sub-groups the collection can be divided into, and p_i means the proportion of the ith sub-group. If we can not classify a collection of components into sub-groups, its entropy is 0 (c=1, p_1=1). After we acquire information on slot n, the collection can be classified into different sub-groups according to their various values on slot n, and the collection's entropy is increased. Information gain of slot n is defined as:

$$InformationGain(slot\_n) = Entropy(S_{have-\inf ormation-about-slot\_n}) - Entropy(S_{have-no-\inf ormation-about-slot\_n})$$

Different slots have different information gain. The larger the information gain one slot has, the more information it can provide if we know the value for this slot. That is, to find the most informative question is to find the slot with the largest information gain.

For instance, there is a candidate component collection with the number of 100, and there are two unknown slots in the new case, "has-image-file-type" and "has-image-color-space". According to the different values of the slot "has-image-file-type", appearing in the candidate components, "BMP", "TIFF" and "JPEG", the collection can be divided into three sub-groups with the numbers, 30, 30 and 40 respectively. According to the different values on "has-image-color-space", "RGB" and "XYZ", the collection can be divided into two sub-groups with the numbers, 30 and 70 respectively. In this case, the information gains of these two slots are calculated using the above formulae:

$$InformationGain(S_{has-image-file-type}) = 1.5711 \quad InformationGain(S_{has-image-color-spacee}) = 0.8814$$

So the question based on the slot "has-image-file-type" is more informative than that of "has-image-color-space". The question, "what type of images do you want to deal with in this component?" is displayed to users with three possible answers, "BMP", "TIFF", or "JPEG".

4. RELATED RESEARCH AND DISCUSSION

Software is used to resolve practical problems, and software components are existing solutions to previous problems, so component reuse can be described as "trying to use the solutions to previous similar problems to help solve the current problem". Therefore, it is very natural to use CBR methods to support component reuse. In fact, various types of CBR methods have been explored and found useful for component reuse.

Object Reuse Assistant (ORA)[2] is a hybrid framework to use CBR to locate appropriate components in an object-oriented software library (small-talk component library). In this framework, both small-talk classes and small-talk methods take the form of stored cases. The concepts in small-talk, for instance, c-class, c-method and c-data-spec, and their instantiated objects are connected together as a conceptual hierarchy. Though the conceptual hierarchy can be seen as a representation method combining case-specific knowledge and general knowledge, the retrieval process is knowledge-poor (a new case is compared with stored cases based on how many attributes two cases have in common).

IBROW[22] is an automated software application configuration project. Users' tasks (queries) can be decomposed into sub-tasks by matched task decomposers, and sub-tasks can be decomposed further. Tasks or subtasks can finally be solved by matched stored components. Both task decomposers and components are referred to as PSMs (problem solving methods). The output is an application configuration composed of stored components, which satisfies users' query. CBR is used at two levels in IBROW. The high level is called constructive adaptation. In this level, PSMs take the form of cases, which are represented using feature terms, and a knowledge-poor matching method (term subsumption) is adopted when searching the possibly applied

PSMs. At the low level, CBR is used as a heuristic algorithm to realize the best-first searching strategy. Previously solved configurations are stored as cases, and represented as feature terms. For each intermediate state, the newly added PSM is considered. The stored configurations in which the same PSM appears as a part are identified, and the similarities between each of these configurations and the new problem are calculated. The most similar configuration is selected, and its similarity value is taken as the heuristic value to the involved intermediate state. As the ORA system, IBROW uses a knowledge-poor retrieval process and only supports tentative and manual interactions between users and the system.

Compared with these two CBR-based component retrieval systems, our proposed conversational component retrieval model (CCRM) has two advantages:

The first is that components are selected based on both their syntactical similarities and semantic similarities. Selecting components based on their semantic similarities with users' query rather than only on syntactical similarities is a promising research topic. However, the existing research concerned with this topic mainly use domain knowledge to refine users' queries before the searching process[5, 17, 23]. In CCRM, besides the query refinement using general thesaurus and domain-ontology, a special type of knowledge-intensive CBR method, explanation-driven CBR, is adopted to explore components' context-based semantic similarities with a query during the retrieval process.

The second is that users' requirements are acquired interactively and incrementally. Normally, component users prefer to provide their initial query only based on their necessary requirements in order to avoid excluding possibly appropriate components. Because of the looseness of the initial query and the large number of available components, users usually still get numerous candidate components. In CCRM, instead of letting users guess and try what requirements they should specify further, an information gain algorithm is used to provide users with the most discriminative questions to refine their query interactively and incrementally.

A limitation of our method is its dependence on knowledge engineering. The knowledge base combining both component specific cases and general domain knowledge is assumed to exist initially. The construction of this initial knowledge base puts a significant workload on the knowledge engineering process.

5. FUTURE WORK

The evaluation of CCRM is in process. The knowledge-intensive similarity measuring process has been realized in the CREEK system, and the conversational process is being added. We are building a knowledge base for the components existing in the DynamicImager system, a visual and dynamic image processing experimentation environment, in which there are about 200 different image operating components.

Our current research focus is to use the knowledge-intensive method to facilitate the discriminative question selection. Though the information gain algorithm can select the most discriminative question automatically and incrementally, it is knowledge-poor essentially. We plan to use knowledge-intensive methods, especially the explanation-driven method, to remove the candidate slots (questions) whose values can be inferred from users' initial query or previously answered questions, and to adjust the priorities between slots which represent semantic relations, such as, abstraction, causality, dependency and part-of relations. The hypothesis is that this will help to identify the most informative question, shorten dialog length, and reduce users' cognitive workload.

References

1. Mili, A., R. Mili, and R.T. Mittermeir, (1998), A survey of software reuse libraries. Annals of Software Engineering, 5(0): p. 349 - 414.
2. Fernández-Chamizo, C., et al. (1996), Supporting Object Reuse through Case-Based Reasoning. European Workshop on Case-Based Reasoning. Lausanne, Switzerland.
3. Iribarne, L., J.M. Troya, and A. Vallecillo. (2002), Selecting software components with multiple interfaces. 28th Euromicro Conference.
4. Ravichandran, T. and M.A. Rothenberger, (2003), Software reuse strategies and component markets. Communications of the ACM, 46(8): p. 109 - 114.
5. Klein, M. and A. Bernstein. (2001), Searching for Services on the Semantic Web Using Process Ontologies. The First Semantic Web Working Symposium. Stanford, CA, USA.
6. Aamodt, A. and E. Plaza, (1994), Case-Based Reasoning: Foundational Issue, Methodological Variations, and System Approaches. AI Communications, 7(1): p. 39-59.
7. Aamodt, A. (1994), Explanation-driven case-based reasoning. Topics in Case-based reasoning: Springer Verlag.
8. Aha, D.W. and L.A. Breslow, (2001), Conversational Case-Based Reasoning. Applied Intelligence, 14(1): p. 9-32.
9. Muňoz-Avila, H., et al., (1999), HICAP: An Interactive Case-Based Planning Architecture and its Application to Noncombatant Evacuation Operations.
10. Shimazu, H., (2002), ExpertClerk: A Conversational Case-Based Reasoning Tool for Developing Salesclerk Agents in E-Commerce Webshops. Artificial Intelligence Review, 18(3-4): p. 223 - 244.
11. Frakes, W.B. and B.A. Nejmeh, (1987), Software reuse through information retrieval. ACM SIGIR Forum, 21(1-2): p. 30-36.
12. Helm, R. and Y.S. Maarek. (1991), Integrating information retrieval and domain specific approaches for browsing and retrieval in object-oriented class libraries. Conference on Object Oriented Programming Systems Languages and Applications. Phoenix, Arizona, United States.
13. Magnini, B., (1999), Use of a lexical knowledge base for information access systems. Journal of Theoretical & Applied Issues in Specialized Communication, 5(2): p. 203 - 228.
14. Prieto-Daz, R., (1991), Implementing faceted classification for software reuse. Communications of the ACM, 34(5): p. 89 - 97.
15. Zaremski, A.M. and J.M. Wing, (1995), Signature matching: a tool for using software libraries. ACM Transactions on Software Engineering Methodology, 4(2): p. 146 -170.
16. Park, Y., (2000), Software retrieval by samples using concept analysis. Systems & Software, 54(3): p. 179-183.
17. Sugumaran, V. and V.C. Storey, (2003), A Semantic-Based Approach to Component Retrieval. The DATA BASE for Advances in Information Systems, 34(3): p. 8-24.
18. Vitharana, P., F.M. Zahedi, and H. Jain, (2003), Design, retrieval, and assembly in component-based software development. Communications of the ACM, 46(11): p. 97-102.
19. Redondo, R.P.D., et al. (2002), Approximate Retrieval of Incomplete and Formal Specifications Applied to Vertical Reuse. International Conference on Software Maintenance. Montreal, Quebec, Canada.
20. Quanlan, J.R., (1986), Induction of decision trees. Machine Learning, 1(1): p. 81-106.

21. Aamodt, A. (1994), A Knowledge Representation System for Integration of General and Case-Specific Knowledge. International Conference on Tools with Artificial Intelligence. New Orleans.
22. IBROW, (2004), Website. http://www.swi.psy.uva.nl/projects/ibrow/home.html.
23. Bernstein, A. and M. Klein. (2002), Towards High-Precision Service Retrieval. International Semantic Web Conference. Sardinia Italy.

Paper C

Comparing Similarity Calculation Methods in Conversational CBR

```
@INPROCEEDINGS{Gu2005b,
  author = {Mingyang Gu and Xin Tong and Agnar Aamodt},
  title = {Comparing Similarity Calculation Methods
          in Conversational CBR},
  booktitle = {Proceedings of the 2005 IEEE International
          Conference on Information Reuse and Integration},
  year = {2005},
  editor = {Du Zhang and Taghi M. Khoshgoftaar
          and Mei-Ling Shyu},
  pages = {427 - 432},
  address = {Hilton, Las Vegas, Nevada, USA},
  month = {August},
  publisher = {IEEE Press},
}
```

Comparing Similarity Calculation Methods in Conversational CBR

Mingyang Gu, Xin Tong, and Agnar Aamodt

Department of Computer and Information Science, Norwegian University of Science and Technology, Sem Saelands vei 7-9, N-7491, Trondheim, Norway
Email: {mingyang, tongxin, agnar}@idi.ntnu.no

Abstract. Conversational Case-Based-Reasoning (CCBR) provides a mixed-initiative dialog for guiding users to construct their problem description incrementally through a question-answering sequence. Similarity calculation in CCBR, as in traditional CBR, plays an important role in the retrieval process since it decides the quality of the retrieved case. In this paper, we analyze the different characteristics of the query (new case) between CCBR and traditional CBR, and argue that the similarity calculation method that only takes the features appearing in the query into account, so called query-biased, is more suitable for CCBR. An experiment is designed and executed on 36 datasets. The results show us that on 31 datasets out of the total 36, the CCBR system using the query-biased similarity calculation method achieves more effective performance than those using case-biased and equally-biased similarity calculation methods.

1 Introduction

The basic idea underlying case-based reasoning (CBR) [1, 2] is to reuse the solution to the previous most similar problem in helping solve the current problem. Before we can reuse any existing solutions, we have to find the most similar previous case based on the current problem description.

In traditional CBR processes, users are assumed to be able to provide a well-defined problem description, and based on such a description a CBR system can find the most appropriate previous case (base case). But this assumption is not always realistic. In some situations, users only have vague ideas about their target problems at the beginning of retrieval, and tend to describe them using surface features.

Conversational Case-Based Reasoning (CCBR) [3] provides a mixed-initiative dialog for guiding users to construct their problem description incrementally through a question-answering sequence. In CCBR, a user provides one or several explicit features as her initial query (new case). The CCBR system uses the initial query to retrieve the first set of candidate cases, and identifies a group of informative features from them to generate discriminative questions. Both the retrieved cases and identified discriminative questions are ranked and shown to the user. The user either finds the base case to terminate the retrieval process or chooses a question, which she considers relevant to her task and can answer explicitly, and provides the answer to it (CCBR systems usually also prompt the alternative answer options that correspond to the feature values available in the case base). An updated query is constructed through combining the previous query with the newly gained answer. Subsequent rounds of retrieving and question-answering will cut down the returned case set iteratively until the user finds her desired base case, or no discriminative questions are available. That is, instead of letting a user guess how to describe her target problem, CCBR discovers a sequence of discriminative questions helping extract information from the user to construct the problem description incrementally. CCBR applications have been successfully fielded, e.g., in the troubleshooting domain [4, 5] and in the products and services selection [6, 7].

In both traditional CBR and CCBR, one key research topic is to calculate the similarities between a query and stored cases to decide which case is most similar to the current problem. Normally, the similarity between a query and a stored case is measured by the accumulated similarities on all counted features. On the one hand, the similarity is influenced by different methods to calculate the similarity on each feature. For example, in syntactic methods two cases can be thought similar on one nominal feature only when they have the same value on that feature [8], while in knowledge-intensive methods, two cases with various values on one nominal feature can possibly be considered as similar through exploring general domain knowledge [9, 10]. On the other hand, the similarity is also influenced by the counted feature scope, i.e. set of the features appearing in the query, in the case, or in both of them. In this paper, from the perspective of counted feature scope, we provide a framework to classify the similarity calculation methods into three categories: case-biased (features in the stored case), query-biased (features in the query) and equally-biased (features in both the query and the stored case).

CCBR research is currently to a large extent focusing on the discriminative question selecting and ranking to minimize the cognitive load demanded on users to retrieve the base case [6, 11], for example, selecting the most informative questions to ask [6, 12–15], or using feature inferencing to avoid asking users the questions which can be answered implicitly using the currently known information [12, 15]. To our knowledge, there are so far no published results on how different similarity calculation methods influence on the performance of a CCBR system.

In this paper, we analyze the differences on query characteristics between traditional CBR and CCBR, and hypothesize that the similarity calculation method only taking the query features into account is more suitable for CCBR. An experiment is designed and executed in an attempt to evaluate this hypothesis.

The rest of this paper is organized as follows. In Section 2, we present a formal framework to classify the similarity calculation methods in CBR into three categories from the feature scope point of view; in Section 3, we focus on the query differences between CCBR and traditional CBR, and hypothesize that query-biased similarity calculation method is more suitable for CCBR; in Section 4, we design an experiment to evaluate this hypothesis, and the results are also analyzed and discussed. At the end we draw our conclusions in Section 5.

2 Similarity Calculation Framework in CBR

Generally, a case in CBR can be represented using the following three parts conceptually [2]:

Problem description: the state of the world at the time the case was happening and, if appropriate, what problem needed to be solved at that time

Solution description: the stated or derived solution to the problem specified in the problem description

Outcome: the resulting state of the world after the solution was carried out

A query (new case) only has the first part. The similarity measurement between a query and a stored case is based on the comparison of the problem description part of them. In our research, we assume that the problem description of a case takes the form of a set of $< feature, value >$ pairs. It is not necessary for both a query and a case to have the same feature set.

We further define that:

Nq: set of features appearing in a query

Nc: set of features appearing in a stored case

One concept that is closely related to similarity is distance. The greater the distance between a query and a stored case, the less the similarity between them is. The main use of the similarity measurement in CBR is to sort the retrieved cases. From that point of view, the similarity and distance measurements have an inverse relationship, and either of them may be chosen. We adopt the distance measurement in our research, as defined by the following formula:

$$distance(q, c) = \sqrt{\frac{\sum_{f \in FS} w_f dif^2(q_f, c_f)}{\sum_{f \in FS} w_f}} \tag{1}$$

where q, c, f, FS and w_f denote a query, a stored case, a particular feature, a selected feature set and the weight for the feature f respectively.

In addition, $dif(q_f, d_f)$ is a function used to compute the difference between a query and a stored case on a feature f, which is defined as following in our research:

$$dif(q_f, c_f) = \begin{cases} |q_f - c_f| & f \text{ is a numerical feature} \\ & \quad \text{(normalized)} \\ 0 & f \text{ is a nominal feature,} \\ & \quad \text{and } q_f = c_f \\ 1 & f \text{ is a nominal feature,} \\ & \quad \text{and } q_f \neq c_f \\ 1 & \text{c or q has missing value on } f \end{cases} \tag{2}$$

Based on three types of value assignment methods to FS in Equation 1, we divide the similarity measurement methods in CBR into three categories.

2.1 Case-Biased Similarity Calculation Methods

In case-biased similarity calculation methods, $FS = N_c$, and Equation 1 is transformed as follows:

$$distance(q, c) = \sqrt{\frac{\sum_{f \in N_c} w_f dif^2(q_f, c_f)}{\sum_{f \in N_c} w_f}} \tag{3}$$

In this type of calculation method, the features appearing in the stored case are the basis for the similarity calculation (here comes the name of 'case-biased'). This type of methods are used in [3, 8]. The basic idea behind it is that the problem description of a stored case is a sufficient condition for the corresponding solution actions, so to what degree the problem description is satisfied by the query decides whether the solution in the stored case is suitable for the current problem.

2.2 Query-Biased Similarity Calculation Methods

In query-biased similarity calculation methods, $FS = N_q$, only the features appearing in the query are taken into account. This type of similarity calculation method focuses on the query, and the intuitive idea underlying it is that whether a stored case can be retrieved is decided by to what degree the query specified by a user is satisfied by this stored case.

2.3 Equally-Biased Similarity Calculation Methods

In equally-biased similarity calculation methods, $FS = N_q \cup N_c$, that is, both the features appearing in the query and those in the stored case are taken into account (the case and the query are treated equally). This type of similarity calculation methods are used in [16, 17]. The basic idea behind such type of methods is that the degree to which the query and the case are similar decides whether the solution to that case can be reused in the current target problem.

3 Using Query-Biased Similarity Calculation Methods in CCBR

CCBR considers the situation where users can not well define their queries, and alternatively provides a multi-retrieval process to help users construct their queries incrementally through a sequence of question-answering cycles. The important difference between CCBR and traditional CBR is that *the query used in CCBR is assumed incomplete, that is, the CCBR query only represents the user's currently identified features.*

Table 1. A fruit retrieval example in CCBR

	Taste	Color	Place	With Fur	With Core	Water Inside	Shape
Query	sweaty	red	Asia		yes		
apple	sweaty	red	Asia	yes	yes	no	round
kiwi	sweaty	brown	America				
banana	sweaty				no		

The features without specified values in a CCBR query do not necessarily mean that they have "missing-value" as in traditional CBR. They may have values, but we have not assigned the value for them in current CCBR stage. The case-biased and equally-biased similarity calculation methods assume that the query is fully specified and all the features that appear in the case but not in the query are considered to have the "missing value". So the difference measurement on each such feature is assigned the same value, e.g. 1. If we proceed further in CCBR, and specify values on more features, the really calculated distance measurement will benefit the base case since it has the higher potential to get less difference measurements on these newly specified features than other cases. The distance between the partially specified query and a stored case is heavily influenced by the number of the features appearing in the case but not yet specified in the query. The query-biased similarity calculation method can avoid the influence of these features, and rank the case that most satisfies the currently partially specified query with the highest priority.

For example, as illustrated by Table 1, the potential fully specified query for searching a desired fruit has four features. The distance measurements using three different similarity calculation methods are shown in 2, in which we assume that each feature has the same weight ($\frac{1}{7}$). We can see that the most similar case with the fully specified query is apple, no matter which similarity calculation method is adopted.

Table 2. Distance measurements with four features in the example

	apple	kiwi	banana
Query-biased	0	$\frac{3}{4}$	$\frac{3}{4}$
Case-biased	$\frac{3}{7}$	$\frac{2}{3}$	$\frac{1}{2}$
Equally-biased	$\frac{3}{7}$	$\frac{3}{4}$	$\frac{3}{4}$

At one stage of the conversation in CCBR a user may only specify two features, "Taste=sweaty, Color=red". The distance measurements between the query and each fruit are shown in Table 3. If query-biased method is adopted, the most similar case to the query has already been the base case, i.e. apple.

Table 3. Distance measurements with two features in the example

	apple	kiwi	banana
Query-biased	0	$\frac{1}{2}$	$\frac{1}{2}$
Case-biased	$\frac{5}{7}$	$\frac{2}{3}$	$\frac{1}{2}$
Equally-biased	$\frac{5}{7}$	$\frac{2}{3}$	$\frac{2}{3}$

The most similar case will be kiwi or banana if equally-biased method is used, and banana if case-biased method is adopted. With the conversation going on, the system further prompts two questions "where does the fruit come from?" and "does the fruit have a core inside?", and the user answers these questions with "Place=Asia, With Core=yes". Until this stage, the base case, i.e. apple, can be ranked with the highest priority (as shown in Table 2). But the user may be angry at her 'tricky searching assistant', "since only apple satisfies the first two specified features, why do dot you show me at that time and still bother me to answer extra two questions?", and her satisfaction level will be reduced.

We can see from the above example that the query-biased similarity calculation method can avoid the influence of the features that appear in the case but not in the partially specified query in CCBR. Since the base case also has a higher potential to have higher similarity on partial set of query features than other cases, it is reasonable to believe that a CCBR system that uses the query-biased similarity calculation method can show users the base case on earlier conversation stage than those using case-biased or equally-biased methods. So our hypothesis tested in this paper is that using the query-biased similarity calculation method, a CCBR system can improve its performance, that is, using less conversation sessions to find the base case than using equally-biased or case-biased methods.

4 Experiment Design and Results Analysis

Our experiment is designed with the objective to compare the conversation lengths of CCBR systems using different similarity calculation methods. The best way to do that is with human subjects. Unfortunately, we can not get sufficient subjects to run the experiment. Therefore, we use a variant of the leave-one-out cross validation (LOOCV) method to simulate the human-computer conversation process, the similar methods to which have been successfully used by the CCBR community [12]. The designed evaluation process is carried out on 36 datasets, and the results provide a significant support to our hypothesis.

4.1 Experiment Design

The LOOCV proceeds with a series of conversations, each conversation starting with selecting a case from the case base as the target case and the remaining cases forming the case base to be searched. The initial query is constructed through selecting the predefined number of features from that target case. Based on this initial query, a retrieval process is carried out and the first k most similar cases are returned. If the base case is included in the returned case set, which means users find their desired case, the conversation process is finished successfully. Otherwise, a new feature is selected from the target case and added into the query to simulate a question-answering session between a human subject and a computer, and the updated query is used to start a new round of retrieval. The selecting, adding, and retrieving cycle continues until the base case appears in the returned case set, or no features are left to be selected.

There are three questions we further need to address in the experiment design: the retrieval algorithm, the feature selection strategy, and the base case determination.

Retrieval Algorithm A weighted k-NN algorithm is introduced in our experiment to complete the case retrieval task, in which the first k most similar cases are returned. The number k is used to simulate the number of cases that will be shown to users on each conversation session in CCBR. In our experiment setting, we set k to 3. We use a feature weighting method, similar to EACH [18], to get a set of global weights, each corresponding to one feature appearing in the case base.

In EACH, given a test case from the case base, its most similar case is selected from the remaining stored cases using a weighted 1-NN algorithm. If the most similar case suggests the same solution as the test case, the weight of each matched feature is increased by a fixed positive amount, while weights for mismatched features are decreased by the same amount.

Three variants of this basic algorithm are constructed based on the three different similarity calculation methods introduced in Section 2.

Feature Selection Strategy The feature selection strategy is used to decide which feature should be selected from a set of candidate features, and added into the current query to simulate a question-answering process. In our experiment, a weight-biased random selection strategy is designed. For example, there are three features, A, B, and C in the candidate feature set with the weight values, 0.1, 0.2, and 0.05, respectively (learned from the feature weighting process). According to the weight-biased random feature selection strategy, feature A, B, C will be selected with the possibilities $\frac{2}{7}$, $\frac{4}{7}$, and $\frac{1}{7}$, respectively. Such a feature selection strategy simulates a question-answering process: a CCBR system ranks the more informative questions (transformed from features) with higher priority, and a user prefers to select the most relevant or important feature to answer first.

Base Case Determination For each case in the case base, its base case is defined as the one returned by a weighted 1-NN algorithm using equally-biased similarity calculation (here the query is fully specified and complete) and with the same solution-feature value. Therefore, not all the cases in the case base can act as a target case to simulate a conversation. The cases that can not find its corresponding base case from the remaining cases (its nearest neighbor has a different solution-feature value) will be dropped out from the leave-one-out cross validation.

Since we choose the base case as the retrieval result of the 1-NN algorithm using the equally-biased similarity method, the simulated conversation process using the equally-biased method will terminate with the base case appearing among the returned case set within all the candidate features are added into the query. It is not guarantied that the conversation process using the case-biased or the query-biased method can terminate with the base case found in the returned case set.

In our experiment, we assign the biggest conversation length (the number of conversation sessions when all candidate features are added into the query) to the unsuccessful conversations. That is, the base case selection mechanism benefits the CCBR system using the equally-biased similarity calculation method. However, the experiment results show us that even with such biased base case selection strategy, the average conversation length using the query-biased similarity method is shorter than that using other two methods, and the average conversation length using the case-biased method is almost the same with that using equally-biased method, as illustrated in Table 5 and Table 6.

The pseudo code for the experiment process is listed in Table 4.

Experiment Environment and Dataset

We implement our evaluation algorithm inside the Weka framework [19], and test it using all the 36 datasets provided by Weka project, originally from the UCI repository [20].

All the numeric features in these datasets are normalized using the corresponding filter provided in Weka3.4.3 according to the requirement of the similarity calculation algorithm. The statistical information about the test datasets is illustrated in the left part of Table 5 and Table 6, in which the first 6 columns denote respectively: the name of each dataset (Dataset), the number of the cases (Total cases), the total number of the features excluding the solution feature (Features), the number of the numeric features and nominal features (Numeric/Nominal), the percentage of the missing data (Missing Data) calculated using equation: $\frac{number\ of\ the\ missing\ values}{Total\ cases*Features}$, and the number of solutions (Solutions).

