Conversational Process-Oriented Case-Based Reasoning

Christian Zeyen, Gilbert Müller, and Ralph Bergmann

Business Information Systems II
University of Trier
54286 Trier, Germany
[zeyen][muellerg][bergmann]@uni-trier.de,
http://www.wi2.uni-trier.de

Abstract. Current approaches for retrieval and adaptation in process-oriented case-based reasoning (POCBR) assume a fully elaborated query given by the user. However, users may only have a vague idea of the workflow they desire or they lack the required domain knowledge. Conversational case-based reasoning (CCBR) particularly addresses this problem by proposing methods which incrementally elicit the relevant features of the target problem in an interactive dialog. However, no CCBR approaches exist that are capable of automatically creating questions from the case descriptions that go beyond attribute-value representations. In particular, no approaches exist that are applicable to workflow cases in graph representation. This paper closes this gap and presents a conversational POCBR approach (C-POCBR) in which questions related to structural properties of the workflow cases are generated automatically. An evaluation in the domain of cooking workflows reveals that C-POCBR can reduce the communication effort for users during retrieval.

Keywords: Process-Oriented Case-Based Reasoning, Workflow Retrieval, Conversational Case-Based Reasoning, Workflows

1 Introduction

Process-oriented case-based reasoning (POCBR) [18] addresses the integration of case-based reasoning (CBR) [24,5] with process-aware information systems [26]. A case in POCBR is usually a workflow or process description expressing procedural experiential knowledge. Among other things, POCBR aims at providing experience-based support for the modeling of workflows [11,14]. In particular, new workflows can be constructed by reuse of already available workflows that have to be adapted for new purposes and circumstances. In traditional POCBR, retrieval and adaptation are fully automatic and assume a fully elaborated query from the beginning [6,20,21]. However, in practice, users may only have a vague idea of the workflow they desire or they lack detailed domain knowledge and thus have serious difficulties to provide a precise query.

Conversational case-based reasoning (CCBR) [1,2,9] addresses this problem by focusing on the interactive nature of problem solving in particular. CCBR approaches include methods which incrementally elicit the relevant features of the target problem in an interactive dialog, often with the aim of minimizing the communication effort for the user. The basic assumption behind CCBR is that guided question answering requires less domain expertise than providing detailed queries from scratch. CCBR research so far focuses on methods for question selection and dialog inferencing and is mainly applied to diagnosis, help-desk support, and product recommendation [12,15,17,16]. Only very few approaches have been proposed that address synthetic applications [13,23,27]. Today, no CCBR approaches exist that automatically elicit questions from case descriptions that go beyond attribute-value representation to construct queries for retrieval. In particular, no such approach exists so far that is applicable for workflow representations as required for POCBR.

We present a new conversational POCBR approach, called C-POCBR. We consider graph-based workflow representations for cases and we propose an approach that considers the structural properties of workflows during the C-POCBR retrieval. Questions related to structural properties of cases are automatically constructed based on extracted workflow fragments and a respective question selection strategy is proposed. Thereby, we aim at reducing the effort and the required expertise for the definition of queries in POCBR. We illustrate and evaluate the approach in the cooking domain [19].

In the following, section 2 briefly introduces POCBR and CCBR before section 3 describes our C-POCBR approach. An experimental evaluation is presented in section 4 while section 5 summarizes our findings and discusses future work.

2 Foundations and Related Work

We now briefly describe relevant foundations and related work in the fields of POCBR and CCBR.

2.1 Process-Oriented CBR

POCBR [18] aims at supporting various tasks in process-aware information systems [26] such as process and workflow modeling, monitoring, analysis, or execution. In this paper, we focus on workflow modeling by reuse of best-practice workflows from a repository (case base). Thus, we aim at retrieving a workflow from a repository for reuse that is best suited to a specific situation.

In POCBR, cases are often represented as processes or workflows. Broadly speaking, a workflow describes a logical or chronological order (referred to as the control-flow) of tasks that are needed to reach a certain outcome – the workflow output [26]. Tasks exchange physical products or data, which is defined by the data-flow. In cooking workflows, tasks represent required cooking steps and exchange ingredients in order to produce a certain dish. We describe workflows as semantically labeled directed graphs by adopting the representation by Bergmann and Gil [6].

Definition 1. *A workflow is a directed graph $W = (N, E)$ with a set of nodes N and a set of edges $E \subseteq N \times N$. Nodes $N = N^D \cup N^T \cup N^C$ can be data nodes N^D, task nodes N^T, or control-flow nodes N^C. Each node $n \in (N^D \cup N^T)$ has a semantic label $S(n) \in \Sigma$, where Σ is a language for semantic annotations. Edges $E = E^C \cup E^D$ can be control-flow edges $E^C \subseteq (N^T \cup N^C) \times (N^T \cup N^C)$, which define the order of the tasks and control-flow nodes or data-flow edges $E^D \subseteq (N^D \times N^T) \cup (N^T \times N^D)$, which define how the data is shared between the tasks.*

A workflow $W' = (N', E')$ is a *partial workflow* (we write $W' \subseteq W$) of a workflow $W = (N, E)$ if W' is a subgraph of W with $N' \subseteq N$ and $E' \subseteq E$. Figure 1 gives an example of a purely sequential cooking workflow describing the preparation of a tomato sandwich.

