Cooking made easy: On a novel approach to complexity-aware recipe generation

Gilbert Müller and Ralph Bergmann

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

Abstract. This paper presents an approach to generate easy-to-prepare cooking recipes represented as workflows. A novel complexity-aware generation approach is described that considers various aspects such as preparation time, number of ingredients, and difficulty of preparation to optimize the complexity of the recipe. Based on a user query specifying the desired and undesired ingredients or preparation steps, easy-to-prepare dishes are generated automatically.

Keywords: workflow complexity, workflow adaptation, cooking, process-oriented case based reasoning

1 Introduction

Nowadays, an increasing amount of amateur chefs become fascinated by the world of cooking. Traditional cooking websites support these chefs in finding suitable cooking recipes. However, the recipes need to match several criteria, which sometimes require recipes to be adapted to the individual demands of the user. These demands include contained ingredients, required preparation tools, or dietary restrictions. Thus, several novel approaches have been presented aiming at supporting the user beyond traditional recipe search (e.g., [5,7,3,6]). In certain situations, amateur chefs may prefer easy-to-prepare cooking recipes with a short preparation time, low required cooking skills, or a small amount of ingredients for a variety of reasons.

In this paper we will describe a novel approach that automatically constructs individual and easy-to-prepare cooking recipes based on ingredients and preparation steps specified as desired or undesired. The approach is based on our CookingCAKE framework [10], which will be extended by a new complexity-aware recipe generation. The remainder of this paper is organized as follows: The next section presents the foundations of the CookingCAKE framework. Then, we introduce a complexity assessment for cooking recipes represented as workflows, which will be applied during CookingCAKE's recipe generation. Finally, we present our prototypical implementation for competing in the *Easy Steps Challenge* of the *Computer Cooking Contest*.

2 CookingCAKE

CookingCAKE constructs individual cooking recipes represented as workflows by means of Process-oriented Case-based Reasoning [8]. In a nutshell, CookingCAKE selects the best matching cooking workflow from the workflow repository (case base) and subsequently adapts it according to a query specified by the user.

2.1 Cooking Workflows

In our approach a cooking recipe is represented as a workflow describing the process to prepare a particular dish [13] (see Fig. 1). A cooking workflow $W = (N, E)$ consists of nodes $N = N^T \cup N^D$ and edges $E = E^C \cup E^D$. Nodes of the workflow represent *preparation steps* N^T (also called *tasks*) or *ingredients* N^D (also called *data nodes*). The execution order of preparation steps is defined by *control-flow edges* $E_C \subseteq N^T \times N^T$ and the consumption or production of an ingredient is specified by *data-flow edges* $E_D \subseteq (N^T \times N^D) \cup (N^D \times N^T)$. Furthermore, we enforce that the workflow is executable, which means here that it consists of a single sequence of tasks such that each task $t \in N^T$ consumes ($\exists d \in N^D : (d,t) \in E^D$) and produces (i.e., $\exists d \in N^D : (t,d) \in E^D$) at least one ingredient, respectively. An example cooking workflow for a sandwich recipe is illustrated in Fig. 1.

2.2 Ingredient and Preparation Step Similarity

To support retrieval and adaptation of workflows, the individual workflow elements are annotated with ontological information resulting in a *semantic workflow* [2]. CookingCAKE uses a taxonomy of ingredients to define the semantics of data items and a taxonomy of preparation steps to define the semantics of tasks. These taxonomies are employed for the similarity assessment between tasks and data items. An example ingredient taxonomy is given in Figure 2. A taxonomy is ordered by terms that are either a generalization or a specialization of a specific other term within the taxonomy, i.e., an inner node represents a generalized term that stands for the set of most specific terms below it. For example, the generalized term $vegetarian$ in the illustrated taxonomy

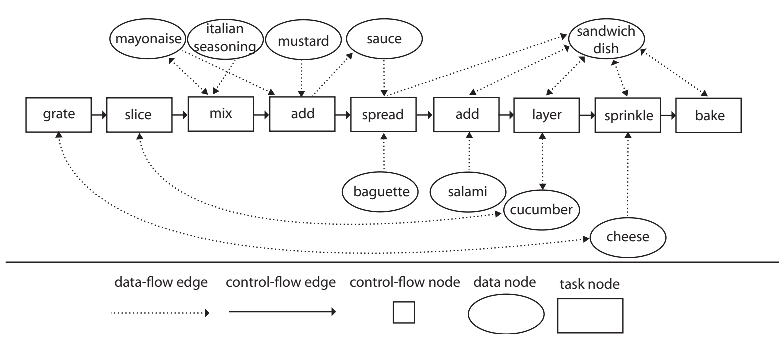

Fig. 1. Example of a block-oriented cooking workflow

stands for the set $\{potatoes, rice, noodles, \ldots\}$. Inner nodes in generalized workflows represent that an arbitrary ingredient from the set of its specializations can be chosen.

