# Workflow Clustering Using Semantic Similarity Measures

Ralph Bergmann, Gilbert Müller, and Daniel Wittkowsky

University of Trier
Business Information Systems II
D-54286 Trier, Germany
Email: bergmann@uni-trier.de, Web: www.wi2.uni-trier.de

**Abstract.** The problem of clustering workflows is a relatively new research area of increasing importance as the number and size of workflow repositories is getting larger. It can be useful as a method to analyze the workflow assets accumulated in a repository in order to get an overview of its content and to ease navigation. In this paper, we investigate workflow clustering by adapting two traditional clustering algorithms ($k$-medoid and AGNES) for workflow clustering. Clustering is guided by a semantic similarity measure for workflows, originally developed in the context of case-based reasoning. Further, a case study is presented that evaluates the two algorithms on a repository containing cooking workflows automatically extracted from an Internet source.

## 1 Introduction

Cluster analysis is an established method that allows to discover the structure in collections of data by exploring similarities between data points. The goal of cluster analysis is to group data objects in such a way that data objects within a cluster are similar while data objects of different clusters are dissimilar to one another [6]. Cluster analysis has already been applied to different types of data, such as relational data, textual data, and even multi-media data.

In this paper, we address the problem of clustering workflows, which is a relatively new area of increasing importance [7,8,13,5]. Traditionally, workflows are "the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules" [16]. Recently, more and more repositories are constructed by companies and organizations to capture their procedural knowledge as a starting point for reuse and optimization. For example, the myExperiment[1] virtual research environment enables the publication, search, and reuse of scientific workflows providing a repository of more than 2000 workflows. Recent efforts on workflow sharing supported by new standards for workflow representation will likely lead to repositories of larger scale. Further, research

[1] www.myexperiment.org

on methods for automatic workflow extraction from text [12] enables obtaining workflow repositories from how-to descriptions on the Internet. Also process mining [1,10,14] can infer process models (which are similar to workflows) by analyzing event logs, thus producing a large number of processes.

Clustering of workflows will likely become relevant as the size of workflow repositories increases. It can be useful to analyze the workflow assets accumulated in a repository in order to get an overview of its content and to ease navigation [7,8,13]. Identifying clusters of similar workflows could highlight the opportunity to unify similar workflows [5], thus reducing the number of workflows that must be supported in a company. Further, clustering might be used as an index structure for a workflow repository that can help to speed-up workflow retrieval [4]. Please note that workflow clustering [7,8,13,5] is significantly different from process mining [14,1,10] as process mining analyzes execution log data while workflow clustering analyzes the workflows themselves.

In this paper, we investigate workflow clustering by applying selected traditional clustering algorithms (in particular k-medoid and AGNES) to the clustering of workflows. The core of the application to workflows is the availability of an appropriate similarity measure for workflows, which replaces the traditional distance measure for $n$-dimensional data points. We propose to use a semantic similarity measure for workflows which we have developed and evaluated in our previous research as part of a similarity-based retrieval method [4]. This similarity measure can be configured according to a particular domain, based on an ontology of tasks and data items.

The next section presents our previous work on workflow representation and semantic workflow similarity. Then, section 3 describes our approach to workflow clustering before section 4 presents a case study investigating the proposed cluster algorithms to analyze a repository of 1729 cooking workflows. This paper ends with a conclusion and a description of potential future work.

## 2 Workflow Representation and Semantic Similarity

We now briefly describe our previous work on semantic workflow similarity [4], which is a cornerstone of the proposed clustering algorithms. We illustrate our approach by an example form the domain of cooking recipes. In this domain a cooking recipe is represented as a workflow describing the instructions for cooking a particular dish [12].

### 2.1 Representation of Semantic Workflows

Broadly speaking, workflows consist of a set of *activities* (also called *tasks*) combined with *control-flow structures* like sequences, parallel (AND split/join) or alternative (XOR split/join) branches, and loops. Tasks and control-flow structures form the *control-flow*. In addition, tasks exchange certain *products*, which can be of physical matter (such as ingredients for cooking tasks) or information. Tasks, products, and relationships between the two of them form the *data flow*.