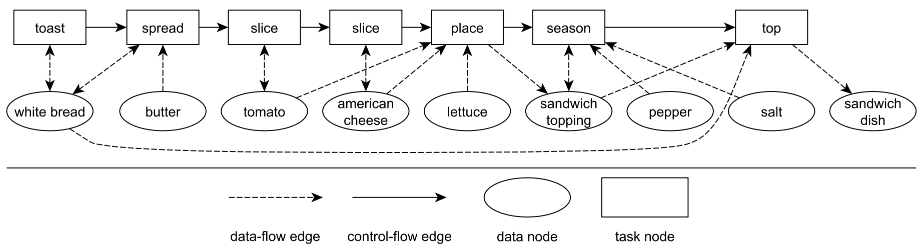

Fig. 1. Example of a Cooking Workflow

The language Σ for the semantic labels of nodes is structured hierarchically in two distinct domain-specific taxonomies, i.e., a data taxonomy of cooking ingredients and a task taxonomy of cooking steps. Thereby, workflows can be generalized regarding their semantic labels [20]. Generalized workflows provide a more general description and thus stand for a set of more specific workflows. For example, the workflow in Figure 1 can be generalized by generalizing the ingredient *american cheese* to the more general ingredient *cheese* from the data taxonomy. A workflow W^* is a generalization of a workflow W (we write $W^* \sqsupseteq W$) if there exists a total mapping of data nodes and task nodes from W^* to W, in which each semantic label in W^* is more general (or equal) according to the taxonomies than the respective label in W.

In order to obtain a reusable workflow, similarity search or process model querying can be applied [8]. Outside of POCBR, various query languages have been proposed [3,4,25], which are used in visual query editors to formulate graph-based queries. Matching workflows from a repository are then obtained by applying graph edit measures [7] or graph/subgraph similarity measures [6,10]. We focus on similarity search as it is able to provide results even if exact matches are not available.

Queries in POCBR are used to describe the users' requirements for retrieving the most useful workflows [22]. In previous work [22], we proposed a process-oriented query language (POQL) to specify such queries. A POQL query $Q = (Q^+, Q^-)$ consists of a query part $Q^+ = (q^+)$ with a single query workflow and a restriction part $Q^- = (q_1^-, \ldots, q_n^-)$ with several restriction workflows q_i^-. The query workflow represents properties the searched workflow should fulfill. Each restriction workflow represents one undesired situation that should be avoided. For a POQL query Q and a case workflow W_c, we define the following similarity measure:

$$\begin{aligned} \mathrm{sim}(Q, W_c) = & \frac{\mathrm{sim}^+(q^+, W_c) \cdot \mathrm{size}_{wf}(q^+)}{\mathrm{size}_q(Q)} \\ & + \frac{\sum_{q^- \in Q^-} (\mathrm{sim}^-(q^-, W_c) \cdot \mathrm{size}_{wf}(q^-))}{\mathrm{size}_q(Q)} \end{aligned} \tag{1}$$

The similarities sim^+ and sim^- are weighted with the number of nodes and edges contained in a query's workflow q, i.e., $\mathrm{size}_{wf}(q) = |N|+|E|$. They are normalized with the overall size of the query Q, i.e., $\mathrm{size}_q(Q) = \mathrm{size}_{wf}(q^+) + \sum_{q^- \in Q^-} \mathrm{size}_{wf}(q^-)$.

The query similarity sim^+ is assessed according to our similarity measure [6], which treats the similarity computation $\mathrm{sim}^+(q^+, W_c) \in [0,1]$ between the query workflow $q^+ = (N_q, E_q)$ and a case workflow $W_c = (N_c, E_c)$ as an optimization problem:

$$\mathrm{sim}^+(q^+, W_c) = \max\{\mathrm{sim}_m(q^+, W_c) | \text{ admissible mapping } m\} \tag{2}$$

The similarity computation requires a search for the best possible admissible mapping $m : N_q \cup E_q \rightarrow N_c \cup E_c$ of nodes and edges of q^+ to those of W_c. A mapping is admissible, if it is type-preserving, partial, and injective. The core of the similarity model is a local similarity measure for semantic descriptions $\mathrm{sim}_\Sigma : \Sigma^2 \rightarrow [0,1]$. In our example domain, similarity values between semantic labels are derived from the data and task taxonomy that reflect the closeness of the concepts (refer to [5] for more details).

The restriction similarity sim^- is assessed with a binary measure that returns 1 if W_c does not fulfill the restriction q^-. If a restriction workflow q^- is a generalization of a partial workflow of W_c, the similarity is 0.

$$\mathrm{sim}^-(q^-, W_c) = \begin{cases} 0.0 & \text{if } \exists W' \subseteq W_c : q^- \sqsupseteq W' \\ 1.0 & \text{otherwise} \end{cases} \tag{3}$$

2.2 Conversational CBR

While in many CBR applications a complete description of the target problem is assumed to be available in advance, CCBR [1,15,2,9] particularly addresses the interactive nature of problem solving. In CCBR, the user only has to answer posed questions, which presumably requires less domain expertise than providing queries from scratch. The CCBR dialog [1] often begins by asking the user to specify a brief textual description of her problem. Subsequently, a dialog is started consisting of a sequence of questions to be answered by the user. A goal in a conversation is to pose relevant questions, potentially suitable to elaborate the query and to determine the most useful case efficiently. The user interface consists of a question and solution display. If the user answers a question, the dialog component extends the query based on the given answer, performs a similarity-based retrieval, and updates the solution and question displays. Users can delete or alter answers to previously asked questions at any time. By selecting a solution, the conversation terminates.

