Project EVER: Extraction and Processing of Procedural Experience Knowledge in Workflows

Ralph Bergmann[1], Mirjam Minor[2], Gilbert Müller[1], Pol Schumacher[3]

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

[2]Goethe University, Department of Informatics
60629 Frankfurt/Main, Germany
[minor]@informatik.uni-frankfurt.de
http://www.wi.informatik.uni-frankfurt.de

[3]Celonis SE
81373 München, Germany
p.schumacher@celonis.com

Abstract. The goal of the EVER project (Extraction and Processing of Procedural Experience Knowledge in Workflows), funded by the German Research Foundation, is to investigate new methods in Process-Oriented Case-Based Reasoning and related fields for extracting, representing, and processing procedural experiential knowledge in Internet communities. This paper summarizes the main achievements of the first funding period of this project. The main research addressed the extraction of workflows from textual sources in Internet Communities, the similarity-based retrieval of workflows for a particular goal of a user, and the automatic adaptation of retrieved workflows.

1 Introduction

Today's Social Web allows people in a community of practice to post their own experiences in a diversity of content repositories such as blogs, forums, or Q&A websites [22]. However, today there is no automated support for reusing these rich collections of personal experience. Current search functions available merely consider experience as text to be indexed as any other text and searched as any other document. The objective of the EVER project (Extraction and Processing of Procedural Experience Knowledge in Workflows) is the analysis, the development, and the experimental application and evaluation of new knowledge-based methods, particularly from process-oriented case-based reasoning (POCBR) [7,8,13], information extraction, and machine learning.

The EVER project is funded by the German Research Foundation (DFG) and led by the Universities of Trier and Frankfurt. During the first funding period from 2011 – 2016, the project focused on the reuse of procedural experiences published by private people in Internet Communities such as cooking web sites. In this regard, it was investigated whether workflow technology and POCBR can help to analyze and reuse procedural experiential knowledge from these Internet communities. In the course of this project, several significant contributions to POCBR research have been made, particular in the fields of workflow extraction from text, workflow retrieval, and workflow adaptation. The methods have been consistently evaluated in the domain of cooking recipes. This paper presents a summary of those achievements and shows, how they are connected to draw an overall picture of POCBR.

2 Architecture for POCBR

The overall architecture of our POCBR approach in the EVER project is illustrated in Fig. 1. First, procedural experience is gathered from Internet communities (or alternatively from repositories of workflows in the Business Process Model and Notation (BPMN) format) and stored in a suitable representation. More precisely, a case-base of semantic workflows is constructed by extracting workflows from textual sources. The workflows in this repository can be reused, i.e., for a particular problem situation a suitable process represented as workflow can be suggested. This is primarily achieved by retrieving the best matching workflow from the repository. If required, the workflow is automatically adapted according to the requirements and restriction in the particular scenario. The required adaptation knowledge is automatically learned from the case-base. In addition to these steps (which basically correspond to the phases of the R^4-CBR cycle [1]) we also include specific methods for user interaction, enabling a conversational POCBR approach [30]. In the following section, we will summarize our research related to various components of the architecture.

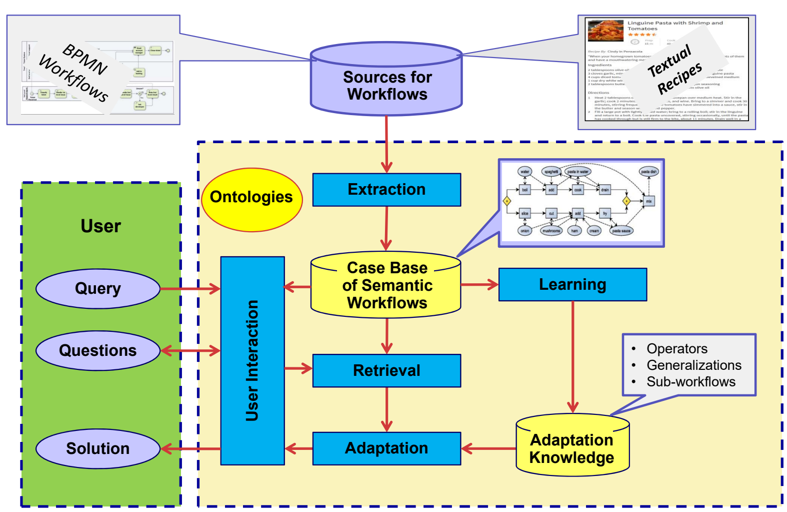


Fig. 1. EVER architecture for POCBR.

