MAC/FAC Retrieval of Semantic Workflows

Ralph Bergmann and **Alexander Stromer**
University of Trier - Department of Business Information Systems II
D-54286 Trier, Germany
Email: bergmann@uni-trier.de, Web: www.wi2.uni-trier.de

Abstract

This paper presents a novel two-step retrieval method for semantic workflow cases, inspired by the MAC/FAC (many are called, but few are chosen) model proposed by Gentner and Forbus. MAC/FAC retrieval is motivated by the computational complexity of graph matching, which is usually involved in the similarity-based retrieval of workflows. An additional computationally efficient retrieval step (MAC stage) is introduced prior to the graph-based retrieval (FAC stage) to perform a pre-selection of potentially relevant cases. The MAC stage is based on a feature representation of the workflows automatically derived from the original graph-based representation. In the paper, we briefly introduce previous work on the semantic workflow retrieval and then we describe the pre-selection step in more detail. A comprehensive evaluation with case bases from the cooking domain is reported with demonstrates that the retrieval time can be significantly reduced without significant negative impact on the retrieval quality.

Introduction

Process-oriented case-based reasoning (POCBR) addresses the challenges that occur when applying case-based reasoning (CBR) to process-oriented areas such as business-process management or workflow management. One major difficulty arises due to the fact that in POCBR, cases involve complete or partial descriptions of processes or workflows, which leads to complex case representations involving structural information. As a consequence, the case retrieval time can become very high, because a complex case representation leads to similarity measures, which are computationally expensive. Currently, graph-based approaches are used to represent and retrieve workflow cases (Kendall-Morwick and Leake 2011; Bergmann and Gil 2011; Montani and Leonardi 2012). The graph-based retrieval is computationally expensive as the similarity computation involves a kind of graph matching. Current experiments have shown that graph-based approaches work sufficiently fast only for quite small case bases.

Today, however, the size of workflow repositories (case bases) is significantly increasing in many domains and consequently, current graph-based retrieval methods reach their limits. For example, recent research on methods for automatic workflow extraction from text (Schumacher et al. 2012) enables obtaining large workflow repositories from textually described workflows on the Internet. Also, recent efforts on workflow sharing supported by new standards for workflow representation easily lead to repositories of larger scale and ask for methods that enable an efficient knowledge-intensive search.

This paper addresses the problem of efficient retrieval from case bases containing semantically annotated workflow cases. In line with recent similar research in this area (Kendall-Morwick and Leake 2011; Kendall-Morwick, J. and Leake, D. 2012; Bergmann, R. et al. 2012), we developed and investigate a novel, two-step retrieval approach for workflows, inspired by the MAC/FAC ("Many are called, but few are chosen") model originally proposed by Gentner and Forbus (1991). The first retrieval step (MAC phase) performs a rough and efficient pre-selection of a small subset of cases from a large case base. Then, the second step (FAC phase) is executed to perform the computationally expensive graph-based similarity computation on the pre-selected cases only. This method improves the retrieval performance, if the MAC stage can be performed efficiently and if it results in a sufficiently small number of pre-selected cases. However, there is a risk that the MAC phases introduces retrieval errors, as it might disregard highly similar cases due to its limited assessment of the similarity. Hence, the retrieval approach for the MAC phase must be carefully designed such that it is efficient and sufficiently precise in assessing the similarity. This is the major challenge addressed in this paper. Our focus is to design a MAC phase for semantic workflow retrieval such that retrieval errors are mostly avoided. The MAC phase is based on a simplified case representation using features that can be derived from the workflow representation without introducing additional domain knowledge.

The remainder of this paper is organized as follows: In the next section, we briefly introduce the graph-based approach for similarity-based retrieval of semantic workflows (Bergmann and Gil 2011; 2012). We then describe the developed MAC/FAC approach in detail. Finally, we present an experimental evaluation in which we analyze retrieval quality and retrieval time of MAC/FAC compared to the graph-based retrieval. The evaluation is performed in the domain

of cooking, using a case base of 1729 cases. Each case is a cooking recipe in which are workflows are used to describe the cooking instructions for cooking a particular dish (Schumacher et al. 2012).