Table 4. Pseudo Code of the Evaluation Process.

```
Procedure evaluation(CaseBase)
    SuccessOnEqually,SuccessOnQuery,SuccessOnCase=false
    TestCases,SessionsOnEqually,SessionsOnCase,SessionsOnQuery=0
    GlobalWeights=weighting(CaseBase)
    for each case X ∈ CaseBase
        X_n=weighted1NNOnEqually(X,CaseBase-X)
        if Solution(X_n) = Solution(X) then
            TestCases=TestCases+1
            X_q=featureSelection(InitialFeatureNumber)
            do while not (SuccessOnEqually and SuccessOnCase
                and SuccessOnQuery) and X_q ≠ null
                if not SuccessOnEqually then
                    ReturnedCasesOnEqually=
                      weightedkNNOnEqually(X_q, CaseBase-X)
                    SessionsOnEqually=SessionsOnEqually+1
                    if X_n ∈ ReturnedCasesOnEqually then
                        SuccessOnEqually=true
                    End If
                End If
                if not SuccessOnCase then
                    ReturnedCasesOnCase=
                      weightedkNNOnCase(X_q, CaseBase-X)
                    SessionsOnCase=SessionsOnCase+1
                    if X_n ∈ ReturnedCasesOnCase then
                        SuccessOnCase=true
                    End If
                End If
                if not SuccessOnQuery then
                    ReturnedCasesOnQuery=
                      weightedkNNOnQuery(X_q, CaseBase-X)
                    SessionsOnQuery=SessionsOnQuery+1
                    if X_n ∈ ReturnedCasesOnQuery then
                        SuccessOnQuery=true
                    End If
                End If
                X_q=X_q+featureSelection(1)
            End Loop
        End IF
    End Loop
Return SessionsOnEqually/TestCases, SessionsOnCase/TestCases, SessionsOnQuery/TestCases
```

Table 5. Dataset Description and Experiment Results (first part)

Dataset	Total cases	Features	Numeric/ Nominal	Missing data	Solut -ions	Test cases	Equally Biased	Case Biased	Query Biased	Case- Query	Equally -Query	Case- Equally
Anneal	898	38	6 / 32	64.98%	5	891	20.76	20.76	20.55	0.21	0.21	0
Anneal Original	898	38	9 / 29	73.13%	6	824	8.36	8.37	7.26	1.10	1.09	0.01
Audiology	226	69	0 / 69	2.03%	24	180	14.01	13.58	17.43	**-3.85**	**-3.42**	-0.43
Autos	205	25	15 / 10	1.15%	7	150	7.13	7.12	2.63	4.49	4.49	-0.01
Balance Scale	625	4	4 / 0	0	3	478	3.58	3.58	3.75	**-0.17**	**-0.17**	0
Breast Cancer	286	9	0 / 9	0.35%	2	190	5.99	6.00	5.88	0.12	0.11	0.01
Breast-W	699	9	9 / 0	0.25%	2	665	8.70	8.70	4.82	3.88	3.88	0
Credit Approval	690	15	6 / 9	0.65%	2	549	11.96	11.98	8.86	3.11	3.10	0.01
Credit German	1000	20	7 / 13	0	2	682	13.11	13.11	11.32	1.79	1.79	0
Diabetes	768	8	8 / 0	0	2	554	7.87	7.87	5.15	2.72	2.72	0
Glass	214	9	9 / 0	0	7	150	8.37	8.37	3.41	4.96	4.96	0
Heart Statlog	270	13	13 / 0	0	2	204	9.51	9.51	6.74	2.77	2.77	0
Heart h	294	13	6 / 7	20.46%	5	220	7.43	7.44	4.92	2.52	2.51	0.01
Heart c	303	13	6 / 7	0.18%	5	233	7.60	7.60	6.04	1.56	1.56	0
Hepatitis	155	19	6 / 13	2.43%	2	125	10.14	10.74	9.06	1.69	1.08	0.61
Horse Colic	368	22	7 / 15	23.80%	2	290	11.77	13.29	9.53	3.76	2.24	1.52
Horse Colic Original	368	27	7 / 20	19.39%	2	258	13.26	13.41	10.61	2.80	2.65	0.15
Hypothyroid	3772	29	7 / 22	5.54%	4	3335	15.22	15.22	14.74	0.48	0.48	0
Ionosphere	351	34	34 / 0	0	2	312	24.95	24.95	7.65	17.29	17.29	0
Iris	150	4	4 / 0	0	3	143	3.75	3.75	2.14	1.61	1.61	0
Kr-vs-kp	3196	36	0 / 36	0	2	3146	21.86	21.86	20.63	1.23	1.23	0
Labor	57	16	8 / 8	35.75%	2	52	5.88	6.60	3.06	3.54	2.83	0.71
Letter	20000	16	16 / 0	0	8	17380	14.09	14.09	9.26	4.84	4.84	0
Lymph	148	18	3 / 15	0	4	123	9.76	9.76	9.82	**-0.07**	**-0.07**	0

Table 6. Dataset Description and Experiment Results (second part)

Dataset	Total cases	Features	Numeric/ Nominal	Missing data	Solut -ions	Test cases	Equally Biased	Case Biased	Query Biased	Case- Query	Equally -Query	Case- Equally
Mushroom	8124	22	0 / 22	1.39%	2	8124	16.07	16.07	13.09	2.97	2.97	0
Primary Tumor	339	17	0 / 17	3.90%	8	125	5.94	5.90	7.22	**-1.31**	**-1.28**	-0.03
Segment	2310	19	19 / 0	0	7	2217	17.55	17.55	7.45	10.10	10.10	0
Sick	3772	29	7 / 22	5.54%	2	3421	14.66	14.67	14.75	**-0.08**	**-0.09**	0.01
Sonar	208	60	60 / 0	0	2	182	39.68	39.68	12.87	26.81	26.81	0
Soybean	683	35	0 / 35	9.78%	19	623	9.44	13.98	6.56	7.42	2.88	4.54
Splice	3190	61	0 / 61	0	3	2877	29.24	29.24	23.59	5.65	5.65	0
Vehicle	846	18	18 / 0	0	4	602	15.93	15.93	6.71	9.22	9.22	0
Vote	435	16	0 / 16	5.63%	2	412	10.15	11.31	8.95	2.36	1.20	1.16
Vowel	990	13	10 / 3	0	11	983	8.58	8.58	3.57	5.01	5.01	0
Wave Form 5000	5000	40	40 / 0	0	3	3157	12.91	12.91	11.98	0.93	0.93	0
Zoo	101	17	1 / 16	0	7	98	7.98	7.98	7.74	0.23	0.23	0
AVERAGE							12.59	12.82	9.16	3.66	3.43	0.23

Table 7. Hypothesis Test Result

Alternative hypothesis	Null hypothesis	Tailed type	Degree of freedom	Significance level	Critical value	t-value	Result
$H1 : V_{Equally-Query} > 0$	$V_{Equally-Query} = 0$	one-tailed	35	0.01	2.44	3.81	refuse null-hypothesis
$H2 : V_{Case-Query} > 0$	$V_{Case-Query} = 0$	one-tailed	35	0.01	2.44	4.05	refuse null-hypothesis
$H3 : V_{Case-Equally} \neq 0$	$V_{Case-Equally} = 0$	two-tailed	35	0.01	2.727	1.68	can not refuse null-hypothesis

Experiment Results

The experiment results are listed in the right part of Table 5 and Table 6, in which the columns: Test cases, Equally Biased, Case Biased, and Query Biased, denote tested cases (corresponding to TestCases in Table 4), the average conversation lengths using the corresponding similarity calculation method for each dataset.

To show the comparison results more clearly, we add three columns into Table 5 and Table 6, Case-Query, Equally-Query, and Case-Equally, to illustrate the differences of the average conversation lengths between each pair of similarity calculation methods. For example, the Case-Query column contains the results of subtracting the average conversation length using the query-biased method from that using the case-biased method on each dataset. And the last row gives the average values of corresponding columns.

Out of 36 datasets, there are 31 datasets in which the query-biased similarity calculation method uses less conversation sessions to find the base case than other two methods (average using 3.66, 3.43 less conversation sessions respectively). That gives us a straightforward evidence that the CCBR system using query-biased method is more effective than those using equally-biased and case-biased methods.

The conversation lengths between CCBR systems using the case-biased method and the equally-biased method do not have clear difference since there are no difference at all on 22 datasets out of 36, and the average difference over 36 datasets is only 0.23. Even if the results show us that the equally-biased method is a little more effective than the case-biased one, but considering that the base case determination strategy benefits the equally-biased similarity calculation method, the experiment results can not provide strong evidence to say that there is performance difference between these two methods.

Further more, we carry out the statistical hypothesis test to evaluate our predefined hypothesis in Section 3. The whole hypothesis is divided into three sub hypotheses to test:

H1: the CCBR system using the query-biased similarity calculation method can achieve more effective performance than that using the equally-biased method, that is, using less conversation sessions to find the base case.

H2: the CCBR system using the query-biased similarity calculation method can use less conversation sessions to find the base case than that using the case-biased method.

H3: there exists performance difference between the CCBR system using the case-biased similarity calculation method and that using the equally-biased method, that is, these two methods use different number of conversation sessions to find the base case.

We choose the values appearing in the column: Equally-Query, Case-Query, and Case-Equally respectively in Table 5 and Table 6, as the parameter values to execute the significance test. The test results (reported in Table 7) show us that the first two sub hypotheses are accepted, and the last one is refused given the significance level of 0.01.

Conclusions

In this paper, we provide a framework to classify the similarity calculation methods used in CBR from the perspective of counted feature scope. And based on the special characteristic of the CCBR query, partially specified and incomplete, we hypothesize that CCBR system using the query-biased similarity calculation method can achieve higher performance than those using case-biased or equally-biased methods. The experiment provides a significant support to our hypothesis. While the conversation process in the experiment is simulated by a leave-one-out cross validation process, an experiment executed on human subjects will provide more evidence to evaluate our hypothesis.

References

1. A. Aamodt and E. Plaza, "Case-based reasoning: Foundational issue, methodological variations, and system approaches," *AI Communications*, vol. 7, no. 1, pp. 39–59, 1994.
2. J. Kolodner, *Case-based reasoning*. Morgan Kaufmann Publishers Inc., 1993.
3. D. W. Aha, L. A. Breslow, and H. Munoz-Avila, "Conversational case-based reasoning," *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, vol. 14, no. 1, p. 9, 2001.
4. K. M. Gupta, "Knowledge-based system for troubleshooting complex equipment," *International Journal of Information and Computing Science*, vol. 1, no. 1, pp. 29–41, 1998.
5. P. Cunningham and B. Smyth, "A comparison of model-based and incremental case-based approaches to electronic fault diagnosis," in *Case-Based Reasoning Workshop*, Seattle, USA, 1994.
6. H. Shimazu, "Expertclerk: A conversational case-based reasoning tool for developing salesclerk agents in e-commerce webshops," *Artificial Intelligence Review*, vol. 18, no. 3-4, pp. 223 – 244, 2002.
7. M. Gu, A. Aamodt, and X. Tong, "Component retrieval using conversational case-based reasoning," in *International Conference on Intelligent Information Systems*, Z. Shi, Ed., Beijing, China, 2004.
8. D. W. Aha, "Case-based learning algorithms," in *the DARPA Case-Based Reasoning Workshop*, Morgan Kaufmann, 1991.
9. A. Aamodt, "Explanation-driven case-based reasoning," *Topics in Case-based reasoning*, pp. 274–288, 1994.
10. A. Aamodt, "Knowledge-intensive case-based reasoning in creek," in *7th European Conference on Case-Based Reasoning*, P. Funk and P. A. G. Calero, Eds., vol. 1 - 15. Madrid, Spain: Spinger, 2004.
11. S. Schmitt, "simvar: A similarity-influenced question selection criterion for e-sales dialogs," *Artificial Intelligence Review*, vol. 18, no. 3-4, pp. 195–221, 2002.
12. D. W. Aha, T. Maney, and L. Breslow, "Supporting dialogue inferencing in conversational case-based reasoning," in *European Workshop on Case-Based Reasoning*, Dublin, Ireland, 1998, pp. 262–273.
13. P. Cunningham, R. Bergmann, S. Schmitt, R. Traphoner, S. Breen, and B. Smyth, "Websell: Intelligent sales assistants for the world wide web," *KI - Kunstliche Intelligenz*, vol. 1, pp. 28–31, 2001.

14. M. H. Goker and C. A. Thompson, "Personalized conversational case-based recommendation," in *the 5th European Workshop on Case-Based Reasoning(EWCBR 2000)*, Trento, Italy, 2000.
15. M. Gu and A. Aamodt, "A knowledge-intensive method for conversational cbr," in *International Conference on Case-Based Reasoning*, Chicago Illinois, 2005.
16. M. M. Richter and S. Web, "Similarity, uncertainty and case-based reasoning in padtex," Kaiserslautern, Tech. Rep., 1993.
17. Q. Yang and J. Wu, "Enhancing the effectiveness of interactive case-based reasoning with clustering and decision forests," *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, vol. 12, pp. 49 – 64, 2001.
18. S. Salzberg, "A nearest hyperrectangle learning method," *Mach. Learn.*, vol. 6, no. 3, pp. 251–276, 1991.
19. I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools with Java implementations.* San Francisco: Morgan Kaufmann, 2000.
20. C. Blake and C. Merz, "Uci repository of machine learning databases [http://www.ics.uci.edu/ mlearn/mlrepository.html]," 1998.

Paper D

A Knowledge-Intensive Method for Conversational CBR

```
@INPROCEEDINGS{Gu2005a,
  author = {Mingyang Gu and Agnar Aamodt},
  title = {A Knowledge-Intensive Method for Conversational CBR},
  booktitle = {Case-Based Reasoning Research and Development,
          Proceedings of the 6th International Conference
          on Case-Based Reasoning},
  year = {2005},
  editor = {H{\'e}ctor Mu{\~n}oz-Avila and Francesco Ricci},
  volume = {3620},
  series = {Lecture Notes in Artificial Intelligence},
  pages = {296-311},
  address = {Chicago, Illinois},
  month = {August},
  publisher = {Springer Verlag},
}
```

A Knowledge-Intensive Method for Conversational CBR

Mingyang Gu and Agnar Aamodt

Department of Computer and Information Science,
Norwegian University of Science and Technology, Sem Sælands vei 7-9,
N-7491, Trondheim, Norway
{Mingyang.Gu, Agnar.Aamodt}@idi.ntnu.no

Abstract. In conversational case-based reasoning (CCBR), a main problem is how to select the most discriminative questions and display them to users in a natural way to alleviate users' cognitive load. This is referred to as the question selection task. Current question selection methods are knowledge-poor, that is, only statistical metrics are taken into account. In this paper, we identify four computational tasks of a conversation process: *feature inferencing*, *question ranking*, *consistent question clustering* and *coherent question sequencing*. We show how general domain knowledge is able to improve these processes. A knowledge representation system suitable for capturing both cases and general knowledge has been extended with meta-level relations for controlling a CCBR process. An "explanation-boosted" reasoning approach, designed to accomplish the knowledge-intensive question selection tasks, is presented. An application of our implemented system is illustrated in the car fault detection domain.

1 Introduction

The basic idea underlying case-based reasoning (CBR) is to reuse the old solution to the previous most similar problem in helping solve the current problem. Before we can reuse any existing solution, we have to find the most similar previous problem, corresponding to the retrieve phase in the standard CBR cycle [5].

In the traditional CBR process, users are assumed to be able to give a well-defined problem description (a new case), and based on such a well-defined description a CBR system can find the most appropriate previous case. But this assumption is not always realistic. Users usually only have vague ideas about their problems when beginning to retrieve cases, and often describe them by surface features, while the previous cases have been described by providers using the essential features. Furthermore, even if users understand what their problems are and what aspects they should describe, they do not know exactly what terms to use to express their problems.

In general, the knowledge gap between case users and case providers is a major cause for the difficulty of case retrieval. Users usually input a problem description by "guessing" the appropriate feature terms, and the system either returns too many matched cases or none. Conversational Case-Based Reasoning (CCBR) [6] has been proposed to bridge this knowledge gap.

H. Muñoz-Avila and F. Ricci (Eds.): ICCBR 2005, LNCS 3620, pp. 296 – 311, 2005.

Conversational CBR provides a mixed-initiative dialog for guiding users to refine their problem descriptions incrementally through a question-answer sequence. In the CCBR process, a user's initial problem description is used to retrieve the first set of candidate cases. Subsequent questions, prompted by the CCBR system, will cut down this case set iteratively until a manageable number of cases remain. That is, instead of letting a user guess how to describe her problem, CCBR discovers a sequence of discriminative questions, which help to extract information from the user, and to construct the problem description automatically and incrementally. CCBR applications have been successfully fielded, e.g., in the troubleshooting domain [11, 16] and in the products and services selection in E-Commerce [23].

A core research concern in conversational CBR is how to minimize the cognitive load demanded on users to retrieve their desired cases [23, 22], which requires to select the most discriminative questions [6, 8, 9] and ask them in a natural way in the conversation process [8, 12].

Up to now, several methods, such as the static decision tree [10], the information gain metric [11, 13, 23], the occurrence frequency metric [6], the information quality metric [9], the similarity variance metric[21], and the attribute-selection strategies [20], have been proposed to support question selection in the conversational CBR process. However, all the methods mentioned above are basically knowledge-poor, that is, they only take statistical information into account. The potential that general domain knowledge has for playing a positive role in the question selection process is little explored. For example, if the answer to question B can be inferred from that of question A, or the answer to question A is easier or cheaper to obtain than that to question B, question A should be prompted to users before question B. Such a knowledge-intensive question selection approach can select and display discriminative questions based on their semantic relations rather than only their statistical metrics.

We have identified four tasks in conversational CBR, for which general domain knowledge has a potential to control and improve the process: feature inferencing, question ranking, question clustering, and question sequencing.

Feature Inferencing (FI). If one feature of a problem can be inferred from the current problem description, this feature can be added to the problem description automatically, instead of posing a question to the user. Users are likely not to trust a communicating partner who asks for information that is easy to infer. General domain knowledge (domain rules or domain models) can be used to infer the features implicit in the problem description.

Question Ranking (QR). In the conversation process, the identified discriminative questions need to be ranked intentionally before displaying them to users. An integrated method should be adopted, which uses not only the superficial statistical metrics of the questions, but also the semantic relations among them. For example, if the answer to question C can be inferred from one of the possible answers to question D, it may be better to ask question D first.

Even though an integrated question ranking module outputs a set of sorted questions, their screen arrangement and questioning sequence should not be decided by such a sorted order alone. The main reason lies in that people always hope to inspect or answer questions in a natural way. They would prefer to see a set of questions that are connected by some semantic relations, grouped together, and to answer them in an uninterrupted sequence. These requirements are captured by the following two tasks:

Consistent Question Clustering (CQC). The arrangement of questions on the screen should be consistent, that is, the questions with some semantic relations among them should be grouped and displayed together, and the order of the questions in each group should be decided intentionally. For example, the questions having dependency relations among them should be grouped and displayed together.

Coherent Question Sequencing (CQS). The questions asked in the sequential question-answer cycles should be as related as possible, that is, the semantic contents of two sequential questions should avoid switching too often. For example, if in the previous question-answer cycle a more general question in an abstraction taxonomy is asked, the downward more specific question should be asked in the succeeding cycle rather than inserting other non-related questions between them.

The suggested knowledge-intensive conversational CBR process is illustrated in Fig. 1. The lines in bold are the modules used to complete the tasks identified above.

```
NewCase := New-Case-Formalize(InitialProblemDescription);
SequentQuestions :=null;          //(CQS)
Repeat:
    NewCase := Feature-Inference(NewCase);  // (FI)
    SortedRetrievedCases := CBR-Retrieve(NewCase);
    DiscriminativeQuestions := Question-Identify(SortedRetrievedCases, NewCase);
    RankedDiscriminativeQuestions := Integrated-Question-Rank(DiscriminativeQuestions);     // (QR)
    RankedDiscriminativeQuestions :=
         Ranked-Questions-Adjust (SequentQuestions, RankedDiscriminativeQuestions);        // (CQS)
    GroupedRankedDiscriminativeQuestions := Question-Group(RankedDiscriminativeQuestions); // (CQC)
    Display(GroupedRankedDiscriminativeQuestions, SortedRetrievedCases);
    If (users find their desired cases or have no question to answer) then
        Return SelectedCases;
    Else
        SelectedQuestionAndAnswer := User-Select-and-Answer-Question();
        SequentQuestions := Sequent-Question-Identify(SelectedQuestionAndAnswer);          // (CQS)
        NewCase :=NewCase-Update(NewCase, SelectedQuestionAndAnswer);
    End If
```

Fig. 1. The knowledge-intensive CCBR process

In this paper we present an explanation-boosted reasoning approach for support of knowledge-intensive question selection. The use of explanation in case-based reasoning is not new, but the meaning of the term differs. In our approach, the explanation part of the process mainly uses general domain knowledge (rather than specific cases), targeted at system internal reasoning (rather than user understanding). However, the explanations constructed can also be displayed to the user for transparency, justification, and increased understanding. What we mean by explanation-boosted reasoning is a particular method for constructing explanation paths that exploit general domain knowledge for the question selection tasks. The method was briefly introduced in an earlier workshop paper [14], in which only two of the four question selection tasks were described. In the presented paper we extend the description to cover more CCBR tasks, we explicitly relate the tasks with meta-level relations for reasoning, and we present the first implemented version of the system.

The rest of this paper is organized as follows. In Section 2, we identify several semantic relations related to question selection. In Section 3, our explanation-boosted question selection method is described from the perspectives of knowledge representation, explanation construction and reasoning method. The system implementation of this approach, and related research, are summarized in Section 4 and Section 5, respectively. Our conclusion is drawn in Section 6.

2 Semantic Relations for Question Selection

General domain knowledge enables question selection to be based on semantic rather than purely syntactic criteria. Below, we describe a set of semantic relations among features, which influence question selection.

- Feature Abstraction. A feature can be described at different abstraction levels that form a subsumption hierarchy. The lower the level a feature belongs to, the more specifically it can describe the case, but the more difficult it will be to obtain. The appearance of a lower level feature can be used to infer the existence of higher level features. For instance, the feature of "Fuel Transmission Faulty" is a lower level feature than that of "Fuel System Faulty". In [17], Gupta argued that the conversations should follow a downward taxonomic traversal to extract questions from general to specific, which prunes questions deemed irrelevant or implicitly inferred by the taxonomy. Here, we define a relation "subclass of" to express the relation of "feature abstraction". "A is a subclass of B" means A is a lower level feature than B.
- Dependency Relations. A dependency relation between two features exists if the appearance of one feature depends on the existence of the other. For instance, the assertion that the fuel pump can pump fuel depends on that the car has fuel in its fuel tank. We define a relation "depends on" to describe dependency relations. "A depends on B" means B is a necessary condition for A.
- Causality Relations. The causality relation means that one feature can cause the occurrence of another feature. For example, an electricity system fault in a car can cause its engine not to start. Here, we define a relation "causes" to express causality relations. "A causes B" means B is the result of A.
- Co-occurrence Relations. A particular relation, "co-occurs with", is defined to express that two features happen together, even though we cannot tell which one causes the other.
- Answer Acquisition Costs. The costs or difficulties of obtaining answers to different questions are various [11]. For instance, to test whether a switch has a fault is more difficult than to test whether the battery has electricity. The relation "is more costly than" is defined to represent that the answer to one question is more difficult or costly to obtain than the answer to another question.

How the above relations can be used to support the knowledge-intensive question selection tasks is illustrated in Table 1.

Our intention here is not to enumerate all the semantic relations that influence the question selection in conversational CBR, but to give some examples and illustrate how our approach can utilize them to improve the question selection process. System implementors can also define their own semantic relations which they think influence the question selection process. We will show that it is straightforward to add a new semantic relation into the question selection application later in the paper.

Table 1. Semantic relations used in the knowledge-intensive question selection

	Feature Inferencing	Knowledge-Intensive Question Ranking	Consistent Question Clustering	Coherent Question Sequencing
Feature Abstraction (A is a subclass of B)	Inference B from A	Ask A after B	Group A and B together	A succeeds B
Dependency Relations (A depends on B)	Inference B from A	Ask A after B	Group A and B together	A succeeds B
Causality Relations (A causes B)	Inference B from A	Ask B after A	Group A and B together	
Co-occurrence Relations (A co-occurs with B)	Inference B from A; Inference A from B		Group A and B together	
Answer Acquisition Costs (A is more costly than B)		Ask A after B		

3 An Explanation-Boosted Question Selection Approach

In this section, the explanation-boosted question selection approach is described, focusing on three architectural and methodological issues: knowledge representation, explanation construction, and explanation-boosted reasoning method.

3.1 Knowledge Representation

A frame-based knowledge representation model, which is a part of the CREEK system [1, 3, 24], is adopted in our system. In CREEK, both case-specific knowledge and general domain knowledge are captured as a network of concepts and relations, each concept and relation is represented as a frame in a frame-based representation language. A frame consists of a set of slots, representing relationships with other concepts or with non-concept values, e.g. numbers. A relationship is described using an ordered triple $<C_f, T, C_v>$, in which C_f is the concept described by this relationship, C_v is another concept acting as the value of this relationship (value concept), and T designates the relation type, simply called relation. The equation $T = C_v$ can also be used to describe a relationship when C_f is default. Viewed as a semantic network, a concept corresponds to a node and a relation corresponds to a link between two nodes.

In the system presented here, knowledge is represented at two levels. The first is the object-level, in which case-specific knowledge and general domain knowledge are represented within a single representation framework. The second is the meta-level, which is used to express the inter-relations of the semantic relations influencing the question selection tasks.

3.1.1 An Object-Level Knowledge Representation Model

As an illustration of how a case is described, Fig. 2 shows, in a frame view, the contents of a new case in the car fault domain, while Fig. 3 shows, in a semantic network view, a part of the integrated knowledge base for that domain. As can be seen, the semantic relations identified in Section 2 are represented as relations connecting different concepts. Cases are integrated into the general domain model, since all case features are defined as concepts within it.

The relationship values, which have corresponding relationships in the retrieved cases, but do not have the same type relationships in the new case, can be converted into discriminative questions. For example, if the relationship value, "Engine Dose Not Turn", has a relationship in one of the retrieved cases, that is, "has engine status = Engine Does Not Turn", but does not have the same type relationship in the new case, then a discriminative question, "What is the engine status of your car?", is added to the discriminative question list.

We define a function that maps a set of relationship values to a set of questions, Q: relationship value set → question set. On this function, we define the following properties:

- The question transformed from one relationship value is the same as those formed by the relationship values that belong to the same relation type. So we only predefine one question for each relation, which is shared by the relationship values belonging to this relation. For example, Q("Engine Fires") = Q("Engine Turns") = Q("Engine Does Not Fire") = Q("Engine Stops After A Few Seconds") = "What is the engine status of your car?".
- The semantic relations that exist between two relationship values are transferred to the two questions transformed by these two relationship values. For instance, the "causes" relation that "Fuel Pump Damaged" "causes" "Engine Stops After A Few Seconds" is transformed to Q("Fuel Pump Damaged") "causes" Q("Engine Stops After A Few Seconds"). Following the "has question" link to the actual question, "What is the fuel pump status of your car?", it follows that this question "causes" the question "What is the engine status of your car?".

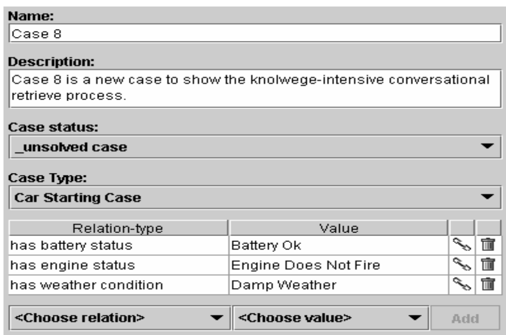

Fig. 2. The frame structure for a car starting case in CREEK

3.1.2 Meta-level Relations and Reflective Reasoning

Four meta-level relations have been defined in order to control the inference processes related to each of the four question selection tasks. For feature inferencing, we define the "*infers*" relation to express that if A infers B, we can get B from the existence of A. This relation has the property of transitivity that if A infers B and B infers C then A infers C. Several semantic relations identified in Section 2, "subclass of", "depends on", "causes" and "co-occurs with" are subclass relations of the "infers" relation since all these relations can be used to infer the existence of the post-condition based on the appearance of the pre-condition.