To perform the dialog, the case representation in CCBR is enriched with an additional set of question-answer pairs stated in natural language. Thus, case authoring can become more demanding in CCBR, because suitable questions need to be formulated. Hence, the automatic creation of questions is desirable and often achieved by deriving questions from case attributes. CCBR research focuses on enhancing case representation to include knowledge relevant for the questioning strategy [9], methods for question selection, and methods for dialog inferencing and termination [12,15,2,9]. Our research is based on the similarity variance measure proposed by Kohlmaier et al. [12]

which prefers questions, whose answers most probably have the highest influence on the similarity distribution of the most similar cases. CCBR finds its application mostly in analytical applications such as sequential diagnosis [17], customer help-desk support, or product recommendation [12,16]. Only very few approaches have been proposed that go beyond interactive query elicitation. Leake and Wilson [13] describe an approach for interactive case acquisition, retrieval, and adaptation for a specific design problem. Muñoz-Avila et al. [23] describe an interactive case-based planner which recursively applies a CCBR approach to guide the planning procedure of a hierarchical task network (HTN) planner. Weber et al. [27] propose a CCBR approach as part of their adaptive workflow system CBRflow. However, they use a traditional case representation consisting of manually defined question-answer pairs to explicitly acquire reasons and constraints for a specific workflow adaptation instance.

Today, no CCBR approaches exist that elicit questions to construct queries for cases represented as workflows as required for POCBR.

3 A Conversational POCBR Approach

Based on the generic CCBR approach, we now present a new approach that is particularly tailored to POCBR and thus named conversational POCBR (C-POCBR). In a nutshell, users are guided through the query process by a sequence of questions about their desired workflows. The more questions are answered, the more knowledge about desired and undesired properties is available, which is stored in an internal query for retrieval. A major focus is put on the automatic creation of questions to avoid that they need to be specified manually. For this purpose, we consider workflow fragments as characteristic properties of a workflow, which we refer to as features. The basic idea is to extract features from the workflows stored in the case base automatically, which are then used as the subject of questions. In order to conduct efficient conversations, we rank features by their ability to distinguish workflows from one another. Furthermore, identified relations between features enable to generate coherent follow-up questions and to infer irrelevant features based on already answered questions.

Figure 2 illustrates the conversational POCBR process. The process is divided into two phases. The *offline phase* comprises pre-computations for the initial setup. During this phase, extraction, ranking, and analysis of features takes place and a feature table is created. Subsequently, the actual conversation is conducted in the *online phase*. In the following, we describe both phases in more detail.

3.1 Offline Phase

At first, features are extracted based on the graph-based representation of the workflows. As those features will occur in the questions posed to the user, they must be as simple and understandable as possible. For this purpose, we consider various design guidelines investigated in related work [1,12,24]. In principle, a feature can be any fragment of a workflow. In a workflow, the smallest possible feature consists of a single workflow item. This can be a single node such as a data or a task node. More complex features can be created by extracting partial workflows. To derive questions on a more general level

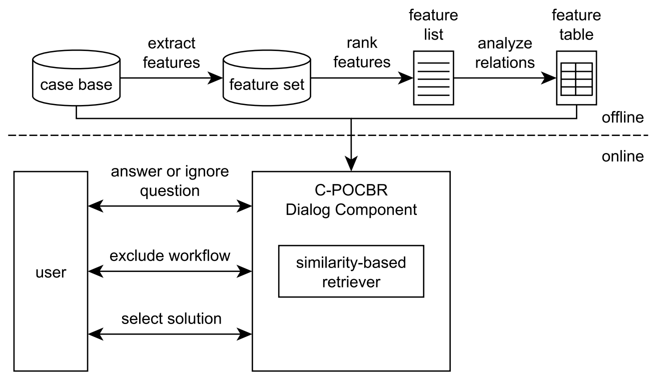

Fig. 2. Conversational POCBR Process

of detail, we apply a generalization algorithm [20], which generalizes semantic labels based on the domain taxonomies. The generalization produces a generalized workflow W^* from the original workflow W, from which more general features can be extracted. We extract and annotate two different kinds of features for each workflow W in the case base:

- *specific feature nodes* and *generalized feature nodes*, i.e., single nodes from W and single nodes for all generalizations within the taxonomy up to the respective node in the generalized workflow W^*
- *specific feature workflows* and *generalized feature workflows*, i.e., partial workflows (consisting of more than one node) from W and W^*, respectively

A *feature workflow* describes structural properties of a workflow. Its definition is inspired by the idea of streamlets [21] proposed for compositional adaptation in POCBR:

Definition 2. *For a workflow $W = (N, E)$ and a data node $d \in N^D$, a feature workflow W_d of W is a partial workflow $W_d = (N_d, E_d)$ that consists of all task nodes $N_d^T \subseteq N^T$ connected to d and connected by control-flow edges. Moreover, W_d comprises all data nodes $N_d^D \subseteq N^D$ connected to N_d^T and the subset of edges $E_d = E \cap ((N_d^T \times N_d^D) \cup (N_d^D \times N_d^T) \cup (N_d^T \times N_d^T))$ connecting the nodes.*

A feature workflow W_d is a workflow according to Definition 1 and consists of at least one task and one data node, i.e., d.

