Semantic Workflow Retrieval Based on Multiple Features

Rujuan Wang [1], Lei Wang [1] and Chunling Xu [1] [+]

[1] College of Humanities & Sciences of Northeast Normal University, Changchun 130012, China

Abstract. Semantic similarity retrieval workflow is the primary task of semantic workflow reuse. Semantic workflow retrieval methods only focus on structural features similar to existing, ignore the behavior characteristics, affect the retrieved similar semantic workflow in order to improve the overall quality of the work flow, semantic reuse cost, multiple workflow semantic similarity retrieval algorithm is proposed combined with behavior the structure and use of task execution behavior. Close relationship set expression semantics of workflow, combined with the domain knowledge base to construct the semantic workflow task tree index and data index. According to the semantic query workflow, first task relation between neighboring tree index and index data are filtered to obtain the candidate set based on Semantic workflow; then use graph matching similarity algorithm to verify the candidate semantic workflow set, get sorted when the semantic workflow ensemble is selected. Experimental results show that compared with the mainstream semantic workflow retrieval algorithm, the retrieval performance of this method is greatly improved, and the efficiency of workflow reuse is improved.

Keywords: TAR, index, workflow, semantic.

1. Introduction

The quality and efficiency of business process operation is one of the key factors of modern enterprises and organizations to maintain the advantage in the competition. Semantic workflow [1] as a knowledge workflow, provide the more sufficient semantics and data for workflow resource reuse, can improve the efficiency of workflow reuse. However, the existing similar semantic retrieval methods only focus on workflow structure, ignore the behavior characteristics, affecting the overall quality of semantic retrieval workflow, but also improve the semantic workflow reuse will cost. Bergmann et al. proposed graph matching method for semantic retrieval based on Workflow case. This method adopts semantic traversal retrieval method of workflow library; for larger semantic workflow library, is actually not feasible. Forbus et al. proposed a method for similarity retrieval of MAC /FAC (many are called, but few are chosen) model [2]. The model consists of 2 phases: 1) MAC phase using nonstructural matching algorithm with less computation from the item pool to filter out the candidate item sets; 2) FAC phase structure matching algorithm to find the matching items from the candidate item sets. .Zha et al. proposed the concept of the transition adjacency relation (TAR) based on the process model of the label Petri net modeling [3]. Transition adjacency relations in close relationship set TARS are the set of all TAR of the process model. Jin et al. constructs a process model TARIndex based on tag Petri net process model, and proposes a second stage process model retrieval method. This method focuses on the execution behavior of business models, but it can't distinguish cyclic and sequential structures [4].

2. Definition

[+] Corresponding author. Tel.: + 0431-84579066; fax: +0431-84579066t.narue@gmail.com.
E-mail address:wangrujuan_1108@sina.com.

Semantic workflow not only describes the tasks and connections in business processes, but also describes important information such as the semantics, data or resources of tasks, and the semantics of connections between tasks. Semantic workflow can express business workflow and scientific workflow.