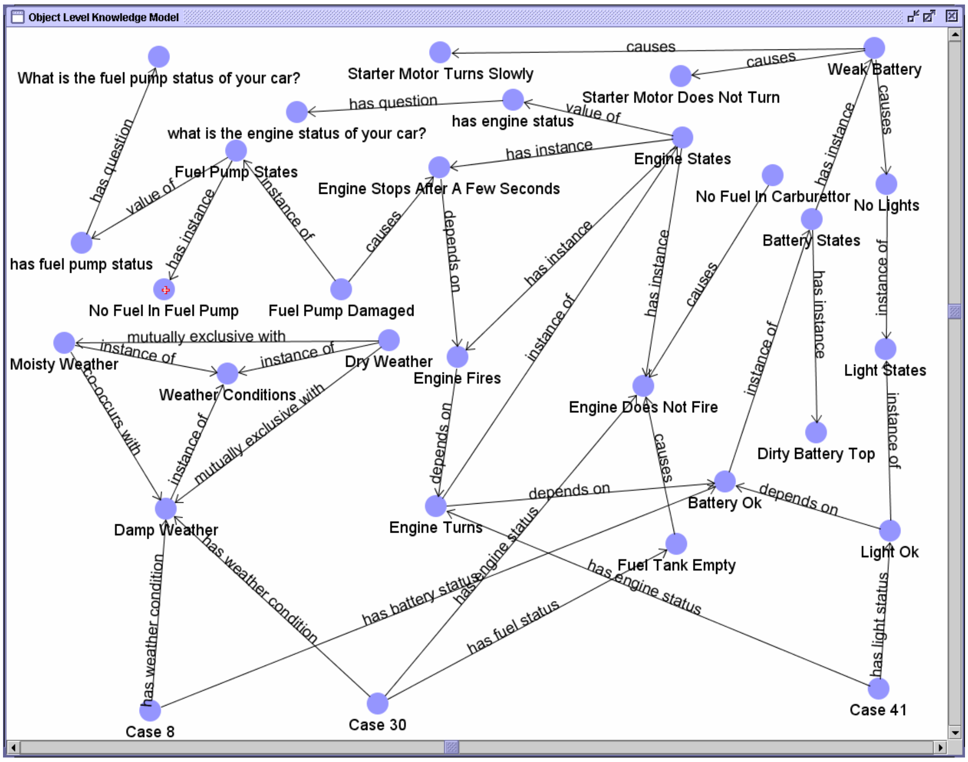

Fig. 3. The "Dialogue" pane in one conversation session

The second metal-level relation, "*appears after*", is defined to complete the question ranking task. "A appears after B" means that Q(A) should be asked after Q(B). This relation also has the property of transitivity that if A appears after B and B appears after C then A appears after C. We define several relations identified in Section 2, "subclass of", "depends on", "caused by" and "is more costly than" as the subclass relations of the "appears after" relation because all these relations can rank the pre-condition question to be asked after the post-condition question.

The third meta-level relation, named "joins", is defined to realize the consistent question clustering task. "A joins B" means that Q(A) should be grouped and displayed together with Q(B). We define several relations identified in Section 2, "subclass of", "depends on", "causes" and "co-occurs with" as subclass relations of the "joins" relation because all the questions connected by these relations should be grouped and displayed together. The transitivity property is not defined on "joins" because we assume that only the questions that have direct "joins" relations between them can be grouped and displayed together.

The last meta-level relation, called "succeeds", is used in the coherent question sequencing task. "A succeeds B" means Q(A) should be asked directly after Q(B) in two sequential question-answer cycles. There are two relations, "subclass of" and "depends on", defined as the subclass relations of this "succeeds" relation. On this basic relation, the transitivity property is also defined, that is, if A succeeds B and B succeeds C, we can get A succeeds C.

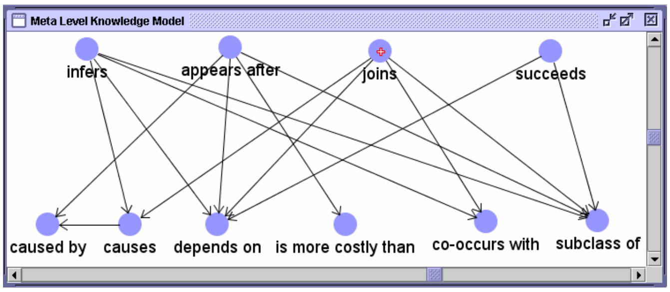

Fig. 4. The structure of the meta-level knowledge representation model

Fig. 4 shows the structure of the meta-level knowledge representation model described above. The top part relations are the meta-level relations defined above, while the bottom part relations are the semantic relations identified in Section 2. The lines from the top part relations to the bottom part relations designate the "has subclass" relations, while the line from "causes" to "caused by" is a "has inverse" relation.

One type of reflective reasoning operation, subclass inheritance, is made explicit in this meta-level knowledge representation model. Subclass inheritance is a special case of the more general "plausible inheritance" mechanism in CREEK [1], and makes subclass relations inherit the properties and reasoning operations (e.g. explanation construction, as introduced in the next sub-section) defined on their parent relation. Thus we need only define the properties and reasoning operations once on the meta-level relations, and all its subclass relations that express much richer domain-specific meanings can inherit them automatically. The other benefit is that new semantic relations can be easily incorporated through defining them as the subclasses of one of the meta-level relations.

3.2 Explanation Construction

Explanation construction is to set up explanation paths between concepts in the semantic network, which are used to explore solutions for particular knowledge-intensive question selection tasks.

We have defined two levels of explanation construction operations. The first level is called "Direct Explanation Construction", which is suitable when there is a direct (local) relation between two concepts. For example, if there are two questions Q(A) and Q(B) and there is a relation "A is a subclass of B", then a direct explanation is constructed that "Q(A) is ranked after Q(B) because A (one possible answer of Q(A)) is a lower level concept than B (one possible answer of Q(B))" in the knowledge-intensive question ranking phase.

The second level is referred to as "Transitive Explanation Construction", which is suitable where there is no direct relation between two concepts in the knowledge base, but we can set up a new semantic relation between them through exploring other relations in the knowledge base.

The transitive explanation construction is based on the transitivity property defined on different relations. In the meta-level knowledge model, we define the transitivity property on the "infers" relation, the "appears after" relation and the "succeeds" relation, and all their sub-class relations can inherit such property from them. So in each relation category (formed by one of these three basic relations and its sub-class relations), all the subclass relations can be transferred on each other to construct new super-class type relations.

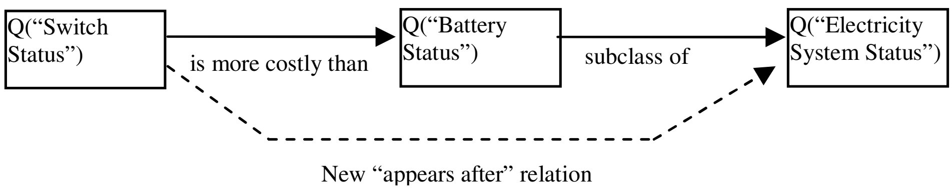

Fig. 5. How to construct a new "appears after" relation

Fig. 5 gives an example of how to build up a new explanation path in the "appears after" relation category through exploring two different subclass relations. In this figure, there are two relations: Q("Switch Status") "is more costly than" Q("Battery Status") and Q("Battery Status") is a "subclass of" Q("Electricity System Status"). Following the "is more costly than" relation and the "subclass of" relation, a new "appears after" relation, Q("Switch Status") "appears after" Q("Electricity System status"), is constructed. Thus if we have two questions Q("Switch Status") and Q("Electricity System Status"), we can rank them through constructing the explanation path that "Q("Switch Status") should be asked after Q("Electricity System Status"), because to answer Q("Switch Status") is more costly than to answer Q("Battery Status"), and Q("Battery Status") is a lower level question than Q("Electricity System Status")" in the concept taxonomy about the electricity system fault.

As discussed in the previous subsection, the "joins" relation does not have the property of transitivity. So we can only use the "Direct Explanation Construction" operation to construct explanations to accomplish the consistent question clustering task.

In the CREEK representation, each relation has a default explanation strength attached to it. The explanation strength of a constructed chain of linked relations, which constitute an explanation path, is calculated on the basis of these defaults (in our implementation introduced in Section 4, we will simply use the product of the defaults to indicate the explanation strength of the constructed explanation path).

3.3 Explanation-Boosted Reasoning Process

The explanation-boosted reasoning process can be divided into three steps: ACTIVATE, EXPLAIN and FOCUS. The three steps, which constitute a general process model for knowledge-intensive CBR, was initially described for the retrieve phase [1], although it applies in principle to all four phases of the CBR cycle. Here this model is instantiated for the different question selection tasks. ACTIVATE determines what knowledge (including case-specific knowledge and general domain knowledge) is involved in one particular task, EXPLAIN builds up explanation paths to explore possible solutions for that task, and FOCUS evaluates the generated explanation paths and identify the best one/ones for that particular task. The operations, done at each step in accomplishing a knowledge-intensive question selection task, are shown in Table 2.

Table 2. Explanation-boosted Reasoning Process in the knowledge-intensive question selection

	Feature Inferencing	Knowledge-intensive Question Ranking	Consistent Question Clustering	Coherent Question Sequencing
ACTIVATE (identify knowledge)	New case features and the related "infers" relations	Discriminative questions and the related "appears after" relations	Sorted questions and the related "joins" relations	Answered questions in the last conversation session and the "succeeds" relations between them and the discriminative questions in current session
EXPLAIN (construct explanation paths)	Feature inferencing explanation paths	Knowledge-intensive question ranking explanation paths	Question clustering explanation paths	Question sequencing explanation paths
FOCUS (evaluate explanation paths and use them to accomplish particular tasks)	The accepted explanations are transformed to new case features	The accepted explanations are combined together with statistical metrics to rank discriminative questions	The accepted explanations are used to group the sorted questions	The accepted explanations are used to re-rank the discriminative question groups

4 System Implementation

We have implemented our proposed approach within the TrollCreek system [2]. TrollCreek is an implementation of CREEK that contains a graphical knowledge model editor and a knowledge-intensive case-based reasoner. Our implementation adds the conversational process with its explanatory mechanism into the retrieve phase.

We are currently exploring two application domains for our CCBR method, car fault detection, and component retrieval for reuse of useful components when developing image processing software [15]. Car fault detection is an example domain adopted in our group for the study of basic knowledge modeling, and representational and reasoning methods, related to particular research directions (e.g. conversational CBR). The knowledge base in this domain incorporates the car fault detection domain knowledge and 29 stored cases. In the graphic window of the knowledge base, we can select an existing case or create a new case to start a knowledge-intensive conversational case retrieve process.

A conversational retrieve process contains one or several conversation sessions (the number of the sessions depends on when the searcher finds her desired case or whether there are still discriminative questions left).

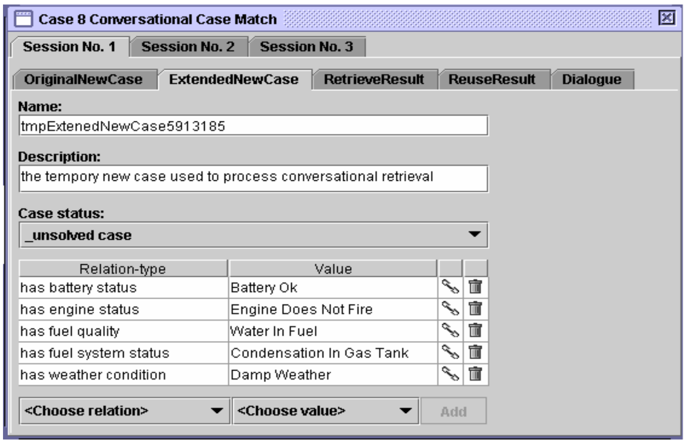

Fig. 6. The "ExtendedNewCase" pane in one conversation session

In the computer interface there are five window panes to move between within each session: The **OriginalNewCase** pane (the example of the content of this pane can be seen in Fig. 2) is used to display the new case in the particular conversation session. The new case extended by the inferred features in the feature inferencing phase is then displayed in the **ExtendedNewCase** pane (as shown in Fig. 6). Based on the extended new case, the CBR retrieve module retrieves a set of sorted cases and displays them in the **RetrieveResult** pane (as illustrated in Fig. 7). In this pane you can inspect the matching details between each retrieved case and the extended new case. The solution for the extended new case is then calculated by the retrieved cases and displayed on the **ReuseResult** pane. If you are not satisfied with the retrieved cases and the reuse result, you can go to the **Dialogue** pane (shown on Fig. 8) to select and answer the discriminative questions, and enter a new conversation session.

The question ranking module divides the identified discriminative questions into two groups: Group one includes the questions that are constrained to be ranked after other questions by some constructed "appears after" explanation paths; Group two contains all the remaining questions. The questions in Group two then gets ranked based on their occurrence frequency metrics [6]. Each question in Group one has one or more "appears after" explanation attached to it. The questions are sorted according to the strongest explanation attached to each questions. Then the ranked questions in Group two are sorted in front of the questions in Group one. If there are some “succeeds” explanation paths between the answered questions in the last conversation session and the current questions, the ranking priority of these involved questions are further increased (putting them in the front of the question queue), and the internal sequence of these “succeeding” questions are decided by their explanation strengths in the “succeeding” explanation paths. The ranked questions are displayed in the Dialogue pane. When each question is selected, its “joined” questions are also displayed in the “Dialogue” pane to prompt the user for further selecting and answering.

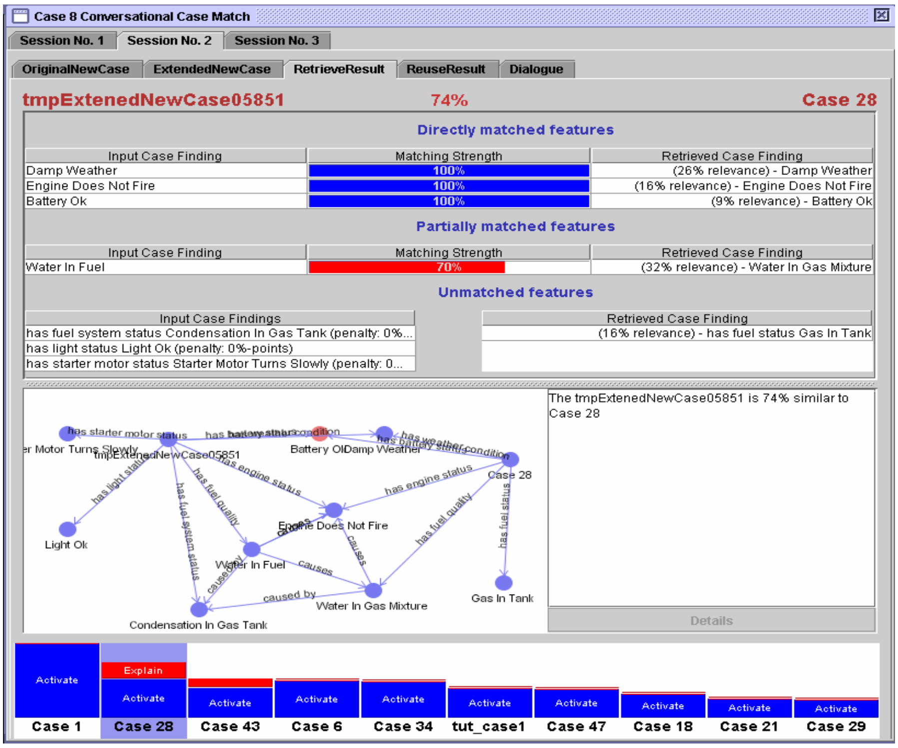

Fig. 7. The “RetrieveResult” pane in one conversation session

Our studies so far indicate that using general domain knowledge as explanatory support in a conversational CBR process improves the focusing of question-asking, and hence reduces the cognitive load needed to identify the best matching case. The target application for empirical testing of our approach will be software component reuse. We are currently building a knowledge base for the components existing in the DynamicImager system [15], a visualization and image processing development environment, in which there are about 200 different image operating components that can be combined in various ways. Our evaluation process will compare component retrieval with and without the explanation method, applied to one-shot vs. conversational CBR retrieval.

5 Related Research

In [22], Schmitt and Bergmann propose a formal model for dialogs between users and a conversation system, in which they identify four important issues in the conversation process: a small number of questions, comprehensible questions, low answering cost of questions and comprehensible question clustering. They also argue that the main reason for the unnatural question sequence during dialogue is due to the ignorance of the relations between different questions. However, they do not give methods about how to incorporate the semantic relations during the dialog process.

In [8], Aha, Maney and Breslow propose a model-based dialogue inferencing (feature inferencing) method. In their method, the general domain knowledge is represented in a library model (including object models and question models) taking the form of a semantic network. At run time, a set of rules are extracted from the library model using an implication rule generator, and the generated rules and the existing problem description are input to a PARKA-DB to infer the implicit knowledge.

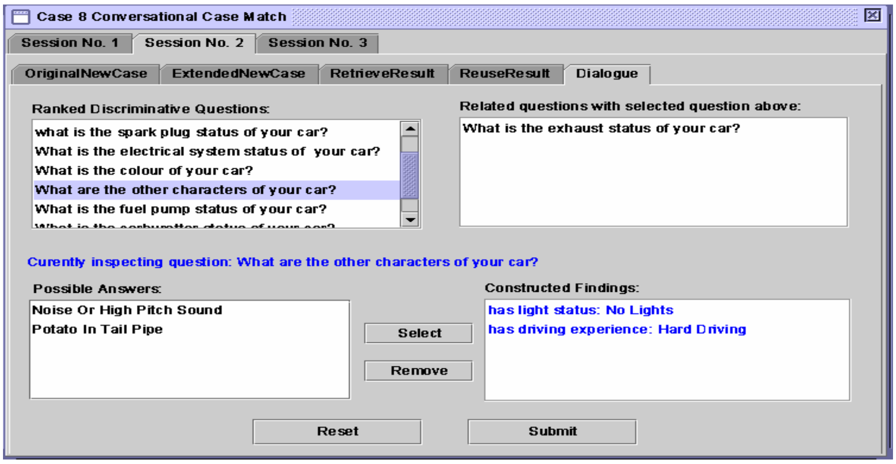

Fig. 8. The “Dialogue” pane in one conversation session

In [17], Gupta proposes a taxonomic conversational CBR approach to tackle the problems caused by the abstraction relations among features. In his approach, cases are described using one or more factors. For each factor, an independent subsumption taxonomy is created by the library designer in advance, and only the most specific feature in each factor taxonomy is selected to describe a case. The similarity between one <question, answer> pair in a case query and one in a case is calculated based on their relative positions in the taxonomy. The question generated from a higher level feature in one factor taxonomy is constrained to be asked before those that come from the lower level features.

Aha, Gupta and Sandhu identify the dependency relation among features [7, 18]. In their method, dependency relations are only permitted to exist between the root nodes among various factor taxonomies and the post-condition node in one dependency relation is excluded from the case representation. In the question ranking step, the question generated from a post-condition node in a dependency relation has higher priority to be asked than the question formalized by the pre-condition node.

Carrick, Yang, Abi-Zeid and Lamontagne try to eliminate the trivial and the repeated questions from users by accessing other information sources to answer them automatically [9]. They take the question answer acquisition costs into account when selecting a task (question) to execute instead of only the information quality metric. In this method, an execution plan is formulated for each question using a hierarchical task network (HTN). The estimated cost for each question is calculated through propagating cost values upward from leaves to the root using the mini-max algorithm.

Comparing with the above knowledge-intensive question selection methods, our approach contributes to the conversational CBR research in two ways: we propose a common integrated framework (including knowledge representation model, explanation construction mechanism and three-step reasoning process) to solve the knowledge-intensive question selection tasks comprehensively (feature inferencing, integrated question ranking, consistent question clustering and coherent question sequencing); and by creating a meta-level knowledge representation model, our approach has the capability to be easily extended to support richer semantic relations that influence the question selection in conversational CBR.

6 Conclusion

The explanation method presented in this paper is based on the CREEK knowledge-intensive CBR approach. The method described extends the existing system with a conversational method and an explanation mechanism targeted at conversational CBR support.

Limitations of the approach include the following two problems. The first is the method's dependence on knowledge engineering. The knowledge base combining both specific cases and general domain knowledge is assumed to exist initially. The construction of this knowledge base puts a significant workload on the development team. However, recent developments in the areas of Knowledge Acquisition and Modeling, as well as Ontology Engineering, provide systematic methods that help reduce this problem [4] . We are also looking into machine learning methods, particularly Bayesian Networks, for solving parts of the problems involved [19] .

The second is conflicting knowledge correction. We store the general domain knowledge in the knowledge base, which explicitly expresses the relations among concepts. However, the knowledge provided by users, including the initial problem description and later answers to discriminative questions, can conflict with this stored general domain knowledge. The problem can be reduced by incorporating an automatic mechanism to detect the knowledge conflicts in order to warn users to revise their new cases, or help knowledge base designers to update the predefined mistaken knowledge.

References

1. Aamodt A. Explanation-driven case-based reasoning. Topics in Case-based reasoning 1994:274-88.
2. Aamodt A. Knowledge-intensive case-based reasoning in Creek. In Funk P, Calero PAG eds. 7th European Conference on Case-Based Reasoning. Madrid, Spain: Spinger, 2004.
3. Aamodt A. A Knowledge Representation System for Integration of General and Case-Specific Knowledge. International Conference on Tools with Artificial Intelligence. New Orleans, 1994:4.
4. Aamodt A. Modeling the knowledge contents of CBR systems. Workshop Program at the Fourth International Conference on Case-Based Reasoning. Vancouver: Naval Research Laboratory Technical Note AIC-01-003, 2001:32 - 7.
5. Aamodt A, Plaza E. Case-Based Reasoning: Foundational Issue, Methodological Variations, and System Approaches. AI Communications 1994;7:39-59.
6. Aha DW, Breslow LA, Munoz-Avila H. Conversational Case-Based Reasoning. Applied Intelligence 2001;14:9-32.
7. Aha DW, Gupta KM. Causal Query Elaboration in Conversational Case-Based Reasoning. International Florida Artificial Intelligence Research Society Conference. Pensacola Beach, Florida, USA, 2002:95-100.
8. Aha DW, Maney T, Breslow L. Supporting Dialogue Inferencing in Conversational Case-Based Reasoning. European Workshop on Case-Based Reasoning. Dublin, Ireland, 1998:262-73.
9. Carrick C, Yang Q, Abi-Zeid I, et al. Activating CBR Systems through Autonomous Information Gathering. International Conference on Case Based Reasoning. Germany, 1999.
10. Cunningham P, Bergmann R, Schmitt S, et al. WEBSELL: Intelligent Sales Assistants for the World Wide Web. KI - Kunstliche Intelligenz 2001;1:28-31.
11. Cunningham P, Smyth B. A Comparison of Model-Based and Incremental Case-Based Approaches to Electronic Fault Diagnosis. Case-Based Reasonging Workshop. Seattle, USA, 1994.
12. Doyle M, Cunningham P. A Dynamic Approach to Reducing Dialog in On-Line Decision Guides. European Workshop on Advances in Case-Based Reasoning. Trento, Italy, 2000:49-60.
13. Göker MH, Thompson CA. Personalized Conversational Case-Based Recommendation. the 5 th European Workshop on Case-Based Reasoning(EWCBR 2000). Trento, Italy, 2000.
14. Gu M, Aamodt A. Explanation-boosted question selection in conversational CBR. ECCBR-04 workshop on Explanation in CBR. Madrid, Spain, 2004:105-14.
15. Gu M, Aamodt A, Tong X. Component retrieval using conversational case-based reasoning. In Shi Z ed. International Conference on Intelligent Information Systems. Beijing, China, 2004.
16. Gupta KM. Knowledge-based system for troubleshooting complex equipment. international Journal of Information and Computing Science 1998;1:29-41.
17. Gupta KM. Taxonomic Conversational Case-Based Reasoning. International Conference on Case-Based Reasoning. Vancouver, BC, Canada, 2001:219-33.
18. Gupta KM, Aha DW, Sandhu N. Exploiting Taxonomic and Causal Relations in Conversational Case Retrieval. European Conference on Case Based Reasoning. Aberdeen, Scotland, UK, 2002:133-47.

19. Langseth H, Aamodt A, Winnem OM. Learning Retrieval Knowledge from Data. In Anand SS, Aamodt A, Aha DW eds. Workshop ML-05: Automating the Consruction of Case-Based Reasoners, in Sixteenth International Joint Conference on Artificial Intelligence. Stockholm, 1999:77 - 82.
20. Mcsherry D. Interactive Case-Based Reasoning in Sequential Diagnosis. Applied Intelligence 2001;14:65-76.
21. Schmitt S. simVar: A Similarity-Influenced Question Selection Criterion for e-Sales Dialogs. Artificial Intelligence Review 2002;18:195-221.
22. Schmitt S, Bergmann R. A Formal Approach to Dialogs with Online Customers. The 14th Bled Electronic Commerce Conference. Bled, Slovenia, 2001:309-28.
23. Shimazu H. ExpertClerk: A Conversational Case-Based Reasoning Tool for Developing Salesclerk Agents in E-Commerce Webshops. Artificial Intelligence Review 2002;18:223 - 44.
24. Sørmo F. Plausible Inheritance: Semantic Network Inference for Case-Based Reasoning. Department of Computer and Information Science. Trondheim: Norwegian University of Science and Technology, 2000:102.

Paper E

Supporting Generalized Cases in Conversational CBR

```
@INPROCEEDINGS{Gu2005d,
  author = {Mingyang Gu},
  title = {Supporting Generalized Cases in Conversational CBR},
  booktitle = {Advances in Artificial Intelligence,
          Proceedings of the Fourth Mexican International
          Conference on Artificial Intelligence (MICAI2005)},
  year = {2005},
  editor = {Alexander Gelbukh and Alvaro de Albornoz
          and Hugo Terashima-Marin},
  volume = {3789},
  series = {Lecture Notes in Artificial Intelligence},
  pages = {544 - 553},
  address = {Monterrey, Mexico},
  month = {November},
  publisher = {Springer Verlag},
}
```

Supporting Generalized Cases in Conversational CBR

Mingyang Gu

Department of Computer and Information Science,
Norwegian University of Science and Technology,
Sem Saelands vei 7-9, N-7491, Trondheim, Norway
mingyang@idi.ntnu.no

Abstract. Conversational Case-Based Reasoning (CCBR) provides a mixed-initiative dialog for guiding users to refine their problem descriptions incrementally through a question-answering sequence. Most CCBR approaches assume that there is at most one discrete value on each feature. While a generalized case (GC), which has been proposed and used in traditional CBR processes, has multiple values on some features. Motivated by the conversational software component retrieval application, we focus on the problem of extending CCBR to support GCs in this paper. This problem is tackled from two aspects: similarity measuring and discriminative question ranking.

1 Introduction

The basic idea underlying case-based reasoning (CBR) [1, 2] is to reuse the solution to the previous most similar problem in helping solve the current problem. In traditional CBR processes, users are assumed to be able to provide a well-defined problem description, and based on such a description a CBR system can find the most appropriate previous case (base case). But this assumption is not always realistic. In some situations, users only have vague ideas about their problems at the beginning of retrieval, and tend to describe them by surface features.