Figure 3 exemplifies all features extracted from a cooking workflow (see dotted rectangles). The specific workflow is depicted in the middle of the figure. Related features (such as specific and generalized features) are arranged near one another. For instance, the specific feature node *pepper* is related to the generalized feature node *flavoring*. Based on the taxonomy, an additional generalized feature node *spice* laying inbetween those two is extracted as well.

With respect to the cooking domain, we applied some domain-specific restrictions. For the sake of simplicity, we assume a simplified workflow structure without control-flow nodes and with the control-flow restricted to a single sequence of tasks. Thus,

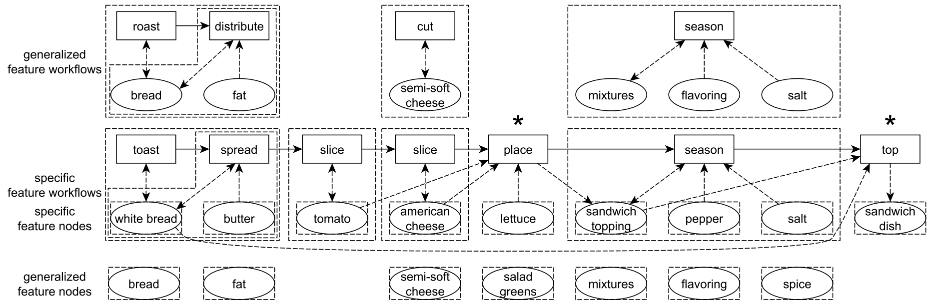

Fig. 3. Examples of a Workflow's Features

parallel or alternative sequences as well as cycles are omitted. For the feature extraction, we omit single task nodes as they are mostly of no relevance when considered on their own. In addition, to obtain easy-to-understand feature workflows, we exclude tasks (marked with "$*$") that produce new data by consuming other data.

In the second step of the offline phase, features are sorted in descending order by their ability to distinguish workflows from one another. By this means, we reduce the length of a conversation. We adopt the simVar measure by Kohlmaier et al. [12], which utilizes the similarity variance as a ranking criterion. It estimates the variance of the similarity of the most similar cases assuming that the value of the respective feature in the query is known. Features with a higher simVar value are preferred.

According to the POQL query (see sect. 2.1), the user can either select a feature as desired or undesired during the conversation. Thus, the similarity variance is precomputed for both situations. To calculate simVar, all similarities between the extracted features and the workflows stored in the case base must be computed. Each feature is added into the query part and the restriction part of an empty query, respectively. Then, for both queries, the similarities to each workflow from the case base are computed (according to equation 1) and cached. For a feature f and a case base CB, we define the similarity variance as follows:

$$\begin{aligned}\mathrm{simVar}(f, CB) &= \frac{1}{|CB|}\sum_{W \in CB}(\mathrm{sim}(Q_f, W) - \mu_f)^2 \\ \mu_f &= \frac{1}{|CB|}\sum_{W \in CB}\mathrm{sim}(Q_f, W)\end{aligned} \tag{4}$$

Q_f denotes the query consisting of the feature f. $\mathrm{sim}(Q_f, W)$ is the semantic similarity between the query Q_f and a workflow W. Moreover, μ_f is the arithmetic mean of the similarities between the query and each workflow W from the case base CB. The simVar value is computed in two ways for each feature f. The feature can either be added to the query part Q^+ of the query Q or it can be added to the restriction part Q^-. Thus, simVar^+ and simVar^- are computed separately and the average simVar is defined by the arithmetic mean:

$$\mathrm{simVarMean}(f, CB) = \frac{\mathrm{simVar}^+(f, CB) + \mathrm{simVar}^-(f, CB)}{2} \tag{5}$$

Initially, features are ranked by their $\mathrm{simVarMean}$ value in descending order. Features with a value of 0 are ignored since they are not suitable to distinguish workflows from one another as they are part of every workflow in the case base.

In the next step, relations between features are analyzed. For each feature f all related features are determined and stored in a feature table FT. Formally, the set of related features $F_{rel}(f)$ of a feature $f \in FT$ contains those features $g \in FT$ that share a common partial workflow with f which is either a generalization of f or g:

$$F_{rel}(f) = \{g \in FT | \exists f' \subseteq f : (f' \sqsupseteq g \vee g \sqsupseteq f') \vee \exists g' \subseteq g : (g' \sqsupseteq f \vee f \sqsupseteq g')\}$$

Related features can be differentiated by their number of nodes and by their generality of nodes. A feature may have related features that are *larger*, *equally large*, or *smaller* as well as related features which are *more specific*, *equally specific*, or *more general*. For example, for the feature workflow $f_1 = \{$*slice, ham*$\}$, the related feature $g_1 = \{$*cut, meat*$\}$ is *more general* and *equally large* while the feature $g_2 = \{$*parma-ham*$\}$ is *more specific* and *smaller*.

3.2 Online Phase

The online phase of the C-POCBR dialog component is described in algorithm 1. The dialog component iteratively creates and displays questions until the user selects a workflow or until stopping criteria are fulfilled. The set of candidate workflows is updated, if a question is answered or if a workflow is excluded by the user.