Definition 1: Semantic Workflow. Semantic workflow can be formalized as a semantic annotation directed graph $SW = (N, E, S, \tau)$. Which in, $N = N_T \cup N_C \cup N_D$, N_T is a collection of task nodes; N_C is the flow of control node (routing node set); N_D is the data set. $E \subseteq N \times N$ is the set of edges; $S: N \cup E \to \Sigma$ will node or edge is mapped to the semantic description; Σ is a domain related semantic metadata language. $\tau: N \cup E \to \Omega$ each node or edge will be mapped into one type, $\Omega =$ {*task node, data node, control-flow node, control-flow edge, dataflow edge*}. Figure 1 is a semantic workflow SW_1 [5] to describe the cooking process in Italy.
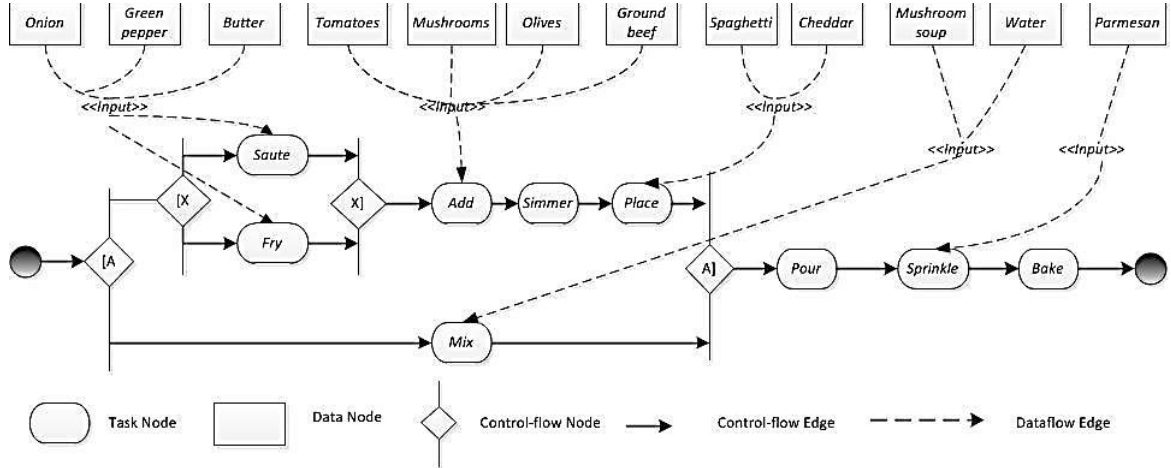

Fig. 1: A sample of semantic workflow.

Definition 2. **Transition Adjacency Relation (TAR)** Assume that *TrS* is a semantic block structured workflow $SW= (N, E, S, t)$ of all possible trajectories set, $\langle a, b \rangle$ is a transition adjacency relation of *SW*, in which, $a, b \in N_T$, N_T is a collection of task nodes, if and only $trace = \langle t_1, t_2, \cdots, t_n \rangle$ an$trace \in T_rS$, $t_i = a, t_{i+1} = b (i \in \{1, 2, \cdots, n - 1\})$, the all sets TAR of SW as TARS.

Semantic workflow can be transformed into a Petri net system with free choice structure. The splitting nodes XORSplit, sink node XORJoin, the LoopStart node and the end node LoopEnd of the block can be mapped to the repository of Petri net system. Complete the prefix TARS. semantic workflow is introduced to calculate the complete prefix semantic workflow launched technology deployment can be obtained by the following methods [6]: 1) to select the n_1 XORJoin node structure block, select a branch is selected as the main branch; a copy of XORJoin n_1' node, select a choice in addition to the main branch outside connect to n_1' and disconnect the branch and n_1, in addition to the main branch until all the selected branch has launched.2) for the n_2 LoopStart node loop structure in the block, determine the main circulation branch; a copy of the LoopStart node n$_2'$, the secondary loop branches connected to n$_2'$ and disconnect the branch and n_2. Fully extended prefix semantic workflow based on the reference literature [7] methods can get the TARS.

Definition 3. The including relationship of TAR. For the semantic workflow $SW_1 = (N_1, E_1, S_1, \tau_1)$, $SW_2 = (N_2, E_2, S_2, \tau_2)$. The task node $a, b \in N_{T.1} \in N_1$, the task node $c, d \in N_{T.2} \in N_2$, $tar_1 = \langle a, b \rangle$, $tar_2 = \langle c, d \rangle$ is transition adjacency relation of SW_1, SW_2 respectively, close to the mission, C_1, C_2, C_3, C_4 are the domain task ontology concept corresponding to the semantic description of *a, b, c, d* respectively. If $C_1 \sqsubseteq C_3$ and $C_2 \sqsubseteq C_4$, said the transition adjacency relation $\langle c, d \rangle$ contains$\langle a, b \rangle$, $\langle a, b \rangle \sqsubseteq \langle c, d \rangle$, or $tar_1 \sqsubseteq tar_2$, the father TAR is tar_2, the son TAR is tar_1.