Today, graph representations for workflows are widely used. In this paper we build upon the workflow representation using semantically labeled graphs [4], which is now briefly summarized. We represent a workflow as a directed graph $W = (N, E, S, T)$ where $N$ is a set of nodes and $E \subseteq N \times N$ is a set of edges. Nodes and edges are annotated by a type from a set $\Omega$ and a semantic description from a set $\Sigma$. Type and semantic description are computed by the two mapping functions $T\colon N \cup E \to \Omega$ and $S\colon N \cup E \to \Sigma$, respectively. The set $\Omega$ consists of the types: *workflow node, data node, task node, control-flow node, control-flow edge, part-of edge* and *data-flow edge*. Each workflow $W$ has exactly one workflow node. The task nodes and data nodes represent tasks and data items, respectively. The control-flow nodes stand for control-flow elements. The data-flow edge is used to describe the linking of the data items consumed and produced by the tasks. The control-flow edge is used to represent the control flow of the workflow, i.e., it links tasks with successor tasks or control-flow elements. The part-of edge represents a relation between the workflow node and all other nodes. $\Sigma$ is a semantic meta data language that is used for the semantic annotation of nodes and edges. In our work we treat the semantic descriptions in an object-oriented fashion to allow the application of well-established similarity measures from case-based reasoning [2,3]. Figure 1 shows a simple fragment of a workflow graph from the cooking domain with the different kinds of nodes and edges. For some nodes semantic descriptions are sketched, specifying ingredients used (data nodes) and tasks performed (cooking steps). The semantic descriptions are based on a domain specific ontology of data items and tasks.

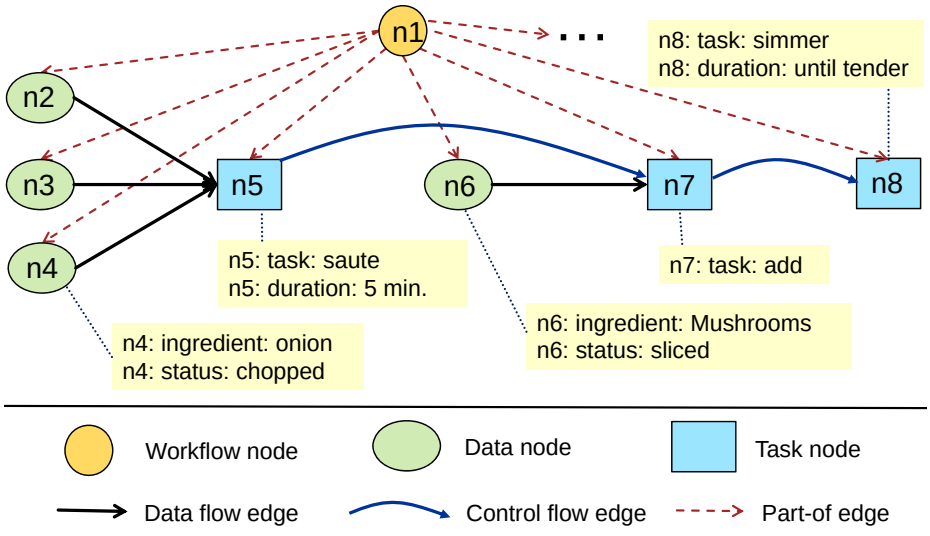


**Fig. 1.** A sample workflow graph

### 2.2 Semantic Similarity

The described graph representation of workflows enables modeling related semantic similarity measures which are well inline with experts assessment of workflow similarity [4]. Our framework for modeling workflow similarity is based on a local similarity measure for semantic descriptions $sim_\Sigma : \Sigma^2 \to [0,1]$ based on which node and edge similarity measures $sim_N : N^2 \to [0,1]$ and $sim_E : E^2 \to [0,1]$ can be easily defined. For example, the node similarity is defined as follows:

$$sim_N(q,c) = \begin{cases} sim_\Sigma(S_q(q), S_c(c)) & \text{if} \quad T_q(q) = T_c(c) \\ 0 & \text{otherwise} \end{cases}$$