The dialog component is always initialized with the full case base CB as well as with the feature table FT. The dialog starts with an empty query Q[1]. Initially, the set of candidate features CF, i.e., relevant features to be asked in a question, is the full set of features from the feature table. The initial set of candidate workflows CW encompasses the whole case base.

In the main loop, the dialog component selects a question based on the candidate features CF considering the $\mathrm{simVarMean}$ scoring, the previously answered questions, as well as the feature relations (details are described below). Each question involves one or in certain cases several candidate features and is displayed to the user. Then, the user has four options to react:

1. *Ignoring a question:* In this case, the feature being subject of the question as well as larger related features and more specific related features are removed from the set of candidate features and the question selection determines the next best question.
2. *Answering a question:* If a question is answered by the user, the query Q is extended and a similarity-based retrieval with the extended query on the current set of candidate workflows CW is performed. After each retrieval, workflows that are less similar than the average of all workflow's similarities are removed from CW and thus are not included in subsequent retrievals. By this means, only the most suitable workflows are retained with respect to the current query. The workflow with the highest similarity from CW is displayed to the user. In addition, the table

[1] In principle an initial pre-modeled query could be used as well, but we have not yet investigated this option.

```
Input  : CB: Case Base, FT: Feature Table
Output: A solution workflow S

C-POCBR Dialog Algorithm(CB, FT)
    Q ← ∅, CW ← CB, CF ← FT
    repeat
        q ← questionSelection(Q, CF)
        displayQuestion(q)
        if userIgnoresQuestion then
            CF ← updateCandidateFeatures(CF, q)
        end
        if a ← userAnswersQuestion then
            Q ← extendQuery(Q, a)
            CW ← retrieveAndDisplayCandidateWorkflows(CW, Q)
            CF ← updateCandidateFeatures(CF, CW, a)
        end
        if W ← userExcludesWorkflow then
            CB ← CB \ {W}
            CW ← retrieveAndDisplayCandidateWorkflows(CB, Q)
            CF ← updateCandidateFeatures(CF, CW)
        end
    until S ← userSelectsSolution(CW) OR stoppingCriteria
    return S
```

Algorithm 1: C-POCBR Dialog Algorithm

of candidate features CF is updated as well. Only features contained in the candidate workflows CW remain in CF, which ensures that only relevant questions are posed. Thus, with each retrieval performed, CW and CF are further reduced. In addition, the ranking of the remaining features CF is updated according to simVarMean (see equation 5) by using CW instead of CB.

3. *Excluding a suggested workflow:* The user may explicitly exclude a suggested workflow as possible solution, which removes the workflow from the case base CB (only temporary for this dialog) and triggers a new retrieval. As a consequence, it is likely that more candidate workflows CW than before exist because of the lower average similarity of all workflows to the current query. Consequently, more candidate features CF may become available.
4. *Selecting a solution:* If the user selects a workflow as the desired solution the retrieval terminates successfully.

We now describe in more detail the question selection method applied. We provide three major types of questions, which are depicted in Table 1. Based on the ranking of the candidate features, the subject matter of a question is determined. If the user answers that the suggested feature is desired, specific follow-up questions are selected in the subsequent iterations of the main loop. Those follow-up questions aim at further refining the previous question asked. Follow-up questions are derived from the set of related features stored in the feature table.

At the beginning of a conversation the highest ranked feature from the candidate features is suggested in a feature question (FQ). This type of question is not related to previously suggested features and it will be asked as long as the user selects the suggested feature as irrelevant or undesired.

In case of a previously answered FQ as desired, a first follow-up question, i.e., a specialization question (SQ), is posed suggesting one or (if available) several equally large but more specific features. Again, the features are sorted by their $\mathrm{simVarMean}$

Table 1. Question Sequence in a Conversation

Order	Question Type	Subject Matter	Example
1.	initial feature question (FQ)	highest ranked feature	Q: Is $\{meat\}$ a desired feature? A: desired, undesired, irrelevant
2.	follow-up specialization question (SQ)	more specific feature(s)	Q: Is there a suitable specialization for $\{meat\}$? $\{poultry\}, \{ham\}, \{chicken\}, \ldots$ A: apply, select undesired feature(s), irrelevant
3.	follow-up enlargement question (EQ)	larger feature(s)	Q: Is there a suitable enlargement for $\{chicken\}$? $\{shred, chicken\}, \{chop, chicken\}, \ldots$ A: apply, select undesired feature(s), irrelevant

value. The user can choose a specialization, select features as undesired, or mark all specializations as irrelevant. This type of question is repeated as long as the user chooses specializations and as long as further specializations are available.

Following the SQs, an enlargement question (EQ) is displayed to the user that suggests, if available, larger and not more general features than the previously selected and/or specialized feature. More general features are omitted since they would widen the current context of the previous feature. Just as in SQ, the user has three different options: choose an enlargement, select an enlargement as undesired, or mark all enlargements as irrelevant. If no more EQs are available, the next initial FQ is selected, addressing a new and potentially unrelated subject matter.