Conversational Case-Based Reasoning (CCBR) [3] provides a mixed-initiative dialog for guiding users to construct their problem descriptions incrementally through a question-answering sequence. In CCBR, as illustrated in Fig. 1, a user provides her initial problem description that is to be transformed as an initial new case. The CCBR system uses the initial new case to retrieve the first set of most similar cases, and identifies a group of informative features from them to generate discriminative questions. Both the retrieved cases and identified discriminative questions are ranked and shown to the user. The user either finds out the base case to terminate the retrieval process or chooses a question to answer. An updated new case is constructed through combining the previous new case with the newly answered question. Subsequent rounds of retrieving and question-answering will iterate until the user finds her desired

A. Gelbukh, A. de Albornoz, and H. Terashima (Eds.): MICAI 2005, LNAI 3789, pp. 544–553, 2005.

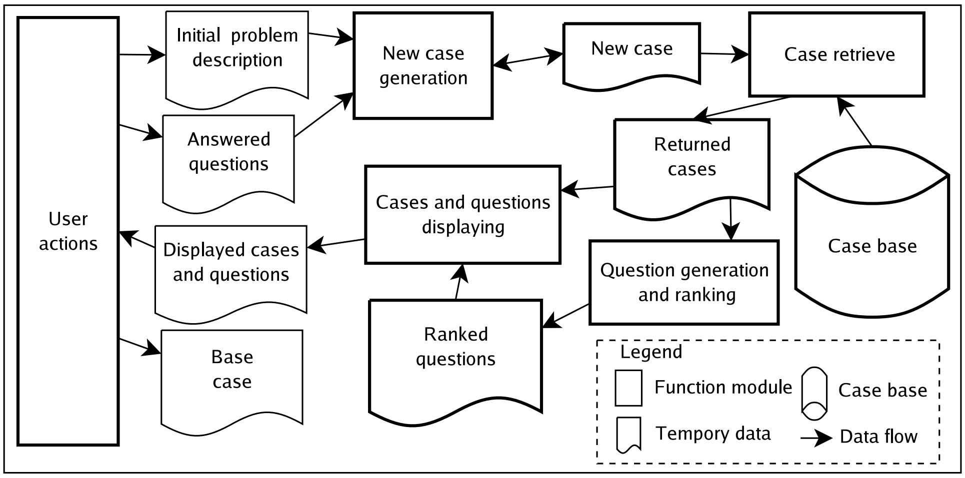

Fig. 1. Conversational Case-Based Reasoning

base case or no discriminative questions are available. CCBR applications have been successfully fielded, such as [3, 4, 5, 6].

Most of the works in CCBR assume that on each feature, there is either missing value or one discrete value (so called point cases, PC). While a GC [7] has multiple values on some features. In the CBR community, there are a considerable amount of research works concerning GCs [8, 9, 10, 11]. While to our knowledge, there are no published results about how to support GCs in CCBR. In this paper, motivated by the software component retrieval application [6], we extend CCBR process to support GCs.

The rest of this paper is organized as follows. In Section 2, we present the software component retrieval application, in which both the component query and stored software components are formalized as GCs; a formal model to represent GCs and a method to calculate the similarity between them are presented in Section 3 and Section 4, respectively; in Section 5, we analyze the feasibilities of applying the current question ranking methods in CCBR to support GCs; related works are listed in Section 6, and we draw our conclusion in Section 7.

2 Motivation to Support Generalized Cases in CCBR

2.1 Software Component Retrieval Using Conversational CBR

Software component retrieval, which is concerned with how to locate and identify appropriate components to satisfy users' requirements, is one of the major problems associated with the software component reuse. With the emergence of several component architecture standards, such as, CORBA, COM, and EJB, software components interoperation becomes more easily. Therefore, component reuse surpasses the limitation of a single software company, that is, instead of getting components from an in-house component library, users search for desired components from component markets (web-based software component collections provided by vendors or third parties). Without the knowledge about how the components are constructed and stored by others, it is very hard for users to well define their component queries. In our research, we propose a conversational software component retrieval approach using the CCBR technology, in which each component takes the form of a stored case, and a component query, as a new case, is constructed by the conversational process.

2.2 Representing Component and Query Using Generalized Cases

Comparing with the PCs used in traditional CCBR works, cases used to support component retrieval need to have multiple values on some features (GCs) for both a new case and stored cases. The semantic for a stored case (a software component) to have multiple values on one feature is that one software component has the capability to function in several situations specified by the multiple values on that feature. For instance, in the image processing domain, if a software component has three values on one feature, 'input image type': BMP, JPEG and TIFF, it means that component has the ability to process all these three different types of images. While the semantic for a new case to have multiple values on one feature is that a user demands all the requirements specified by these values to be satisfied.

3 A Formal Generalized Case Representation Method

Generally, a case in CBR can be represented by a three-item vector $< PD, SD, O >$ [2]:

—PD (Problem Description): the state of the world at the time the case was happening and, if appropriate, what problem needed to be solved at that time.

—SD (Solution Description): the stated or derived solution to the problem specified in the problem description.

—O (Outcome): the resulting state of the world after the solution has been carried out.

In our research, the PD part of a GC takes the form of a set of $< feature, value >$ pairs (fv pairs). Comparing with the fv pairs used for the PD of a PC, those for a GC have the following characteristics:

— For a PC, there is at most one fv pair for each feature. On the contrary, there may be multiple fv pairs for each feature in the PD part of a GC. Each fv pair in a generalized new case presents a specific requirement of the user on this feature, while that in the generalized stored case tells that the software component presented by this case can support the function described by this fv pair.

— For a PC, the *value* in each fv pair takes the form of a single number (numeric feature) or a string (nominal feature). On the contrary, the *value* in a fv pair of a GC may be either a single value (a number or a string) or a numeric interval. When the value takes the form of a numeric interval, it means, for a generalized new case, all the values existing in the interval are the demanded requirements by the user on this feature, and for a generalized stored case, the software component presented by this case can support all the functions or function variables specified by the values contained in the numeric interval.

Since the goal of the CCBR process is to identify the most similar or appropriate stored case in the case base, we use the unique tag of each case as the SD part and drop off the O information from the case description. In the software component retrieval application, we use each component's unique name as the SD part of the corresponding case.

A new case in the CCBR process has only the PD part compared with a stored case that has both the PD and SD parts.

As illustrated by Fig. 1, to complete a CCBR process, we need further defining the following two modules: case retrieve module, and question generation and ranking module.

4 Supporting Generalized Case Retrieval Using a Query-Biased Similarity Calculation Method

One important difference between CBR and CCBR lies on the content of the new case. In CBR, including that supporting GCs, a new case is assumed to be created completely in advance, and the case retrieval process is completed in one shot of retrieval. While a new case in CCBR is incrementally constructed by a sequence of question-answering processes, so it is incompletely or partially specified during the middle-stage retrieval process. There are some features that get values in stored cases, while have not be assigned values in a new case in the middle stage of the conversational retrieval process. The similarities on these features are the same, unmatched, to all the stored cases. In fact, if these features are assigned values in the end, the similarities on these features should benefit the final base case. So in order to avoid the negative influences of these features on the similarity measurement in the middle stage of the conversational retrieval, a query-biased similarity calculation method, which only takes the features appearing in the new case (query) into account, is more suitable for CCBR. The empirical efficiency of this method has been evaluated in [12]. The semantics behind this method is that to which degree the partially specified new case is satisfied by each stored case decides the possibility of that stored case to be selected and shown to the user. In our research, we compute the similarity by counting how many or to what degree the incomplete requirements specified by the fv pairs in a new case are satisfied by the fv pairs in a stored case.

In our approach, the similarity measurement is defined using the concept of distance. The greater the distance between a new case and a stored case, the less the similarity between them is.

$$distance(n, c) = \sqrt{\frac{\sum_{fv \in FVS} w_{fv} dif^2(fv, c)}{\sum_{fv \in FVS} w_{fv}}} \tag{1}$$

In addition, $dif(fv, c)$ is a function used to compute the difference between the new case and a stored case on the fv pair or to what degree this fv pair is satisfied by the stored case, c, which is defined as follows:

$$dif(fv,c) = \begin{cases} 1 & c \text{ has missing value on feature } f \\ 1 & f \text{ is a nominal feature, and } c \text{ has not the } fv \text{ pair} \\ & \quad \text{in its } PD \\ 0 & f \text{ is a nominal feature, and } c \text{ has the } fv \text{ pair in its } PD \\ |v - near(f,v,c)| & f \text{ is a normalized numeric feature, and } v \text{ is} \\ & \quad \text{a single number} \\ 1 - (\frac{len(cov(f,v,c))}{len(v)}) & f \text{ is a normalized numeric features, and v is} \\ & \quad \text{a interval value} \end{cases} \quad (2)$$

where $near(f,v,c)$ is a function to find out the nearest value of v in c on feature f, $cov(f,v,c)$ is a function to return the subintervals of v covered by the values in c on feature f, and $len()$ is a function to compute the length of its parameter (interval/intervals). For example, there are three fv pairs on a normalized numeric feature $f : (f, 0.1), (f, [0.2, 0.3]), and(f, [0.6, 0.8])$ in a stored case c, and two fv pairs in a new case n, $(f, 0.55)and(f, [0.2, 0.7])$. The differences for each fv pair in n are calculated as follows:

$$dif((f, 0.55), c) = |0.55 - near(f, 0.55, c)| = |0.55 - 0.6| = 0.05$$

$$dif((f, [0.2, 0.7]), c) = 1 - (\frac{len(cov(f,[0.2,0.7],c))}{len([0.2,0.7])}) = 1 - (\frac{len(\{[0.2,0.3],[0.6,0.7]\})}{0.5}) \\ = 1 - (\frac{0.2}{0.5}) = 0.6 \quad (3)$$

5 Selecting a Discriminative Question Ranking Metric

In our approach, the features that appear in the currently retrieved cases, but have not been assigned values in the current new case, will be transferred as discriminative questions. For example, in the currently retrieved cases, there is a feature, 'input image dimension', which is used to describe how many dimensions the input images of the corresponding software component should have. This feature has not been assigned values in the current new case, so a discriminative question, 'how many dimensions do you want the input images of your desired component to have?'

Before all identified discriminative questions are displayed to users, they are ranked according to their capabilities of discriminating the stored cases from each other if they are answered. In fact, CCBR research is currently to a large extent focusing on the discriminative question selecting and ranking in order to minimize the cognitive load on users to retrieve the base case. We review the question ranking methods used in traditional CCBR and classify them into four categories: information metric [5, 13, 14, 15, 16], occurrence frequency metric [3, 17], importance weight metric [12], and feature selection strategies [18]. We analyze, in the following subsections, the feasibilities of applying them in the CCBR supporting GCs, and propose the necessary adjustments if they can not be used directly.

5.1 Information Metric

Information is a way of measuring the uncertainty of a user about which stored case is the base case. When there is no clear requirement specified in the query (new case), the uncertainty is highest. With more and more features get their values in a new case, the uncertainty will reduce step by step. Normally, entropy function is used to quantify the uncertainty, and information gain measurement is used to measure which feature, if it is answered or assigned values, can produce most information. In [16], the author further proved that if the SD part of each case in a case base is unique (so called irreducible case base), the information gain one feature may provide is calculated by the following equation:

$$-\sum_i p_i log p_i \tag{4}$$

where $p_1, p_2 ... p_r$ are the propositions of this feature's values in the counted stored cases.

At first glance, it is impossible to apply the information metric-based feature ranking method in the CCBR supporting GCs, since the information metric requires each feature can at most have one single discrete value in a case. While through the following three steps, the generalized case base can be transferred to a state where the information metric can be applied:

—1. if there is a numeric interval value on one feature, we can use the sampling method applying on the value field of that feature to generate a set of sample numbers to represent the numeric interval.

—2. if there is a continuous-valued feature in the case base, Mitchell [19] discussed the methods transferring a continuous-valued feature into a discrete-valued feature through dynamically defining new discrete-valued features that partition the continuous attribute value into a discrete set of intervals.

—3. if there are multiple discrete values on one feature in a case, we can simply combine all these discrete values together to form a new value and add it into the value set for this feature. This step may cause a problem of combination explosion if the discrete values are combined randomly or irregularly. However, in practical application domains, the values, combined together to form a new value, normally have some semantic relations behind them. For instance, in the image processing software component retrieval application, there is a feature called 'the datatype of the input image'. On this feature, there are 12 discrete features totally, such as, byte, word, Dword, signed char, short, long, signed int, signed short, float, double, float complex, and double complex. While the number of the combination values of these discrete values is only 5, such as anyDataType, realDataType, integerDataType, unsignedIntegerDataType, and complexDataType. This is because the software component producers always provide the functions that are similar or related to each other, other than randomly selecting them in order to improve a component's function without increasing too much cost.

5.2 Occurrence Frequency Metric

In NaCoDAE (Navy Conversational Decision Aids Environment) [3], the identified discriminative questions are ranked according to their frequency appearing in the PD part of the returned cases. The larger the number of the returned cases in which one feature is assigned values, the higher ranking priority that feature gets. In this metric, it is assumed that the cases are highly heterogeneous, that is, the features appearing in one case may not appear in another case.

This metric will not deal with PCs or GCs differently, since it only count whether there is assigned value/values on one feature in a case. Therefore, this discriminative question ranking method can be directly applied in the CCBR supporting GCs.

5.3 Importance Weight Metric

One problem in case-based reasoning is the curse of the dimensionality [19], that is, not all the available features are relevant or equally important. Feature weighting method can avoid this problem by assigning different features with various importance weights according to their contributions, such as EACH [20] and Relief [21]. The rational behind the importance weight-based feature ranking method is that the most relevant or important features can provide more information than other features to discriminate cases from one another.

If we can find out features' importance weights, this metric can be used in the CCBR supporting GCs. While the fact, there are multiple values on some features in GCs, may cause difficulties in some weighting methods. For example, Relief requires computing the difference between a new case and a stored case on a particular feature as a whole. This problem can be solved by calculating the average, maximal or minimal value of all the difference values of the fv pairs sharing the same feature f, which are computed by Equation 2. Another point we need pay attention to is that some weight learning methods, for instance EACH and Relief, are supervised-learning process, that is, the cases in a case base should be able to be classified into limited number of groups or categories according to cases' SDs. If the case base is irreducible [16], such as the software component library (case base), these weighting methods are unsuitable.

5.4 Feature Selection Strategies Metric

From the perspective that whether the question ranking process can be explained, McSherry provided a new type of method, Strategist [18]. This method provided four feature-selection strategies, CONFIRM, ELIMINATE, VALIDATE, and OPPOSE, listed in order of priority. A feature supports the CONFIRM strategy, if it has a value that occurs only in the target solution class in the current returned cases, the ELIMINATE strategy if one of its values occurs in the target solution class but not in the likeliest alternative solution class, the VALIDATE strategy if one of its value is more likely in the target solution class than in any others, and the OPPOSE strategy if one of its values is less likely in the likeliest alternative solution class than in any others.

If we follow the three steps listed in the information metric subsection to transform the interval-valued features, continuous-valued, and multiple-valued features into the single-discrete-valued features, this method can be applied to the CCBR supporting GCs. In addition, this feature selection strategies metric uses a supervised process, and can not be applied to the irreducible case base.

5.5 Choosing a Suitable Discriminative Question Ranking Metric

For the software component retrieval application, we choose the information metric as the discriminative question ranking method. The rational behind our decision is that the case base storing software components is irreducible, therefore, we drop the importance weight metric and the feature selection strategies metric that incorporate supervised processes inside. In addition, the heterogeneous level of our case base is quite low, which makes it less efficient to choose the occurrence frequency metric.

6 Related Research

Recently, there are several researches [7, 22, 23] on how to calculate the similarity between GCs. The authors described a GC as a subspace in the case space, constructed by the case's features, instead of a point as in the traditional CBR. They formulate the similarity calculation problem between two GCs as a mathematical optimization problem. Comparing to their works, our method supports similarity measurement of GCs particularly for CCBR application. That is, in CCBR a new case is partially specified and incomplete, so a query-biased similarity calculation method is more suitable. In addition, we also tackle the discriminative question ranking problem in the CCBR supporting GCs.

In [24], Gupta proposed a taxonomic conversational CBR approach to solve the problems caused by the abstraction relations among feature values. In his approach, for each feature, an independent subsumption taxonomy is created by the case base designer in advance, and only the most specific fv pair in each feature taxonomy is selected to describe a case. The similarity between one fv pair in a new case and that in a stored case with the same f is calculated based on their values' relative positions in the taxonomy. The question generated from a higher level feature value in one feature taxonomy is constrained to be asked before those that come from the lower level feature values. If we consider that a higher level feature value in the feature taxonomy may implicitly contain all its lower-level feature values, this method is capable of supporting GCs in some sense. While comparing with our method, his method is unable to support a feature to have multiple values on the same abstraction level.

7 Conclusion

In this paper, we focus on the problem of supporting GCs in CCBR. This problem is tackled from two aspects: similarity measuring and discriminative question ranking. In similarity measuring, we adopt a query-biased similarity calculation method, that is, to count how many or to what degree the requirements specified by the fv pairs in a new case are satisfied by each stored cases. For discriminative question ranking, we analyze the feasibilities of applying four types of question ranking metrics, used in traditional CCBR, in the CCBR supporting GCs. In addition, from the software component retrieval application, we discuss the semantics of GCs, and exemplify how to choose a question ranking metric according to the characteristics of the application.

Recently, how to improve CCBR using knowledge-intensive methods is getting more and more attention [3, 17, 24, 25]. How to support GCs in knowledge-intensive CCBR methods is our further research direction, for instance, the multiple values assign to a feature may have overlapping or conflicting semantic relations.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issue, methodological variations, and system approaches. AI Communications **7** (1994) 39–59
2. Kolodner, J.: Case-based reasoning. Morgan Kaufmann Publishers Inc. (1993)
3. Aha, D.W., Breslow, L.A., Munoz-Avila, H.: Conversational case-based reasoning. Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies **14** (2001) 9
4. Gupta, K.M.: Knowledge-based system for troubleshooting complex equipment. international Journal of Information and Computing Science **1** (1998) 29–41
5. Shimazu, H.: Expertclerk: A conversational case-based reasoning tool for developing salesclerk agents in e-commerce webshops. Artificial Intelligence Review **18** (2002) 223 – 244
6. Gu, M., Aamodt, A., Tong, X.: Component retrieval using conversational case-based reasoning. In Shi, Z., ed.: International Conference on Intelligent Information Systems, Beijing, China (2004)
7. Maximini, K., Maximini, R., Bergmann, R.: An investigation of generalized cases. In: 5th International Conference on Case-Based Reasoning, Trondheim, Norway (2003) 261 – 275
8. Kolodner, J.L., Simpson, R.L.: The mediator: Analysis of an early case-based problem solver. Cognitive Science **13** (1989) 507–549
9. Reinartz, T., Iglezakis, I., Roth-Berghofer, T.: Review and restore for case-based maintenance. Computational Intelligence **17** (2001) 214–234
10. Rayner, N.J.W., Harris, C.J.: Mission management for multiple autonomous vehicles. research journal of University of Southampton (1996)
11. Gebhardt, F., Vob, A., Grather, W., Schmidt-Belz, B.: Reasoning with complex cases. Kluwer Academic Publishers (1997)
12. Gu, M., Tong, X.: Comparing similarity calculation methods in conversational cbr. In: Submitted to the 2005 IEEE International Conference on Information Reuse and Integration, Hilton, Las Vegas, Nevada, USA (2005)
13. Cunningham, P., Smyth, B.: A comparison of model-based and incremental case-based approaches to electronic fault diagnosis. in. In: Case-Based Reasoning Workshop, Seattle, USA (1994)

14. Carrick, C., Yang, Q., Abi-Zeid, I., Lamontagne, L.: Activating cbr systems through autonomous information gathering. In: International Conference on Case Based Reasoning, Germany (1999)
15. Goker, M.H., Thompson, C.A.: Personalized conversational case-based recommendation. In: the 5th European Workshop on Case-Based Reasoning(EWCBR 2000), Trento, Italy (2000)
16. McSherry, D.: Minimizing dialog length in interactive case-based reasoning. In: International Joint Conferences on Artificial Intelligence. (2001) 993–998
17. Gu, M., Aamodt, A.: A knowledge-intensive method for conversational cbr. In: International Conference on Case-Based Reasoning, Chicago Illinois (2005)
18. McSherry, D.: Interactive case-based reasoning in sequential diagnosis. Applied Intelligence **14** (2001) 65–76
19. Mitchell, T.: Decision tree learning. In: Machine Learning. McGraw Hill (1997) 414
20. Salzberg, S.: A nearest hyperrectangle learning method. Mach. Learn. **6** (1991) 251–276
21. Kira, K., Rendell, L.: The feature selection problem traditional methods and a new algorithm. In: Tenth National Conference on Artificial Intelligence. (1992)
22. Mougouie, B., Bergmann, R.: Similarity assessment for generalizied cases by optimization methods. In: 6th European Conference on Case-Based Reasoning, Aberdeen, Scotland, UK (2002) 249 – 263
23. Alexander Tartakovski, Martin Schaaf, R.M., Bergmann, R.: Minlp based retrieval of generalized cases. In: 7th European Conference on Case-Based Reasoning, Madrid, Spain (2004) 404 – 418
24. Gupta, K.M.: Taxonomic conversational case-based reasoning. In: International Conference on Case-Based Reasoning, Vancouver, BC, Canada (2001) 219–233
25. Gupta, K.M., Aha, D.W., Sandhu, N.: Exploiting taxonomic and causal relations in conversational case retrieval. In: European Conference on Case Based Reasoning, Aberdeen, Scotland, UK (2002) 133–147

Paper F

Dialog Learning in Conversational CBR

```
@INPROCEEDINGS{Gu2006a,
  author = {Mingyang Gu and Agnar Aamodt},
  title = {Dialog Learning in Conversational CBR},
  booktitle = {Proceedings of the 19th International
          FLAIRS Conference},
  year = {2006},
  editor = {Geoff C. J. Sutcliffe and Randy G. Goebel},
  pages = {358 - 363},
  address = {Melbourne Beach, Florida},
  month = {May 11-13},
  publisher = {AAAI Press},
}
```

Dialog Learning in Conversational CBR

Mingyang Gu and Agnar Aamodt

Department of Computer and Information Science, Norwegian University of Science and Technology, Sem Saelands vei 7-9, N-7491, Trondheim, Norway
Email: mingyang,agnar@idi.ntnu.no

Abstract. Conversational Case-Based Reasoning (CCBR) provides a mixed-initiative dialog for guiding users to refine their problem descriptions incrementally through a question-answering sequence. In this paper, we argue that the successful dialogs in CCBR can be captured and learned in order to improve the efficiency of CCBR from the perspective of shortening the dialog length. A framework for dialog learning in CCBR is proposed in the present paper, and an instance of this framework is implemented and tested empirically in an attempt to evaluate the learning effectiveness of the framework. The results show us that on 29 out of the 32 selected datasets, CCBR with the dialog learning mechanism uses fewer dialog sessions to retrieve the correct case than CCBR without using dialog learning.

1 Introduction

Reusing the solution to the previous most similar problem in helping solve the current problem is the basic idea underlying case-based reasoning (CBR) [1]. In [2], the authors formalize the CBR cycle into four steps: RETRIEVE the most similar previous case/cases to the current problem, REUSE the information or knowledge to solve the current problem, REVISE the proposed solution and RETAIN the problem solving experience likely to be useful in the future. The latter step is the learning step.

In traditional CBR processes, users are assumed to be able to provide a well-defined problem description, and based on such a description a CBR system can find the most appropriate previous case. But this assumption is not always realistic. In some situations, users only have vague ideas about their problems when beginning to retrieve, and often describe them by surface features.

Conversational Case-Based Reasoning (CCBR) [3] provides a mixed-initiative dialog for guiding a user to construct her problem description incrementally through a question-answering sequence. CCBR research is currently to a large extent focusing on discriminative question selection and ranking to minimize the cognitive load on users to retrieve the case that best matches the problem [4]. For example, selecting the most informative questions to ask [5–7], or using feature inferencing to avoid asking users the questions that can be answered implicitly using the current known information [8, 9].

Successful dialogs that have occurred in a CCBR system can be seen as previous solutions to users' case retrieval tasks and retained as cases. A new type of CBR is thereby introduced into the CCBR process to improve its efficiency. To our knowledge, there are so far no published results on how to improve the dialog efficiency in CCBR through this type of learning.

A framework for dialog learning in CCBR is presented in the next section, followed by a description of an implementation of this framework. Following from that, an experiment design for evaluating the efficiency of the dialog learning method is described. Then the experimental results are described and discussed, followed by our conclusion.

2 A Framework to Support Dialog Learning in CCBR

In CCBR, as illustrated in Fig. 1 (the upper part surrounded by the dashed line), a user provides her initial problem description as an initial new case (target case). The CCBR system uses the initial new case to retrieve the first set of most similar cases, and identifies a group of informative features to generate discriminative questions. Both the retrieved cases and the identified discriminative questions are ranked and shown to the user. The user either finds her desired case, which will terminate the retrieval process, or chooses a question, which she considers relevant to her task, and provides the answer. An updated new case is constructed through combining the previous new case with the answered question. Subsequent rounds of retrieving and question-answering will iterate until the user finds her desired case or no discriminative questions are available.

In order to support dialog learning in CCBR, we introduce a special CBR process, as illustrated in the lower part of Fig. 1, which includes a dialog case base, a dialog case RETRIEVE module and a dialog case RETAIN module. In addition, two other modules used in the standard CCBR process, question generating and ranking, and case and question displaying, are updated to utilize the most similar dialog case in the CCBR process. This utilization process can be seen as the dialog case REUSE process. In order to avoid the conceptual confusion between the conversational CBR process and the dialog learning CBR process, the CBR process where the cases come from the application case base is referred to as application CBR. The CBR where the cases come from the dialog case base is referred to as dialog CBR. Also, correspondingly, the terms used in the two CBR processes are distinguished by adding the adjective 'application' or 'dialog'.

Generally, a case in CBR can be represented by the following three parts [1]:

- Problem description: the state of the world at the time of the case, and, if appropriate, what problem needed to be solved at that time.
- Solution description: the stated or derived solution to the problem specified in the problem description.
- Outcome: the resulting state of the world after the solution was carried out.

Dialog case base. For a dialog case, the problem description part contains the information related to the dialog process: the initial constructed new case, the

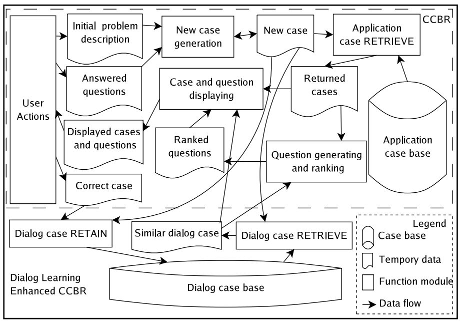

Fig. 1. A framework for the dialog learning enhanced CCBR

later incrementally selected questions, and their answers. The solution description of a dialog case refers to the case successfully retrieved from the application case base. For all dialog cases, their outcomes are the same, that is, the user gets the retrieved application case, and terminates the dialog.

Dialog case RETRIEVE. In CCBR, a new case, describing an application problem, is incrementally constructed and used to retrieve a case from the application case base. The same new case is also used to retrieve a case from the dialog case base. In addition to the features used in application case retrieval, the similarity assessment used in dialog case retrieval also takes the feature sequencing information into account. The positions of various features in a dialog process express the changes of users' focus of attention, which influence the similarity between the new case and a stored dialog case.

Dialog case REUSE. The retrieved most similar dialog case is considered to be able to improve the efficiency of the current conversational application case retrieval process in the following two ways:

The application case contained in the most similar dialog case is to be displayed to users for inspecting, that is, the case and question displaying module in standard CCBR needs to be modified.

The features that appear in the most similar dialog case, but do not appear in the current new case, will get improved ranking priority. That is, the question generating and ranking module in standard CCBR can be influenced.