Similarity-Based Retrieval of Semantic Workflows

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" (Workflow Management Coalition 1999). 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*. 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*. Today, graph representations for workflows are widely used in process-oriented CBR. In this paper we build upon the workflow representation using semantically labeled graphs (Bergmann and Gil 2011; 2012), which is now briefly summarized. This graph representation enables modeling related semantic similarity measures which are well inline with experts assessment. Specific heuristic search algorithms for computing the semantic similarity for graphs have been developed, but their scalability with growing case bases is quite limited. This is caused by the inherent computational complexity of graph similarity.

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 Ω and a semantic description from a set Σ. 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 Ω 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. Σ 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. Figure 1 shows a simple fragement 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).

Based on this representation, a framework for modeling

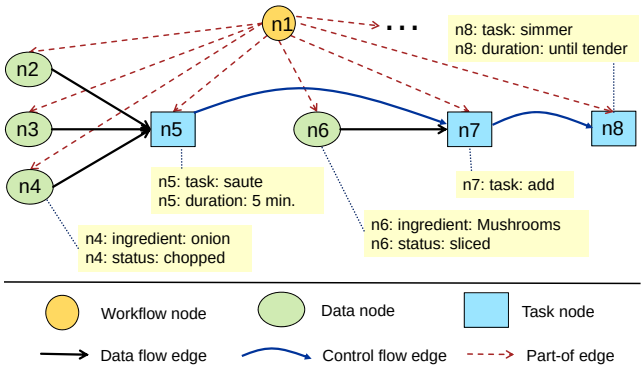

Figure 1: A sample workflow graph

semantic workflow similarity is defined. It 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. In the context of this paper, the node similarity is of importance (as it is also used in the MAC phase), which 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 are considered dissimilar; there similarity is always zero. The similarity of nodes of equal type is defined by the similarity of the semantic descriptions. Due to space the limitations of this paper, we refer to (Bergmann and Gil 2012) for more details and examples of how such local similarity measures look like. The similarity of a query workflow QW and a case workflow CW is then 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 a particular mapping m the overall workflow similarity $sim_m(QW, CW)$ is computed by a particular aggregation (e.g. a weighted average) of the local similarity values computed by using sim_N and sim_E. The overall workflow similarity $sim(QW, CW)$ is then 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, which is the cause for the computational complexity of this approach.

Workflow retrieval requires selecting the k most similar cases from the case base $CB = \{CW_1, \ldots, CW_n\}$ and hence it requires to compute the similarity $sim(QW, CW_i)$ for each case from the case base. Consequently, the retrieval time increases linearly with the size of the case base, but due to construction of the optimal mapping m, it shows a factorial growth with the number of nodes in the worst case.

A MAC/FAC Approach to Workflow Retrieval

To overcome the performance limitations of the described retrieval approach, we now investigate a retrieval method

based on the MAC/FAC (Gentner, D. and Forbus, K.D. 1991) idea. The basic idea behind MAC/FAC is very simple: it is a two-step retrieval approach that first performs a rough pre-selection of a small subset of cases from a large case base. This pre-selection is the MAC stage ("Many Are Called"), which is performed using a selection method which is computationally efficient even for large case bases. For example, cases may be stored in a relational data base and the pre-selection can be performed by an SQL query (Schumacher and Bergmann 2000). Then, the second step called FAC phase "Few Are Choosen") is executed, which only uses the pre-selected cases to perform the computationally expensive similarity computation. This method improves the retrieval performance, if the MAC stage can be performed efficiently and if it results in a sufficiently small number of pre-selected cases that allows applying the complex similarity measure for retrieval.

The major difficulty with MAC/FAC retrieval in general is the definition of the filter condition of the MAC stage. Since cases that are not selected by the MAC stage will not occur in the overall retrieval result, the completeness of the retrieval can be easily violated if the filter condition is too restrictive. Hence, retrieval errors, i.e., missing cases will occur. On the other hand, if the filter condition is less restrictive, the number of pre-selected cases may become too large, resulting in a low retrieval performance. To balance retrieval error and performance, the filter condition should be a good approximation of the similarity measure used in the FAC stage, while at the same time it must be efficiently computable to be applicable to a large case base in the MAC stage.