When the set of candidate features CF is updated due to an ignored or answered question, irrelevant features can be inferred based on the relations between features. If a question is marked as irrelevant, all the related features (e.g., more specific and larger features) are marked as irrelevant, too. If suggested features are selected as undesired, they are added to the restriction part of the current query and related irrelevant features are no longer considered as candidate features, to prevent the system from repetitively asking the user what she does not like. If a feature is marked as desired, also related features such as more general features are removed from the candidates table. If a user chooses a specialization or an enlargement, the target feature that is already present in the query is replaced with the new feature. In this event, related features of the target feature without those that are still relevant for the new feature are removed from the feature table.

4 Evaluation

We now describe the evaluation comparing the presented C-POCBR approach with a traditional POCBR approach in which the user models a POQL query (see sect. 2.1) manually using a query editor. The evaluation aims at testing three hypotheses and is conducted with a simulated user as well as with human users. Hypothesis H1 states that if the user's requirements can be fulfilled by a workflow from the case base, then this workflow must be retrievable by correctly answering all questions posed. However, as questions are created automatically, real users may give wrong answers due to misunderstood questions. Therefore, hypothesis H2 targets the basic utility from the user's perspective. Furthermore, hypothesis H3 relates to the user interaction effort by comparing the conversational approach with the query modeling approach. The following hypotheses are formulated under the assumption that the user's requirements can be fulfilled entirely by one workflow in the case base:

H1 The desired workflow is retrieved with C-POCBR when all questions are answered correctly.
H2 The C-POCBR dialog enables users to retrieve the desired workflow.
H3 C-POCBR reduces the communication effort required to retrieve the desired workflow.

4.1 Evaluation Setup

For the experiments, we used the already existing CookingCAKE system [19], which is part of the CAKE framework[2]. It already includes a graphical POQL editor, which is used as implementation of the POCBR approach. In addition, we implemented the C-POCBR approach[3] as an extension of CookingCAKE. Thus, both systems to be compared use the same case base, similarity measures, and retrieval implementation[4].

In all experiments, we use a case base of 61 cooking workflows that describe the preparation of sandwich recipes. We created search scenarios for the evaluation that describe queries in plain text to be given to the users. According to the structure of POQL queries, a search scenario describes a required workflow together with several restriction workflows. Queries are constructed in a semi-automatic process in which each workflow from the case base is turned into a textual description of a search scenario that contains sufficient information to unambiguously specify it. In this process, feature workflows (see Definition 2) are added iteratively to the query either as requirement or as restriction until a unique specification is obtained. In a last step, textual descriptions are written by hand based on the constructed queries. In total for 60 workflows (out of 61) an appropriate search scenario description could be constructed.

4.2 Experimental Evaluation

Hypothesis H1 is tested using an experiment with a simulated user, which automatically answers the posed questions of the C-POCBR approach correctly. We adopt the methodology by Aha et al. [1], who evaluate a conversational retrieval with a leave-one-in cross validation. Consequently, in each of the 60 search scenarios the corresponding target workflow remains in the case base. During a conversation, the algorithm ignores questions that are not relevant in the specific search scenario; all other questions are answered according to the described search scenario. The conversation for a scenario is considered successful, if the proposed best fitting workflow that is displayed during the conversation is equal to the workflow from which the search scenario was derived. It turned out that the target workflow is retrieved in each of the 60 search scenarios, which fully confirms hypothesis H1. In average, 10.25 questions were asked in the dialog.

[2] See `cake.wi2.uni-trier.de`
[3] Online demo available at `cookingcake.wi2.uni-trier.de/conversation`
[4] During the experiments, the available adaptation methods of CookingCAKE are not used.

Hypotheses H2 and H3 are tested in experiments with eight human users who simultaneously performed the experiments on different computers while all interactions are being logged. After a familiarization phase in which the users are introduced to the usage of the POCBR and the C-POCBR approach, we randomly distributed the eight participants evenly to one of two groups. Furthermore, we randomly chose four textual search scenarios of similar size. Each user evaluated both approaches on the basis of the four scenarios. Thus, each approach was used 16 times in total. The first group evaluated the POCBR approach with two scenarios and conducted the C-POCBR with the other scenarios afterwards. The second group evaluated both approaches in the opposite order. Finally, all users filled in a questionnaire capturing their subjective experience during the experiment.

Table 2. Experimental Results: Avg. Values Across all Successful Retrievals and Users

	POCBR	C-POCBR
number of successful conversations	15/16	14/16
total conversation time	5:34 min.	5:40 min. (30 questions)
required conversation time	4:46 min.	2:16 min. (9 questions)

Table 2 summarizes selected measures extracted from the logged experiment data. The values shown are average results over all successful queries and all users for the POCBR and the C-POCBR approach. The *number of successful conversations* shows that only a few of the 16 query runs were not successful as the target workflow was not identified and selected by a user. Thus, hypothesis H2 can be confirmed.