Dialog case RETAIN. With the conversational application case retrieval going on, more and more successful dialogs will take place, and if we add all of them into the dialog case base as dialog cases, the dialog case base will grow rapidly. A dialog case learning strategy is needed to maintain the dialog case base during the run time to improve its capability without expanding too much.

3 An Implemented Instance of the Dialog Learning Enhanced CCBR Framework

In an attempt to evaluate whether our proposed framework can empirically improve the efficiency of the CCBR process, we have implemented an instance of this framework and evaluated it using a set of test datasets.

3.1 Underlying CCBR Process

We define an application case (ac) as

$$ac :< \{fvw\}, as >$$

Here, $ac, \{fvw\}$, and as express an application case, a set of three-item vectors ($< f, v, w >$) used to express the problem description of the application case ac, and the solution contained in case ac, respectively. In the present paper, our only concern is how to identify the most similar case from the case base (RETRIEVE), not the reuse of the case, so the outcome part of the case is dropped. $< f, v, w >$ is a vector that describes a feature in case ac, in which f denotes the feature name, v the feature value, and w is the importance weight for feature f.

A new case in the CCBR process has only the $\{fvw\}$ part, while a stored application case has both the $\{fvw\}$ and as parts.

To complete a CCBR process, we further define the following two modules: application case RETRIEVE, and question generating and ranking.

Application case RETRIEVE. In this experiment, we adopt a weighted k-NN algorithm to complete the application case retrieval task, in which the first k most similar cases are returned. The number k, typically 7, is used to control the number of cases that will be shown to the user in each conversation session in CCBR.

We use a global feature weighting method, similar to EACH [10], to get a set of global weights, one for each feature appearing in the application case base.

The similarity measurement between a new case and a stored application case is defined using the concept of distance. The greater the distance between the new case and a stored application case, the lower the similarity between them is.

$$distance(an, ac) = \sqrt{\frac{\sum_{f\in\{f\}} w_f dif^2(an_f, ac_f)}{\sum_{f\in\{f\}} w_f}} \tag{1}$$

where $an, ac, \{f\}$ and w_f denote an application new case, a stored application case, a feature set only including features appearing in an and the importance weight for a feature f, respectively. $dif(an_f, ac_f)$ is a function used to compute the difference between the new case and a stored application case on a feature f, and is defined as follows:

$$dif(an_f, ac_f) = \begin{cases} |an_f - ac_f| & f \text{ is numeric (normalized)} \\ max\{an_f, 1 - an_f\} & f \text{ is numeric (normalized)} \\ & \text{and } ac_f \text{ is missing} \\ 0 & f \text{ is nominal and } an_f = ac_f \\ 1 & f \text{ is nominal, and } an_f \neq ac_f \\ & \text{or } ac_f \text{is missing} \end{cases} \quad (2)$$

In [11], the authors argue, supported by experimental evidence, that the query-biased similarity calculation method (only taking the features appearing in the query (new case) into account during similarity computation) is the one most suitable for CCBR applications. The reason is that the new case in CCBR is incomplete and partially specified, and the query-biased similarity method can avoid the negative influence of the features that appear in the stored case but have not been assigned values in the new case. Therefore, in Equation 1, $\{f\}$ takes the value of all the features appearing in the new case.

Question generating and ranking. In our implementation, the features that appear in the application case base but have not been assigned a value in the current new case will be transferred as discriminative questions. Discriminative questions are ranked before being displayed to users. A weight-based question ranking strategy is used in our approach. For example, assume that there are three questions transferred from three features, A, B, and C with the weights values, 0.1, 0.2, and 0.05, respectively (learned from the feature weighting process). According to the weight-based question ranking strategy, their priority to be shown to users will be ranked as B, A, and C. The basic idea underlying this strategy is that the most relevant or important features can provide more information than other features to discriminate one case from others.

So, after a user provides her initial problem description, a case retrieval process will be executed, and the first returned k cases and the ranked discriminative questions are shown. If she can find her desired case, the CCBR process is terminated, otherwise, she will select and answer one question. An updated new case is constructed through adding the answered feature into the previous new case, and a new round of the RETRIEVE process starts. The retrieving, questioning, and answering cycle continues until the case is selected or no question is available.

3.2 Dialog Learning Enhanced Process

According to the framework introduced above, our implemented dialog learning process contains the following four parts:

Dialog case base. A dialog case (dc), in our approach, is defined as:

$$dc :< \{fvwp\}, ds >$$

Here, $dc, \{fvwp\}$, and ds express a dialog case, a set of four-item vectors ($< f, v, w, p >$) describing the problem description of the dialog case dc, and a dialog solution referring to the retrieved application case following the dialog process, respectively. $< f, v, w, p >$ is a vector that describes a feature in the dialog case dc, in which f denotes the feature name, v the feature value, w is the importance weight for feature f, and p is an integer value that expresses the appearance position of feature f in the dialog process.

A new case in the dialog learning enhanced CCBR is similar to that in CCBR introduced in the above subsection, but in order to support the dialog case retrieval, we add the feature position information into it. That is, the form of a new case in the dialog learning enhanced CCBR is '$\{fvwp\}$', instead of '$\{fvw\}$'.

Dialog case RETRIEVE. In our research we define the distance equation between a dialog new case and a stored dialog case as follows:

$$distance(dn, dc) = \sqrt{\frac{\sum_{f\in\{f\}} w_f posw(dn_f, dc_f) dif^2(dn_f, dc_f)}{\sum_{f\in\{f\}} w_f}} \qquad (3)$$

where $dn, dc, \{f\}$, and w_f denote a dialog new case, a stored dialog case, a selected feature set, and the importance weight for the feature f, respectively.

In Equation 3, w_f and $dif(dn_f, dc_f)$ have the similar definition as in Equation 1. In addition, $posw(dn_f, dc_f)$ is a function used to compute the weight concerning the appearance position of feature f in the dialog new case, dn, and the stored dialog case, dc:

$$posw(dn_f, dc_f) = \frac{1}{2} + \frac{1}{2} * \left(1 - \frac{|p(dn, f) - p(dc, f)|}{max(dialoglength(dn), dialoglength(dc))}\right) \qquad (4)$$

where $p(dn, f), p(dc, f), dialoglength(dn)$, and $dialoglength(dc)$ denote the appearance position of feature f in the new case, dn, and that in the dialog case, dc, and the dialog length of the new case and that of the dialog case. In addition, if a dialog case, dc, has missing value on feature, f, we assign $\frac{1}{2}$ to $posw(dn_f, dc_f)$. The underlying idea behind this equation is that the more similar the appearing positions of the feature in the new dialog case and the stored dialog case, the more important the difference of this feature between these two cases is to the similarity calculation.

Following the idea in [11], since we basically use the same new case to retrieve in both application case base and dialog case base, it is reasonable to adopt the query-biased similarity calculation method in the dialog case retrieval process, that is, $\{f\}$ is assigned the same value as in Equation 1: all the features appearing in the new case.

Based on Equation 3, a 1-NN algorithm is used to retrieve the most similar dialog case.

Dialog case REUSE. In our implementation, the most similar dialog case is used for two tasks: adjusting the displayed application cases and adjusting the discriminative question ranking priorities in the current dialog session.

For the first task, if the application case acting as the solution in the most similar dialog case is not included in the k most similar application cases, we use it to replace the least similar application case in the k returned cases. Concerning the second task, the following equation is used to adjust the weights of the candidate questioning features that also appear in the most similar dialog case:

$$\begin{aligned} &w_f = w_f + \\ &(\frac{1}{2} + \frac{1}{2} * (1 - \frac{p(dc, f))}{dialoglength(dc)}))(1/|total\ feature\ set|) \end{aligned} \tag{5}$$

where $|total\ feature\ set|$ is the number of the features that appear in the application case base.

Through increasing the weights of those candidate features that also appear in the retrieved most similar dialog case, those discriminative questions transferred from these features will be ranked with higher priority.

Dialog case RETAIN. If a successful conversational case retrieval takes place, whether this new dialog process should be stored as a dialog case is decided by the dialog case learning strategy. Our dialog learning strategy only stores the most general dialog cases in the case base. The relation, more general than ($\gg$), between two dialog cases is defined as:

$$\begin{aligned} &< \{fvwp\}_1, ds_1 > \gg < \{fvwp\}_2, ds_2 > : \\ &\qquad ds_1 = ds_2\ and\ \{fvwp\}_1 \subseteq \{fvwp\}_2 \end{aligned} \tag{6}$$

4 Experiment Design

Our experiment is designed in an attempt to evaluate the effectiveness of the dialog learning mechanism from the perspective of using fewer dialog sessions to find the correct stored case. We use a leave-one-out cross validation (LOOCV) method to simulate the human-computer dialog process; similar methods have been successfully used by the CCBR community [8, 11].

LOOCV proceeds with a series of simulated dialogs, each dialog starting with selecting a case from the application case base as the target case. The remaining cases form the case base to be searched. The initial new case is constructed

Table 1. Datasets description and experiment results

Dataset	Total Cases	Feat -ures	Solut -ions	DLNL C1	DLL C1	Cases C1	DLNL C2	DLL C2	Cases C2	Ave Shorten
Anneal	898	38	5	22.56	21.69	822	22.54	18.52	530	10.86%
Anneal Original	898	38	6	4.59	4.62	623	4.55	3.76	228	8.30%
Audiology	226	69	24	28.69	27.57	171	27.55	22.70	131	10.59%
Autos	205	25	7	3.07	2.69	147	3.37	3.17	141	9.02%
Balance Scale	625	4	3	3.04	3.20	369	3.06	2.36	-39	9.08%
Breast Cancer	286	9	2	5.42	5.30	153	5.39	4.59	52	8.47%
Breast-W	699	9	2	5.58	5.56	421	5.61	5.10	160	4.74%
Credit Approval	690	15	2	8.49	8.47	507	8.53	5.93	104	15.42%
Credit German	1000	20	2	6.66	6.61	622	6.61	5.65	235	7.67%
Diabetes	768	8	2	5.27	5.39	532	5.37	3.35	114	17.90%
Glass	214	9	7	3.69	3.83	145	4.06	2.91	60	13.16%
Heart Statlog	270	13	2	5.74	5.54	189	5.85	5.47	139	5.07%
Heart-h	294	13	5	5.30	5.08	190	5.37	4.43	79	10.97%
Heart-c	303	13	5	5.73	5.76	215	5.85	5.20	139	5.43%
Hepatitis	155	19	2	7.59	7.82	119	7.69	6.77	79	4.49%
Horse Colic	368	22	2	4.77	4.84	220	4.86	3.61	20	12.25%
Horse Colic Original	368	27	2	7.73	7.56	203	7.79	6.01	53	12.58%
Hypothyroid	3772	29	4	17.91	17.23	2882	17.93	15.61	1116	8.38%
Ionosphere	351	34	2	5.48	5.38	310	5.46	5.22	296	3.02%
Iris	150	4	3	2.31	2.28	123	2.35	1.84	30	11.69%
Kr-vs-kp	3196	36	2	21.13	21.07	2778	21.12	18.62	1337	6.08%
Labor	57	16	2	2.90	2.90	42	2.76	2.98	27	**-3.89**%
Lymph	148	18	4	6.99	7.06	114	7.19	6.56	89	3.96%
Primary Tumor	339	17	8	6.46	6.31	97	6.01	5.51	73	5.25%
Segment	2310	19	7	5.69	5.71	2130	5.81	4.18	1297	14.05%
Sick	3772	29	2	18.05	17.36	2983	18.09	15.82	1062	8.19%
Sonar	208	60	2	5.82	6.18	182	5.66	5.33	179	**-0.24**%
Soybean	683	35	19	14.22	13.22	498	14.42	8.91	177	22.73%
Vehicle	846	18	4	6.23	6.40	594	6.11	4.39	383	12.46%
Vote	435	16	2	7.81	7.42	243	7.80	6.89	69	8.36%
Vowel	990	13	11	3.23	3.19	962	3.27	2.99	707	4.98%
Zoo	101	17	7	5.86	5.92	86	5.52	5.55	62	**-0.81**%
Average				8.25	8.10	614.75	8.24	6.87	285.28	8.44%

through selecting the predefined number of features from the target case. Based on this initial new case, a retrieval process is carried out and the first k most similar cases are returned. If the correct case is included in the returned application case set, which means users find their desired case, the conversation process is finished successfully. Otherwise, a new feature is selected from the target case and added into the current new case, simulating a question-answering session between a human subject and a computer. The updated new case is then used to start a new round of retrieval. The retrieving, selecting, and adding cycle continues until the correct case appears in the returned application case set or there is no feature remaining to select in the target case.

There are two tasks we need to clarify in the above LOOCV process: the feature selection strategy and the correct case determination.

Feature selection strategy. Feature selection strategy is used to decide which feature should be selected from a set of candidate features and added into the current new case to simulate the question-answering process. In our implemented CCBR, features are ranked according to their weights. In LOOCV, we design a weight-biased random selection strategy to simulate the discriminative question selecting and answering process. For instance, suppose there are three features, A, B, and C in the candidate feature set with the weights values, 0.1, 0.2, and 0.3, respectively. According to the weight-biased random feature selection strategy, feature A, B, C will be selected with the possibilities $\frac{1}{6}$, $\frac{1}{3}$, and $\frac{1}{2}$, respectively.

Correct case determination. For each case in the application case base, its correct case, or to be more specific: its correctly matching case, is defined as the case returned by a weighted 1-NN algorithm and with the same solution value. Therefore, not all the cases in the application case base can act as a target case to simulate a dialog. If a case has a nearest neighbor with a different solution value, it will be dropped from LOOCV.

According to whether or not the dialog learning mechanism is used, the above LOOCV gets two variants. To each variant the above LOOCV cycle is executed twice with the aim to inspect the continuous learning characteristics of the dialog learning mechanism. For the LOOCV process without the dialog learning mechanism, the execution contexts are exactly the same for both two cycles. For that with the dialog learning mechanism, the only difference between these two cycles lies in the dialog case base content. That is, for the first cycle, the dialog case base is initially empty, while the dialog case base in the second cycle starts with a set of dialog cases learned from the first cycle[1].

We further identify the following hypothesis to test:

H1: the dialog learning mechanism is effective, that is, the CCBR system with the dialog learning mechanism is able to find the correct case using fewer dialog sessions than the one without dialog learning.

H2: the dialog learning mechanism is sustainable, that is, with the dialog learning process going on and more dialog cases being stored in the dialog case base, the performance of the dialog learning enhanced CCBR system keeps increasing.

H3: the dialog case base is maintainable, that is, with the dialog learning process going on, the dialog learning enhanced CCBR system retains fewer dialog cases to save.

5 Experiment Environment, Datasets and Results

We implement the experiment inside the Weka framework [12], and test it using the datasets provided by the Weka project, originally from the UCI repository [13]. There are 36 datasets available, and we choose 32 from them. The dataset selection criterion is quite simple, that is, the 4 datasets with the largest size are dropped out because they need too much execution time.

All the numeric features in these datasets are normalized using the corresponding filter provided in Weka3.4.3 according to the requirement of the distance calculation algorithm. The detailed information of the selected datasets is

[1] The random feature selection process (weight-biased) leads to different questioning sequences in two LOOCV cycles for each simulated target case. To some extent this compensates for the problem of the test set being biased by the training set.

illustrated in the left part of Table 1, in which the first 4 columns denote respectively: the name of each dataset (Dataset), the number of the cases (TotalCases), the total number of the features excluding the solution feature (Features), and the number of categories or solutions (Solutions).

The experiment results are listed in the right part of Table 1, in which the columns: DLNLC1, DLLC1, CasesC1, and NLDLC2, DLLC2, and CasesC2 denote the average dialog length without the dialog learning mechanism, the average dialog length with the dialog learning mechanism, and the number of the dialog cases obtained in the dialog case base in the first cycle and in the second cycle of LOOCV for each dataset.

To clearly show whether the dialog learning process really improves the dialog efficiency, we add one column into Table 1, AveShorten, to illustrate the percentage of the reduced dialog sessions in CCBR using the dialog learning mechanism for each dataset. And the last row gives the average values of the result parameters for all the 32 datasets.

Out of 32 datasets, there are 29 datasets in which CCBR enhanced by the dialog learning mechanism uses fewer dialog sessions to find the correct case than that without the dialog learning process (average using 8.44% fewer dialog sessions). Comparing the average results of the first LOOCV cycle with those of the second one, we can see that CCBR without the dialog learning mechanism uses almost the same dialog lengths in both the first and the second LOOCV cycle ($8.25 \approx 8.24$), while CCBR with the dialog learning process uses fewer dialog sessions to find the correct case in the second LOOCV cycles than in the first cycle ($6.87 < 8.10$), and the stored dialog cases in the second cycle are also fewer than in the first cycle ($285.28 < 614.75$).

To show how significant the experiment results support the hypothesis identified in last section, we carry out the hypothesis testing (one-tailed t-test with two related samples). The average values of column 'DLNLC1' and column 'DLNLC2', and that of column 'DLLC1' and column 'DLNLC2' for each dataset are calculated and taken as the hypothesis testing parameter for H1; The values in column 'DLLC2' and column 'DLLC1' are selected as the testing parameter for H2; And for H3, the values in column 'CasesC2' and column 'CasesC1' are used as the testing parameter. With the degree of freedom of 31 and the significance level of 0.01, we find out the critical value as 2.457. For the three hypotheses listed above, we get the t values as 5.23, 5.80, and 3.81, respectively. Since all the calculated t values are larger than the critical value, we reject all the null hypotheses and accept the three original hypotheses.

6 Conclusion

In this paper, we propose a dialog learning framework in CCBR, implement it, and evaluate it based on 32 datasets. The evaluation results give us significant evidence to support our hypotheses, that is, the dialog learning mechanism is effective and sustainable, and the dialog case base is maintainable.

Our conclusion is drawn based on the two cycles of LOOCV. A long term real human-subject based experiment would give us more solid evidence to our hypotheses. In addition, though the dialog learning enhanced CCBR stores fewer dialog cases in the second evaluation cycle than in the first, the size of the dialog case base is comparable to or even larger than the application case base, which demands a considerable memory space and CPU time to retrieve inside. We are now focusing on designing a better dialog learning strategy to retain fewer dialog cases without reducing the system efficiency significantly.

In a practical CCBR application, whether this dialog learning mechanism should be adopted depends on the tradeoff between the dialog efficiency improvement and the resource cost (both CPU time and memory space). In addition, if a knowledge-intensive question selection method [9] is used, in which discriminative questions are ranked also based on the semantic relations among them, more research is needed on how to combine the semantic question ranking with the question ranking priority adjustment in dialog case REUSE.

References

1. Kolodner, J.: Case-based reasoning. Morgan Kaufmann Publishers Inc. (1993)
2. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issue, methodological variations, and system approaches. AI Communications **7**(1) (1994) 39–59
3. Aha, D.W., Breslow, L., Muñoz-Avila, H.: Conversational case-based reasoning. Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies **14**(1) (2001) 9
4. Schmitt, S.: simvar: A similarity-influenced question selection criterion for e-sales dialogs. Artificial Intelligence Review **18**(3-4) (2002) 195–221
5. Shimazu, H.: Expertclerk: A conversational case-based reasoning tool for developing salesclerk agents in e-commerce webshops. Artificial Intelligence Review **18**(3-4) (2002) 223 – 244
6. Cunningham, P., Bergmann, R., Schmitt, S., Traphoner, R., Breen, S., Smyth, B.: Websell: Intelligent sales assistants for the world wide web. KI - Kunstliche Intelligenz **1** (2001) 28–31
7. Göker, M.H., Thompson, C.A.: Personalized conversational case-based recommendation. In: the 5th European Workshop on Case-Based Reasoning(EWCBR 2000), Trento, Italy (2000)
8. Aha, D.W., Maney, T., Breslow, L.: Supporting dialogue inferencing in conversational case-based reasoning. In: European Workshop on Case-Based Reasoning, Dublin, Ireland (1998) 262–273
9. Gu, M., Aamodt, A.: A knowledge-intensive method for conversational cbr. In Muñoz-Avila, H., Ricci, F., eds.: Case-Based Reasoning Research and Development, Proceedings of the 6th International Conference on Case-Based Reasoning. Volume 3620 of Lecture Notes in Artificial Intelligence., Chicago, Illinois, Springer Verlag (2005) 296–311 ISBN: 3-540-28174-6, ISSN: 0302-9743.
10. Salzberg, S.: A nearest hyperrectangle learning method. Mach. Learn. **6**(3) (1991) 251–276
11. Gu, M., Tong, X., Aamodt, A.: Comparing similarity calculation methods in conversational cbr. In Zhang, D., Khoshgoftaar, T.M., Shyu, M.L., eds.: Proceedings of the 2005 IEEE International Conference on Information Reuse and Integration,

Hilton, Las Vegas, Nevada, USA, IEEE Press (2005) 427 – 432 ISBN: 0-7803-9093-8.
12. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools with Java implementations. Volume 10 of Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers, Inc., San Francisco (1999)
13. Newman, D., Hettich, S., Blake, C., Merz, C.: Uci repository of machine learning databases [http://www.ics.uci.edu/ mlearn/mlrepository.html] (1998)

Paper G

Component Retrieval Using Knowledge-Intensive Conversational CBR

```
@INPROCEEDINGS{Gu2006b,
  author = {Mingyang Gu and Ketil B\o},
  title = {Component Retrieval Using Knowledge-Intensive
          Conversational CBR},
  booktitle = {Proceedings of the 19th International Conference
          on Industrial, Engineering and Other Applications
          of Applied Intelligent Systems},
  year = {2006},
  volume = {4031},
  series = {Lecture Notes in Artificial Intelligence},
  pages = {554 - 563},
  address = {Annecy, France},
  month = {27 - 30, June},
  publisher = {Springer Verlag},
}
```

Component Retrieval Using Knowledge-Intensive Conversational CBR

Mingyang Gu and Ketil Bø

Department of Computer and Information Science, Norwegian University of Science and Technology, Sem Saelands vei 7-9, N-7491, Trondheim, Norway
{mingyang, ketilb}@idi.ntnu.no

Abstract. One difficulty in software component retrieval comes from users' incapability to well define their queries. In this paper, we propose a conversational component retrieval model (CCRM) to alleviate this difficulty. CCRM uses a knowledge-intensive conversational case-based reasoning method to help users to construct their queries incrementally through a mixed-initiative question-answering process. In this model, general domain knowledge is captured and utilized in helping tackle the following five tasks: feature inferencing, semantic similarity calculation, integrated question ranking, consistent question clustering and coherent question sequencing. This model is implemented, and evaluated in an image processing component retrieval application. The evaluation result gives us positive support.

1 Introduction

Component retrieval, how to locate and identify appropriate components for current software development, is one of the major problems in component reuse [1]. This problem becomes more critical with the emergence of several component architecture standards, such as, CORBA, COM, DCE, and EJB. These standards make software components inter-operate more easily. Therefore component reuse surpasses the limitation of a single software company. Instead of getting components from an in-house component library, users search for desired components from component markets [2] (web-based software component collections provided by vendors or third parties), which separate component users and component developers from each other. In addition, a large and rapidly increasing number of reusable components put more strict demands on the retrieval efficiency [3].

Several methods have been put forward to address the component retrieval problem [4], such as the free-text-based retrieval method, the pre-enumerated vocabulary method, the signature matching method, the behavior-based retrieval method, and the faceted selection method. Most of them assume that users can define their component queries clearly and accurately, and get their desired components based on such well defined queries. However, before users know the components available for them to choose, they often lack clear ideas about what they need, and usually can not define their queries accurately. In addition, the huge number of available components prevents users from knowing all of them.

M. Ali and R. Dapoigny (Eds.): IEA/AIE 2006, LNAI 4031, pp. 554–563, 2006.

One promising solution to this problem can be that we invite an expert (or construct an intelligent system) who knows the characteristics of all the components. If one user needs a component, she can consult this expert. The expert extracts the requirement information from the user through conversation, and suggests appropriate components for her. Conversational case-based reasoning can be used to construct such an intelligent component retrieval system.

Case-Based Reasoning (CBR) is a problem solving method [5]. The main idea underlying CBR is that when facing a new problem, we search in our memory to find the most similar previous problem, and reuse the old solution to help solve the current problem. Conversational case-based reasoning (CCBR) [6] is an interactive form of CBR. It is proposed to deal with problems where users can not pose well defined queries (new cases) or where constructing well-defined new cases are expensive. CCBR uses a mixed-initiative dialog to guide users to facilitate the case retrieval process through a question-answering sequence. CCBR has been probed in several application domains, for instance, in the troubleshooting domain [7, 8], in the products and services selection [9, 10], and recently in workflow management [11].

In our research, we apply the CCBR method to software component retrieval, and propose a conversational component retrieval model (CCRM), where each component is described as a stored case, and a component query is formatted as a new case [4]. This CCRM model can help users construct their component queries incrementally through a dialog process, and find the appropriate components for them. In this paper, we identify six tasks in the component retrieval application, and extend the CCBR method to satisfy these identified tasks through incorporating general domain knowledge.

The rest of this paper is organized as follows. In Section 2, we present the framework of CCRM; in Section 3, comparing with the traditional CCBR process, a set of tasks are further identified in the software component retrieval application; in Section 4, we describe the design of CCRM focusing on how to solve the identified tasks; in Section 5, an implementation of CCRM is described and evaluated in an image processing software component retrieval application; at the end, related research is described and compared with our method in Section 6.

2 CCRM Overview

As illustrated in Fig. 1, the conversational component retrieval model (CCRM) includes six parts: a knowledge base, a query generating module, a similarity calculation module, a question generating and ranking module, a component displaying module, and a question displaying module.

The knowledge base stores both component-specific knowledge (cases) and general domain knowledge. After a user provides her initial requirement specification (arrow A), the query generating module uses it to construct a component query. Given a query, the similarity calculation module calculates the similarities between the query and each stored component, and returns a set of components whose similarities surpass a threshold (the threshold is pre-defined and can be

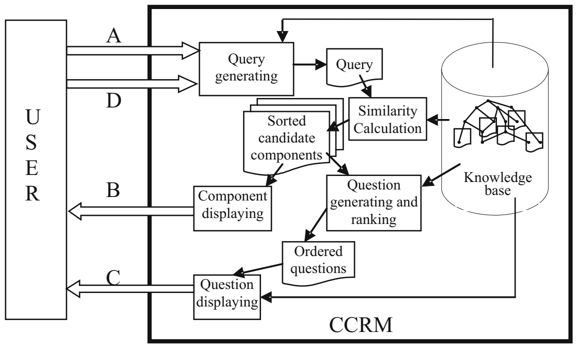

Fig. 1. The architecture of conversational component retrieval model (CCRM)

adjusted following the execution). In the question generating and ranking module, discriminative questions are identified from the returned components and ranked. The component displaying module displays the returned components, ordered by their similarities, to the user (arrow B). The question displaying module displays the ranked questions (arrow C). If the user finds her desired component in the displayed components, she can select it and terminate the retrieval process. Otherwise, she chooses a question, and provides the answer to the system (arrow D). Then the query generating module combines the previous query and the newly gained answer to construct a new query, and a new round of retrieving and question-answering is started until the user finds her desired component (success) or there are no questions left for her to choose (fail).