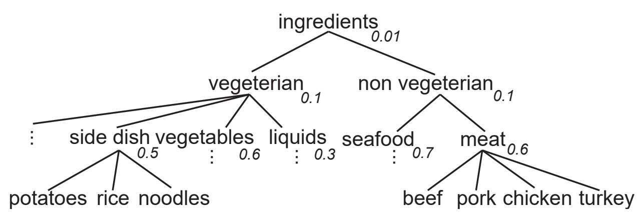

Fig. 2. Example of an ingredient taxonomy

In our previous work, we developed a semantic similarity measure for workflows that enables the similarity assessment of a case workflow W_c w.r.t a query workflow W_q [2], i.e. $sim(W_c, W_q)$. Each query workflow element $x_q \in W_q$ is mapped by the function $m : W_q \rightarrow W_c$ to an element of the case workflow $x_c \in W_c$, i.e., $x_c = m(x_q)$. The mapping is used to estimate the similarity between the two workflow elements utilizing the taxonomy, i.e., $sim(x_q, x_c)$. The similarity of preparation steps or ingredients reflects the closeness in the taxonomy and further regards the level of the taxonomic elements. In general, the similarity is defined by the attached similarity value of the least common ancestor, e.g., $sim(beef, pork) = 0.6$. If a more general query element such as *meat* is compared with a specific element below it, such as *pork*, the similarity value is 1. This ensures that if the query asks for a recipe containing meat, any recipe workflow containing any kind of meat is considered highly similar. All the similarity values of the mappings are then aggregated to estimate an overall workflow similarity.

2.3 Workflow Query Language

CookingCAKE uses the query language POQL [12] to capture desired and undesired ingredients or preparation steps of a cooking workflow as query q. The ability to specify preparation steps is useful as certain tools might not be available or their usage is desired (e.g., oven). Let $q_d = \{x_1, \ldots, x_n\}$ be a set of desired ingredients or preparation steps and $q_u = \{y_1, \ldots, y_n\}$ be a set of undesired ingredients or preparation steps, respectively. A query q is then defined as $(x_1 \wedge \ldots \wedge x_2) \wedge \neg y_1 \wedge \ldots \wedge \neg y_n$. POQL further enables the specification of generalized terms, i.e., if a vegetarian dish is desired, this can be defined by $\neg meat$. The query q is used to guide retrieval, i.e., to search for a workflow which at best contains all desired elements but no undesired element. Based on the query q the not matching elements can be identified, enabling to determine the elements to be deleted or added to the retrieved workflow during the subsequent adaptation stage. The query fulfillment of a workflow W for a query q is defined as the similarity between the desired ingredients/preparation steps as well as the workflow W and the number of undesired ingredients/preparation steps not contained in W according to the workflow similarity (see Sec. 2.2) in relation to the size of the query (see Formula 1).

$$QF(q,W) = \frac{\sum_{x \in q_d} sim(x, m(x)) + |\{y \in q_u | sim(y, m(y)) \neq 1\}|}{|q_d| + |q_u|} \quad (1)$$

Consequently, similar desired ingredients or preparation steps increase the query fulfillment, while matching undesired ingredients or preparation steps reduce the query fulfillment between the POQL query and the workflow.