To assess the communication effort, the conversation time used in the POCBR and the C-POCBR approach were compared. In addition, the number of questions posed in the C-POCBR approach were determined. The *total conversation time* is the time span from the start of the conversation (in C-POCBR) or point in time when the user begins to enter the query in the POQL editor (in POCBR) until the desired workflow is retrieved and identified by the user. For POCBR and C-POCBR those time spans are quite comparable. We discovered that users following the POCBR approach tend to completely model the given query scenario, before they start the retrieval for the first time. Sometimes, the first retrieval does not lead to the desired workflow and modifications of the query have to be performed until the desired workflow is retrieved. In C-POCBR the users follow the dialog and investigate the presented workflow. In average 30 questions are answered before the user identified that the desired workflow is displayed. When analyzing these results in more detail, we found out that quite often the desired workflow is presented to the user but she did not recognize it as the desired result. In those cases, the dialog could have been terminated earlier if the user would have analyzed the displayed result more thoroughly. We analyzed this effect in detail and determined the *required conversation time*, i.e., the time until the desired workflow is displayed the first time in the dialog loop. We also determined the number of questions the user was asked during this period. We can see that if users would have checked the displayed workflows more thoroughly, the C-POCBR approach could have been more than twice as fast as the POCBR approach. The fact that this does not happen is an indication that the workflow presentation in the C-POCBR implementation needs additional explanation functions that better allow the user to identify how the presented workflow relates to the answers of her query. However, with respect to the dialog component, we consider hypothesis H3 at least partially confirmed.

Figure 4 shows the results obtained from the questionnaires that the users filled after they performed the conversation. The values are average ratings over 16 conversations with C-POCBR. Users consider the majority of the posed C-POCBR questions to be comprehensible and relevant. Moreover, the retrieval results were also rated to be reasonable with respect to the answered questions. The results indicate that the automatic creation of questions provides useful questions for the conversation. Users did state different opinions on whether the question sequences are sensible.

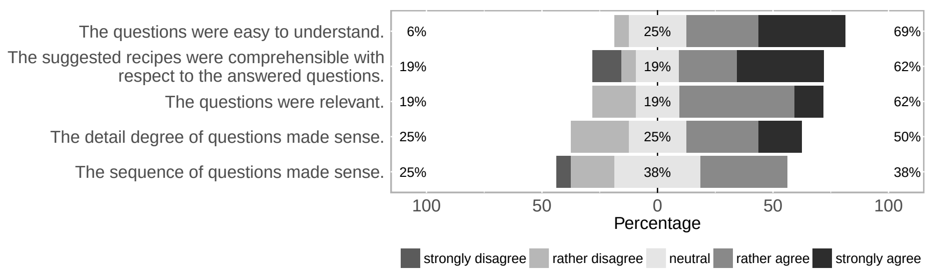

Fig. 4. Average User Ratings of C-POCBR Conversations on a Five-Point Likert Scale

In addition, users compared both approaches at the end of all experiments. Three out of eight users stated that the C-POCBR approach could be enhanced by allowing the user to model an initial query. Five out of eight participants stated the POCBR approach to be easier to use than the C-POCBR approach. Users indicated that sometimes general concepts in questions were difficult to understand as the subsumption of concepts was not always clear to them. Six out of eight participants claimed that POCBR is faster than C-POCBR, although this subjective assessment clearly conflicts with the measured values. One reason may be that the users are more actively involved during the modeling with the POCBR approach.

5 Conclusions and Future Work

We presented a novel approach to conversational POCBR that conducts an interactive dialog with users to facilitate the retrieval of workflows. To save effort for defining suitable questions, a method for the automatic creation of questions based on extracted features was described. Our work showed that those features are meaningful subjects of questions and that they are suitable to distinguish workflow cases from one another. The quality and performance of conversations was improved by ranking, analyzing, and selecting relevant features. We evaluated the approach with simulated and real users and showed that when questions are always answered correctly, the desired workflow is found in a straight-forward manner. Furthermore, our results indicate that the conversational query process has the potential to be faster than the traditional query approach and thus is able to reduce the communication effort for users.

The evaluation revealed some issues that should be investigated in future work. We discovered that users did not recognize the target workflow immediately and that more general questions caused problems of comprehension. Thus, the presentation and explanation of workflows and features should be improved. For the sake of simplicity, we omitted control-flow elements such as loops, parallel, and alternative sequences and restricted the description of nodes to their semantic label. Thus, it is desirable to evaluate the approach in domains with more complex workflows. In such domains, we assume that the conversation more strongly outperforms the modeling due to the increased complexity involved in modeling. In addition to POCBR domains, we assume the questioning strategy presented in this work to be also applicable more broadly in CCBR with complex case representations, provided that feature vocabularies are organized in a hierarchy and co-occurring features are identified. Also, future work should investigate how adaptability of workflows can be considered during a conversation. By this means, interactive retrieval could be combined with interactive adaptation to provide an even more powerful problem solver for users.

Acknowledgments. This work was funded by the German Research Foundation (DFG), project number BE 1373/3-3.

References

1. Aha, D.W., Breslow, L.A., Muñoz-Avila, H.: Conversational case-based reasoning. Applied Intelligence 14(1), 9–32 (2001)
2. Aha, D.W., McSherry, D., Yang, Q.: Advances in conversational case-based reasoning. Knowledge Engingeering Review 20(3), 247–254 (2005)
3. Awad, A.: BPMN-Q: a language to query business processes. In: Enterprise Modelling and Information Systems Architectures - Concepts and Applications. LNI, vol. P-119, pp. 115–128. GI (2007)
4. Beeri, C., Eyal, A., Kamenkovich, S., Milo, T.: Querying business processes. In: Dayal, U., Whang, K., Lomet, D.B., Alonso, G., Lohman, G.M., Kersten, M.L., Cha, S.K., Kim, Y. (eds.) Proceedings of the 32nd International Conference on Very Large Data Bases, 2006. pp. 343–354. ACM (2006)
5. Bergmann, R.: Experience Management: Foundations, Development Methodology, and Internet-based Applications. Springer-Verlag, Berlin, Heidelberg (2002)
6. Bergmann, R., Gil, Y.: Similarity assessment and efficient retrieval of semantic workflows. Inf. Syst. 40, 115–127 (2014)
7. Dijkman, R.M., Dumas, M., García-Bañuelos, L.: Graph matching algorithms for business process model similarity search. In: Dayal, U., Eder, J., Koehler, J., Reijers, H.A. (eds.) Business Process Management, 7th International Conference, BPM 2009. LNCS, vol. 5701, pp. 48–63. Springer (2009)
8. Dijkman, R.M., Rosa, M.L., Reijers, H.A.: Managing large collections of business process models - current techniques and challenges. Computers in Industry 63(2), 91–97 (2012)