3 Semantic Workflows as Case Representation

In order to formalize procedural experience, we employed *semantic workflows* as case representation. Broadly speaking, a workflow consists of a set of *activities* (also called *tasks*) combined with *control-flow structures* like sequences, parallel (AND) or alternative (XOR) branches, as well as repeated execution (LOOP). In addition, tasks consume and produce certain *data items*, or objects, depending on the workflow domain (e.g., ingredients in the cooking domain). Tasks, data items, and relationships between the two form the *dataflow*. For the given application domain, a cooking workflow describes the preparation steps required and ingredients used in order to prepare a particular dish. Here, the tasks represent the cooking steps and the data items refer to the ingredients being processed by the cooking steps. An example cooking workflow for a sandwich recipe is illustrated in Fig. 2.

As a basis for the project, we developed a graph-based representation of semantic workflows that further enables to compute similarities between two workflows [2]. In a semantic workflow the individual workflow elements are annotated with ontological information. In particular, tasks and data nodes are linked into domain-specific task and data ontology and can be further specified by properties, e.g. to represent context factors or resources. In the cooking domain a taxonomy of cooking ingredients and cooking steps is consequently constructed. Within the developed POCBR system CAKE ontologies are represented in an object-oriented fashion while a (partial) transformation into OWL has been developed.

4 Automatic Workflow Extraction from Text

Prior to reasoning with procedural knowledge, the available experience is transformed into a suitable and formal process representation. More precisely, we developed a novel framework for automated workflow extraction [23], which transforms textual descriptions of processes into semantic workflows. Here, from the textual description the

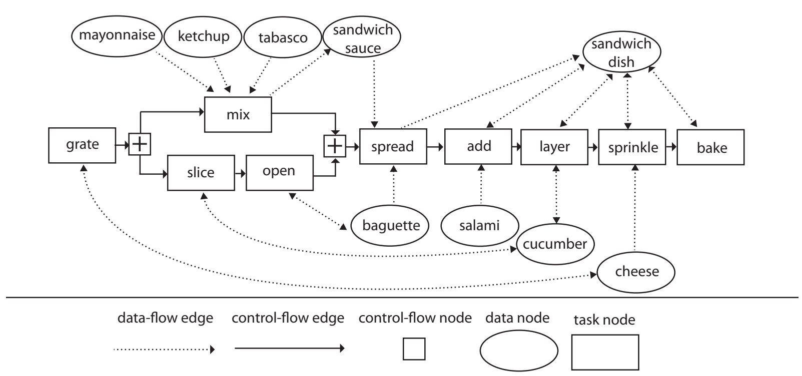


Fig. 2. Example workflow from the domain of cooking.

preparation step (saute) and the ingredients consumed (onion, green pepper) are identified and transformed into a workflow fragment. A stepwise extraction of the entire process description thereby constructs a complete workflow.

The developed extraction methods are able to identify the activities of the process [27], organizing them in a control flow [24], and enriching the control flow by data flow information [26]. For the latter, we additionally investigated an alternative approach to complete missing data-flow information [21] by learning completion operators from a set of revised workflows within the repository. The framework implements a pipe-and-filters architecture. Different extraction steps can be implemented as independent components (filters), which can be composed to an extraction sequence (pipe). Consequently, this allows the flexible reuse and exchange of filters. For the basic linguistic analysis of the textual descriptions, methods from natural language processing have been applied. We used the developed framework to extract a repository of cooking workflows from 35,000 online recipes. The source code of the workflow extraction framework as well as the repository are available for download under open source licenses[1].

5 Similarity-based Workflow Retrieval

For reusing the extracted procedural experiences, the workflow repository is searched for the best matching workflow using similarity-based retrieval methods. In order to capture the scenario or problem situation, a specific workflow query language POQL [20] was developed. The query may include single workflow elements as well as entire workflow fragments (e.g., sub-workflows), which are either marked as desired or undesired. Furthermore, also generalized workflow elements such as generalized tasks and generalized data items can be specified.

POQL can then be used to trigger a similarity-based retrieval for the workflow best matching the requirements and restrictions defined, for which several methods have been developed. Most basically, we developed a semantic similarity measure for semantic workflows [2] which is based on a workflow ontology. The semantic similarity of workflows is defined as an optimization problem for the mapping of workflow elements from the query to the mostly similar elements of case workflow. Various search algorithms and respective heuristics have been developed to efficiently compute this similarity [2]. As an alternative approach to the developed semantic similarity measures, we investigated similarity measures based on the trace index of a workflow [25]. A trace index is created by analyzing all potential execution traces. Similarity of workflows is then computed by comparing the trace indices of workflows.

Moreover, several methods have been developed aiming at improving the efficiency of similarity search within the repository, which is particularly important when the workflow repository grows. For this purpose, a two-level retrieval method has been developed [6]. Additionally, we investigated new methods for workflow clustering based on the developed semantic similarity measures [4]. In particular, we developed various algorithms that explore this cluster structure as an index structure for retrieval [14].

[1] `www.wi.informatik.uni-frankfurt.de/index.php?option=com_content&view=article&id=126`

6 Automatic Workflow Adaptation

We aim at supporting the users in situations in which the best matching workflow from the case base does not sufficiently fulfill the query. This requires that the workflow is automatically adapted according to the given restrictions and requirements, i.e., workflow elements or fragments are added or deleted according to the particular needs.

For that purpose, we developed several workflow adaptation methods. Since such adaptation methods usually require a significant amount of domain-specific adaptation knowledge, we additionally developed new methods that allow to automatically learn the required adaptation knowledge from the workflow repository. Hence, we distinguish between a learning phase of adaptation knowledge and a problem solving phase in which for a given query the best matching workflow is adapted such that it matches the particular problem scenario at best (see Fig. 3). The developed adaptation methods can mostly be classified into *transformational adaptation*, *compositional adaptation* and *adaptation by generalization* [9].

More precisely, we developed two *transformational adaptation* methods, which differ in the representation of the adaptation knowledge. In both approaches, adaptation of workflow cases is performed by chaining several transformation steps $w \stackrel{\alpha_1}{\rightarrow} w_1 \stackrel{\alpha_2}{\rightarrow} \ldots \stackrel{\alpha_n}{\rightarrow} w_n = w'$ which iteratively transform the retrieved workflow w towards the adapted workflow w'. This process is a search process with the goal to achieve an adapted workflow which is as similar as possible to the query. Thus adaptation is considered an optimization problem. In *case-based adaptation* [10] the individual transformation steps are represented as so called *adaptation cases* which are learned automatically from the workflow repository [12]. An adaptation case represents a particular previous adaptation scenario by capturing the information about how to transform a particular

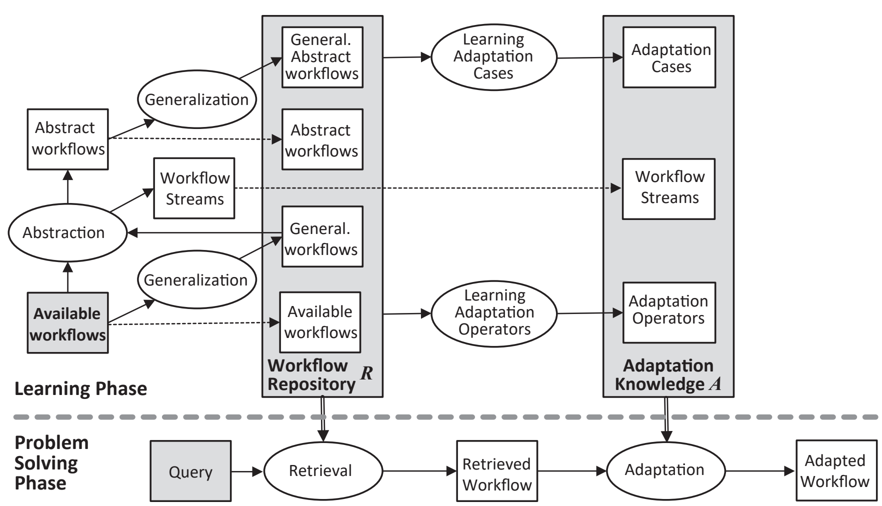


Fig. 3. Integration of adaptation approaches.