Nodes with different types (e.g. a task node compared with a data node) are considered dissimilar; there similarity is always zero. The similarity of nodes of equal type is defined by the similarity of the semantic descriptions. In particular the taxonomical structure of the data and task ontology (ingredients and cooking steps ontology in the cooking recipe domain) is employed to derive a similarity value that reflects the closeness in the ontology as well as additional parameters such as the quantity of an ingredient used in a recipe. Due to the space limitations of this paper, we refer to [4] for more details and examples of how such local similarity measures look like.

The similarity measure for workflows allows to compare two complete workflows. Motivated by its use for similarity-based retrieval, one workflow is considered a query workflow $QW$ and the second workflow is considered a case workflow $CW$ from a repository. The similarity measure assess how well the query workflow is covered by the case workflow. In particular, the similarity should be 1 if the query workflow is exactly included in the case workflow as a subgraph. Consequently, the proposed similarity measure is not symmetrical.

The similarity $sim(QW, CW)$ is computed by means of a legal mapping $m : N_q \cup E_q \to N_c \cup E_c$, which is a type-preserving, partial, injective mapping function of the nodes and edges of the query workflow to those of the case workflow. For each query node or edge $x$ mapped by $m$, the similarity to the respective case node or edge $m(x)$ is computed through $sim_N(x, m(x))$ and $sim_E(x, m(x))$, respectively. The overall workflow similarity with respect to the mapping $m$, named $sim_m(QW, CW)$ is computed by an aggregation function (e.g. a weighted average) combining the previously computed similarity values.

Finally, the overall workflow similarity $sim(QW, CW)$ is determined by the best possible mapping of that kind, i.e.,

$$sim(QW, CW) = \max\{sim_m(QW, CW) \mid \text{legal map } m\}.$$

As a consequence of this definition, the computation of the similarity requires the systematic construction of such mappings $m$. While the similarity computation by exhaustive search guarantees to find the optimal match, it is computationally not feasible. Hence, we developed a memory-bounded A* search algorithm with an appropriate admissible heuristic to keep similarity computation feasible [4].

## 3 Workflow Clustering

We now describe our approach for using the described semantic similarity measure for clustering workflows. The goal of cluster analysis is to group data objects in such a way that data objects within a cluster are similar while data objects of different clusters are dissimilar. According to Han et al. [6] the following types of clustering algorithm can be distinguished: *Partitioning-based methods* structure the objects into $k$ clusters based on centroids or representatives (algorithms $k$-means and $k$-medoid) while *hierarchical methods* build a tree of clusters in a top-down (e.g. algorithm DIANA) or bottom-up fashion (e.g. algorithm AGNES). *Density-based methods* (e.g. algorithm DBSCAN) identify regions with a high density separated by less dense regions to define clusters. *Conceptual clustering methods* (e.g. algorithms UNIMEM or COBWEB) do not only identify clusters, moreover they identify characteristic descriptions (a concept) for each cluster. As these methods handle the task of clustering differently, they also have different properties with regard to performance, calculation costs and requirements. For this reason there is no clear recommendation for a specific clustering method in general [6]. This leads to the problem of selecting a suitable method for a specific clustering scenario. Additionally, some methods need specific input parameters, e.g. the number of clusters $k$.

### 3.1 Selection of clustering method

To make use of the proposed semantic similarity measure, we selected clustering techniques that are capable of dealing with similarities. We decided to examine two clustering methods of different type. Hence, we selected AGNES as a hierarchical algorithm and $k$-medoid as a partitioning-based algorithm. Both algorithms are based on a distance/similarity computation between the data items. As both algorithms are well known [6], we now describe them only briefly.

***k*-medoid** is a partitioning-based clustering method that separates the objects into a given number of $k$ clusters. First, it randomly chooses $k$ data objects, so-called medoids. The remaining data objects are then assigned to the closest medoid using a distance function $d(x, y)$ that assesses the distance of two data points. Then the total quality of the clustering is calculated. Traditionally, the quality of the clustering is calculated by summing the absolute value of the distance between the medoid and the data points belonging to the cluster. This initial clustering is iteratively improved. Therefore, for each medoid $m$ and each non-medoid data point $o$ a swap operation of $m$ and $o$ is performed and the resulting cluster quality is computed. The clustering representing the best possible is retained and the algorithm continues with the next swapping phase until the total quality of configuration does not improve anymore.

**AGNES** is a agglomerative hierarchical clustering method that starts with creating one cluster for each individual data point. Then the existing clusters

are aggregated in a bottom-up fashion until a complete binary cluster tree is constructed. This aggregation process is performed iteratively. In each iteration a pair of clusters is selected and a new cluster is constructed by merging the data points of the two clusters. The two original clusters are linked as sub-clusters to the new cluster. The selection of the pair of clusters to be merged in each iteration is determined by the closeness of the clusters. Therefore, the set of unlinked clusters is searched for the closest pair of clusters. To assess the closeness of two clusters, several variants are established, called linkage criteria. They assess cluster closeness based on a measure of distance $d(x, y)$ of two data points. *Single linkage* defines cluster closeness as the minimum distance between the points of the two clusters, *complete linkage* uses the maximum distance, while *average linkage* computes the average distance.

### 3.2 Integrating Semantic Workflow Similarity

To apply $k$-medoid and AGNES for clustering workflows is quite straight forward. We assume that the given repository of workflows is represented as a set of workflow graphs as defined in section 2.1. This set is then clustered using the selected clustering algorithm. Hence, instead of n-dimensional data points the graphs are used. Further, the definition of distance $d(W_1, W_2)$ of two data points (here workflows) is replaced by the semantic similarity measure by $d(W_1, W_2) = 1 - sim(W_1, W_2)$. However, a difficulty with this approach arises because the distance functions used in the clustering algorithms are assumed to be symmetric ($d(x, y) = d(y, x)$) while the semantic similarity measure as defined in section 2.2 is asymmetric as it distinguishes a query from a case workflow. To address this problem, several approaches can be taken.

**Modification of the Semantic Similarity Measure.** The definition of the mapping function $m$ and the aggregation function could be modified such that a bidirectional mapping is enforced. In addition, the local similarity measure $sim_\Sigma$ must be restricted to symmetric measures only. While this approach is feasible in principle, it has the significant disadvantage that for applications that require both, retrieval and clustering, two different similarity measures must be modeled, which leads to additional effort.

**Modification of the Clustering Algorithms.** The clustering algorithms can be slightly modified in order to deal with the asymmetric nature of the similarity. As $k$-medoid always compares a medoid with a non-medoid data point, this comparison is already asymmetric. We can apply the semantic similarity measure such that the medoid becomes the query workflow and the data point becomes the case workflow. For AGNES the distance is used to compute the cluster closeness according to the selected linkage criterion. To achieve a symmetric definition of cluster closeness based on the asymmetric semantic similarity measure, the linkage computation can me modified such that for each two workflows $W_1$ and $W_2$ the two similarity values $sim(W_1, W_2)$ and $sim(W_2, W_1)$ are considered.

**Symmetrization of the Similarity Measure.** Instead of modifying the semantic similarity computation or the clustering algorithm, the similarity measure itself can be symmetrized by aggregating the two similarity values $sim(W_1, W_2)$ and $sim(W_2, W_1)$ into a single value by a symmetric aggregation function $\alpha$. For $\alpha$ we consider three different options, namely: $min$, $max$, and $average$. Which option is selected has an impact on the similarity. For example, consider two recipe workflows with a high similarity value in either direction. Thus, these recipes are very similar and contain almost the same cooking steps and ingredients, i.e. one may be used as a replacement of the other. Independent of the choice of $\alpha$, the symmetric similarity value is high as well. However, if one recipe workflow is contained in the other recipe workflow (e.g. a bolognese sauce recipe in a recipe for spaghetti bolognese) the situation is different as one of the two similarity values is high while the other is low. Now, the overall symmetric similarity assessment differs strongly depending on $\alpha$.

### 3.3 Performance Considerations

As the distance/similarity computation is quite frequently called within both clustering algorithms, the computational complexity of the semantic similarity measure involving the search algorithm for the optimal map $m$ is a problem. We address this problem by caching, i.e., we perform the similarity computation in a pre-processing step before the clustering algorithm is started. Thus, a similarity matrix is pre-computed that stores the similarity value of each pair of workflows from the repository. As the individual similarity computations are independent from one another, they can be easily parallelized, taking advantage of multi-core CPUs. Additionally, we improved the performance of the clustering algorithms themselves. We focused on some of the computationally intensive calculation steps such as the estimating of the best swap operation in $k$-medoid and parallelized them as well.

## 4 Case Study: Clustering Cooking Workflows

The aim of this case study was to achieve a first evaluation of the proposed clustering approach on a specific workflow repository. Therefore, the approach was implemented, a repository was created, and a semantic similarity measure was defined. Then, the two proposed algorithms were tested using various variations of their parameters on the workflow repository. We aim at assessing whether the clustering algorithms are helpful to get an insight into the workflow data. As literature emphasizes both the importance and the difficulty of evaluation of clusterings [15], we focus on two purposive evaluations. In an internal evaluation we want to find out, how well the clustering results fulfill the usual requirements of homogeneity and heterogeneity. This evaluation is internal as it is based on indices derived from the clustering results themselves. An external evaluation is also performed to examine how well the clustering resembles a given human

classification. In combination these evaluations should provide a better understanding of the clustering methods and the structure of the specific workflow repository.

### 4.1 Implementation and Repository Creation

We implemented the described clustering algorithms within the CAKE framework[2] [3] that already includes a process-oriented case-based reasoning component for similarity-based retrieval of workflows. The already implemented algorithms for similarity computation are used for clustering. For the domain of cooking recipes, a cooking ontology containing 208 ingredients and 225 cooking preparation steps was developed manually. A specific similarity measure for workflow similarity was defined according to the described framework. This includes the definition of local similarity measures $sim_{\Sigma}$ as well as the definition of a weighting scheme. According to common practice in case-based reasoning, this similarity measure has been optimized manually for the retrieval of recipe workflows. We created a workflow repository containing 1729 workflows (on the average, 11 nodes per workflow) by automated workflow extraction [12]. The workflows have been extracted from `allrecipes.com` by applying a frame-based approach for information extraction using the SUNDANCE parser [11]. Each cooking workflow was automatically annotated by an appropriate semantic description formed by the ontology concepts. The quality of the resulting semantic workflows was ensured by manual postprocessing.

### 4.2 Internal Evaluation

For $k$-medoid clustering the number of clusters $k$ is of high importance. Although it is not that essential for AGNES, it might be also useful to limit the number of clusters. Either the number of clusters can be used as a stopping criterion or an extract of a hierarchical clustering tree can be chosen [6]. In the following experiments we performed clustering with different values for the number of clusters $k$ ranging from 2 to 100. Due to the fact that the results of $k$-medoid depend on the initial random selection of medoids, we repeated each run of $k$-medoid 5 times and selected the best clustering result. We applied the symmetrization approach for the similarity measure using all three variants: $min$, $max$, and $average$ and in addition the asymmetric variant of each algorithm. For AGNES we also varied the linkage approach to test all three variants.

For each clustering we determined three internal measures, namely cohesion, separation, and the silhouette coefficient. The cohesion of a single cluster is equivalent to the average similarity of all pairs of objects of the cluster. The total cluster cohesion measure is the weighted average of the cohesion of the individual clusters. The cohesion values range from 0 to 1, while a high value indicates highly homogenous clusters. The separation of two clusters is defined as the average of all distances (1 - similarity) between all pairs of elements from

[2] cake.wi2.uni-trier.de

both clusters. The total cluster separation measure, which ranges from 0 to 1, is the weighted average of the separation of all pairs of clusters. The silhouette coefficients [9] combines the idea of cohesion and separation into a single value ranging from -1 to 1. A negative value corresponds to a case in which the average distance to points in the cluster is greater than the minimum average distance to points in another cluster. A high positive value $> 0.75$ is an indication of a good homogeneity and a good separation.

**Table 1.** Cluster Results for $k$-Medoid

| | Min Symmetr. | Mean Symmetr. | Max Symmetr. | Asymm. $k$-Medoid |
|---|---|---|---|---|
| Optimal $k$ | 2 | 2 | 3 | 2 |
| Silhouette | 0.16 | 0.15 | 0.07 | 0.06 |
| Cohesion | 0.22 | 0.29 | 0.36 | 0.28 |
| Separation | 0.82 | 0.75 | 0.67 | 0.74 |

Table 1 shows the results for the different variants of $k$-medoid. For each algorithm the results for the number of clusters $k$ is shown which leads to the highest value of the silhouette coefficient. The best silhouette coefficients vary from 0.06 to 0.16, typically suggesting solutions with 2 - 3 clusters. While the silhouette coefficient enables evaluating how well a clustering result fulfills the goals of heterogeneity and homogeneity, it can be stated that even the best clustering results of these combinations don't reveal a strong cluster structure in the workflow repository. According to Kaufmann and Rousseeuw [9] a strong structure leads to silhouette values between 0.75 to 1. This interpretation is supported by cohesion and separation values. The clusters found by $k$-medoid are quite heterogenous due to high separation values ranging from 0.67 to 0.82 but not very homogenous as the cohesion values ranges from 0.22 to 0.36 only.

Table 2 shows the results for AGNES for each combination of algorithm and linkage (SL=single linkage, AL=average linkage, CL=complete linkage). For each combination the results for the number of clusters $k$ is shown which leads to the highest value of the silhouette coefficient. The best silhouette coefficients vary from 0.05 to 0.22, which is in line with the results from $k$-medoid confirming that there is no strong structure in the workflow repository. Examining the other measures also confirms this interpretation. Cohesion varies from 0.20 to 0.39,

**Table 2.** Cluster Results for AGNES

| Algorithm | Min Symmetr. | | | Mean Symmetr. | | | Max Symmetr. | | | Asymm. AGNES | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Linkage | SL | AL | CL | SL | AL | CL | SL | AL | CL | SL | AL | CL |
| optimal $k$ | 2 | 3 | 2 | 4 | 23 | 2 | 2 | 9 | 2 | 3 | 2 | 2 |
| Silhouette | 0.21 | 0.22 | 0.16 | 0.11 | 0.16 | 0.05 | 0.09 | 0.16 | 0.06 | 0.05 | 0.18 | 0.13 |
| Cohesion | 0.23 | 0.20 | 0.20 | 0.29 | 0.35 | 0.27 | 0.35 | 0.39 | 0.34 | 0.29 | 0.27 | 0.27 |
| Separation | 0.82 | 0.86 | 0.86 | 0.73 | 0.75 | 0.76 | 0.68 | 0.69 | 0.70 | 0.74 | 0.77 | 0.76 |

hence homogeneity is very limited. Separation varies from 0.69 to 0.86 which means that there is quite a heterogeneity between different clusters.