3 Requirements for Conversational CBR to Support Software Component Retrieval

3.1 Supporting Component Retrieval Using Generalized Cases

Most of the applications in CCBR assume that on each feature, there is either missing value or one discrete value (so called point cases, PC). However, for the cases used in CCRM (either a new case or a stored case), it is necessary to have multiple values on some features. The semantic for a stored case with multiple values for one feature is that the corresponding component has the capability to function in several situations, specified by multiple values for that feature. The multiple values on one feature in a new case means the user demands all the requirements specified by these values to be satisfied. The cases that can have multiple values on some features are named generalized cases (GC) [12]. In [13], we discussed how to support GCs in conversational CBR in a knowledge-poor context. In this paper, we will show how the GCs can be represented and utilized in a knowledge-intensive context.

To our knowledge, most of the applied CCBR methods are, to a large extent, knowledge-poor, that is, they only take the syntactical information or statistical metrics into account. The potential that general domain knowledge has for playing a positive role in the conversation process is little explored. In our research, we identify the following five tasks in CCRM, for which general domain knowledge is able to help controlling and improving the conversation process.

3.2 Feature Inferencing

In CCBR, the features that appear in the returned cases but not in the new case are selected and transformed into discriminative questions. However, if one feature can be inferred from the current features of a new case, this feature should be added to the new case automatically, instead of repeatedly inquiring it from the user. Users are likely not to trust a communicating partner who asks for information that is easy to infer, and the conversation efficiency will also be decreased by asking such "repeating" questions. Feature inferencing [14] is designed to extend a new case by adding the features that can be inferred by the current new case description.

3.3 Knowledge-Intensive Similarity Calculation

Selecting components based on their semantic similarities to user's query rather than syntactical similarities only is an active research topic [3, 15]. However, existing research concerned this topic mainly use domain knowledge to refine user's query before the searching process. In our research, besides the query refinement process (feature inferencing), we are using abductive inference [16] to exploit the general domain knowledge during the similarity calculation process. The similarity calculation process is divided into two steps: in the first step, similarities are calculated syntactically based on how high percentage of features specified in the query are matched by those in a component. In the second step, the abductive inference mechanism is adopted to exploit the general domain knowledge to construct the possible explanation paths trying to bridge the unmatched features [17].

3.4 Integrated Question Ranking

In CCBR, a main research topic is how to select the most discriminative questions and prompt them in a natural way to alleviate users' cognitive load. The feature inferencing process removes the questions that can be answered implicitly. Before the remaining questions are displayed to users, they need to be ranked intentionally. Currently, most of the question ranking metrics are knowledge-poor, for example, information metric, occurrence frequency metric, importance weight metric, and feature selection strategies [13]. The general domain knowledge, particularly the semantic relations between questions, can also be used to rank the discriminative questions. For example, if the answer to question B can be inferred from that of question A, or the answer to question A is easier or cheaper to obtain than that to question B, question A should be prompted to users before question B. In CCRM, an integrated question ranking method is designed, which uses not only the superficial statistical metrics of questions, but also the semantic relations among them.

Even though an integrated question ranking module outputs a set of sorted questions, their screen arrangement and questioning sequence should not be decided by such a sorted order alone. The main reason lies in that people always hope to inspect or answer questions in a natural way. They would prefer to see a set of questions, connected by some semantic relations, grouped together, and answered in an uninterrupted sequence. These requirements are captured by the following two tasks:

3.5 Consistent Question Clustering

The arrangement of questions on the screen should be consistent, that is, the questions with some semantic relations among them should be grouped and displayed together. For example, the questions having dependency relations among them should be grouped and displayed together. The order of the questions in each group should be decided intentionally.

3.6 Coherent Question Sequencing

The questions asked in the sequential question-answering cycles should be as related as possible, that is, the semantic contents of two sequential questions should avoid unnecessarily switching. For example, if in the previous question-answering cycle a more general question in an abstraction taxonomy is asked, the downward more specific question should be asked in the succeeding cycle rather than inserting other non-related questions between them.

4 CCRM Design

4.1 Knowledge Representation Model

In CCRM, knowledge is represented on two levels. The first is the object-level, in which both general domain knowledge and case-specific knowledge are represented within a single representation framework. The second is the meta-level, which is used to organize the semantic relations to complete the knowledge-intensive tasks identified above.

Object-Level Knowledge Model. A frame-based knowledge representation model, which is a part of the CREEK system [17], is adopted in CCRM. In this representation model, both case-specific knowledge and general domain knowledge are captured as a network of concepts and relations. Each concept and relation is represented as a frame in a frame-based representation language. A frame consists of a set of relationships, representing connections with other concepts or non-concept values, e.g. numbers. A relationship is described using an ordered triple $< C_f, T, C_v >$, in which C_f is the concept described by this relationship, C_v is another concept acting as the value of this relationship (value concept), and T designates the relation type. Viewed as a semantic network, a concept corresponds to a node and a relation corresponds to a link between two nodes.

Both a new case and stored cases are represented as concepts, and the features inside a case are represented as relationships starting from the concept representing this case. In CCRM, it is permitted for one case concept to have more than one of the same type of relationships in order to support generalized cases. The semantic relations among concepts are also represented using relationships, which can be used to support knowledge-intensive reasoning, for example, feature inferencing and semantic question ranking.

Meta-Level Knowledge Model. To organize general domain knowledge (semantic relations) to complete the knowledge-intensive tasks, we design a meta-level knowledge model. In this model, the semantic relations are defined as the subclasses of the meta-level relations, each of which corresponds to a knowledge-intensive task. So we only need to define the properties and operations once on a super-class meta-level relation, all its subclass semantic relations can inherit these properties and operations automatically. The separation of this meta-level representation model from the object-level model makes CCRM easy to be extended through introducing new semantic relations as the subclasses of some meta-level relations, and easy to be transplanted between different component retrieval application domains.

4.2 Explanation-Boosted Reasoning Process

An explanation-boosted reasoning process [14] is adopted in CCRM to complete the five knowledge-intensive tasks. This process can be divided into three steps: ACTIVATE, EXPLAIN and FOCUS. These three steps, which constitute a general process model, were initially described for knowledge-intensive CBR [17]. Here this model is instantiated for the identified five knowledge-intensive CCBR tasks. ACTIVATE determines what knowledge (including case-specific knowledge and general domain knowledge) is involved in one particular task, EXPLAIN builds up explanation paths to explore possible knowledge-intensive solutions for that task, and FOCUS evaluates the generated explanation paths and identify the best one/ones for that particular task.

5 Implementation and Evaluation

5.1 CCRM Implementation

We have implemented CCRM within the TrollCreek system [17]. TrollCreek is a knowledge-intensive case-based reasoner with a graphical knowledge model editor, where the knowledge-intensive similarity calculation has been realized. Our implementation adds the conversational process into the retrieval phase, and extends it to support generalized cases and complete the other four knowledge-intensive tasks.

In this implementation, a conversational retrieval process contains one or several conversation sessions. As illustrated in Fig. 2, in the computer interface there

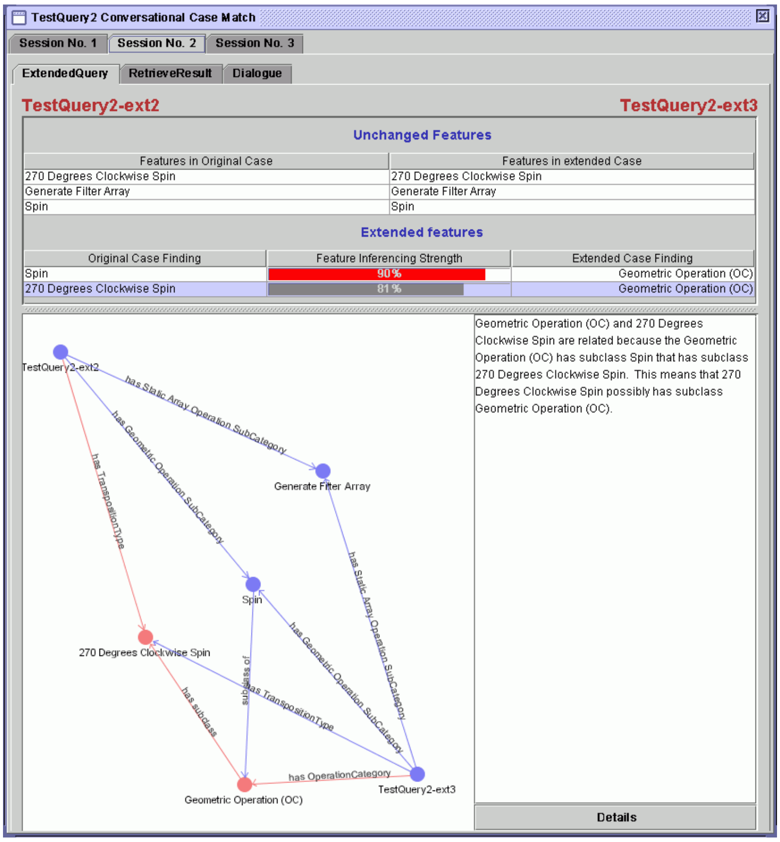

Fig. 2. The conversational retrieval process implemented in TrollCreek

are three window panes to move between within each session. The "Extended-Query" pane is used to display the original query and the extended query, and show the detailed explanation about how a feature is inferred from the original features. Based on the extended query, the similarity calculation module retrieves a set of components, and displays them in the "RetrieveResult" pane. In this pane a user can inspect the explanations about how the similarity is computed between each retrieved component and the extended query. If the user is not satisfied with the retrieved components, she can go to the "Dialogue" pane, where the discriminative questions are ranked using the integrated question ranking process, and adjusted by the consistent question clustering and the coherent question sequencing processes. After the user selects a discriminative question and submits the answer, a new conversation session is started based on a constructed new query through combining the provided answer with the previous query.

5.2 Evaluation

We choose image processing software component retrieval, particularly the components in the DynamicImager system [4], as the evaluation application.

DynamicImager is a visualization and image processing development environment, in which different image processing components can be combined in various ways. Currently, the components in the system are categorized according to their functions, and users select each component by exploring through the category structure manually.

A knowledge base is constructed by combining the image processing domain knowledge and 118 image processing components extracted from DynamicImager. In this knowledge base, there are 1170 concepts, 104 features and 913 semantic relationships.

For the evaluation of CCRM, we choose a relatively weak evaluation method, so called direct expert evaluation [18]. We invited two experts from the image processing domain and two experts from the software engineering domain to test our system. Given a set of image processing tasks, these domain experts were asked to retrieve image processing components using both a one-shot CBR-based retrieval method and the multiple shots knowledge-intensive CCBR based method (CCRM). After that, they were required to fill in a form to describe their subjective evaluation of the implemented system. The resulting analysis of the collected feedback forms shows us that:

- Based on the same initial new case, the CCRM method can achieve more useful results;
- The reasoning transparency provided by the explanation mechanisms in CCRM improves users' confidence in the retrieved results;
- The feature inferencing, consistent question clustering and coherent question sequencing mechanisms provide users' with a natural question-answering process, which helps to alleviate their cognitive loads in retrieving components interactively;
- The straight-forward question-answering query construction process is able to reduce users' cognitive load to guess the query, and help users with limited domain knowledge to retrieve the suitable components.

6 Related Research and Conclusion

Software is used to solve practical problems, and software components are existing solutions to previous problems, so component reuse can be described as "trying to use the solutions to previous similar problems to help solving the current problem". Therefore, it is very natural to use the CBR method to support component reuse. Various types of CBR methods have been explored and found useful for component reuse.

Object Reuse Assistant (ORA) [19] is a hybrid framework that uses CBR to locate appropriate components in an object-oriented software library (small-talk component library). In this framework, both small-talk classes and small-talk methods take the form of stored cases. The concepts in small-talk, for instance, c-class, c-method and c-data-spec, and their instantiated objects are connected together as a conceptual hierarchy. Though the conceptual hierarchy can be seen as a representation method combining case-specific knowledge and general knowledge, the retrieval process is knowledge-poor (a new case is compared with stored cases based on how many attributes two cases have in common).

IBROW [20] is an automated software configuration project. Users' tasks (queries) can be decomposed into sub-tasks by matched task decomposers, and sub-tasks can be decomposed further. Tasks or subtasks can finally be solved by matched software components. Both task decomposers and components are referred to as PSMs (problem solving methods). CBR is used at two levels in IBROW. The high level is called constructive adaptation. In this level, PSMs take the form of cases, which are represented using feature terms, and a knowledge-poor matching method (term subsumption) is adopted when searching the possibly applied PSMs. At the low level, CBR is used as a heuristic algorithm to realize the best-first searching strategy. Previously solved configurations are stored as cases, and represented as feature terms. In an intermediate stage of a configuration task, for each possible further configuration, C, the PSM, through applying which C is produced, is considered. The stored configurations in which the same PSM appears as a part are identified, and the similarities between each of these configurations and the semi-finished configuration C are calculated. The most similar configuration is selected, and its similarity value is taken as the heuristic value for this PSM to be applied. As the ORA system, IBROW uses a knowledge-poor retrieval process and only supports tentative and manual interactions between users and the system.

Comparing with these two CBR-based component retrieval systems, CCRM has two advantages: providing a conversational process helping users to construct their component queries incrementally and find out their desired component at the same time; providing integrated knowledge-intensive solutions to identified knowledge-intensive tasks: feature inferencing, knowledge-intensive similarity calculation, integrated question ranking, consistent question clustering and coherent question sequencing.

A limitation of our method is its dependence on knowledge engineering. The knowledge base combining both component specific knowledge and general domain knowledge is assumed to exist initially. The construction of this knowledge base puts a significant workload on the knowledge engineering process.

Our future work focuses on integrating this CCRM into the DynamicImager system to help users constructing their queries and finding out their desired components through a conversation process instead of manually searching through the categories.

References

1. Mili, A., Mili, R., Mittermeir, R.: A survey of software reuse libraries. Annals of Software Engineering **5** (1998) 349 – 414
2. Ravichandran, T., Rothenberger, M.A.: Software reuse strategies and component markets. Communications of the ACM **46** (2003) 109 – 114
3. Klein, M., Bernstein, A.: Searching for services on the semantic web using process ontologies. In: The First Semantic Web Working Symposium, Stanford, CA, (2001).

4. Gu, M., Aamodt, A., Tong, X.: Component retrieval using conversational case-based reasoning. In : Procedings of International Conference on Intelligent Information Processing, (2004).
5. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issue, methodological variations, and system approaches. AI Communications **7** (1994) 39–59
6. Aha, D.W., Breslow, L., Muñoz-Avila, H.: Conversational case-based reasoning. Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies **14** (2001) 9
7. Gupta, K.M.: Knowledge-based system for troubleshooting complex equipment. International Journal of Information and Computing Science **1** (1998) 29–41
8. Cunningham, P., Smyth, B.: A comparison of model-based and incremental case-based approaches to electronic fault diagnosis. In: Case-Based Reasoning Workshop, Seattle, USA (1994)
9. Cunningham, P., Bergmann, R., Schmitt, S., Traphoner, R., Breen, S., Smyth, B.: Websell: Intelligent sales assistants for the world wide web. KI - Kunstliche Intelligenz **1** (2001) 28–31
10. Shimazu, H.: Expertclerk: A conversational case-based reasoning tool for developing salesclerk agents in e-commerce webshops. Artificial Intelligence Review **18** (2002) 223 – 244
11. Weber, B., Rinderle, S., Wild, W., Reichert, M.: Ccbr-driven business process evolution. In: Case-Based Reasoning Research and Development, Proceedings of the 6th International Conference on Case-Based Reasoning, (2005).
12. Maximini, K., Maximini, R., Bergmann, R.: An investigation of generalized cases. In: 5th International Conference on Case-Based Reasoning, (2003), 261 – 275
13. Gu, M.: Supporting generalized cases in conversational cbr. In : Proceedings of the Fourth Mexican International Conference on Artificial Intelligence, (2005).
14. Gu, M., Aamodt, A.: A knowledge-intensive method for conversational cbr. In : Case-Based Reasoning Research and Development, Proceedings of the 6th International Conference on Case-Based Reasoning, (2005).
15. Sugumaran, V., Storey, V.C.: A semantic-based approach to component retrieval. The DATA BASE for Advances in Information Systems **34** (2003) 8–24
16. Öztürk, P.: Abductive inference - an evidential approach. In: A knowledge level model of context and context use in diagnositic domains - Doctoral Thesis. Norwegian University of Science and Technology (2000) 49 – 60
17. Aamodt, A.: Knowledge-intensive case-based reasoning in creek. In : Proceedings of the 7th European Conference on Case-Based Reasoning. Madrid, Spinger (2004)
18. Cohen, P.R., Howe, A.E.: How evaluation guides ai research. AI Mag. **9** (1988) 35–43
19. Fernández-Chamizo, C., González-Calero, P.A., Gámez-Albarrán, M., Hernández-Yáñez, L.: Supporting object reuse through case-based reasoning. In: EWCBR '96: Proceedings of the Third European Workshop on Advances in Case-Based Reasoning, London, UK, Springer-Verlag (1996) 135–149
20. IBROW-project: http://www.swi.psy.uva.nl/projects/ibrow/home.html (2005)

Paper H

Evaluating CBR Systems Using Different Data Sources: A Case Study

```
@INPROCEEDINGS{Gu2006c,
  author = {Mingyang Gu and Agnar Aamodt},
  title = {Evaluating CBR Systems Using Different Data
          Sources: A Case Study},
  booktitle = {accepted by the 8th European Conference
          on Case-Based Reasoning},
  year = {2006},
  volume = {4106},
  series = {Lecture Notes in Artificial Intelligence},
  pages = {121 - 135},
  address = {Fethiye, Turkey},
  month = {4 - 7, September},
  publisher = {Springer Verlag},
}
```

Evaluating CBR Systems Using Different Data Sources: A Case Study

Mingyang Gu and Agnar Aamodt

Department of Computer and Information Science, Norwegian University of Science and Technology, Sem Saelands vei 7-9, N-7491, Trondheim, Norway
{mingyang, agnar}@idi.ntnu.no

Abstract. The complexity and high construction cost of case bases make it very difficult, if not impossible, to evaluate a CBR system, especially a knowledge-intensive CBR system, using statistical evaluation methods on many case bases. In this paper, we propose an evaluation strategy, which uses both many simple case bases and a few complex case bases to evaluate a CBR system, and show how this strategy may satisfy different evaluation goals. The identified evaluation goals are classified into two categories: domain-independent and domain-dependent. For the evaluation goals in the first category, we apply the statistical evaluation method using many simple case bases (for example, UCI data sets); for evaluation goals in the second category, we apply different, relatively weak, evaluation methods on a few complex domain-specific case bases. We apply this combined evaluation strategy to evaluate our knowledge-intensive conversational CBR method as a case study.

1 Introduction

As summarized in [1], AI research is an empirical process: selecting a task incorporating intelligence features, building a system exhibiting these features, and evaluating the system in different task environments. After an intelligent system is constructed, it is necessary to evaluate whether it does what we expect it to do and how good its performances is. Cohen and Howe [2] extend the importance of evaluation from assessing the performance to guiding the different AI research phases.

Evaluation methods for intelligent systems include statistical evaluation (inductive evaluation), theoretical analysis, ablation evaluation, tuning evaluation, limitation evaluation, direct expert evaluation and characteristic analysis [3,4,5]. The ideal evaluation method among them is statistical evaluation; that is, to execute the constructed system in different task environments in order to investigate its performance in different application contexts. This method is difficult to apply, in general, to case-based reasoning (CBR) [6,7] because of the typical complexity of CBR applications. The complexity comes from two aspects [8]: the CBR system itself is complex, and the task domain where it operates is also typically complex and ill-structured. The complexity of the application domain makes it difficult and expensive to construct a case base, especially for

T.R. Roth-Berghofer et al. (Eds.): ECCBR 2006, LNAI 4106, pp. 121–135, 2006.

knowledge-intensive CBR systems [9] that demand a significant knowledge engineering effort. Because of the complexity and heterogeneity of CBR systems, transplanting a case base from one CBR system to another also needs considerable adaptation work. Therefore, it is very hard to construct or transplant a number of complex case bases to use in a statistical evaluation. For these reasons, the evaluation of a CBR system is, to a large extent, based on one or a few case bases, which can only provide limited evidence.

Aha [10] provides a method to generalize the evaluation result of an AI system, which is based on one (or a few) data sets. In this method, a set of dimensions are identified to describe the original data set, and a data set generator is created to produce many artificial data sets with predefined values on the identified dimensions. The target system is executed on the artificial data sets, and its performances are recorded. The relations between differences of the system performance and changes of the dimension values are studied, and a set of rules are generated to describe the conditions under which the performance differences hold. Applying this method into CBR researches needs substantial efforts since it is difficult to artificially construct a set of complex case bases with the predefined dimension values.

When we look into the details of the evaluation process for CBR research, we find that there are usually multiple evaluation goals. For instance, this includes the efficiency of the similarity calculation method, the validity of the adaptation method, the problem solving efficiency on the target application domain, the usability or human friendliness, and the individual contributions of various system components. Further, different evaluation goals are related to different application scopes. Some goals are domain-dependent; that is, they need to be evaluated on the target specific application domain, for example to determine whether the general domain knowledge can improve, for instance, the similarity matching using a knowledge-intensive method [11], or make an explanation to the user more understandable [12]. Other goals are domain-independent, for instance whether the sustained learning process in CBR can improve problem solving capability. For the domain-independent goals, we can evaluate them on either complex case bases or simple case bases. There are plenty of such simple case bases, for instance, the data sets available at the UCI repository [13], and there are many examples of research contributions evaluated by this data sets within CBR community [14,15,16,17].

We propose an evaluation strategy for CBR research aiming to assess these two types of evaluation goals (domain-dependent and domain-independent) based on different data sources and using different evaluation methods. For the domain-independent evaluation goals, we use the statistical evaluation over many simple data sets, while domain-dependent goals are evaluated on one or a limited number of complex case bases using multiple weak evaluation methods. That is, this strategy combines a statistical evaluation method with many simple case bases, and alternatively combines limited number of complex case bases with multiple weak evaluation methods. This evaluation strategy can provide solid evidence for both the domain-independent goals and the domain-dependent goals. For the domain-independent goals, the evaluation power comes from the statistical justification. For the domain-dependent goals, the solidity comes from whether all the multiple weak evaluation methods can output positive outcomes.

As part of our current research, we have designed and implemented a knowledge-intensive conversational case-based reasoning (KI-CCBR) system which can capture and utilize general domain knowledge to support an efficient and natural conversation process to complete the case retrieval task. In this paper, we use our proposed evaluation strategy to evaluate this KI-CCBR method as a case study.

In the next section, we give a short introduction to the evaluation methods we have used. In Section 3, we briefly introduce our KI-CCBR method and identify the relevant evaluation goals. In Section 4, we report how we use 36 UCI data sets to show that the two domain-independent evaluation goals, lazy dialog learning and query-biased similarity calculation, can improve conversation efficiency of CCBR in general. We also evaluate the KI-CCBR method on a case base of image processing software components, within a system designed to support component reuse in software design. Three different evaluation methods are used: a characteristic analysis is used to see whether the system meets the requirements of a conversational diagnosis system; a direct domain expert assessment is used in order to see whether the KI-CCBR method can provide a natural conversation process; and a simulated ablation study is adopted to evaluate whether KI-CCBR can improve the conversation efficiency and how much each knowledge-intensive module contributes to the total improvement. We conclude in Section 6.

2 Introduction to the Evaluation Methods

The purpose of an evaluation process is to assess a system, with reference to some selected baseline, to see whether the performance of the system is accepted or improved. In this section, we introduce the evaluation methods used in our study.

2.1 Statistical Evaluation (Inductive Evaluation)

The basic statistical evaluation process is one in which we define one or more performance measures, execute both the new system and the baseline system on many different data sets, and calculate the percentage of the data sets on which the new system gives better performance, or test statistical significance in relation to predefined hypotheses. Statistical evaluation is a proper method to support the claim of generality of a system's benefits or advantages. This method is a strong evaluation method and is frequently used in many scientific disciplines such as psychology or biology. Cohen [4] gives detailed information about how to apply this evaluation method for AI research.

2.2 Characteristic Analysis

For a certain type of intelligent system, what characterizes it are usually discussed and gradually agreed upon by researchers in that field. Analyzing whether and to what degree a system can support the relevant characteristics is one approach to evaluate the system with respect to its possible performance. For example, if a CBR system support all the four 'RE-' phases [6], we may claim that this system is a full-cycle CBR system.

2.3 Direct Expert Evaluation

When a test system can produce more acceptable solutions than we can possibly generate beforehand as a baseline [3], or the evaluation measures involve human common sense or psychology aspects, one method is to invite domain experts to use the system and report back their subjective assessments. This is a weak evaluation method because of experts' overly generosity and their unrepresentativeness of typical users. One way to balance this shortcoming is to select experts using different criteria, or experts from different related domains.

2.4 Ablation (Lesion, Substitution) Evaluation

Ablation evaluation [2,8] is a method for analyzing the contributions of different modules of a system to the total performance improvement. In this type of evaluation, one or more modules are de-activated, removed or replaced by other comparable modules to observe changes on system performance. This method was used to evaluate the PROTOS system [18] and the SIROCCO system [19]. One difficulty in applying this evaluation method is that it is not always feasible to remove or de-active particular modules in a system because of the interdependence among modules.

3 Knowledge-Intensive Conversational Case-Based Reasoning

3.1 Research Overview

Conversational case-based reasoning (CCBR) [20] is a special type of CBR, which emphasizes the difficulty to appropriately describe a new problem, i.e. to define a new case. CCBR alleviates it through providing a mixed-initiative interactive process to guide users to incrementally construct a new case description that is sufficient to complete the case retrieval task.

In CCBR, an initial new case (only one or few features) is specified and used to retrieve a set of most similar cases from the case base. A group of discriminative questions are identified based on the returned cases (transformed by the features that have values in the returned cases but not in the current new case), and ranked according to their capabilities to discriminate the stored cases. Both the returned cases, sorted according to their similarity values, and the ranked questions are displayed to the user. The user either finds a satisfactory stored case, which then terminates the case retrieval phase, or chooses a question to answer. The newly gained answer and the current new case are combined together to construct an updated new case. A new round of retrieval and question-answering is started, and this continues until the user finds a satisfactory stored case or there are no discriminative questions left for the user to choose.

A major research concern in CCBR is how to select the most discriminative questions [14,21] and ask them in a natural way [20,22,5] to alleviate users' cognitive load demanded in the conversation process. Most of the methods used to select questions now are knowledge-poor (KP); that is, only statistical metrics are used. In our research, we study the possibility of using general domain knowledge in the conversation process [23]. We identify the following four tasks for which general domain knowledge can be used to improve the conversation process:

- **Feature Inferencing:** The features that can be inferred from the current new case description should be added into the new case description, instead of posting users questions.
- **Knowledge-Intensive Question Ranking:** The semantic relations among discriminative questions should be taken into account during question ranking. For instance, if one answer of question A, A_a, can be inferred out by one answer of question B, B_a, question B should be asked before question A.
- **Consistent Question Clustering:** The questions that are connected by some semantic relations, for example, a causal relation or subclass relation, should be grouped and displayed together, so that users can inspect them together and select which one to answer first.
- **Coherent Question Sequencing:** If a question from a higher level node in a taxonomic structure is asked in the current question answering cycle, the question one level lower should be asked in the next cycle, instead of inserting other unrelated questions between them.