We address this problem by proposing an additional feature-based case representation of workflows, which simplifies the original representation while maintaining the most important properties relevant for similarity assessment. This representation is automatically derived from the original graph-based representation. The MAC stage then selects cases by performing a similarity-based retrieval using an traditional similarity measure. This similarity measure will partially use the local similarity functions of the graph-based retrieval but in a more simple manner, ignoring the structural properties of the workflow graph. The resulting retrieval method is thus more efficient. A further important property of this realization of the MAC stage is that the number of selected cases can be easily controlled. Therefore, we introduce a parameter we call *filter size s*, which specifies the number of cases resulting from the MAC stage. Hence, the MAC stage retrieves the s-most similar cases using feature-based retrieval. The choice of the filter size determines the behavior of the overall retrieval method with respect to retrieval speed and error in the following manner: the smaller the filter size, the faster the retrieval but the larger the retrieval error will become. Hence, an appropriate choice of the filter size is important.

Feature Representation

We now introduce our approach in more detail. A feature-based case base $CB' = \{CW'_1, \ldots, CW'_n\}$ is computed offline, i.e., prior to performing the retrieval. Therefore, each case CW'_i is derived from the corresponding case CW_i of the original graph-based case base CB. In the representation of a feature-based case CW', two types of features are considered: *semantic features* and *syntactic features*. A vector V_{sem} represents the semantic features derived from the workflow graph, while a vector V_{syn} represents the syntactic features, thus $CW' = (V_{sem}, V_{syn})$.

Currently, four semantic features are considered, i.e., $V_{sem} = (D, A, D^*, A^*)$. The first feature D is related to the individual data nodes and just stores them as a set of nodes (while maintaining the link with their semantic descriptions from Σ). In the same manner, the second feature A consists of the set of task nodes.

$$D = \{d \in N | T(d) = DataNode\}$$

$$A = \{a \in N | T(a) = TaskNode\}$$

With these two features, the linking of the nodes is completely ignored. In order to include at least local information about the direct neighbors of a node, the features D^* and A^* represent for each node in the set also the set of its direct neighbors in the graph.

$$D^* = \{(d, con_T(d)) | d \in D\}$$

$$A^* = \{(a, con_A(a)) | a \in A\}$$

with

$$con_T(d) = \{a \in A | (d, a) \in E \vee (a, d) \in E\}$$

$$con_A(a) = \{d \in D | (d, a) \in E \vee (a, d) \in E\}$$

Consequently, D^* is the set of data nodes together with the directly linked task nodes and A^* is the set of task nodes together with the directly linked data nodes.

The syntactic features, however, are simple numerical features that together build a kind of profile reflecting the size of the graph. Hence, V_{syn} is defined as $V_{syn} \in \mathbb{R}^f$, with f being the number of features. These features reflect the number of the various components the graph consists of. Currently, we use $f = 9$ syntactic features such as the total number of nodes, the number of specific types of nodes, the number of data flow edges, and the average number and size of subsequences occurring between split and join control flow nodes. The left column of Table 1 gives the list of all features.

Similarity Assessment for Feature Representation

To perform the MAC/FAC retrieval for a given query workflow QW the related feature-based representation QW' of the query is derived in the same manner as for cases in the case base. The similarity measure sim' that compares a query $QW' = (V_{sem_q}, V_{syn_q})$ with a case $CW' = (V_{sem_c}, V_{syn_c})$ is further specified as follows: For both vectors, separate similarity functions are specified. The computed similarity values are then aggregated into the overall similarity. For the two semantic features D and A, the local similarity measure sim_N modeled for the graph-based retrieval is used, but without applying any mapping m. Let's assume, a set of data nodes in the query $D_q = \{q_1, \ldots, q_u\}$ and a set of data nodes in the case $D_c = \{c_1, \ldots, c_v\}$. The

measure sim_N is used to assess the similarity between each pair of nodes (q_i, c_j). Based on this, a local similarity measure for D is specified as follows:

$$sim'_D(D_q, D_c) = \frac{1}{u} \cdot \sum_{i=1}^{u} max_{j=1...v}\{sim_N(q_i, c_j)\}$$

Hence, for each data node in the query, the best matching data node in the case is selected. Their similarity is aggregated into the overall similarity for D. This is obviously still a kind of mapping, but it is less constrained with respect to the mapping m computed in the graph-based approach, because each node is mapped independent of the mapping of the other nodes and independent of any linking. Thus, the time complexity is only polynomial with the number of nodes. The local similarity measure $sim'_A(A_q, A_c)$ for A, the set of task nodes, is specified analogously.

The local similarity measures for D^* and A^* require in addition the similarity assessment of the linked nodes $con_T(d)$ and $con_A(a)$, respectively. Assume $D^*_q = \{(q_1, con_T(q_1)) \ldots, (q_u, con_T(q_u))\}$ be the query value of D^* and $D^*_c = \{(c_1, con_T(c_1)) \ldots, (c_v, con_T(c_v))\}$ be the case value of D^*. Then, the local similarity is defined as follows:

$$sim'_D*(D^*_q, D^*_c) = \tfrac{1}{u} \cdot \sum_{i=1}^{u} max_{j=1...v}\{ \\ 0.5 \cdot (sim_N(q_i, c_j) + sim'_A(con_T(q_i), con_T(c_j)))\}$$

In the same manner, the local similarity measure sim'_{A^*} is defined. The time complexity of the similarity computation for D^* and A^* is still polynomial with the number of nodes.

In addition, the similarity of the syntactic features is considered. Here, we apply a standard similarity measure $sim' : \mathbb{R}^2 \rightarrow [0, 1]$. In order to aggregate the local similarity values into the global similarity, feature weights are considered for the features in V and for the semantic features D and A. Let's assume, $W = (w_1, \ldots, w_f)$ is a vector of feature weights for the syntactic features and w_d, w_a, w^*_d, w^*_a are the feature weights for the semantic features, respectively. Then, the global similarity between the query and the case for feature-based retrieval is specified as follows:

$$sim'(QW', CW') = (w_a \cdot sim'_A(A_q, A_c) + \\ w_d \cdot sim'_D(D_q, D_c) + w^*_a \cdot sim'_{A^*}(A_q, A_c) + \\ w^*_d \cdot sim'_{D^*}(D_q, D_c) + \sum_{i=1}^{f}(w_i \cdot sim'(v_{q_i}, v_{c_i})))/ \\ (w_d + w_a + w^*_a + w^*_d + \sum_{i=1}^{f} w_i)$$

MAC-Phase Retrieval

The selection of cases $CW'_1, ..., CW'_s$ during the MAC phase is performed by a similarity-based retrieval from CB' using the similarity measure $sim'(QW', CW'_i)$. Thereby, the s most-similar cases are retrieved (s is the filter size), which requires to compute the similarity between the query and each case of the case base. Hence, the computation time of the MAC phase increases linearly with the size of case base, but each similarity computation is less costly than the similarity computation for the graph-based retrieval.

Experimental Evaluation

The aim of the following experiments is to evaluate whether the proposed MAC/FAC retrieval approach using the suggested feature representation helps to improve the retrieval time without significantly reducing the retrieval quality caused by retrieval errors. We created a workflow repository containing 1729 workflows (on the average, 11 nodes per workflow) by automated workflow extraction from cooking recipes (Schumacher et al. 2012). The quality of the automatically extracted workflows was ensured by manual post-processing. Further, a cooking ontology containing 208 ingredients and 225 cooking preparation steps was developed manually. Each cooking workflow was automatically annotated by an appropriate semantic description formed by the ontology concepts. From this repository, a set of case bases with increasing sizes (from 200 to 1700 cases) was randomly generated. As part of the development of a workflow-based recipe retrieval system which participated in the computer cooking contest in 2012[1], specific similarity measures for workflow similarity according to the described framework were developed, which we use in the following evaluation. We implemented our MAC/FAC retrieval approach in the CAKE framework[2] and executed the following experiments on Windows 7 Enterprise 64-bit, running on a PC with an Intel Core i5-750 CPU @ 2.4GHz and 8.00 GB RAM.

Quality Criterion

In order to assess the impact of the retrieval error introduced in the MAC phase, we used a quality criterion which weighs the retrieval error with respect to the position in the retrieval list: missing cases at the top of the result list have a larger impact on the quality than missing cases at the end of the list:

$$q(err, r) = \sum_{i=1}^{r} err(i) \cdot (2 + r - i) / \sum_{i=1}^{r} (2 + r - i)$$

Here, the quality q for a particular retrieval result list with size r is a value within $[0, 1]$, while the error function $err(i) = 1$ if the case at the retrieval position i from the original retrieval (without MAC filtering) is missing due to a retrieval error caused in the MAC phase, otherwise $err(i) = 0$.

Selection of Weights

As the resulting retrieval quality depends on the used feature weights, we first aim at finding a good weight vector. For this purpose, we evaluated the retrieval quality on the case base with 1700 cases and filter size $s = 30$ by applying a MAC phase using a single feature only. We iteratively determined the resulting quality value (for retrieval result list size $r = 1$) for each feature. The higher the quality value, the more important is the feature. Due to this idea, we define the weight vector according to the quality values for the features, i.e., the quality value is used directly as weight value

[1] www.computercookingcontest.net

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

in the vector. The resulting quality values are shown in Table 1 in the column weight setting 1. We found that the semantic features lead to a much better quality value than the syntactic features. Further, among the semantic features D^* and A^* outperform D and A. In order to better understand the impact of the semantic features, we define three feature subsets of semantic features (by setting the weight of features that are not used to 0) that allow to asses the impact of the second and third best feature (D^* and A), while always including the best feature A^*. The three additional weight settings are shown in Table 1.

	Weight setting			
Feature	1	2	3	4
Semantic Features				
D	0.595	0	0	0
A	0.7	0	0.7	0.7
D^*	0.755	0.755	0	0.755
A^*	0.91	0.91	0.91	0.91
Syntactic Features				
No. of Nodes	0.035	0	0	0
No. of Data Nodes	0.045	0	0	0
No. of Task Nodes	0.04	0	0	0
No. of Control Nodes	0.055	0	0	0
No. of AND Nodes	0.055	0	0	0
No. of XOR Nodes	0.045	0	0	0
No. of Dataflow Edges	0.035	0	0	0
Avg. No. of Sequences	0.055	0	0	0
Avg. Len. of Sequences	0.065	0	0	0

Table 1: Feature vector with 4 chosen weight settings

For each of the four weight settings, we evaluated the impact on the retrieval quality as shown in Figure 2. Instead of investigating the retrieval quality directly, we determined the filter size s required in the MAC phase to reach a quality of 1, i.e., to guarantee that the MAC phase does not cause any retrieval error. The lower the filter size, the faster the subsequent FAC phase will be. The results clearly demonstrate the advantage of weight setting 1, i.e., when all features are

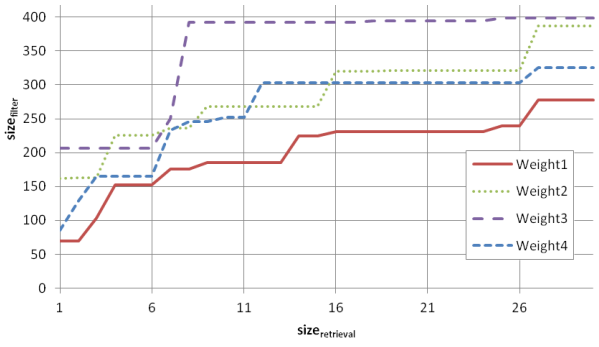

Figure 2: Required filter size s to achieve quality $q = 1$ for four weight settings and different sizes of the result list ($\text{size}_{\text{retrieval}}$), on a case base with 500 cases.

used. Hence, this feature vector is used in the remainder of this evaluation.

Retrieval Quality

The next evaluation aims at analyzing the relationship between retrieval quality and filter size. Clearly, increasing the filter size should increase the retrieval quality monotonously as more cases are added as outcome of the MAC phase. Further, we expect that for larger case bases larger filter sizes are required in order to achieve the same quality. Figure 3 shows the retrieval quality depending on the filter size ($\text{size}_{\text{retrieval}}$) for case bases of different sizes. In this experiment, the length of the result list ($\text{size}_{\text{retrieval}}$) is fixed to 15.

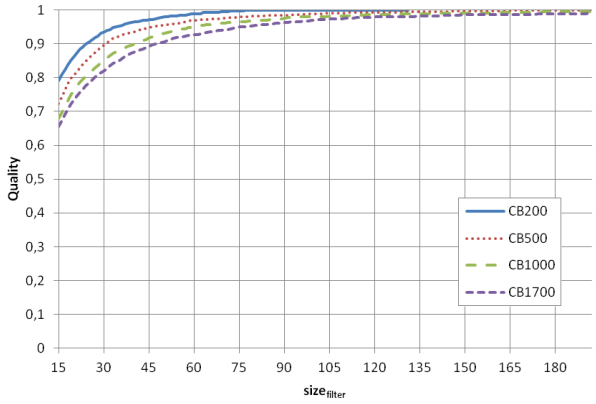

Figure 3: MAC/FAC quality for $\text{size}_{\text{retrieval}} = 15$

The figure shows in general quite high quality values. For each cases base, they converge quite fast to the value of 1. For example, for the case base with 200 cases, a filter size of 63 is needed to achieve a quality of 0.99.

Performance Improvement

Finally, we investigate the improvement in the retrieval performance caused by the MAC/FAC approach over the original graph-based similarity. The retrieval time for the MAC/FAC approach consists of the time to derive the feature-based representation from the graph-based representation of the query, the time to perform the feature-based retrieval (MAC phase), and the time to perform the graph-based retrieval on the selected cases (FAC phase). The focus of this evaluation is to analyze the retrieval time if the MAC phase is configured such that a high retrieval quality is achieved. We define high retrieval quality by a quality threshold of 0.99. In the following, we report in the results of a series of experiments with the 16 case bases of different size (200 to 1700 cases). For each case base, we experimentally determined the smallest possible filter size such that a retrieval quality of at least 0.99 is achieved. Then, for each case base, the average retrieval performance using MAC/FAC with the identified filter size over 200 different queries is determined. Figure 4 shows the average retrieval time for MAC/FAC together with the retrieval time of the original graph-based retrieval as well as the retrieval time of

the feature-based MAC phases alone. Hence, the difference between the MAC curve and the MAC/FAC curve reflects the retrieval time caused by the FAC phase.

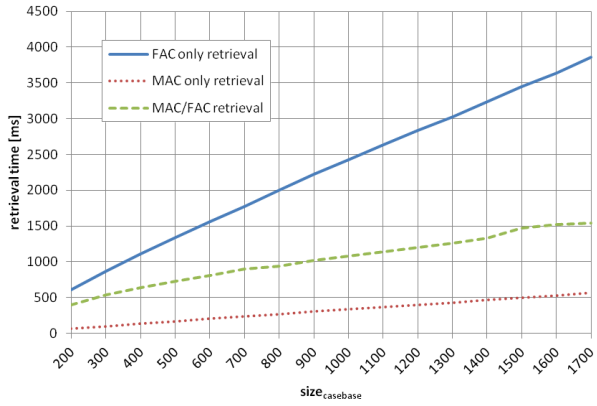

Figure 4: MAC/FAC, MAC only and FAC only retrieval time for $\text{size}_{\text{retrieval}} = 15$

The figure clearly confirms that the MAC retrieval is significantly more efficient than the graph-based (FAC) retrieval: for the case base with 1700 cases it is approximately 7 times faster. More interestingly, also the MAC/FAC approach significantly outperforms graph-based retrieval, while ensuring a very high retrieval quality of 0.99. For the case base of 1700 cases, it is 2.5 times faster than the graph-based retrieval. If the quality threshold is reduced to 0.9, the retrieval speedup for the case base of 1700 cases is approximately 5 (result from an additional experiment, not shown in the figures).