Definition 4. Similarity of TAR. Semantic workflow $SW_1 = (N_1, E_1, S_1, \tau_1)$, $SW_2 = (N_2, E_2, S_2, \tau_2)$. The task node $a, b \in N_{T.1} \in N_1$, the task node $c, d \in N_{T.2} \in N_2$, $tar_1 = \langle a, b \rangle$, $tar_2 = \langle c, d \rangle$ is transition adjacency relation of SW_1, SW_2 respectively, C_1, C_2, C_3, C_4 are the domain task ontology concept corresponding to the semantic description of a, b, c, d respectively. If $tar_1 \sqsubseteq tar_2$, then $sim(tar_1, tar_2) = 1$, else $sim(tar_1, tar_2) = (sim(C_1, C_3) + sim(C_2, C_4))/2$. Where the compute method of $sim(C_i, C_j)$ can refer to the literature [8].

Definition 5. TARTree. The TARTree of semantic workflow is a multifork tree based on the inclusion relationship of TAR. Each node of it is an item like （*tar, tar.list*）, and a mapping from *tar* to semantic workflow set is established. *tar* is the task neighbor relation, and *tar.list* is a semantic workflow set containing *tar*.

Definition 6. TARTreeIndex. The TARTreeIndex of semantic workflow is TARTree set, which is used to get the set of semantic workflow that contains the set of TAR.

Definition 7. DataIndex. The structure is like (*data, data.list*). In which *data* represents the input data object, and *data.list* represents a semantic workflow set containing data.

3. Multiple Similarity Semantic Workflow Retrieval Method

We find a similar semantic workflow containing a semantic workflow fragment as the main retrieval requirement. This paper presents a MAC/FAC model of multiple similarity retrieval method based on semantic workflow. It is consists of two stages. First, MAC phase consists of 2 steps: 1) filtering operation according to specific semantic workflow database, combining domain task ontology construct the TarTreeIndex, combined with field data to construct DataIndex.2) Computing semantic query workflow TARS, TarTreeIndex filter to the semantic workflow library gets a collection that contains the TARS semantic workflow based on rough selection; then DataIndex filter for roughing the semantic workflow based on set, get the collection of selected semantic workflow contains input data query semantic workflow, called the set of candidate semantic workflow. In the second, FAC using graph edit distance graph matching [9] similar to the set of candidate semantic workflow verification method based on similarity, according to the ranking of candidate semantic workflow.

3.1. TarTreeIndex

The pseudo code of the construction algorithm is shown in algorithm 1.

Algorithm 1
Input: semantic workflow library SWB, domain task ontology TaskOnto;
Output: task close relationship tree index TarTreeIndex.

1） $tarTreeIndex = \emptyset$
2） if $SWB==\emptyset$ then
3）   return $tarTreeIndex$;
4） end if
5） for each $SW \in SWB$ do
6）   *tarS=computeTARS(SW).*
7）   for each $tar \in tarS$ do
8）   if $tarTreeIndex == \emptyset$ then
9）   *tree$_1$ .root=new node(tar), n$_1$ .list.add(SW), tarTarTreeIndex =tarTreeIndex$\cup$ {tree$_1$}*
10）   else n$_1$=new *node(tar), n$_1$.list.add(SW), insertTartreeIndex(SW,n$_1$)*
11）   end if
12）   end for
13） end for
14) return $tarTreeIndex$

In algorithm 1, the function *computeTARS (SW)* is used to calculate the semantic workflow SW. Function *insertTARTree-Index(SW, tarNode)* is used for the TAR node of SW *tarNode* insertion the *tarTreeIndex*, function *insertTARTree (treeRoot,SW, tarNode)* is used for the *tarNode* insert to the TARTree with *treeRoot* as the root node. In the process of constructing tarTreeIndex, each TAR in the *tarS* is inserted into the TARTree or a TARTree as the root node. When a new semantic workflow adds to the semantic library, the update process of tarTreeIndex is basically the same as the process of its establishment. When a semantic workflow is deleted from the library, the TARTree corresponding node is required to be deleted. This can immediately execute or delete all the corresponding nodes at once when the semantic workflow needs to be deleted to a certain amount.

Based on TARTreeIndex, semantic workflow similar to a query semantic workflow can be transformed to semantic workflow that searches and queries semantic workflow execution behavior in semantic workflow

library. The input is semantic workflow library *SWB*, tarTreeIndex, query semantic workflow SW_q, the similarity threshold θ_1 of TAR, output is a semantic workflow set *WFS*. Similar to SW_q execution behavior.

3.2. DataIndex

The pseudo code for establishing the data index DataIndex algorithm is shown in algorithm 2.

Algorithm 2
Input: semantic workflow library *SWB*;
Output: data index dataIndex.

1） $dataIndex = \emptyset$
2） if *SWB*==$\emptyset$ then
3）   return $dataIndex$;
4） end if
5） for each $SW \in SWB$ do
6） $dataS=computeDS(SW)$.
7）   for each $data \in dataS$ do
8）   if $data \notin dataIndex$ then
9） $data.list.add(SW), dataIndex.add(data)$
10）   else $data.list.add(SW)$
11）   end if
12）   end for
13） end for
14） return tarTreeIndex

Retrieval of a similar semantic workflow containing an input data objects that contains a query semantic workflow. The input contains dataIndex SW_q, DataOnto, the similarity threshold θ_2 of data object; the output is data object contains the set of SW_q is similar to the semantic workflow set *WFS*. Here on the semantic workflow consists of a collection of objects for each data set operation is to obtain the input and the number of SW_q objects according to the similar semantic workflow.

3.3. Multiple Similarity Semantic Workflow Retrieval Method

The MAC/FAC model similarity retrieval method is used. Considering the behavior and structure characteristics of domain knowledge, semantic workflow, proposed multiple semantic similarity retrieval method for semantic query workflow. Workflow SW_q, TarTreeIndex and DataIndex in MAC stage based on semantic workflow library filtering, semantic workflow gets the candidate set; using graph matching similarity method to verify the candidate semantic workflow set in FAC phase, the candidate semantic workflow scheduling, alternative semantic workflow workflow reuse.

- Semantic workflow filtering. For the query semantic workflow SW_q, filtering operations by *B* and *D*. The first use of task relation between neighboring tree index TarTreeIndex roughing filter operation, by *B*, get the semantic workflow set *WFS₁*; then using the data object index selection of DataIndex filter for *WFS₁* operation, get the semantic workflow set *WFS₂*, *WFS₂* is called a candidate semantic workflow.
- Validation of candidate semantic workflow set. In the verification phase, the query on semantic workflow SW_q, using semantic workflow graph matching similarity method of candidate semantic workflow set *WFS₂* is verified and sorted according to the degree of similarity and SW_q. Semantic matching workflow graph similarity graph edit distance based on, and graph edit distance reflects the difference between the structural features of semantic workflow in acquiring semantic workflow graph matching, the matching routing node priority semantic workflow, based on this and then matching task nodes, and finally gets the mapping among all nodes. The semantic similarity of task nodes is described by its semantic similarity. Due to the use of the greedy strategy may generate non optimal matching, but in MAC the 2 step filtering operation, which can reduce the probability of pseudo code omitted here.

4. Experiment and Result Analysis

The experimental data set from WikiTaaable[i]and RecipeandRecipesource[ii], which has a total of more than 1000 Italy pasta recipes (pasta). The field task ontology in TaskOnto experiment from WikiTaaable Culinary actions ontology, the field data objects ontology DataOnto from WikiTaaable Food ontology.