We classify similarity calculation methods in CBR into three categories, according to the scope of the features that are taken into account during similarity calculation:

- **Query-Biased Similarity Methods:** Only the features appearing in the current new case (query) are taken into account during similarity calculation.
- **Case-Biased Similarity Methods:** Only the features appearing in the current stored case are considered during similarity calculation.
- **Equally-Biased Similarity Methods:** All the features appearing in both the current new case and the current stored case are taken into account during similarity calculation.

We emphasize the special characteristic of the new case, partially specified, in CCBR. If the features which have not yet been assigned values in the new case, are considered in the similarity calculation, the similarity method will be biased to those cases with fewer such features, instead of to those that most satisfy the current new case (users' attention focus). So in order to avoid the negative influence of these features, we argue that the query-biased similarity calculation method is more suitable for CCBR than the case-biased or equally-biased similarity calculation methods [24].

In addition, we introduce a lazy dialog learner into CCBR [25], which is capable of capturing and storing previous successful conversational case retrieval processes and reusing them in the later conversational case retrieval tasks.

The implemented KI-CCBR method has been recently tested in an image processing software component retrieval application [26].

3.2 Identified Evaluation Goals

As discussed in Section 1, the evaluation goals of this KI-CCBR method are classified into two categories. The first category contains the evaluation goals that are valid for CCBR research in general; that is, domain-independent: whether the query-biased similarity calculation method and the lazy dialog learner can improve the efficiency of CCBR. The second category includes the evaluation goals that rely on a specific application domain, the image processing component retrieval application. This includes whether the KI-CCBR method meets the requirements of a conversational diagnosis system, whether the KI-CCBR method can provide users with a natural question answering process, whether the KI-CCBR method can achieve higher efficiency compared to the knowledge-poor CCBR method, and how the different knowledge-intensive modules contribute to the total achievement.

In Section 4 and Section 5, we will report how we choose different evaluation methods and test case bases for the identified evaluation goals.

4 Statistical Evaluation on UCI Data Sets

In an attempt to evaluate whether the query-biased similarity calculation method and the lazy dialog learner can improve the efficiency of CCBR in general, we choose the statistical evaluation method to see whether these methods can achieve higher efficiency than their competitors on multiple simple case bases.

In order to evaluate which similarity calculation method (query-biased, case-biased, or equally-biased) is more suitable for CCBR, we implement three variants of CCBR within Weka [27], each of which uses a particular similarity calculation method. In order to evaluate whether the dialog learning mechanism can improve the conversation efficiency, we implement two more variants of CCBR also within Weka, one of them using our dialog learning mechanism and the other not. We summarize the statistical evaluation to these two topics in this paper, and more detailed information can be found in our earlier studies [24,25].

The simple case bases we test are 36 classification data sets[1] provided by Weka, originally from the UCI repository [13]. Some of these case bases have been used to test conversational CBR methods in [14,16,17]. Aha, McSherry and Yang [28] argued that the typical case bases in CCBR applications are irreducible and heterogeneous. From our perspective, it is not necessary for case bases in CCBR to have these characteristics. For instance, in one typical CCBR

[1] For the evaluation of the lazy dialog learner, we drop off the 4 biggest case bases simply because they need too much execution time.

application domain, fault diagnosis, it is natural for two types of faults to share the same solution, which means the case base is reducible. Heterogeneity is only the characteristic of one type of case bases in CCBR, which is the necessary condition to apply the occurrence-frequency metric [20] in discriminative question selection. However, the entropy based question selection methods, which are adopted by more CCBR researches [29,21,30], require all the cases having the same structure (homogeneous).

The human-computer conversation process is simulated using leave-one-out cross validation (LOOCV). LOOCV is an extreme variant of K-fold cross validation, which splits the entire n cases in one case base into n subsets, each containing only one case. In each evaluation cycle of LOOCV, the test case, q, is taken as a description of a new problem, referred to as the target case. Before the retrieval starts, a part of the problem description of q, a subset of the $< feature, value >$ pairs (10%), is taken out to construct an initial new case. This initial new case is used for retrieval from the test case base containing the remaining cases in the original case base. If the base case, with respect to the target case, is returned as the most similar case, or is in the returned most similar case group, the retrieval process is terminated successfully. Otherwise, the question generating and ranking module will output a set of sorted discriminative questions. A predefined question selection strategy is used to select a question from the discriminative question list, for example selecting the first question. The $< feature, value >$ pair corresponding to the selected question is chosen from the target case q and added into the current new case to form an updated new case. Based on the updated new case, a new round of retrieval is started. The retrieval, question selection and answering process will continue until the successful condition or failed condition (there are no $< feature, value >$ pairs left to choose) is met.

The average session number of the conversations simulated by the total set of cases in one case base is taken as the main criterion to assess the performance of a CCBR method on that case base [20,14].

The successful termination condition of LOOCV is that the base case appears in the first returned case group (k cases). If the query biased similarity method is used, especially in the beginning phase of the retrieval process, the number, m, of the cases that exactly match the partially-specified new case (and are thus equally similar) may be larger than k. In this situation, the simulated process randomly returns k out of them. This setting may be arbitrary. Ferguson and Bridge [31] suggest a method to abandon exact similarities in favor of preference relations between cases. In our case, the successful termination condition is acceptable since the final statistical result is computed from multiple cases and case bases using the same successful condition.

For the evaluation of the similarity calculation methods, in 31 out of total 36 case bases, the CCBR using query-biased similarity method gets better performance than the other two methods (case-biased and equally-biased similarity methods). For the assessment of the lazy dialog learner, in 29 out of total 32 case bases, the CCBR process with the lazy learner gets better performance than that without the learner. In this experiment, we execute the LOOCV two rounds with the aim to evaluate the ability of the lazy learner to learn in a long term. The results show that the lazy dialog learner is sustainable and the dialog case base is maintainable; that is, with the dialog learning going on, the dialog learner achieves better performance and the lazy learner requires fewer dialog cases to be stored in the dialog case base. For all the above comparisons, we have carried out the significance tests on the tested case bases (t-test with the significance level 0.01), and the results give us supportive evidence (all the observed differences in performance are significant).

5 Evaluating the KI-CCBR Method on an Image Processing Software Component Retrieval Application

We have implemented our KI-CCBR within the CREEK system [26][2]. We choose image processing software component retrieval, exemplified by retrieving components from the DynamicImager system [32], as the evaluation domain to assess the domain-dependent evaluation goals. DynamicImager is an image processing development and visualization environment, in which different image processing components can be combined in various ways. Currently, the components in the system are categorized according to their functions, and users select each component by exploring the category structure manually. A knowledge base has been constructed through combining image processing domain knowledge with 118 image processing components extracted from DynamicImager. In this knowledge base, there are 1170 concepts, 104 features and 913 semantic relationships, using approximately 20 relation types (e.g. has subclass, has part, causes).

As illustrated in Fig. 1, a conversational retrieval process contains one or several conversation sessions, and for each session, there are three window panes to move between in the computer interface. The ExtendedQuery pane is used to show how a new case is extended through feature inferencing, and to display a detailed explanation of why a new feature is added into the case. Based on the extended new case, a set of stored cases are retrieved and displayed in the RetrieveResult pane. In this pane the user can inspect the explanations about how the similarity values are computed. If a user is not satisfied with the retrieved cases, she can go to the Dialogue pane, where the discriminative questions are ranked using both the knowledge-intensive question ranking method and statistical metrics, and adjusted by the consistent question clustering and coherent question sequencing processes. After the user selects a discriminative question and submits her answer, a new conversation session is started based on the updated new case by combining the newly gained answer with the previous new case.

5.1 Characteristic Analysis as a Sequential Diagnosis System

The CCBR process is basically a sequential diagnosis process: as more and more problem features (evidence) are identified and added into the new case, the system can identify the correct diagnosis (the base case) with more confidence.

[2] The dialog learning mechanism is not implemented in our KI-CCBR.

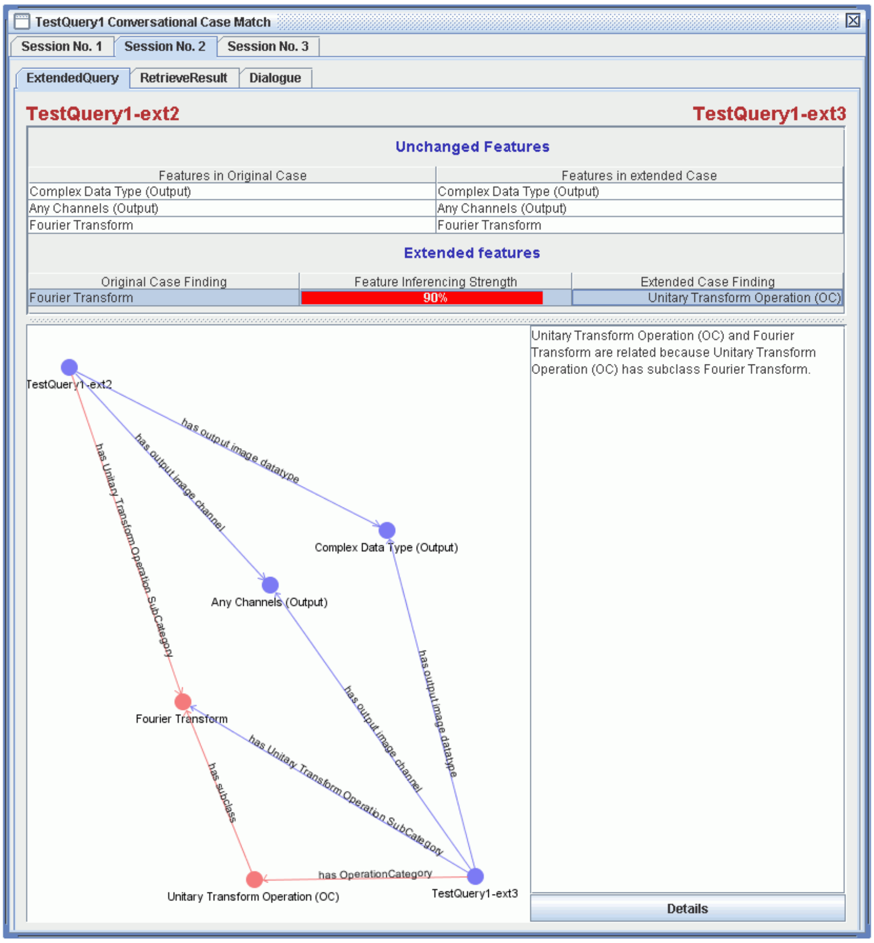

Fig. 1. The conversational case retrieval process in our KI-CCBR

McSherry [5] identifies seven desirable features (or characteristics) of an intelligent sequential diagnosis system. Our implemented system supports all of these characteristics.

– **Mixed-Initiative Dialogue:** Users, particularly professional users, are unlikely to accept a conversation partner (or intelligent system) who keeps asking a lot of questions. Instead, users prefer playing a positive role in the conversation, i.e. to control the conversation or to volunteer data at any stage of the conversation. Permitting users to select a question to answer from a list is a form of mixed-initiative dialogue which is supported by most CCBR applications. In addition, our method allows users to cancel or modify their answered questions (go to the specific session window and remove or reselect the answered entities). Furthermore, our method clusters related questions together, so that users can inspect the questions with different difficulty levels, and select one to answer according to their expertise levels.

- **Feedback on the Impact of Reported Evidence:** It is unacceptable if users get no feedback from an expert (or intelligent system) after they provide more evidence. In our method, after the user answers a question, or modifies the initial new case or previous answered questions, the case retrieval process and the question ranking, sequencing, clustering processes will run immediately. The returned cases and discriminative questions will be based on the updated evidence.
- **Relevant Dialogue:** The questions asked by an intelligent partner should be relevant to the context of the problem provided by the user. We assume that only the features appearing in the most similar cases are relevant. Therefore, our method generates discriminative questions based only on the most similar cases, instead of all the cases in the case base.
- **Consistent Dialogue:** The questions that can be answered implicitly by the current partially specified new case should not be prompted again. Otherwise, the conversation efficiency is reduced, and users are unlikely to trust a conversation partner that repeats previously implicitly answered questions. Furthermore, if a user provides an answer to a question that is not consistent with that inferred from the current new case, the content of the new case is not consistent any more. The feature inferencing process in our method guarantees that this type of dialog inconsistency will not occur, by ensuring that these types of questions will not be asked.
- **Explanation of Reasoning:** In order to improve users' confidence in the results of an intelligent system, it is important to provide an explanation of how results are derived [33,34,5,12]. Our KI-CCBR method provides the following explanations: why a new feature is added into the current new case description through feature inferencing, why two different feature values are partially matched through knowledge-intensive case matching [11], why a question is ranked with highest priority in the coherent question sequencing, and why two questions are grouped and displayed together through consistent question clustering.
- **Tolerance of Missing Data:** Missing data stem from two aspects. First, the cases in the case base may contain missing features. Our system's partial matching process can tolerate this type of missing data. We adopt the occurrence-frequency metric [20] as the knowledge-poor question ranking method, which basically takes the advantage of the presence of missing features. In addition, our explanation-driven reasoning process [23] exploits general domain knowledge, which may itself be incomplete. Another source of missing data is the user's incapability to answer every question due to the unavailability of some observations, the user's lack of expertise, or need for an expensive test to obtain the answer. Our method tolerates this type of missing data through permitting the user to choose candidate questions to answer, instead of forcing her to answer them in a fixed sequence.
- **Sensitivity Analysis:** The uncertainty that is inherent in the dialogue process, as well as the possible uncertainty in the user's answers to questions, means that support for sensitivity analysis is essential. Our method supports sensitivity analysis through allowing users to modify previously speci-

fied features (answered questions) and re-execute the retrieval and question-answering process in order to inspect the possible influences of the updated information.

5.2 Domain Expert Evaluation of the Psychological Goals

Evaluation Goals of KI-CCBR related to psychology include the user's cognitive load, the 'natural' question-answering process, and the user's confidence in the final results. We adopt a relatively simple or weak evaluation method, a so-called direct expert evaluation [2], for these evaluation goals.

We invited two experts from the software engineering domain, and two experts from the image processing domain, to test our system. Given a set of image processing tasks, these domain experts were asked to retrieve image processing components using both a one-shot CBR-based retrieval method and the multiple shots KI-CCBR method. After doing so, they were required to fill in a form to describe their subjective evaluation of the implemented system[3]. The resulting analysis of the collected feedback forms suggests that:

- Based on the same initial new case, the KI-CCBR method can achieve more useful results;
- The reasoning transparency provided by the explanation mechanisms in KI-CCBR improves user confidence in the retrieved results;
- The feature inferencing, consistent question clustering and coherent question sequencing mechanisms provide users with a natural question-answering process, which helps to alleviate their cognitive loads in retrieving components interactively;
- The straightforward question-answering query construction process helps to reduce users' cognitive load in constructing queries, thus enabling users with limited domain knowledge to retrieve suitable components.

5.3 Ablation Evaluation Using Leave-One-In Cross Validation

In order to show that the KI-CCBR method does improve the conversation efficiency by reducing the length of conversation sessions compared to knowledge-poor CCBR, we execute another cross validation on the image processing component retrieval application. Unlike the LOOCV we introduced in Session 4, we adopt leave-one-in cross validation (LOICV) to simulate the human-computer conversation. The difference between them is that, in LOOCV, the test case (target case) is taken away from the case base during the case retrieval process, while in LOICV, the test case is kept in the case base, and acted as the base case for the simulated retrieval process[4]. The LOICV has been successfully used in the CCBR community [20,22].

---

[3] The hypotheses list and the feedback form can be found at http://www.idi.ntnu.no/∼mingyang/research/CCRM_Evaluation.pdf

[4] The query-biased similarity method ensures that the test case is always included in the case group with highest similarity value, so the successful termination condition in LOICV, unlike that in LOOCV, is that the case group with the highest similarity value only contains the test case itself.

The reason why we switch from LOOCV to LOICV lies in that:

– In the UCI case bases we use in LOOCV, many of the cases in a case base have the same solutions, so we can evaluate variant CCBR applications in a classification context. In this context, we can choose a case, which shares the same solution as the target case, as the base case of the target case. That is, it is possible to execute a simulated CCBR retrieval with the target case out of the case base.
– In the image processing software component case base, each software component has a unique solution (i.e., the software component itself). McSherry [21] refers to a case base with this property as an *irreducible* case base. The component retrieval problem is basically an identification problem rather than a classification problem. It is impossible to carry out a simulated CCBR retrieval with the target case being removed from an irreducible case base, as its unique solution is no longer represented in the case base.

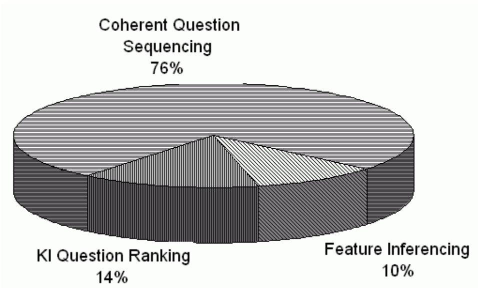

Fig. 2. Relative contribution of each KI-CCBR module to overall improvement in conversation efficiency

In our KI-CCBR method, if we disable the four knowledge-intensive modules, Feature Inferencing, Knowledge-Intensive Question Ranking, Consistent Question Clustering, and Coherent Question Sequencing, the system becomes a knowledge-poor CCBR system (use only the statistical metric (occurrence frequency) to rank questions). Instead of enabling all these four modules at the same time, we enable them in a sequence of Feature Inferencing, Knowledge-Intensive Question Ranking, and Coherent Question Sequencing[5], respectively. With the above module enabling sequence, the average conversation session numbers needed to find the base case are 3.70, 3.64, 3.56, and 3.12, respectively, the latter with all modules enabled. That is, our knowledge-intensive CCBR method improves the efficiency by using 16% fewer conversation sessions (questions) to find the base case compared with the knowledge-poor CCBR method. Fig. 2 shows us that the relative improvements from Feature Inferencing, Knowledge-Intensive Question Ranking, and Coherent Question Sequencing are 10%, 14%,

[5] In the simulated question-answering process, only the question with the highest priority is selected to be answered, so the enabling status of the consistent question clustering module has no influence on the evaluation results.

and 76%, respectively. The underlying reason why the coherent question sequencing module has such a major impact is that it guides users to answer the discriminative questions using a sequence ranging from general to specific and insisting on one description aspect instead of allowing a jump from one aspect to another which may be unrelated. However, the degree to which each module contributes to overall performance may depend on the different application domains and the contents of the knowledge bases.

6 Conclusion

In this paper, we note the difficulty of evaluating CBR systems using multiple case bases, and propose an evaluation strategy to use different data sources to assess different evaluation goals of a CBR system. First, all the evaluation goals are divided into two categories: domain-independent goals and domain-dependent goals. For domain-independent goals, we can choose many simple case bases and a statistical evaluation method for testing. For domain-dependent goals, we can choose one or a few target domain case bases and use multiple weak evaluation methods for testing. This evaluation strategy is applied to a knowledge-intensive conversational CBR method as a case study. The results of our case study indicate that such a combination of evaluation methods and test data sources can provide more solid evaluation results than is possible with a single evaluation method.

References

1. Simon, H.A.: Artificial Intelligence: an Empirical Science. Artif. Intell. **77** (1995) 95–127
2. Cohen, P.R., Howe, A.E.: How Evaluation Guides AI Research. AI Mag. **9** (1988) 35–43
3. Cohen, P., Howe, A.: Toward AI Research Methodology: Three Case Studies in Evaluation. Systems, Man and Cybernetics, IEEE Transactions on **19** (1989) 634–646
4. Cohen, P.R.: Empirical Methods for Artificial Intelligence. MIT Press, Cambridge, MA, USA (1995)
5. McSherry, D.: Interactive Case-Based Reasoning in Sequential Diagnosis. Applied Intelligence **14** (2001) 65–76
6. Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Communications **7** (1994) 39–59
7. Kolodner, J.: Case-Based Reasoning. Morgan Kaufmann Publishers Inc. (1993)
8. Santamaria, J.C., Ram, A.: Systematic Evaluation of Design Decisions in CBR Systems. In: Proceedings of the AAAI Case-Based Reasoning Workshop, Seattle, Washington (1994) 23 – 29
9. Díaz-Agudo, B., González-Calero, P.A.: An Architecture for Knowledge Intensive CBR Systems. In: Proceedings of the 5th European Workshop on Case-Based Reasoning. Trento, Italy (2000) 37–48

10. Aha, D.W.: Generalizing from Case Studies: A Case Study. In Sleeman, D.H., Edwards, P., eds.: Proceedings of the Ninth International Workshop on Machine Learning, Aberdeen, Scotland, UK, Morgan Kaufmann (1992) 1–10
11. Aamodt, A.: Knowledge-Intensive Case-Based Reasoning in Creek. In Funk, P., González-Calero, P.A., eds.: 7th European Conference on Case-Based Reasoning. Madrid, Spain, (2004) 1–15
12. Sørmo, F., Cassens, J., Aamodt, A.: Explanation in Case-Based Reasoning-Perspectives and Goals. Artificial Intelligence Review **24** (2005) 109 – 143
13. Newman, D., Hettich, S., Blake, C., Merz, C.: UCI Repository of Machine Learning Databases [http://www.ics.uci.edu/ mlearn/mlrepository.html] (1998)
14. Doyle, M., Cunningham, P.: A Dynamic Approach to Reducing Dialog in On-line Decision Guides. In: European Workshop on Advances in Case-Based Reasoning, Trento, Italy (2000) 49–60
15. Tong, X., Öztürk, P., Gu, M.: Dynamic Feature Weighting in Nearest Neighbor Classifiers. In: Proceedings of the 3rd International Conference on Machine Learning and Cybe (ICMLC2004). Volume 4., Shanghai, China, (2004) 2406 – 2411
16. Yang, Q., Wu, J.: Enhancing the Effectiveness of Interactive Case-Based Reasoning with Clustering and Decision Forests. Applied Intelligence, **12** (2001) 49 – 64
17. Bogaerts, S., Leake, D.: Facilitating CBR for Incompletely-Described Cases: Distance Metrics for Partial Problem Descriptions. In: Proceedings of the 7th European Conference on Case-Based Reasoning, Springer - Verlag (2004) 62–76
18. Bareiss, R.: The Experimental Evaluation of a Case-Based Learning Apprentice. In: the proceedings of the Case-Based Reasoning Workshop, Pensacola Beach, Florida (1989) 162 – 167
19. McLaren, B.M.: Extensionally Defining Principles and Cases in Ethics: an AI Model. Artificial Intelligence Journal **150** (2003) 145 – 181
20. Aha, D.W., Breslow, L., Muñoz-Avila, H.: Conversational Case-Based Reasoning. Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies **14** (2001) 9
21. McSherry, D.: Minimizing Dialog Length in Interactive Case-Based Reasoning. In: International Joint Conferences on Artificial Intelligence. (2001) 993–998
22. Gupta, K.M., Aha, D.W., Sandhu, N.: Exploiting Taxonomic and Causal Relations in Conversational Case Retrieval. In: European Conference on Case Based Reasoning, Aberdeen, Scotland, UK (2002) 133–147
23. Gu, M., Aamodt, A.: A Knowledge-Intensive Method for Conversational CBR. In Muñoz-Avila, H., Ricci, F., eds.: Case-Based Reasoning Research and Development, Proceedings of the 6th International Conference on Case-Based Reasoning. Chicago, Illinois, Springer Verlag (2005) 296–311
24. Gu, M., Tong, X., Aamodt, A.: Comparing Similarity Calculation Methods in Conversational CBR. In Zhang, D., Khoshgoftaar, T.M., Shyu, M.L., eds.: Proceedings of the 2005 IEEE International Conference on Information Reuse and Integration, Hilton, Las Vegas, Nevada, USA, (2005) 427 – 432
25. Gu, M., Aamodt, A.: Dialog Learning in Conversational CBR. To appear in the Proceedings of the 19th International FLAIRS Conference, Melbourne Beach, Florida, AAAI Press (2006)
26. Gu, M., Bø, K.: Component Retrieval Using Knowledge-Intensive Conversational CBR. To appear in the Proceedings of the 19th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Annecy, France, Springer (2006)

27. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, Second Edition. 2 edn. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers (2005)
28. Aha, D.W., McSherry, D., Yang, Q.: Advances in Conversational Case-Based Reasoning. Knowledge Engineering Review **20** (2006) 7
29. Göker, M.H., Thompson, C.A.: Personalized Conversational Case-Based Recommendation. In: the 5th European Workshop on Case-Based Reasoning(EWCBR 2000), Trento, Italy (2000)
30. Shimazu, H.: Expertclerk: A Conversational Case-Based Reasoning Tool for Developing Salesclerk Agents in E-Commerce Webshops. Artificial Intelligence Review **18** (2002) 223 – 244
31. Ferguson, A., Bridge, D.G.: Partial Orders and Indifference Relations: Being Purposefully Vague in Case-Based Retrieval. In: EWCBR '00: Proceedings of the 5th European Workshop on Advances in Case-Based Reasoning, London, UK, Springer-Verlag (2000) 74–85
32. Gu, M., Aamodt, A., Tong, X.: Component Retrieval Using Conversational Case-Based Reasoning. In Shi, Z., He, Q., eds.: Intelligent Information Processing II, Volume 163 of IFIP International Federation for Information Processing., Springer Science + Business Media Inc (2004)
33. McSherry, D.: Explanation in Recommender Systems. Artificial Intelligence Review **24** (2005) 179 – 197
34. Reilly, J., McCarthy, K., McGinty, L., Smyth, B.: Explaining compound critiques. Artificial Intelligence Review **24** (2005) 199 – 220

Part III

Bibliography

Bibliography

Aamodt, Agnar. 1991. "A Knowledge-Intensive, Integrated Approach to Problem Solving and Sustained Learning (PhD thesis)." Ph.D. diss., University of Trondheim.

Aamodt, Agnar. 1994a, November 1–5. "Explanation-Driven Case-Based Reasoning." Edited by Stefan Wess, Klaus-Dieter Althoff, and Michael M. Richter, *Topics in Case-based reasoning, Proceedings of the first European Workshop on Case-Based Reasoning*, Volume 837 of *Lecture Notes in Computer Science*. Kaiserslautern, Germany: Springer, 274–288.

Aamodt, Agnar. 1994b, November 5–12. "A Knowledge Representation System for Integration of General and Case-Specific Knowledge." *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence*. New Orleans, 4.

Aamodt, Agnar. 2001. "Modeling the Knowledge Contents of CBR Systems." *Workshop Program at the Fourth International Conference on Case-Based Reasoning*. Vancouver: Naval Research Laboratory Technical Note AIC-01-003, 32–37.

Aamodt, Agnar. 2004. "Knowledge-Intensive Case-Based Reasoning in Creek." Edited by Peter Funk and Pedro A. González-Calero, *7th European Conference on Case-Based Reasoning*, Volume 3155. Madrid, Spain: Springer, 1–15.

Aamodt, Agnar, and Enric Plaza. 1994. "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches." *AI Communications* 7 (1): 39–59.

Aha, D. W. 1991. "Case-Based Learning Algorithms." *the DARPA Case-Based Reasoning Workshop*. Morgan Kaufmann, 147–158.

Aha, David W. 1992, July 1–3. "Generalizing from Case Studies: A Case Study." Edited by Derek H. Sleeman and Peter Edwards, *Proceedings of the Ninth International Workshop on Machine Learning*. Aberdeen, Scotland, UK: Morgan Kaufmann, 1–10.

