Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 407809, 9 pages
http://dx.doi.org/10.1155/2014/407809

Research Article

Measuring Semantic and Structural Information for Data Oriented Workflow Retrieval with Cost Constraints

Yinglong Ma, Liying Zhang, and Moyi Shi

School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China

Correspondence should be addressed to Yinglong Ma; gtmikema@gmail.com

Received 20 February 2014; Revised 13 April 2014; Accepted 28 April 2014; Published 23 June 2014

Academic Editor: M. Chadli

The reuse of data oriented workflows (DOWs) can reduce the cost of workflow system development and control the risk of project failure and therefore is crucial for accelerating the automation of business processes. Reusing workflows can be achieved by measuring the similarity among candidate workflows and selecting the workflow satisfying requirements of users from them. However, due to DOWs being often developed based on an open, distributed, and heterogeneous environment, different users often can impose diverse cost constraints on data oriented workflows. This makes the reuse of DOWs challenging. There is no clear solution for retrieving DOWs with cost constraints. In this paper, we present a novel graph based model of DOWs with cost constraints, called constrained data oriented workflow (CDW), which can express cost constraints that users are often concerned about. An approach is proposed for retrieving CDWs, which seamlessly combines semantic and structural information of CDWs. A distance measure based on matrix theory is adopted to seamlessly combine semantic and structural similarities of CDWs for selecting and reusing them. Finally, the related experiments are made to show the effectiveness and efficiency of our approach.

1. Introduction

Data oriented workflows (DOWs) nowadays have been adopted in diverse areas such as high quality computation [1–4] and supply chain process [5, 6]. The reuse of DOWs can reduce the cost of workflow system development and control the risk of project failure and is crucial for accelerating the automation of business processes. For example, managers want to find and customize a logistic workflow such that customers can quickly perform the tasks of inquiry/quotation, order conformation, payment, and delivery within the least cost. Because each step in a logistic workflow system has to spend some cost to perform, managers are concerned about how to make the least cost of the whole logistic workflow by selecting suitable tasks from those developed logistic workflows and assembling a new one instead of developing a new workflow. The reuse of workflows with cost constraints is a challenging problem.

Reusing workflows can be achieved by measuring the similarity among candidate workflows and selecting the workflow satisfying requirements of users from them. Most existing approaches of retrieving DOWs are made based on their structures [7–9] because of most real-life workflows such as [10]. However, the structure based approaches often concentrate on data flows and invocation relations between services, but they neglect the semantic information that services and data rely on. As a result, the accuracy of retrieving DOWs is low, which is hard to satisfy the users' requirements indeed. Semantic based approaches make full use of semantic information of services and data in workflows and retrieve workflows by comparing them from the perspective of semantics [8], where ontologies are used to represent semantic information of services and data [11, 12]. These approaches have illustrated their potential in reusing DOWs.

However, due to DOWs being often developed based on an open, distributed, and heterogeneous environment, different users often can impose diverse cost constraints on data oriented workflows. This makes the reuse of DOWs challenging. There is no clear solution for retrieving DOWs with diverse cost constraints. On one hand, there is no formal representation model that not only represents both semantic and structure information within DOWs, but also encodes diverse cost constraints that are imposed on DOWs by

different users. The vendors of DOWs cannot represent and publish more semantic enriched information for their DOWs that will be reused by users. This will straightforwardly bring about the poor accuracy in retrieving DOWs. On the other hand, we lack a feasible method to seamlessly combine the semantic and structure based approaches for retrieving and reusing DOWs. This makes the retrieval results full of uncertainty. These problems will impede the more accurate DOW retrieval and therefore have a significance to the reuse of DOWs.

In this paper, we present a novel graph based model of DOWs with diverse cost constraints, called constrained data oriented workflow (CDW), which can express cost constraints that users are often concerned about. An approach is proposed for retrieving CDWs, which seamlessly combines semantic and structural information of workflows by computing the similarities between CDWs. A distance measure based on matrix theory is adopted to seamlessly combine semantic and structural similarities of CDWs for selecting and reusing them. Finally, the related experiments are made to show the effectiveness and efficiency of our approach.