2.4 Recipe Construction

Based on the defined POQL query, CookingCAKE constructs a workflow automatically by retrieving the best matching workflow from the repository (*case base*) and adapting it according to the query fulfilment. Consequently, the adaptation process of CookingCAKE aims at adding missing desired ingredients/preparation steps and at removing undesired contained ingredients/preparation steps. In a nutshell, the adaptation process uses three different adaptation methods that are subsequently executed. First, entire components of the cooking dish such as the *sandwich sauce* or *sandwich topping* are replaced by matching components from other recipes [9]. Next, adaptation is performed by use of operators that define possible and valid modifications on the cooking workflows. Finally, the cooking recipes are adapted by replacing single ingredients and preparation steps by means of the specified taxonomy, assuming that similar terms can most likely be replaced with each other [11]. In all approaches, adaptation of a workflow is performed by chaining several adaptation steps $W \stackrel{\alpha_1}{\rightarrow} W_1 \stackrel{\alpha_2}{\rightarrow} \ldots \stackrel{\alpha_n}{\rightarrow} W_n = W'$, which iteratively transforms the retrieved workflow W towards an adapted workflow W'. This process solves an optimization problem aiming at maximizing the specified criterion, which is so far implemented by the query fulfillment. Thus, the recipe construction is a search process with the goal to achieve an adapted workflow with the highest query fulfillment possible. The overall recipe construction process ensures the syntactical correctness of the workflows, i.e., that the workflows are executable. More detailed information on the construction process of CookingCAKE can be found in the corresponding publication [10].

In the next section, we introduce a new criterion for the retrieval and adaptation process that considers the complexity of workflows. Thus, retrieval as well as the adaptation become complexity-aware and aim at optimizing the constructed workflow with regard to the new defined criterion during recipe construction.

3 Complexity Assessment

In the literature various approaches to asses the complexity of workflows exist (see [4]). In this approach, we rather focus on a domain-specific complexity measure for cooking workflows. During recipe construction, this complexity criterion is considered to generate easy-to-prepare recipes automatically. We assume that the complexity of a recipe is less focused on one single feature, but is composed by several criteria. Thus, we deploy a complexity measure that covers five different indicators for determining the complexity of the recipe (see Table 1).

Table 1. complexity criteria

critera description	criteria measure
number of ingredients	$\frac{\|N^D\|}{max\{\|N_1^D\|,\ldots,\|N_n^D\|\}}$
number of preparation steps	$\frac{\|N^T\|}{max\{\|N_1^T\|,\ldots,\|N_n^T\|\}}$
complexity of ingredient processing	$1 - \frac{2\cdot\|N^T\|}{\|E^D\|}$
complexity of preparation steps	$\frac{\sum_{t\in N^T} taskComplexity(t)}{\|N^T\|}$
duration of preparation	$\frac{preparationTime(W)}{max\{preparationTime(W_1),\ldots,preparationTime(W_n)\}}$

The first two criteria measure basic complexity properties, i.e., the *number of preparation steps* as well as the *number of ingredients* in the particular cooking workflow $W = (N, E)$. Both measures are normalized by the highest amount of ingredients or preparation steps contained in the workflows from the workflow repository. Consequently, cooking workflows with more ingredients or more preparation steps are assumed to be more complex. Furthermore, the *complexity of preparation steps* as well as the *complexity of ingredient processing* represent two additional complexity criteria. The complexity measure for *ingredient processing* considers the average amount of ingredients consumed and produced by the preparation steps, which assigns a high complexity value to those workflows in which the preparation steps N^T consume and produce a large amount of ingredients[1] E^D. In contrast, for computing the *complexity of preparation steps* each task t in the taxonomy (see Sec. 2.2) is annotated by an estimated task complexity value $taskComplexity(t) \in [0, 1]$. As an example, the preparation step *blanche* is considered to be more complex than the preparation step *mix*. The criterion is then defined as the average complexity of the preparation steps in the workflow W. Finally, the duration for preparing a particular dish is also a factor affecting the complexity. Therefor, also approximated execution times $taskPreparationTime(t) \in \mathbb{N}$ are annotated to each task t in the taxonomy. Here, for example, *baking* is annotated by a long execution time, while *season* is considered as a rather short preparation step. The *duration of preparation* for a workflow W is then heuristically measured by aggregating the execution times of the preparation steps, i.e., $preparationTime(W) = \sum_{t\in N^T} taskPreparationTime(t)$. To assess the corresponding complexity, this value is normalized in relation to the workflows from the repository as defined in Table 1.