Aha, David W., and Len Breslow. 1997, July 25–27. "Refining Conversational Case Libraries." *ICCBR '97: Proceedings of the Second International Con-*

ference on Case-Based Reasoning Research and Development. Providence, Rhode Island, USA: Springer-Verlag, 267–278.

Aha, David W., Len Breslow, and Héctor Muñoz-Avila. 2001. "Conversational Case-Based Reasoning." *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies* 14 (1): 9.

Aha, David W., and Kalyan Moy Gupta. 2002. "Causal Query Elaboration in Conversational Case-Based Reasoning." *International Florida Artificial Intelligence Research Society Conference*. Pensacola Beach, Florida, USA, 95–100.

Aha, David W., Tucker Maney, and Len Breslow. 1998. "Supporting Dialogue Inferencing in Conversational Case-Based Reasoning." *European Workshop on Case-Based Reasoning*. Dublin, Ireland, 262–273.

Aha, David W., David McSherry, and Qiang Yang. 2006. "Advances in Conversational Case-Based Reasoning." *Knowledge Engineering Review* 20 (3): 7.

Bareiss, Ray. 1989, May 31–June 2. "The Experimental Evaluation of A Case-Based Learning Apprentice." *the proceedings of the Case-Based Reasoning Workshop*. Pensacola Beach, Florida, 162–167.

Bergmann, Ralph, Michael M. Richter, Sascha Schmitt, Armin Stahl, and Ivo Vollrath. 2001. "Utility-Oriented Matching: A New Research Direction for Case-Based Reasoning." *9th German Workshop on Case-Based Reasoning (GWCBR'2001)*.

Bergmann, Ralph, Ralph Traphöner, Sascha Schmitt, Pádraig Cunningham, and Barry Smyth. 2002. Chapter Knowledge-Intensive Product Search and Customization in Electronic Commerce of *E-Business Applications: Technologies for Tomorrow's Solutions*, edited by Jorge Gasos and Klaus-Dieter Thoben, Volume Advanced Information Processing, 89–103. Springer Verlag.

Bernstein, A., and M. Klein. 2002. "Towards High-Precision Service Retrieval." *International Semantic Web Conference*. Sardinia Italy.

Bogaerts, Steven, and David Leake. 2004, Aug. "Facilitating CBR for Incompletely-Described Cases: Distance Metrics for Partial Problem Descriptions." *Proceedings of the 7th European Conference on Case-Based Reasoning*. Springer - Verlag, 62–76.

Branting, Karl, James Lester, and Bradford Mott. 2004, January. "Dialogue Management for Conversational Case-Based Reasoning." *the 7th European Conference on Case-Based Reasoing*, Volume 3155. 77–90.

Burke, R.D., K.J. Hammond, and B.C. Yound. 1997. "The FindMe Approach to Assisted Browsing." *Expert, IEEE* 12 (4): 32–40.

Burke, Robin. 2002. “Interactive Critiquing for Catalog Navigation in E-Commerce.” *Artificial Intelligence Review* 18 (3–4): 245–267.

Carrick, Christina, Qiang Yang, Irene Abi-Zeid, and Luc Lamontagne. 1999. “Activating CBR Systems through Autonomous Information Gathering.” *International Conference on Case Based Reasoning.* Germany.

Cohen, Paul R. 1995. *Empirical Methods for Artificial Intelligence.* Cambridge, MA, USA: MIT Press.

Cohen, Paul R., and Adele E. Howe. 1988. “How Evaluation Guides AI Research.” *AI Mag.* 9 (4): 35–43.

Cohen, P.R., and A.E. Howe. 1989. “Toward AI Research Methodology: Three Case Studies in Evaluation.” *Systems, Man and Cybernetics, IEEE Transactions on* 19 (3): 634–646.

Cooper, Donald R., and Pamela S. Schindler. 2003. Chapter 17: Hypothesis Testing of *Business Research Methods (Eighth Edition)*, The McGraw-Hill/Irwin Series, 518–565. Brent Gordon.

Cunningham, P., and B. Smyth. 1994. “A Comparison of Model-Based and Incremental Case-Based Approaches to Electronic Fault Diagnosis.” *Case-Based Reasoning Workshop.* Seattle, USA.

Cunningham, Padraig, Ralph Bergmann, Sascha Schmitt, Ralph Traphoner, Sean Breen, and Barry Smyth. 2001. “WEBSELL: Intelligent Sales Assistants for the World Wide Web.” *KI - Kunstliche Intelligenz* 1:28–31.

Díaz-Agudo, Belén, and Pedro A. González-Calero. 2000, September. “An Architecture for Knowledge Intensive CBR Systems.” *5th European Workshop on Case-Based Reasoning*, Volume 1898 of *Lecture Notes in Computer Science.* Trento, Italy, 37–48.

Doody, John, Edwin Costello, Lorraine McGinty, and Barry Smyth. 2006, May. “Combining Visualization and Feedback for Eyewear Recommendation.” Edited by Geoffrey C. J. Sutcliffe and Randy G. Goebel, *Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference.* Melbourne Beach, Florida: AAAI, 6.

Doyle, Michelle, and Pádraig Cunningham. 2000. “A Dynamic Approach to Reducing Dialog in On-Line Decision Guides.” *European Workshop on Advances in Case-Based Reasoning.* Trento, Italy, 49–60.

DynamicImager. 2006. TROLLHETTA-Corporation. http://trollhetta.tripod.com/DI.htm.

Ferguson, Alex, and Derek G. Bridge. 2000. “Partial Orders and Indifference Relations: Being Purposefully Vague in Case-Based Retrieval.” *EWCBR '00: Proceedings of the 5th European Workshop on Advances in Case-Based Reasoning.* London, UK: Springer-Verlag, 74–85.

Fernández-Chamizo, Carmen, Pedro A. González-Calero, Mercedes Gámez-Albarrán, and Luis Hernández-Yáñez. 1996. "Supporting Object Reuse through Case-Based Reasoning." *EWCBR '96: Proceedings of the Third European Workshop on Advances in Case-Based Reasoning.* London, UK: Springer-Verlag, 135–149.

Frakes, W B, and B A Nejmeh. 1987. "Software Reuse through Information Retrieval." *ACM SIGIR Forum* 21 (1–2): 30–36.

Gebhardt, Friedrich, Angi Vob, Wolfgang Grather, and Barbara Schmidt-Belz. 1997. *Reasoning with Complex Cases.* Kluwer Academic Publishers.

Göker, Mehmet H., and Cynthia A. Thompson. 2000. "Personalized Conversational Case-Based Recommendation." *the 5th European Workshop on Case-Based Reasoning (EWCBR 2000).* Trento, Italy.

Gómez-Gauchía, Hector, Belén Díaz-Agudo, Pedro Pablo Gómez Martín, and Pedro González-Calero. 2005, September. "Supporting Conversation Variability in COBBER Using Causal Loops." *Case-Based Reasoning Research and Development, Proceedings of the 6th International Conference on Case-Based Reasoning*, Volume 3620 of *Lecture Notes in Artificial Intelligence.* 252–266.

Grimnes, Morten, and Agnar Aamodt. 1996. "A Two-layer Case-Based Reasoning Architecture for Medical Image Understanding." *European Workshop on Case-Based Reasoning*, Volume 1168 of *Lecture Notes in Artificial Intelligence.* Springer Verlag, 164–178.

Gu, Mingyang. 2004. "A Case to Do Empirical Study Using Educational Projects." *Issues in Informing Science and Information Technology Journal* 1:509–520. Also presented in InSITE2004, Rockhampton, Australia, Jun. 25–28.

Gu, Mingyang. 2005, November. "Supporting Generalized Cases in Conversational CBR." Edited by Alexander Gelbukh, Alvaro de Albornoz, and Hugo Terashima-Marin, *Advances in Artificial Intelligence, Proceedings of the Fourth Mexican International Conference on Artificial Intelligence (MICAI2005)*, Volume 3789 of *Lecture Notes in Artificial Intelligence.* Monterrey, Mexico: Springer Verlag, 544–553.

Gu, Mingyang, and Agnar Aamodt. 2004, 30th August–2nd September. "Explanation-boosted question selection in conversational CBR." Edited by Pablo Gervás and Kalyan Moy Gupta, *Proceedings of the ECCBR 2004 Workshops, the 7th European Conference on Case-Based Reasoning*, Volume 142 of *Technical report.* Universidad Complutense de Madrid, Departemento de Sistemas Informaticos y Programacion, Madrid, Spain, 105–114.

Gu, Mingyang, and Agnar Aamodt. 2005, August. "A Knowledge-Intensive Method for Conversational CBR." Edited by Héctor Muñoz-Avila and

Francesco Ricci, *Case-Based Reasoning Research and Development, Proceedings of the 6th International Conference on Case-Based Reasoning*, Volume 3620 of *Lecture Notes in Artificial Intelligence*. Chicago, Illinois: Springer Verlag, 296–311.

Gu, Mingyang, and Agnar Aamodt. 2006a, May 11–13. "Dialog Learning in Conversational CBR." Edited by Geoff C. J. Sutcliffe and Randy G. Goebel, *Proceedings of the 19th International FLAIRS Conference*. Melbourne Beach, Florida: AAAI Press, 358–363.

Gu, Mingyang, and Agnar Aamodt. 2006b, September. "Evaluating CBR Systems Using Different Data Sources: A Case Study." Edited by T.R. Roth-Berghofer et al., *accepted by the 8th European Conference on Case-Based Reasoning*, Volume 4106 of *Lecture Notes in Artificial Intelligence*. Fethiye, Turkey: Springer Verlag, 121–135.

Gu, Mingyang, Agnar Aamodt, and Xin Tong. 2004, 21–23 Oct. "Component Retrieval using Conversational Case-Based Reasoning." Edited by Zhongzhi Shi and Qing He, *Intelligent Information Processing II - International Conference on Intelligent Information Processing (IIP2004)*, Volume 163 of *IFIP International Federation for Information Processing*. Beijing, China: Springer Science + Business Media Inc, 259–271.

Gu, Mingyang, and Ketil Bø. 2006, 27–30, June. "Component Retrieval Using Knowledge-Intensive Conversational CBR." *accepted by the 19th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, Volume 4031 of *Lecture Notes in Artificial Intelligence*. Annecy, France: Springer Verlag, 554–563.

Gu, Mingyang, and Xin Tong. 2004a, Sep. 28–Sep. 30. "An Intelligent Component Retrieval System Using Conversational CBR." *the 28th International Conference on Computer Software and Application (COMPSAC2004)*, Volume 2. Hong Kong, China: IEEE Press, 47–48.

Gu, Mingyang, and Xin Tong. 2004b, April 5–8. "Towards Hypotheses on Creativity in Software Development." Edited by Frank Bomarius and Hajimu Iida, *Proceedings of the 5th International Conference on Product Focused Software Process Improvement (PROFES 2004)*, Volume 3009 of *Lecture Notes in Computer Science*. Kansai Science City, Japan: Springer-Verlag, 47–61.

Gu, Mingyang, Xin Tong, and Agnar Aamodt. 2005, August. "Comparing Similarity Calculation Methods in Conversational CBR." Edited by Du Zhang, Taghi M. Khoshgoftaar, and Mei-Ling Shyu, *Proceedings of the 2005 IEEE International Conference on Information Reuse and Integration*. Hilton, Las Vegas, Nevada, USA: IEEE Press, 427–432.

Gupta, Kalyan Moy. 1998. "Knowledge-Based System for Troubleshooting

Complex Equipment." *International Journal of Information and Computing Science* 1 (1): 29–41.

Gupta, Kalyan Moy. 2001. "Taxonomic Conversational Case-Based Reasoning." *International Conference on Case-Based Reasoning.* Vancouver, BC, Canada, 219–233.

Gupta, Kalyan Moy, David W. Aha, and Nabil Sandhu. 2002. "Exploiting Taxonomic and Causal Relations in Conversational Case Retrieval." *European Conference on Case Based Reasoning.* Aberdeen, Scotland, UK, 133–147.

Helm, Richard, and Yoelle S. Maarek. 1991. "Integrating Information Retrieval and Domain Specific Approaches for Browsing and Retrieval in Object-Oriented Class Libraries." *Conference on Object Oriented Programming Systems Languages and Applications.* Phoenix, Arizona, United States, 47–61.

IBROW-project. 2005. http://www.swi.psy.uva.nl/projects/ibrow/ home.html.

Iribarne, L., J.M. Troya, and A. Vallecillo. 2002, September. "Selecting Software Components with Multiple Interfaces." *28th Euromicro Conference.* Dortmund, Germany: IEEE Computer Society Press, 27–33.

Kira, Kenji, and L.A. Rendell. 1992. "The Feature Selection Problem Traditional Methods and A New Algorithm." *Tenth National Conference on Artificial Intelligence.*

Klein, Mark, and Abraham Bernstein. 2001. "Searching for Services on the Semantic Web Using Process Ontologies." *The First Semantic Web Working Symposium.* Stanford, CA, USA.

Kofod-Petersen, Anders, and Agnar Aamodt. 2003, October. "Case-Based Situation Assessment in a Mobile Context-Aware System." *Workshop on Artificial Intelligence for Mobil Systems.* Seattle.

Kolodner, Janet. 1993. *Case-Based Reasoning.* Morgan Kaufmann Publishers Inc.

Kolodner, Janet L., and Robert L. Simpson. 1989. "The MEDIATOR: Analysis of an Early Case-Based Problem Solver." *Cognitive Science* 13 (4): 507–549.

Kusnierczyk, Waclaw, Agnar Aamodt, and Astrid Lægreid. 2004, March. "Towards Automated Explanation of Gene-Gene Relationships." *the Eighth International Conference on Computational Molecular Biology.* San Diego.

Kusnierczyk, Waclaw, Agnar Aamodt, and Astrid Lægreid. 2005, August. "Knowledge-Intensive Case-Based Support for Automated Explanation of Biological Phenomena." *the 6th International Conference on Case-Based Reasoning, Workshop Proceedings.* Depaul University, Chicago, 62–71.

Magnini, B. 1999. "Use of a Lexical Knowledge Base for Information Access Systems." *International Journal of Theoretical & Applied Issues in Specialized Communication* 5 (2): 203–228.

Maximini, Kerstin, Rainer Maximini, and Ralph Bergmann. 2003, June. "An Investigation of Generalized Cases." *5th International Conference on Case-Based Reasoning*. Trondheim, Norway, 261–275.

McLaren, Bruce M. 2003. "Extensionally Defining Principles and Cases in Ethics: an AI Model." *Artificial Intelligence Journal* 150:145–181.

McSherry, David. 1999. "Strategic Induction of Decision Trees." *Knowl.-Based Syst.* 12 (5–6): 269–275.

McSherry, David. 2001a. "Interactive Case-Based Reasoning in Sequential Diagnosis." *Applied Intelligence* 14 (1): 65–76.

McSherry, David. 2001b. "Minimizing Dialog Length in Interactive Case-Based Reasoning." *International Joint Conferences on Artificial Intelligence*. 993–998.

McSherry, David. 2003, August 9–15, 2003. "Increasing Dialogue Efficiency in Case-Based Reasoning Without Loss of Solution Quality." *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*. Acapulco, Mexico, 121–126.

McSherry, David. 2004a. "Balancing User Satisfaction and Cognitive Load in Coverage-Optimized Retrieval." *Knowl.-Based Syst.* 17 (2–4): 113–119.

McSherry, David. 2004b, August. "Incremental Relaxation of Unsuccessful Queries." *7th of the European Conference on Case-Based Reasoning*, Volume 3155. 331–345.

McSherry, David. 2005. "Explanation in Recommender Systems." *Artificial Intelligence Review* 24 (2): 179–197.

Mcsherry, David. 2005. "Retrieval Failure and Recovery in Recommender Systems." *Artificial Intelligence Review* 24 (3–4): 319–338 (November).

Mili, A., R. Mili, and R.T. Mittermeir. 1998. "A Survey of Software Reuse Libraries." *Annals of Software Engineering* 5 (0): 349–414.

Miller, George A. 1956. "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information." *Psychological Review* 63:81–97.

Mitchell, Tom. 1997. "Decision Tree Learning." In *Machine Learning*, 414. McGraw Hill.

Mougouie, Babak, and Ralph Bergmann. 2002, Sep. "Similarity Assessment for Generalized Cases by Optimization Methods." *6th European Conference on Case-Based Reasoning*. Aberdeen, Scotland, UK, 249–263.

Muñoz-Avila, Héctor, David W. Aha, Len Breslow, and Dana Nau. 1999. "HICAP: An Interactive Case-Based Planning Architecture and its Application to Noncombatant Evacuation Operations." *Proceedings of the Eleventh Conference on Innovative Applications of AI*. Orlando, FL: AAAI Press, 870–875.

Newman, D.J., S. Hettich, C.L. Blake, and C.J. Merz. 1998. UCI Repository of Machine Learning Databases.

Öztürk, Pinar. 2000. "A Knowledge Level Model of Context and Context Use in Diagnostic Domains - Doctoral Thesis." Ph.D. diss.

Öztürk, Pinar, and Agnar Aamodt. 1997, February. "Towards Model of Context for Case-Based Diagnostic Problem Solving." *the interdisciplinary conference on modeling and using context.* Rio de Janeiro, 198–208.

Park, Y. 2000. "Software Retrieval by Samples Using Concept Analysis." *Systems & Software* 54 (3): 179–183.

Prieto-Daz, Rubn. 1991. "Implementing Faceted Classification for Software Reuse." *Communications of the ACM* 34 (5): 89–97.

Quinlan, J. R. 1986. "Induction of Decision Trees." *Machine Learning* 1 (1): 81–106 (March).

Ravichandran, T., and Marcus A. Rothenberger. 2003. "Software Reuse Strategies and Component Markets." *Communications of the ACM* 46 (8): 109–114.

Rayner, N. J. W., and C. J. Harris. 1996. "Mission Management for Multiple Autonomous Vehicles." Technical Report, University of Southampton.

Redondo, Rebeca P. Díaz, Josè Juan Pazos Arias, Ana Fernández Vilas, and Belèn Barragáns Martínez. 2002, October. "Approximate Retrieval of Incomplete and Formal Specifications Applied to Vertical Reuse." *Proc of IEEE International Conference on Software Maintenance (ICSM).* Montreal, Canada., 618–627.

Reilly, James, Kevin McCarthy, Lorraine McGinty, and Barry Smyth. 2005. "Explaining Compound Critiques." *Artificial Intelligence Review* 24 (2): 199–220.

Reinartz, Thomas, Ioannis Iglezakis, and Thomas Roth-Berghofer. 2001. "Review and Restore for Case-Based Maintenance." *Computational Intelligence* 17 (2): 214–234.

Ricci, Francesco, Bora Arslan, Nader Mirzadeh, and Adriano Venturini. 2002. "ITR: A Case-Based Travel Advisory System." *ECCBR '02: Proceedings of the 6th European Conference on Advances in Case-Based Reasoning.* London, UK: Springer-Verlag, 613–627.

Richter, Michael M. 1995, October 23–26. "The Knowledge Contained in Similarity Measures." Edited by Manuela Veloso and Agnar Aamodt, *Proceedings of the 1st International Conference on Case-Based Reasoning (ICCBR1995)*, Volume 1010 of *Lecture Notes in Artificial Intelligence.* Sesimbra, Portugal: Springer.

Richter, Michael M., and Stefan Web. 1991. "Similarity, Uncertainty and Case-Based Reasoning in PADTEX." Seki-report SR-91-01 (SFB), Fachbereich Informatik, Universität Kaiserslautern, Kaiserslautern, Germany.

Rudnicky, A., and W. Xu. 1999, Dec. "An Agenda-Based Dialog Management Architecture for Spoken Language Systems." *Proceedings of the International Workshop on Automatic Speech Recognition and Understanding (ASRU).* Keystone, Colorado, 337–340.

Salzberg, Steven. 1991. "A Nearest Hyperrectangle Learning Method." *Mach. Learn.* 6 (3): 251–276.

Santamaria, Juan Carlos, and Ashwin Ram. 1994. "Systematic Evaluation of Design Decisions in CBR Systems." *Proceedings of the AAAI Case-Based Reasoning Workshop.* Seattle, Washington, 23–29.

Schmitt, Sascha. 2002. "simVar: A Similarity-Influenced Question Selection Criterion for e-Sales Dialogs." *Artificial Intelligence Review* 18 (3–4): 195–221.

Schmitt, Sascha, and Ralph Bergmann. 2001. "A Formal Approach to Dialogs with Online Customers." *The 14th Bled Electronic Commerce Conference*, Volume 1. Bled, Slovenia, 309–328.

Shimazu, Hideo. 2002. "ExpertClerk: A Conversational Case-Based Reasoning Tool for Developing Salesclerk Agents in E-Commerce Webshops." *Artificial Intelligence Review* 18 (3–4): 223–244.

Simon, Herbert A. 1995. "Artificial Intelligence: an Empirical Science." *Artif. Intell.* 77 (1): 95–127.

Skalle, Pål, and Agnar Aamodt. 2004, October. "Knowledge-Based Decision Support in Oil Well Drilling." Edited by Zhongzhi Shi, *the International Conference on Intelligent Information Systems.* Beijing, China.

Smyth, Barry, and Paul McClave. 2001. "Similarity vs. Diversity." *ICCBR 2001: Proceedings of the 4th International Conference on Case-Based Reasoning.* London, UK: Springer-Verlag, 347–361.

Sørmo, Frode. 2000. "Plausible Inheritance: Semantic Network Inference for Case-Based Reasoning." Master Thesis, Norwegian University of Science and Technology.

Sørmo, Frode, and Agnar Aamodt. 2002, September. "Knowledge Communication and CBR." *Workshop in the 6th European Conference on Case-Based Reasoning.* Aberdeen, 47–59.

Sørmo, Frode, Jorg Cassens, and Agnar Aamodt. 2005. "Explanation in case-based reasoning-perspectives and goals." *Artificial Intelligence Review* 24 (2): 109–143.

Sugumaran, Vijayan, and Veda C. Storey. 2003. "A Semantic-Based Approach to Component Retrieval." *The DATA BASE for Advances in Information Systems* 34 (3): 8–24.

Tartakovski, Alexander, Martin Schaaf, Rainer Maximini, and Ralph Bergmann. 2004, Aug-Sep. "MINLP Based Retrieval of Generalized Cases." *7th European Conference on Case-Based Reasoning.* Madrid, Spain, 404–418.

Tong, Xin, and Mingyang Gu. 2005, February 14–16. "A Framework for Grouping and Comparing Feature Weighting Methods." Edited by M. H. Hamza, *IASTED International Conference on Artificial Intelligence and Applications.* Innsbruck, Austria, 697–702.

Tong, Xin, Pinar Öztürk, and Mingyang Gu. 2004, 26–29 Aug. "Dynamic Feature Weighting in Nearest Neighbor Classifiers." *Proceedings of the 3rd International Conference on Machine Learning and Cybe (ICMLC2004)*, Volume 4. Shanghai, China: IEEE Press, 2406–2411.

TrollCreek. 2005. TrollCreek Website. http://creek.idi.ntnu.no/.

Vitharana, Padmal, Fatemah Mariam Zahedi, and Hemant Jain. 2003. "Design, Retrieval, and Assembly in Component-Based Software Development." *Communications of the ACM* 46 (11): 97–102.

Vollrath, Ivo. 1998, September. "Reuse of Complex Electronic Designs: Requirements Analysis for a CBR Application." Edited by Barry Smyth and Padraig Cunningham, *Advances in Case-Based Reasoning, Proceedings of the 4th European Workshop on Case-Based Reasoning (EWCBR-98)*, Volume 1488 of *Lecture Notes in Computer Science.* Dublin, Ireland: Springer, 136–147.

Weber, Barbara, Stefanie Rinderle, Werner Wild, and Manfred Reichert. 2005. "CCBR-Driven Business Process Evolution." *ICCBR.* 610–624.

Witten, Ian H., and Eibe Frank. 2005, Jun. 2005. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition.* 2. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers.

Yang, Qiang, and Jing Wu. 2001. "Enhancing the Effectiveness of Interactive Case-Based Reasoning with Clustering and Decision Forests." *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies* 12:49–64.

Zaremski, A. M., and J. M. Wing. 1995. "Signature Matching: a Tool for Using Software Libraries." *ACM Transactions on Software Engineering Methodology* 4 (2): 146–170.

Part IV

Appendix

Appendix A

Domain Expert Evaluation Hypotheses

- **Hypothesis 1:** The TrollCCRM system gives more appropriate result.

  Rationality: The query used by the one-shot CBR-based method is guessed by users and may be less over-specified, under-specified or ill-specified. On the contrary, the query in TrollCCRM is constructed incrementally by answering the user understandable questions (the valid alternative answers for each question are also displayed for users to select in an easily understandable way), which can help finding out the appropriate component as soon as possible, and as good as possible.

- **Hypothesis 2:** Users' cognitive load demanded in guessing the component query is alleviated.

  Rationality: Being not clear about their component requirements and not knowing what component available for them to choose, users suffer a lot of cognitive difficulties to 'guess' their component queries. On the contrary, TrollCCRM alleviates this difficulty through moving their focus from guessing the component requirements to selecting alternative answers to prompted discriminative questions.

- **Hypothesis 3:** The feature inferencing module, the question clustering module and the question sequencing module help alleviate users' cognitive load to inspect and answer questions.

  Rationality: The feature inferencing module removes the questions that can be answered implicitly, which prevents users' wasting their energy to answer the unnecessary questions. The question clustering module groups questions according to their semantic relations and displays them together, and the question sequencing module asks users the semantic related questions in an uninterrupted way. These two modules can prevent users' focus jumping too much among semantic topics.

- **Hypothesis 4:** The explanations about how the component features are extended, how the query is matched to each returned component, why a question is given higher priority to be asked by the question sequencing module and why a question is clustered together with another question by the question clustering module give users more confidence in the system results.

  Rationality: Explanation mechanisms about the system inside reasoning process provide users the reasoning transparency instead of a black box, which clarifies users' possible doubts about system's executing process and improves their confidence in the results.

- **Hypothesis 5:** TrollCCRM helps improving the retrieval results for users with limited domain knowledge.

  Rationality: Instead of the systematic domain knowledge, users only need to understand the meaning of each question and their alternative answers, which alleviates the knowledge learning load for the users with limited domain knowledge to get good component retrieval results.

Appendix B

Domain Expert Evaluation Feedback Form

- **Question 1**: Which type of retrieval process can give you more appropriate components?

  A. One-shot CBR-based method.

  B. Conversational CCBR-based method.

  C. No difference.

  Comments:

- **Question 2**: Does TrollCCRM reduce your cognitive load in constructing your component requirements?

  A. Reduce a lot.

  B. Not so much.

  C. Not at all.

  Comments:

- **Question 3**: Can the feature inferencing, the question clustering, and the question sequencing modules help providing you a natural question-answering environment, and therefore alleviate your cognitive load?

  A. Help a lot.

  B. Help a little bit.

  C. Not at all.

  Comments:

- **Question 4**: Do the explanation mechanisms about the feature inferencing, the knowledge-intensive case matching, the question sequencing and clustering help improving your confidence in the results?

  A. Help a lot.

  B. Help a little bit.

  C. Not at all.

  Comments:

- **Question 5**: Is TrollCCRM able to help improving the retrieval results for users with limited domain knowledge?

  A. Help a lot.

  B. Help a little bit.

  C. Not at all.

  Comments: