Project Summary: EVER*

Extraction and Processing of Procedural Experience Knowledge in Workflows

http://ever.wi2.uni-trier.de

Ralph Bergmann
bergmann@uni-trier.de
University of Trier
Business Information Systems II
54286 Trier, Germany

Mirjam Minor
minor@cs.uni-frankfurt.de
Goethe University
Department of Informatics
60629 Frankfurt/Main, Germany

Gilbert Müller
University of Trier
Business Information Systems II
54286 Trier, Germany

Pol Schumacher
Goethe University
Department of Informatics
60629 Frankfurt/Main, Germany

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, Q&A websites, etc. However, today there is no real support in finding and 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 and found as any other document. The objective of EVER is the analysis, the development, and the experimental application and evaluation of new knowledge-based methods, particularly from case-based reasoning (CBR), information extraction, and machine learning to extract and process procedural experiences in Internet communities.

Overview

The EVER project (Extraction and Processing of Procedural Experience Knowledge in Workflows) was funded by the German Research Foundation (DFG) from 2011 to 2017 and led by the Universities of Trier and Frankfurt. It focused on the reuse of procedural experiences published by common people in Internet Communities such as cooking web sites. In this regard, it was investigated whether workflow technology and case-based reasoning can help to analyze and reuse procedural experiential knowledge from these Internet communities. Cooking has been chosen as a joint application domain to demonstrate and to empirically evaluate the developed methods.

In order to formalize procedural experience, we employed *workflows* which can be used to represent processes. 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 exchange certain *data items*,

*This work was funded by the German Research Foundation (DFG), project number BE 1373/3-1.

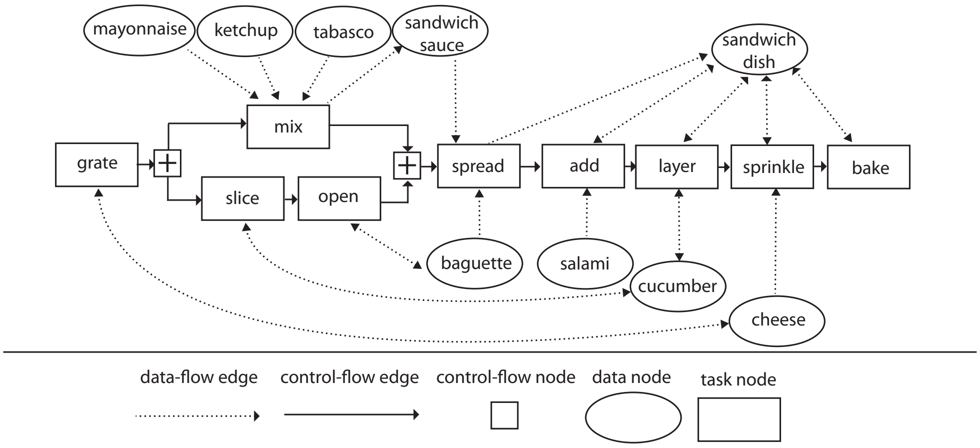

Figure 1: Example Workflow [19]

which can also be of physical matter, depending on the workflow domain (e.g., ingredients in the cooking domain). Tasks, data items, and relationships between the two of them 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 Figure 1.

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, 3]. 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 CBR system CAKE [1], ontologies are represented in an object-oriented fashion while a (partial) transformation into OWL has been developed [20].

The overall process of the EVER project is illustrated in Figure 2. First, procedural experience is gathered from Internet communities and stored in a suitable representation. More precisely, a procedural knowledge base is constructed by extracting workflows from textual sources. Based on the constructed workflow repository procedural experience 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. In overall, EVER provides search methods for processes and further supports the adaptation of processes to particular needs. Consequently, extraction of workflows from textual sources, the similarity-based retrieval for a particular purpose, and the automatic adaptation of retrieved workflows were the core activities that have been investigated in the course of this project. They will be illustrated in more detail below.

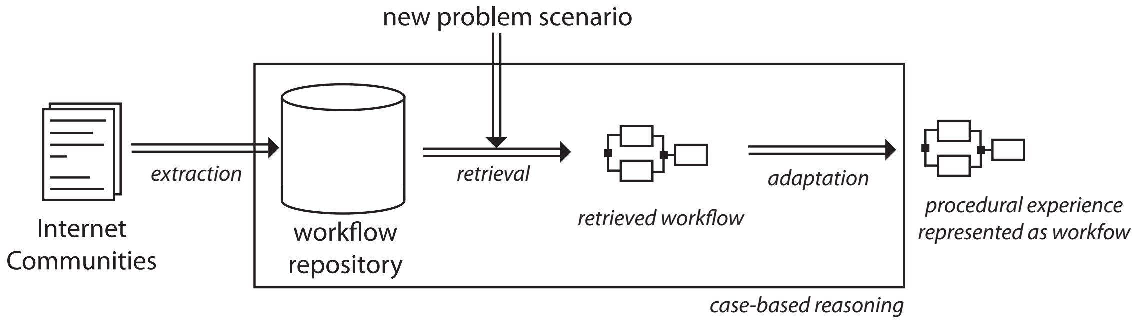

Figure 2: EVER overview

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 [21], which transforms textual descriptions of processes into semantic workflows, as illustrated in Figure 3 by an example in the cooking domain. Here, from the textual description the 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 [25], organizing them in a control flow [22], and enriching the control flow by data flow information [24]. For the latter, we additionally investigated an alternative approach to complete missing data-flow information [19] by learning completion operators from a set of revised workflows within the repository.

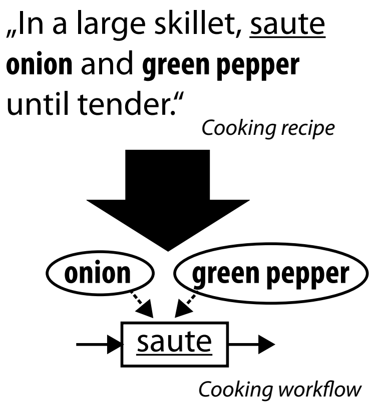

Figure 3: Extraction Example [26]

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].

Similarity-based Workflow Retrieval

For reusing the extracted procedural experiences in other scenarios, 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 [18] was developed. The novel query is able to consider single workflow elements as well as entire workflow fragments (e.g., sub-workflows), which are 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, 3] 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, 3]. We further developed an exchange format for such similarity measures based on OWL [8]. As an alternative approach to the developed semantic similarity measures, we investigated similarity measures based on the trace index of a workflow [23]. 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 [7]. Additionally, we investigated new methods for workflow clustering based on the developed semantic similarity measures [5]. In particular, we developed various algorithms that explore this cluster structure as an index structure for retrieval [13].

Automatic Workflow Adaptation

We further aimed at supporting the user during new problem solving scenarios in which the best matching workflow (identified during similarity-based retrieval) cannot be applied. 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

[1] `http://www.wi.informatik.uni-frankfurt.de/index.php?option=com_content&view=article&id=126&Itemid=533`

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 Figure 4). The developed adaptation methods can mostly be classified into *transformational adaptation*, *compositional adaptation* and *adaptation by generalization*.

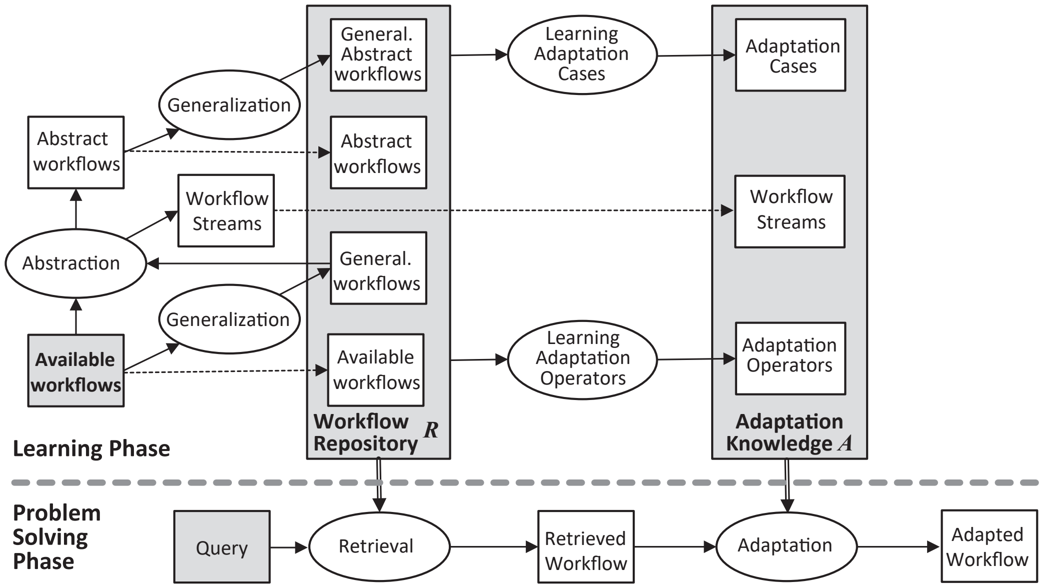

Figure 4: Integration of adaptation approaches

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 $w \overset{\alpha_1}{\rightarrow} w_1 \overset{\alpha_2}{\rightarrow} \ldots \overset{\alpha_n}{\rightarrow} w_n = w'$ 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* [9, 10] the individual transformation steps are represented as so called *adaptation cases* which are learned automatically from the workflow repository [11, 30]. An adaptation case represents a particular previous adaptation scenario by capturing the information about how to transform a particular 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* [17] 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* [15][2]. Such workflow streams can be automatically discovered from the workflow repository. Workflow streams represent valuable adaptation knowledge which are used as "chunks"

[2] The paper [15] won the best PhD-student paper award of the International CBR conference (ICCBR) in 2014 as well as the 2nd prize for the best faculty publication awarded by the postgraduate center of the University of Trier.

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 less 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 [27]. 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 [16]. 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 [12] as shown in Figure 4. 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 [6], 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.

CookingCAKE

The approaches developed throughout the whole project have been continuously integrated in a prototype system called CookingCAKE for participation in the Computer Cooking Contest in 2011 [29], 2012 [4], 2014 [14, 28] and 2015 [12]. An online demonstration of CookingCAKE for workflow retrieval and adaptation is available under `http://cookingcake.wi2.uni-trier.de`. Using the previously sketched retrieval and adaptation methods, CookingCAKE provides a platform for generating sandwich recipes considering ingredients and preparation steps that are desired or undesired.

References

[1] Ralph Bergmann, Sarah Gessinger, Sebastian Görg, and Gilbert Müller. The Collaborative Agile Knowledge Engine CAKE. In *Proceedings of the 18th International Conference on Supporting Group Work*, GROUP '14, pages 281–284, New York, NY, USA, 2014. ACM.

[2] Ralph Bergmann and Yolanda Gil. Retrieval of Semantic Workfows with Knowledge Intensive Similarity Measures. In *Case-Based Reasoning. Research and Development, 19th International Conference on Case-Based Reasoning, ICCBR 2011*, volume 6880 of *Lecture Notes in Computer Science*, pages 17–31. Springer, 2011.

[3] Ralph Bergmann and Yolanda Gil. Similarity Assessment and Efficient Retrieval of Semantic Workfows. *Information Systems*, 40:115–127, 2014.

[4] Ralph Bergmann, Mirjam Minor, Pol Schumacher, and Siblee Islam. Cooking CAKE. Presentation and Demonstration at the Computer Cooking Contest at ICCBR 2012, 2012.

[5] Ralph Bergmann, Gilbert Müller, and Daniel Wittkowsky. Workflow clustering using semantic similarity measures. In Ingo Timm, editor, *KI 2013: Advances in Artificial Intelligence - 36th Annual German Conference on AI*, volume 8077 of *Lecture Notes in Computer Science*, pages 13–24. Springer, 2013.

[6] Ralph Bergmann, Gilbert Müller, Christian Zeyen, and Jens Manderscheid. Retrieving adaptable cases in process-oriented case-based reasoning. In Zdravko Markov and Ingrid Russell, editors, *Proceedings of the Twenty-Ninth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2016, Key Largo, Florida, May 16-18, 2016.*, pages 419–424. AAAI Press, 2016.

[7] Ralph Bergmann and Alexander Stromer. MAC/FAC retrieval of semantic workflows. In Chutima Boonthum-Denecke and G. Michael Youngblood, editors, *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.

[8] Kerstin Frei and Benjamin Porz. Modellierung von Ähnlichkeitsmaßen in OWL, 2014. Technical Report - Forschungspraktikum Universität Trier.

[9] Mirjam Minor, Ralph Bergmann, and Sebastian Görg. Case-based adaptation of workflows. *Information Systems*, 40:142–152, 2014.

[10] Mirjam Minor, Ralph Bergmann, Sebastian Görg, and Kirstin Walter. Reasoning on Business Processes to Support Change Reuse. In *13th IEEE Conference on Commerce and Enterprise Computing, CEC 2011, Luxembourg-Kirchberg, Luxembourg, September 5-7, 2011*, pages 18–25. IEEE Computer Society, 2011.

[11] Mirjam Minor and Sebastian Görg. Acquiring Adaptation Cases for Scientific Workflows. In *Case-Based Reasoning. Research and Development, 19th International Conference on Case-Based Reasoning, ICCBR 2011*, volume 6880 of *LNCS*, pages 166–180. Springer, 2011.

[12] Gilbert Müller and Ralph Bergmann. CookingCAKE: A framework for the adaptation of cooking recipes represented as workflows. In *submitted to Computer Cooking Contest 2015*, Springer.

[13] Gilbert Müller and Ralph Bergmann. 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*, pages 639–644, 2014.

[14] Gilbert Müller and Ralph Bergmann. Compositional Adaptation of Cooking Recipes using Workflow Streams. In *Computer Cooking Contest, Workshop Proceedings ICCBR 2014*, Springer, 2014.

[15] Gilbert Müller and Ralph Bergmann. 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*, LNCS 8766, pages 315–329. Springer, 2014.

[16] Gilbert Müller and Ralph Bergmann. Generalization of Workflows in Process-Oriented Case-Based Reasoning. In *Proceedings of the Twenty-Eighth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2015*, pages 391–396, Hollywood (Florida), USA, 2015. AAAI Press.

[17] Gilbert Müller and Ralph Bergmann. 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*, LNCS 9343, pages 259–274. Springer, 2015.

[18] Gilbert Müller and Ralph Bergmann. POQL: A new query language for process-oriented case-based reasoning. 2015. Submitted.

[19] Gilbert Müller and Ralph Bergmann. Case completion of workflows for process-oriented case-based reasoning. In Ashok K. Goel, M. Belén Díaz-Agudo, and Thomas Roth-Berghofer, editors, *Case-Based Reasoning Research and Development - 24th International Conference, ICCBR 2016, Atlanta, GA, USA, October 31 - November 2, 2016, Proceedings*, volume 9969 of *Lecture Notes in Computer Science*, pages 295–310. Springer, 2016.

[20] Peter Pazen. Design und Implementierung eines Konverters zur Transformation von OWL-Ontologien in die objektorientierte Fallrepräsentation von CAKE, 2013. Masterthesis, University of Trier.

[21] Pol Schumacher. *Workflow extraction from textual process descriptions*. PhD thesis, Goethe University Frankfurt am Main, 2015.

[22] Pol Schumacher and Mirjam Minor. Extracting control-flow from text. In *Proc. of the 2014 IEEE 15th International Conference on Information Reuse and Integration*, pages 203 – 210, San Francisco, California, USA, 2014. IEEE.

[23] Pol Schumacher and Mirjam Minor. Towards a Trace Index Based Workflow Similarity Function. In Carsten Lutz and Michael Thielscher, editors, *KI 2014: Advances in Artificial Intelligence - 37th Annual German Conference on AI*, volume 8736 of *LNCS*, pages 225–230. Springer, 2014.

[24] Pol Schumacher, Mirjam Minor, and Eric Schulte-Zurhausen. Extracting and enriching workflows from text. In *Information Reuse and Integration (IRI), 2013 IEEE 14th International Conference on*, pages 285–292. IEEE, 2013.

[25] Pol Schumacher, Mirjam Minor, Kirstin Walter, and Ralph Bergmann. 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)*, pages 739–747, New York, 2012. ACM.

[26] Pol Schumacher, Mirjam Minor, Kirstin Walter, and Ralph Bergmann. Extraction of procedural knowledge from the web: a comparison of two workflow extraction approaches. In Alain Mille, Fabien L. Gandon, Jacques Misselis, Michael Rabinovich, and Steffen Staab, editors, *Proceedings of the 21st World Wide Web Conference, WWW 2012, Lyon, France, April 16-20, 2012 (Companion Volume)*, pages 739–747. ACM, 2012.

[27] Alexander Spät. Adaption von Datenanalyseprozessen mit kontextfreien Grammatiken, 2015. Masterthesis, Goethe University Frankfurt.

[28] Alexander Spät, Marius Keppler, Matthias Schmidt, Moritz Kohlhase, Niels Lauritzen, and Pol Schumacher. GoetheShaker - 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)*, pages 101–108, 2014.

[29] Kirstin Walter, Mirjam Minor, and Ralph Bergmann. Workflow Extraction from Cooking Recipes. In Belen Diaz-Agudo and Amelie Cordier, editors, *Proceedings of the Computer Cooking Contest Workshop at the International Conference on Case-Based Reasoning (ICCBR 2011)*, pages 207–216, 2011.

[30] Stephan Zonker. Automatische Generierung von Adaptionswissen zur fallbasieren Adaption von Workflows, 2013. Masterthesis, Universität Trier.