Each of these five complexity measures determines a complexity value within the interval $[0, 1]$. Based on these measures, we constructed an overall complexity measure $complexity(W) \rightarrow [0, 5]$ which adds up all complexity criteria to a single value. The

[1] Please note that each task in a workflow consumes and produces at least one ingredient, respectively (see Sec. 2.1)

overall complexity measure specifies the corresponding difficulty level of the recipe preparation and distinguishes between *very easy* ($[0,1[$), *easy* ($[1,2[$), *medium* ($[2,3[$), *difficult* ($[3,4[$) and *very difficult* ($[4,5]$).

$$QF_{\text{complexity}}(q,W) = \alpha \cdot QF(q,W) + (1-\alpha) \cdot (1 - complexity(W)/5) \quad (2)$$

Based on this overall complexity measure, we defined a new complexity-aware query fulfilment measure $QF_{\text{complexity}}(q,W) \rightarrow [0,1]$ (see Eq. 2) for the retrieval and adaptation process. It replaces the query fulfillment measure specified in formula 1, thus considering complexity as well. Both criteria may be weighted by a parameter $\alpha \in [0,1]$. The workflow construction process of CookingCAKE as described in Section 2.4 then aims at optimizing the constructed workflow with regard to this new criterion. Please note that this is a multi-objective optimization problem and thus the adaptation may not be able to maximize the query fulfillment and to reduce the complexity of the workflow at the same time.

4 Computer Cooking Contest: Easy steps challenge

We created a new user interface for the CookingCAKE system in order to address the *Easy Steps Challenge* of the *Computer Cooking Contest*, which applies the previously described complexity assessment. A running prototype of the implementation is available under (`http://cookingCAKE.wi2.uni-trier.de/complexity`), which uses a workflow repository of 61 sandwich recipes manually modelled from various Internet sources (e.g., sandwich recipes on WikiTaaable [1][2]). The employed taxonomies of preparation steps and ingredients (see Sec. 2.2) are based on the WikiTaaable ontology and were manually annotated with similarity, preparation time, and task complexity values.

The query of CookingCAKE involves desired and undesired ingredients as well as desired and undesired preparation steps. An example query (`http://cookingCAKE.wi2.uni-trier.de/complexity?d=cherry%20tomato|salmon&u=cheese`), generates a *salmon* and *cherry tomato* recipe without using any kind of *cheese*. CookingCAKE then selects the best matching workflow from the repository and subsequently adapts it according to the novel criterion $QF_{complexity}(q,W)$. Thus, the system tries to maximize the query fulfilment on the one hand and on the other hand aims at reducing the complexity of the workflow to generate an appropriate easy-to-prepare recipe for an amateur chef. The result page of the novel CookingCAKE interface also displays the estimated difficulty of preparation, the computed duration time as well as the single complexity values (see Sec. 3) for the constructed recipe.

To evaluate our new complexity-aware approach for recipe construction, we generated 61 queries automatically. More precisely, for each workflow W, a corresponding query was constructed by selecting the most similar workflow W' from the repository and by determining the difference between the two workflows. The constructed query considers workflow elements as desired that are only contained in the workflow W while the elements only contained in workflow W' are considered as undesired. At

[2] http://wikitaaable.loria.fr

most 4 randomly selected ingredients and 2 preparation steps are determined as desired or undesired respectively. For each of the queries we performed a leave-one-out test, i.e., the corresponding workflow was removed from the repository. Then, we executed the recipe generation process with the standard approach as well as the complexity-aware approach. For the complexity-aware recipe construction we chose the parameter $\alpha = 0.5$ to consider the query fulfillment and the complexity in equal shares. For both approaches, we measured the query fulfillment, the complexity, and the combined complexity-aware criterion of the retrieved as well as of the adapted workflow.

Table 2. evaluation results

	query fulfillment	complexity	combined	computation time
standard retrieval	0.83	0.43	0.70	1.15 s
standard adaption	0.92	0.48	0.72	18.73 s
complexity-aware retrieval	0.75	0.28	0.74	1.42 s
complexity-aware adaption	0.87	0.29	0.79	9.49 s

The evaluation results illustrated in Table 2 clearly show that already during complexity-aware retrieval, a less complex workflow is selected. Furthermore, the computation time of the subsequent adaptation stage is significantly decreased[3]. The most important observation, however, is that with the new complexity-aware approach, the final complexity is significantly reduced (-40%), while the query fulfillment is only slightly decreased (-5%). Altogether it can be concluded that the complexity-aware approach presented in this paper enables the individual construction of easy-to-prepare cooking recipes with a low preparation complexity.

5 Conclusions and Future Work

This paper presents a new approach to generate easy-to-prepare cooking recipes based on cooking workflows. The new approach considers a query specified by the user to automatically generate a cooking workflow matching the users demands and further considers the complexity of the cooking workflow as an additional criterion. The complexity measure is composed of several criteria including the number of ingredients, the preparation time and the complexity of preparation steps.

In future work we aim at providing an interface for choosing the desired recipe complexity. Furthermore, the complexity assessment will be improved and evaluated by comparing various complexity measures. Finally, we will investigate several other factors that could be considered during the construction of recipes such as nutritions and dietary restrictions.

[3] The adaptation time depends on the size of the workflow, which is usually smaller, if the workflow is less complex.

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

References

1. Badra, F., Bendaoud, R., Bentebibel, R., Champin, P., Cojan, J., Cordier, A., Desprès, S., Jean-Daubias, S., Lieber, J., Meilender, T., Mille, A., Nauer, E., Napoli, A., Toussaint, Y.: TAAABLE: text mining, ontology engineering, and hierarchical classification for textual case-based cooking. In: Schaaf, M. (ed.) ECCBR 2008, Trier, Germany, Workshop Proceedings. pp. 219–228 (2008)
2. Bergmann, R., Gil, Y.: Similarity assessment and efficient retrieval of semantic workflows. Inf. Syst. 40, 115–127 (Mar 2014)
3. Bridge, D., Larkin, H.: Creating new sandwiches from old. In: Nauer, E., Minor, M., Cordier, A. (eds.) Procs. of the Seventh Computer Cooking Contest (Workshop Programme of the Twenty-second International Conference on Case-Based Reasoning). pp. 117–124 (2014)
4. Cardoso, J.S., Mendling, J., Neumann, G., Reijers, H.A.: A discourse on complexity of process models. In: Eder, J., Dustdar, S. (eds.) Business Process Management Workshops, BPM 2006 International Workshops, BPD, BPI, ENEI, GPWW, DPM, semantics4ws, Vienna, Austria, September 4-7, 2006, Proceedings. Lecture Notes in Computer Science, vol. 4103, pp. 117–128. Springer (2006), http://dx.doi.org/10.1007/11837862_13
5. Gaillard, E., Lieber, J., Nauer, E.: Improving ingredient substitution using formal concept analysis and adaptation of ingredient quantities with mixed linear optimization. In: Kendall-Morwick, J. (ed.) Workshop Proceedings from (ICCBR 2015), Frankfurt, Germany, 2015. CEUR Workshop Proceedings, vol. 1520, pp. 209–220. CEUR-WS.org (2015)
6. Grace, K., Maher, M.L., Wilson, D.C., Najjar, N.A.: Combining CBR and deep learning to generate surprising recipe designs. 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. 154–169. Springer (2016), http://dx.doi.org/10.1007/978-3-319-47096-2_11
7. Keppler, M., Kohlhase, M., Lauritzen, N., Schmidt, M., Schumacher, P., Spät, A.: Goethe-Shaker - Developing a rating score for automated evaluation of cocktail recipes. In: ICCBR Workshop Proceedings. pp. 101–108 (2014)
8. Minor, M., Montani, S., Recio-Garca, J.A.: Process-oriented case-based reasoning. Information Systems 40(0), 103 – 105 (2014)
9. Müller, G., Bergmann, R.: Workflow Streams: A Means for Compositional Adaptation in Process-Oriented Case-Based Reasoning. In: Proceedings of ICCBR 2014. Cork, Ireland (2014)
10. 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), Frankfurt, Germany, 2015. CEUR Workshop Proceedings, vol. 1520, pp. 221–232. CEUR-WS.org (2015)
11. Müller, G., Bergmann, R.: Generalization of Workflows in Process-Oriented Case-Based Reasoning. In: 28th International FLAIRS Conference. AAAI, Hollywood (Florida), USA (2015)
12. Müller, G., Bergmann, R.: POQL: A New Query Language for Process-Oriented Case-Based Reasoning (2015), to be submitted to LWA 2015 Trier, Germany
13. Schumacher, P., Minor, M., Walter, K., Bergmann, R.: Extraction of procedural knowledge from the web. In: Workshop Proceedings: WWW'12. Lyon, France (2012)