origin workflow to a corresponding goal workflow. It can be applied if it matches at a certain position within the workflow to be adapted. The *operator-based adaptation* [19] represents the individual transformation steps as so called *workflow adaptation operators*. They are denoted in a STRIPS-like manner, i.e., by specifying a fraction of the workflow to be deleted and a fraction to be added to the workflow. A learning algorithm was also developed that allows to automatically acquire adaptation operators from pairs of similar cases from the workflow repository.

In addition, we developed a method for *compositional and hierarchical adaptation*. It is based on the idea that each workflow can be decomposed into meaningful sub-workflows called *workflow streams* [16]. Such workflow streams can be automatically discovered from the workflow repository. Workflow streams represent valuable adaptation knowledge which is used as "chunks" that can be inserted or used as replacement during compositional adaptation. Compositional adaptation is also implemented as a search process, but it replaces larger portions of a workflow than the transformational adaptation approaches. In addition, workflow streams provide a means for abstraction. An abstracted workflow, is a structurally simplified workflow, i.e., containing fewer nodes or edges. *Abstraction* is achieved by replacing each discovered workflow stream in a case by a single abstract task. As further background knowledge for abstraction, domain-specific abstraction rules have been introduced, describing how to map a sub-workflow to a domain-specific abstract task linked with an appropriate semantic description from the domain ontology. The abstraction rules consist of elementary abstractions such as sequential abstraction, block abstraction, and elimination. Abstraction can be performed hierarchically, i.e., a rule can abstract also non-primitive tasks. During problem solving, abstract cases (which are also stored in the workflow repository) can be retrieved and reused by refining the occurring abstract tasks, e.g. by using workflow streams as refinement operators, best suited to the current query.

Finally, *generalization and specialization* was investigated as a third adaptation approach [18]. A *generalized workflow* is structurally identical to the base workflow but the semantic descriptions of task and data items are generalized. We generalize a workflow by considering a set of similar workflows as training samples and employ the ontology as generalization hierarchy from which generalized semantic descriptions are selected. The computed generalized cases are added to the workflow repository. During problem solving, adaptation is performed by specializing a previously generalized workflow in a manner, best suited to the current query.

The adaptation methods just described have also been integrated [17] as shown in Fig. 3. In particular adaptation cases and adaptation operators can be learned not only from the available concrete-level cases, but also from cases resulting from abstraction or generalization. Also, case generalization can be performed on top of abstraction. As a consequence, a large spectrum of possible ways arise for learning adaptation knowledge. As a result of the integrated learning process, the workflow repository R consists of four type of cases: 1. the available concrete cases, 2. generalized cases, 3. abstracted cases, and 4. generalized abstract cases. The adaptation knowledge A consists of adaptation operators, adaptation cases, and streams. During problem solving, i.e., when a new workflow for a given new query must be determined, the most similar (generalized/abstract) workflow from the workflow repository R is retrieved. Then, during adaptation the available adaptation knowledge from A is applied in a local search process in order to achieve an adapted workflow which is most similar to the query.

The availability of the previously introduced adaptation methods changes the utility of the workflows stored within the repository. A workflow with a lower similarity value during retrieval might more likely be adaptable to the particular problem situation. Hence, we developed a novel approach for the adaptation-guided retrieval of workflows [5], aiming at identifying the workflow which can at best be adapted to the particular situation during retrieval. The approach basically assesses the adaptability of the workflows by performing several example adaptations.

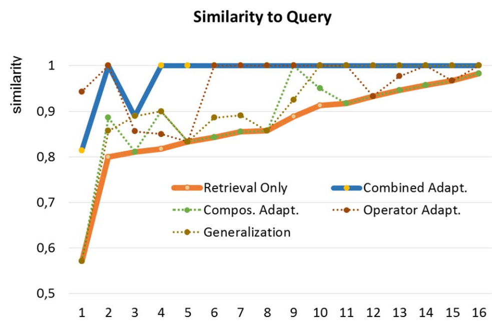
7 Implementation and Experimental Evaluation

The approaches developed throughout the whole project have been continuously integrated in a prototype system called CookingCAKE[2] for participation in the Computer Cooking Contest in 2011 [29], 2012 [3], 2014 [15,28] and 2015 [17]. Using the previously sketched retrieval and adaptation methods, CookingCAKE demonstrates the generation of sandwich recipes considering ingredients and preparation steps that are desired or undesired. A large number of experimental evaluations have been performed which are reported in the papers describing the individual methods. In the following, we show some preliminary experimental results of a preparatory study we performed in the process of the preparation of a more comprehensive systematic trial. In this experiment we used a case-base of 60 extracted pasta recipe workflows that have been further improved manually. In a study with human users of CookingCAKE we elaborated 16 realistic queries representing the user's desires for cooking. CookingCAKE was used in various conditions (pure retrieval, use of all adaptation methods in isolation, and the combined adaptation approach) to produce the desired recipe workflow. We compared the system (see Fig. 4) in the various conditions a) by assessing the similarity of the resulting solution workflow to the query and b) by asking the users to assess query fulfillment and quality of the resulting recipes on a 5-point Likert scale. The indicated values for workflow quality and query fulfillment are the difference resulting from adaptation, compared to pure retrieval, thus indicating the impact of adaptation. These initial results indicate that the adaptation methods improve the workflow w.r.t. the degree to which the requirements in the query are fulfilled. On the other hand, workflow quality is decreased to a certain degree. Overall, the combined approach performs best and in particular only leads to a minimal reduction of the quality. These results look promising, but a final assessment and a clear view of the various benefits and shortcomings of the methods can only become substantiated after the final trial is completed.

8 Future Work

The EVER project is currently in its second funding phase (2017 – 2020). During this phase, we aim at working on four novel issues.

[2] https://www.uni-trier.de/index.php?id=40545



Overall Impact on Workflow Quality and Query Fulfillment

Adaptation Approach	Workflow Quality	Query Fulfillment
Combined	-0.17	+1.14
Stream-based	-0.34	+0.25
Operator-based	-0.61	+0.92
Generalization	-0.11	+1.02

Fig. 4. Experimental results: a) similarity comparison, b) user assessment.

- *Adaptation Quality:* While in our previous research, we developed methods that enable the automatic adaptation of workflows by using adaptation knowledge automatically acquired by machine learning methods from workflow repositories, the quality of the adapted workflows is difficult to control. Therefore, we aim at investigating new methods for assessing the quality of automatically adapted workflows as well as methods to assess the impact of each piece of learned adaptation knowledge on the resulting workflow quality. This allows to better control which adaptation knowledge to retain and which to discard.
- *Interactivity:* The retrieval and adaptation methods developed so far are fully automatic, i.e., they adapted a retrieved workflow according to a specified change request (or goal) without further user interaction. However, specifying a workflow goal or even a change request for an existing workflow in sufficient detail turned out to be quite difficult. Therefore, we aim at developing new methods for conversational POCBR [30] that enable fully interactive problem solving involving retrieval and adaptation of workflows.
- *Transfer Learning:* The adaptation methods investigated so far require existing procedural knowledge of significant volume in order to learn enough adaptation knowledge. This makes it difficult to address small or newly emerging domains in which procedural knowledge is still sparse. Therefore, we aim at investigating whether transfer learning methods can be used to improve learning of adaptation knowledge by transferring knowledge from a different, but related domain with substantial procedural knowledge [11].
- *Exploring New Application Domains:* So far, we demonstrated our methods primarily in the domain of cooking workflows. In the second funding period, we aim at broadening the experimental basis for the whole project by exploring workflow and business process model repositories available in existing repository collections. Furthermore, we will expore the field of scientific text mining workflows in more detail.

Acknowledgements. This work was funded by the German Research Foundation (DFG), project numbers BE 1373/3-1, BE 1373/3-3, MI 1455/1-1, MI 1455/2-3.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: foundational issues, methodological variations, and system approaches. AI Communications 7(1), 39–59 (1994)
2. Bergmann, R., Gil, Y.: Similarity Assessment and Efficient Retrieval of Semantic Workfows. Information Systems 40, 115–127 (2014)
3. Bergmann, R., Minor, M., Schumacher, P., Islam, S.: Cooking CAKE. Presentation and Demonstration at the Computer Cooking Contest at ICCBR 2012 (2012)
4. Bergmann, R., Müller, G., Wittkowsky, D.: Workflow clustering using semantic similarity measures. In: Timm, I. (ed.) KI 2013: Advances in Artificial Intelligence - 36th Annual German Conference on AI. Lecture Notes in Computer Science, vol. 8077, pp. 13–24. Springer (2013)
5. Bergmann, R., Müller, G., Zeyen, C., Manderscheid, J.: Retrieving adaptable cases in process-oriented case-based reasoning. In: Markov, Z., Russell, I. (eds.) Proceedings of the Twenty-Ninth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2016, Key Largo, Florida, May 16-18, 2016. pp. 419–424. AAAI Press (2016)
6. Bergmann, R., Stromer, A.: MAC/FAC retrieval of semantic workflows. In: Boonthum-Denecke, C., Youngblood, G.M. (eds.) Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2013, St. Pete Beach, Florida. May 22-24, 2013. AAAI Press (2013)
7. Dufour-Lussier, V., Ber, F.L., Lieber, J., Nauer, E.: Automatic case acquisition from texts for process-oriented case-based reasoning. Inf. Syst. 40, 153–167 (2014)
8. Kendall-Morwick, J., Leake, D.B.: Facilitating representation and retrieval of structured cases: Principles and toolkit. Inf. Syst. 40, 106–114 (2014)
9. Lopez Mantaras, R., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M.L., Cox, M.T., Forbus, K., Keane, M., Aamodt, A., Watson, I.: Retrieval, reuse, revision and retention in case-based reasoning. The Knowledge Engineering Review 20(03), 215–240 (2005)
10. Minor, M., Bergmann, R., Görg, S.: Case-based adaptation of workflows. Information Systems 40, 142–152 (2014)
11. Minor, M., Bergmann, R., Müller, J.M., Spt, A.: On the Transferability of Process-oriented Cases. In: Goel, A.K., Diaz-Agudo, B., Roth-Berghofer, T. (eds.) Case-Based Reasoning Research and Development - 24nd International Conference, ICCBR 2016, Atlanta, USA. Proceedings. Lecture Notes in Artificial Intelligence, vol. 9969, pp. 281–294. Springer (2016)
12. Minor, M., Görg, S.: Acquiring Adaptation Cases for Scientific Workflows. In: Case-Based Reasoning. Research and Development, 19th International Conference on Case-Based Reasoning, ICCBR 2011. LNCS, vol. 6880, pp. 166–180. Springer (2011)
13. Minor, M., Montani, S., Recio-García, J.A.: Process-oriented case-based reasoning. Inf. Syst. 40, 103–105 (2014)
14. Müller, G., Bergmann, R.: A cluster-based approach to improve similarity-based retrieval for Process-Oriented Case-Based Reasoning. In: 20th European Conference on Artificial Intelligence (ECAI 2014), IOS Press. pp. 639–644 (2014)
15. Müller, G., Bergmann, R.: Compositional Adaptation of Cooking Recipes using Workflow Streams. In: Computer Cooking Contest, Workshop Proceedings ICCBR 2014. Springer (2014)
16. Müller, G., Bergmann, R.: Workflow Streams: A Means for Compositional Adaptation in Process-Oriented Case-Based Reasoning. In: Case-Based Reasoning. Research and Development, 22th International Conference on Case-Based Reasoning, ICCBR 2014. pp. 315–329. LNCS 8766, Springer (2014)

17. Müller, G., Bergmann, R.: CookingCAKE: A Framework for the adaptation of cooking recipes represented as workflows. In: Computer Cooking Contest, Workshop Proceedings ICCBR 2015. Springer (2015)
18. Müller, G., Bergmann, R.: Generalization of Workflows in Process-Oriented Case-Based Reasoning. In: Proceedings of the Twenty-Eighth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2015. pp. 391–396. AAAI Press, Hollywood (Florida), USA (2015)
19. Müller, G., Bergmann, R.: Learning and Applying Adaptation Operators in Process-Oriented Case-Based Reasoning. In: Case-Based Reasoning. Research and Development, 23th International Conference on Case-Based Reasoning, ICCBR 2015. pp. 259–274. LNCS 9343, Springer (2015)
20. 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 Workshops: KDML, FGWM, IR, and FGDB, Trier, Germany, October 7-9, 2015. CEUR Workshop Proceedings, vol. 1458, pp. 247–255. CEUR-WS.org (2015)
21. Müller, G., Bergmann, R.: Case completion of workflows for process-oriented case-based reasoning. In: Goel, A.K., Díaz-Agudo, M.B., Roth-Berghofer, T. (eds.) Case-Based Reasoning Research and Development - 24th International Conference, ICCBR 2016, Atlanta, GA, USA, October 31 - November 2, 2016, Proceedings. Lecture Notes in Computer Science, vol. 9969, pp. 295–310. Springer (2016)
22. Plaza, E.: Semantics and experience in the future web. In: Althoff, K.D., Bergmann, R., Minor, M., Hanft, A. (eds.) Advances in case-based reasoning, 9th European conference, ECCBR 2008. Lecture Notes in Artificial Intelligence, vol. 5239, pp. 44–58. Springer, Berlin (2008)
23. Schumacher, P.: Workflow extraction from textual process descriptions. Ph.D. thesis, Goethe University Frankfurt am Main (2015)
24. Schumacher, P., Minor, M.: Extracting control-flow from text. In: Proc. of the 2014 IEEE 15th International Conference on Information Reuse and Integration. pp. 203 – 210. IEEE, San Francisco, California, USA (2014)
25. Schumacher, P., Minor, M.: Towards a Trace Index Based Workflow Similarity Function. In: Lutz, C., Thielscher, M. (eds.) KI 2014: Advances in Artificial Intelligence - 37th Annual German Conference on AI. LNCS, vol. 8736, pp. 225–230. Springer (2014)
26. Schumacher, P., Minor, M., Schulte-Zurhausen, E.: Extracting and enriching workflows from text. In: Information Reuse and Integration (IRI), 2013 IEEE 14th International Conference on. pp. 285–292. IEEE (2013)
27. Schumacher, P., Minor, M., Walter, K., Bergmann, R.: Extraction of Procedural Knowledge from the Web. In: Proc. of the 21st World Wide Web Conference, WWW 2012, Lyon, France, April 16–20, 2012 (Companion Volume). pp. 739–747. ACM, New York (2012)
28. Spät, A., Keppler, M., Schmidt, M., Kohlhase, M., Lauritzen, N., Schumacher, P.: Goethe-Shaker - Developing a rating score for automatedevaluation of cocktail recipes. In: Proceedings of the Computer Cooking Contest Workshop at the International Conference on Case-Based Reasoning (ICCBR). pp. 101–108 (2014)
29. Walter, K., Minor, M., Bergmann, R.: Workflow Extraction from Cooking Recipes. In: Diaz-Agudo, B., Cordier, A. (eds.) Proceedings of the Computer Cooking Contest Workshop at the International Conference on Case-Based Reasoning (ICCBR 2011). pp. 207–216 (2011)
30. Zeyen, C., Müller, G., Bergmann, R.: Conversational Process-Oriented Case-Based Reasoning. In: Case-Based Reasoning Research and Development - 25th International Conference, ICCBR 2017, Trondheim, Norway. Proceedings. Lecture Notes in Artificial Intelligence, Springer, accepted for publication (2017)