### 4.3 External Evaluation

The goal of the external evaluation was to evaluate whether the clustering methods produce clusters of recipe workflows similar to the structure in a cookbook, i.e., a classification into salads, soups, etc. Because this classification information is not available in the current repository and because of the lack of structure identified in the internal evaluation, we decided to manually select and classify a small subset of recipe workflows. We defined 5 classes, namely salads, soups, cake, bread and pork and selected 25 recipes for each class. Then we applied the clustering methods for the selected workflows only. Due to space limitations we just present the results of $k$-medoid using the mean symmetrization approach, however the results for AGNES are quite comparable.

**Table 3.** Clustering results (3 classes)

| | | Classification | | |
|---|---|---|---|---|
| | | Cake | Salad | Soup |
| Cluster | 1 | 20 | 0 | 0 |
| | 2 | 5 | 21 | 0 |
| | 3 | 0 | 4 | 25 |

**Table 4.** Clustering results (5 classes)

| | | Classification | | | | |
|---|---|---|---|---|---|---|
| | | Cake | Salad | Soup | Bread | Pork |
| Cluster | 1 | 15 | 0 | 0 | 3 | 0 |
| | 2 | 2 | 14 | 1 | 1 | 7 |
| | 3 | 0 | 8 | 14 | 0 | 8 |
| | 4 | 8 | 2 | 0 | 21 | 3 |
| | 5 | 0 | 1 | 10 | 0 | 7 |

Table 3 shows the results of clustering 75 workflows belonging to the classes cake, salad, and soup. The results show that the found classification of the clustering algorithm matches well with the external classification. No soup recipe was classified wrong, while approx. 80% of the salad and cake recipes were classified correctly, leading to an overall classification accuracy of 88%. Table 4 shows the results of clustering all 125 workflows. It turns out that the classification structure is less well present. For example, the soup recipes, which were well classified in table 3, are nearly equally divided among two clusters. The pork recipes are spread among 4 clusters. The overall classification accuracy drops down to 56.8%.

To examine the clustering result more deeply we decided to determine the cohesion of each of the five classes and to compare them with the cohesion of the found five clusters (see table 5). Overall the cohesion of the classification is only 0.34 and is thus not very high. The class of soup and pork recipes have the lowest cohesion, which explains that they are not well clustered as shown in table 4. The overall cohesion that results from the clustering is even slightly higher than the cohesion of the manual classification of the recipes.

Finally, it can be concluded that clustering algorithms using the semantic similarity won't classify all recipes as one would expect in a cookbook. This

**Table 5.** Cohesion of classification and clusters

| Classification | | | | | | Clustering | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Cake | Salad | Soup | Bread | Pork | Average | 1 | 2 | 3 | 4 | 5 | Average |
| 0.40 | 0.30 | 0.34 | 0.37 | 0.30 | 0.34 | 0.47 | 0.32 | 0.31 | 0.35 | 0.33 | 0.35 |

is because the semantic similarity measure is based on the workflow structure and thus includes the preparation of the dishes. Contrary to this, the cookbook classification aims at the purpose or outcome of the recipes. However, two recipes with different purpose could be prepared quite similarly (e.g. diced beef and beef salad). On the other hand, two recipes with the same purpose could be prepared quite differently (e.g. a tomato salad and a beef salad).

## 5 Conclusions, Related and Future Work

Workflow clustering is a new research topic, which is particularly important when the size and the availability of workflow repositories is further increasing. We have explored how the $k$-medoid and the AGNES algorithms can be adapted for workflow clustering based on a semantic similarity measure. Unlike previous work on workflow and process clustering [7,8,13,5] our approach enables to configure the semantic similarity measure in relation to a domain ontology of data and task items. Thereby it allows to control the cluster algorithm such that the workflows are clustered according to the intended meaning of similarity. Thus, our approach is generic and can be applied to various domains by adapting the ontologies and similarity measures. As we have already applied our similarity measure to scientific workflows [4], we believe that our method could be considered an alternative to the work proposed by Silva et al. [13] specifically for this domain.

We have systematically applied the algorithms to analyze an automatically extracted set of cooking workflows, which is a workflow domain that has not yet been investigated by previous work on workflow clustering. The analysis revealed that there is only little cluster structure in the examined workflows, i.e., that the kind of preparation of the recipes varies a lot from recipe to recipe. The application of the algorithms to the analysis of a reduced set of workflows that have been manually classified according to five classes was only able to partially discover the given classification. However, it also discloses the fact that traditional classifications of recipes in a cookbook don't always resemble with the similarity of the preparation workflows.

Future work will include applying the presented methods on different data sets in different domains, e.g. for scientific workflows. The myExperiment platform could provide a good source of workflows for this purpose [13]. Further, we will investigate whether the cluster methods can be applied to derive an index structure for the repository that can be exploited to improve the performance of similarity-based workflow retrieval. Further, we aim at extending our approach to density-based clustering as proposed by Ekanayake et al. [5].

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

## References

1. Van der Aalst, W.M.: Process mining. Springerverlag Berlin Heidelberg (2011)
2. Bergmann, R.: Experience Management - Foundations, Development Methodology, and Internet-Based Applications, vol. LNAI 2432. Springer (2002)
3. Bergmann, R., Freßmann, A., Maximini, K., Maximini, R., Sauer, T.: Case-based support for collaborative business. In: Roth-Berghofer, T.R., Göker, M.H., Güvenir, A.H. (eds.) Advances in CBR. vol. 4106, pp. 519–533. Springer (2006)
4. Bergmann, R., Gil, Y.: Similarity assessment and efficient retrieval of semantic workfows. Information Systems Journal (2012), `http://www.wi2.uni-trier.de/publications/2012_BergmannGilISJ.pdf`
5. Ekanayake, C.C., Dumas, M., García-Bañuelos, L., Rosa, M.L., ter Hofstede, A.H.M.: Approximate clone detection in repositories of business process models. In: Business Process Management - 10th International Conference. Lecture Notes in Computer Science, vol. 7481, pp. 302–318. Springer (2012)
6. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann (2000)
7. Jung, J.Y., Bae, J.: Workflow clustering method based on process similarity. In: Computational Science and Its Applications - ICCSA 2006, pp. 379–389. Lecture Notes in Computer Science, Springer (2006)
8. Jung, J.Y., Bae, J., Liu, L.: Hierarchical clustering of business process models. International Journal of Innovative Computing, Information and Control 6(12 A) (2009)
9. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data - An Introduction to Cluster Analysis. John Wiley, New York (1990)
10. Montani, S., Leonardi, G.: Retrieval and clustering for business process monitoring: Results and improvements. In: Díaz-Agudo, B., Watson, I. (eds.) ICCBR. Lecture Notes in Computer Science, vol. 7466, pp. 269–283. Springer (2012)
11. Riloff, E., Phillips, W.: An introduction to the sundance and autolog systems. Tech. rep., School of Computing, University of Utah. (2004)
12. Schumacher, P., Minor, M., Walter, K., Bergmann, R.: Extraction of procedural knowledge from the web. In: WWW'12 Workshop Proceedings. ACM (2012)
13. Silva, V., Chirigati, F., Maia, K., Ogasawara, E., de Oliveira, D., Braganholo, V., Murta, L., Mattoso1, M.: Similarity-based workflow clustering. Journal of Computational Interdisciplinary Sciences 2(1), 23–35 (2011)
14. Song, M., Günther, C.W., van der Aalst, W.M.P.: Trace clustering in process mining. In: Ardagna, D., Mecella, M., Yang, J. (eds.) Business Process Management Workshops. Lecture Notes in Business Information Processing, vol. 17, pp. 109–120. Springer (2008)
15. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison-Wesley (2005)
16. Workflow Management Coalition: Workflow management coalition glossary & terminology. http://www.wfmc.org/standars/docs/TC-1011_term_glossary_v3.pdf (1999), last access on 05-23-2007