Conclusion

We presented a new MAC/FAC approach to scale the similarity-based retrieval of semantic workflows. A similar method was proposed by Leake and Kendall-Morwick (Leake and Kendall-Morwick 2008; Kendall-Morwick and Leake 2011; Kendall-Morwick, J. and Leake, D. 2012), but they use a different filter method in the MAC phase. Our approach is based on a feature-based representation of workflows, which includes properties that are relevant for the similarity assessment. We found that the semantic features, particularly the features D^* and A^* strongly contribute to the quality of the MAC phase. However, due to the fact that these two features are sets of sets, the resulting similarity computation involves matching as well. Hence, more than 80% of the MAC retrieval time is caused by D^* and A^*. Unfortunately, the set-based nature of these features also prevent us from applying any indexing of the case base or the use of SQL queries (Schumacher and Bergmann 2000) to further improve the retrieval speed of the MAC stage, which is clearly a disadvantage of the proposed features. Hence, future work must focus on a more careful evaluation of the cost and benefits of these features in terms of retrieval quality and time. Further, our investigation so far avoids domain specific features. Hence, the proposed method is domain independent. However, to further improve speed and quality of the MAC phase, domain specific features might be considered in the future. Further, automated methods for optimizing the feature weights and the filter size should be investigated to optimize the performance of the MAC phase.

Acknowledgements

This work was funded by the German Research Foundation (DFG), project number BE 1373/3-1 and by Stiftung Rheinland-Pfalz für Innovation, project WEDA, No. 974.

References

Bergmann, R., and Gil, Y. 2011. Retrieval of semantic workflows with knowledge intensive similarity measures. In *19th International Conference on Case-Based Reasoning, ICCBR 2011*, 17–31. Springer.

Bergmann, R., and Gil, Y. 2012. Similarity assessment and efficient retrieval of semantic workfows. *Information Systems, Special Issue on Process-Oriented Case-Based Reasoning, forthcomming*.

Bergmann, R.; Minor, M.; Islam, S.; Schumacher, P.; and Stromer, A. 2012. Scaling similarity-based retrieval of semantic workflows. In *ICCBR-Workshop on Process-oriented Case-Based Reasoning*, 15–24.

Gentner, D., and Forbus, K.D. 1991. MAC/FAC: a model of similarity-based retrieval. In *Proceedings of the Thirtheenth Annual Conference of the Cognitive Science Society*. Cognitive Science Society.

Kendall-Morwick, J., and Leake, D. 2011. A toolkit for representation and retrieval of structured cases. In *Proceedings of the ICCBR 2011 Workshops*, 111–120.

Kendall-Morwick, J., and Leake, D. 2012. On tuning two-phase retrieval for structured cases. In *ICCBR-Workshop on Process-oriented Case-Based Reasoning*, 25–34.

Leake, D. B., and Kendall-Morwick, J. 2008. Towards Case-Based support for e-Science workflow generation by mining provenance. In *Advances in Case-Based Reasoning, 9th European Conference, ECCBR 2008, Trier, Germany, September 1-4, 2008. Proceedings*, 269–283.

Montani, S., and Leonardi, G. 2012. Retrieval and clustering for business process monitoring: Results and improvements. In *Case-Based Reasoning Research and Development*, volume 7466. Springer. 269–283.

Schumacher, J., and Bergmann, R. 2000. An efficient approach to similarity-based retrieval on top of relational databases. In *Proceedings of the 5th European Workshop on CBR*, volume 1898, 273–284. Springer.

Schumacher, P.; Minor, M.; Walter, K.; and Bergmann, R. 2012. Extraction of procedural knowledge from the web. In *WWW'12 Workshop Proceedings*. ACM.

Workflow Management Coalition. 1999. Workflow management coalition glossary & terminology. http://www.wfmc.org/standards/docs/TC-1011_term_glossary_v3.pdf. last access on 05-23-2007.