9. Gu, M., Aamodt, A.: A knowledge-intensive method for conversational CBR. In: Muñoz-Ávila, H., Ricci, F. (eds.) Case-Based Reasoning, Research and Development, ICCBR 2005. LNCS, vol. 3620, pp. 296–311. Springer (2005)
10. Kapetanakis, S., Petridis, M., Knight, B., Ma, J., Bacon, L.: A case based reasoning approach for the monitoring of business workflows. In: Bichindaritz, I., Montani, S. (eds.) Case-Based Reasoning. Research and Development, ICCBR 2010,. LNCS, vol. 6176, pp. 390–405. Springer (2010)
11. Kim, J., Suh, W., Lee, H.: Document-based workflow modeling: a case-based reasoning approach. Expert Systems with Applications 23(2), 77 – 93 (2002)
12. Kohlmaier, A., Schmitt, S., Bergmann, R.: A similarity-based approach to attribute selection in user-adaptive sales dialogs. In: Aha, D.W., Watson, I. (eds.) Case-Based Reasoning Research and Development, ICCBR 2001. LNCS, vol. 2080, pp. 306–320. Springer (2001)
13. Leake, D.B., Wilson, D.C.: Combining CBR with interactive knowledge acquisition, manipulation and reuse. In: Case-Based Reasoning and Development, ICCBR '99. pp. 203–217. Springer (1999)
14. Madhusudan, T., Zhao, J.L., Marshall, B.: A case-based reasoning framework for workflow model management. Data and Knowledge Engineering 50(1), 87–115 (2004)
15. Mcsherry, D.: Increasing dialogue efficiency in case-based reasoning without loss of solution quality. In: IJCAI-03. pp. 121–126. Morgan Kaufmann (2003)
16. McSherry, D.: Explanation in recommender systems. Artif. Intell. Rev. 24(2), 179–197 (2005)
17. McSherry, D.: Conversational case-based reasoning in medical decision making. Artificial Intelligence in Medicine 52(2), 59–66 (2011)
18. Minor, M., Montani, S., Recio-Garca, J.A.: Process-oriented case-based reasoning. Information Systems 40, 103 – 105 (2014)
19. Müller, G., Bergmann, R.: CookingCAKE: A framework for the adaptation of cooking recipes represented as workflows. In: Kendall-Morwick, J. (ed.) Workshop Proceedings from (ICCBR 2015). CEUR, vol. 1520, pp. 221–232. CEUR-WS.org (2015)
20. Müller, G., Bergmann, R.: Generalization of workflows in process-oriented case-based reasoning. In: Russell, I., Eberle, W. (eds.) Proceedings of the 28th Int. Florida Artificial Intelligence Research Society Conference, FLAIRS 2015. pp. 391–396. AAAI Press (2015)
21. Müller, G., Bergmann, R.: Learning and applying adaptation operators in process-oriented case-based reasoning. In: Hüllermeier, E., Minor, M. (eds.) Case-Based Reasoning Research and Development, ICCBR 2015. LNCS, vol. 9343, pp. 259–274. Springer (2015)
22. Müller, G., Bergmann, R.: POQL: A new query language for process-oriented case-based reasoning. In: Bergmann, R., Görg, S., Müller, G. (eds.) Proceedings of the LWA 2015. CEUR Workshop Proceedings, vol. 1458, pp. 247–255. CEUR-WS.org (2015)
23. Muñoz-Avila, H., Aha, D.W., Breslow, L.A., Nau, D.S.: HICAP: an interactive case-based planning architecture and its application to noncombatant evacuation operations. In: Proc. of the 16th Nat. Conf. on Artif. Intell., AAAI/IAAI 1999. pp. 870–875. AAAI Press (1999)
24. Richter, M.M., Weber, R.O.: Case-Based Reasoning - A Textbook. Springer (2013)
25. Störrle, H., Acretoaie, V.: Querying business process models with VMQL. In: Proceedings of the 5th ACM SIGCHI Annual International Workshop on Behaviour Modelling - Foundations and Applications. pp. 4:1–4:10. BMFA '13, ACM, New York (2013)
26. Van Der Aalst, W.M.: Business process management: a comprehensive survey. ISRN Software Engineering 2013 (2013)
27. Weber, B., Wild, W., Breu, R.: Cbrflow: Enabling adaptive workflow management through conversational case-based reasoning. In: Funk, P., González Calero, P.A. (eds.) Advances in Case-Based Reasoning, LNCS, vol. 3155, pp. 434–448. Springer (2004)