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

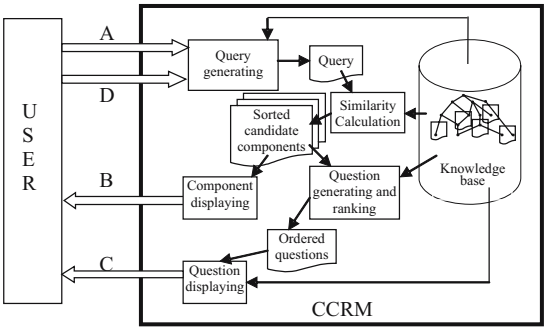

Fig. 1. The architecture of conversational component retrieval model (CCRM)

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

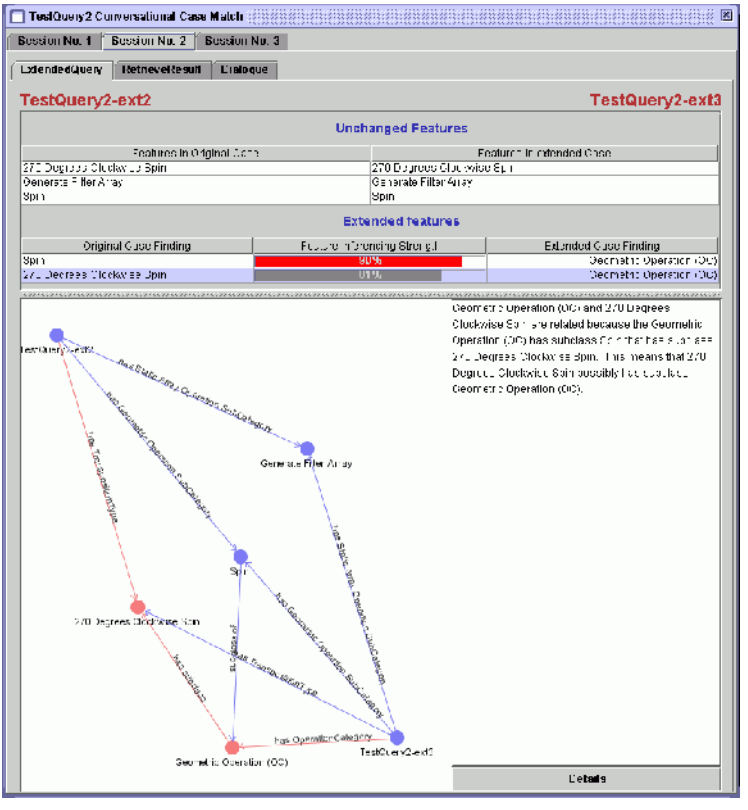

Fig. 2. The conversational retrieval process implemented in TrollCreek

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)