In this experiment the semantic workflow library randomly selected 10 semantic workflows, component size is 10 sets. Set the similarity threshold $\theta_1 = 0.6$, the similarity threshold of data objects $\theta_2 = 0.5$, workflow semantic similarity threshold $\theta_3 = 0.5$ for each query semantic workflow this paper respectively carry out the retrieval, rendering algorithm and the TARTreeIndex+DataIndex P-R (precision-recall) curve of $k = 1$, based on the TARIndex algorithm is applied to semantic similarity retrieval method of workflow similarity algorithm, greedy strategy map, such as the use of semantic TARS, semantic workflow task nodes ignore the data stream such as the similarity of the drawing. P-R curve by TARIndex algorithm and graph matching algorithm, as shown in figure 2:
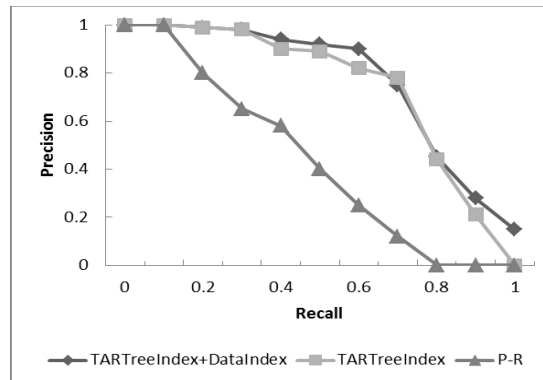

Fig. 2: P-R curves of three retrieval algorithms.

Can be seen from Figure 2, TARTreeIndex+DataIndex algorithm and TARTreeIndex algorithm in P-R curve above of graph matching algorithm, shows that the introduction of behavioral characteristics index based on semantic similarity improves the workflow retrieval performance of the TARIndex algorithm of P-R curve in R=0.7 is higher than that of TARTreeIndex+DataIndex algorithm, this is because the semantic workflow in library the 2 execution behavior sufficiently similar data set have different acceptable semantic workflow. But the P-R curve of the TARTreeIndex+DataIndex algorithm in the most P-R above the curve TARIndex algorithm, according to structural characteristics and introducing the data object index to improve the retrieval performance of similar semantic workflow from the whole. The above shows that the TARTreeIndex+DataIndex algorithm of semantic similarity workflow retrieval performance is better than the other two algorithms.

5. Conclusion

This paper puts forward the semantic workflow tasks close to the concept of relationship and construct with the domain knowledge task relation between neighboring tree index TARTreeIndex+DataIndex, is used to filter the semantic workflow specified behavior characteristics; constructed with data index DataIndex domain knowledge, get used to filter contains semantic workflow specific data; the TARTreeIndex+DataIndex and DataIndex filter graph matching multiple similarity retrieval algorithm to verify the workflow semantic similarity algorithm; through the experiments, this algorithm with acceptable cost to improve the retrieval performance of similar semantic workflow.

6. Acknowledgment

7. References

[i] http://wikitaaable.loria.fr
[ii] http://www.recipesource.com

[1] Bergmann R, Gil Y. Similarity assessment and efficient retrieval of semantic vorkflows [J]. Information Systems, 2014 40(1):115-127.

[2] Forbus K D, Gentner D, Law K. MAC/FAC: A model of similarity based retrieval [J]. Cognitive Science, 1995, 19(2): 141-205

[3] Zha Haiping, Wang Jianming, Wen Lijie, et al. A workflow net similarity measure based on transition adjacency relations [J]. Computers in Industry , 2010, 61(5): 463-471

[4] Jin Tao, Wang Jianmin, Wen Lijie. Efficient retrieval of similar workflow models based on behavior [G] //Web Technologies and Applications. Berlin: Springer, 2012:677-684

[5] Dufour--Lussier V, Leber F, Lieber J, et al. Automatic case acquisition from texts for process-oriented case-based reasoning [J]. Information Systems, 2014, 40(1):153-167

[6] Esparza J, Romer S, vogler W. An improvement of McMillan's unfolding algorithm [G] // Tools and Algorithms for the construction and Analysis of Systems. Berlin:Springer, 1996:87-106

[7] Song Jinfeng, Wen Lijie, Wang Jianmin. A similarity measure for process models based on the occurrence relations of tasks [J]. Jouranl of Computer Research and Development, 2017,54(4);832-843

[8] Sun Jinyong, Gu Tianlong, Wen Lijie, et al. Similarity algorithm for semantic workflows used in process-oriented case-based reasoning[J]. Computer Integrated Manufacturing Systems 2016,22(2):381-394

[9] Dijkman R, Dumas M, Garciabanuelos L. Graph matching algorithms for business process model similarity search [C]// Proc of the 7th Int Conf on Business Process Management. Berlin: Springer, 2009:48-63