This paper is organized as follows. Section 2 is the related work of the development for data oriented workflow retrieval. An overview of our work will be discussed in Section 3. Section 4 is some basic notations about constrained data oriented workflows. Semantic similarity computation of task nodes will be discussed in Section 5. In Section 6, we discuss CDW structure based similarity and comparison. Section 7 is the experiments and evaluation with other methods. Section 8 is the conclusion and future work.

2. Related Work

The main actuating force of workflow retrieval is naturally derived from the nonlinearities and uncertainties of planning and modeling of real world applications, which have been recognized and researched in some interesting work [13–18]. We also need to consider such nonlinearities and uncertainties in business process management. Traditional workflow representations mainly focus on activities/tasks and control oriented flow [19, 20]. Dataflow is not paid more attention to because traditional workflows are executed on a closed environment within a specific corporation rather than an open one. Some methods consider the semantic information but ignore the constraint which can reflect the quality of services.

Many workflow systems have emerged based on the recent research within semantic community [21–24]. Bergmann and Gil proposed a novel representation and retrieval method based on graph [20]. They extended a traditional workflow into a new representation whose nodes had three types: task node, dataflow node, and semantic node represented by a RDF file. However, specific applications or services cannot be provided by one single corporation due to the open, distributed, and heterogeneous execution environment. They need the cooperation of different kinds of service providers. Most of them did not consider constraint information which reflects requirements of specific users [25]. Some constraints such as time and cost, are important to reflect the quality of services provided by workflows. They cannot be defined and formulated in advance. Internal dependencies between semantic information residing in tasks/services such as hierarchical relationship and primary and secondary relationship must also be considered. Seamless integration of semantic and structure information is necessary to represent, execute, and reuse workflows because of the open, heterogeneous, and distributed environment on the Web.

Graph matching plays a key role to measure the similarity of two workflow models. It is a popular and mutual research topic. There are two classes of graph matching. One is the exact graph matching which includes graph isomorphism and subgraph isomorphism. The other is the inexact graph matching which includes attributed graph matching and attributed subgraph matching. Time complexity is a difficult problem in real-life applications that should be considered. From the perspective of implementing algorithms, graph matching can be also classified into three classes: graph isomorphism, feature extraction, and iterative methods. Graph isomorphism is a common approach introduced in [26]. Feature extraction [27] uses the idea that certain properties might be shared by similar graphs. This method has been widely used in the applications of character recognition, fingerprint images. In the iterative method, it is assumed that the similarity of two nodes depends on the similarity of their adjacent nodes. After multiple iterations, the similarity between two graphs will be obtained. Key words instead of data and semantic information play an important role in traditional workflow retrieval. The seamless combination of structure and semantic similarities is an urgent challenge for workflow retrieval. Bergmann and Gil presented a new similarity model which could seem as an enhancement of the well-known local/global approach [28]. However, it seems that they had not considered the constraints between semantic information of nodes and data types. In essence, these approaches are control flow based rather than dataflow based. They cannot seamlessly combine semantic and structure information together for representation and similarity comparison of constrained data oriented workflows. In our paper, a holistic approach based on matrix norm is proposed to measure the similarity of two workflow models. The time complexity can be also acceptable for workflow retrieval. Furthermore, the increasing speed of execution time is no faster than the growing speed of graph size. Seamless integration of semantic and structure similarities leads to a high retrieval accuracy and efficiency.

3. Overview of Process for Retrieving CDWs

In the section, we give an overview of the whole procedure of our approach using a formal graph based representation model called CDW for data oriented workflows retrieval with constraints, where a CDW model seamlessly integrates the structural and semantic information, as well as the cost constraints, and so forth. We attempt to effectively and efficiently retrieve CDWs by measuring and comparing the

similarities of both the semantic and structural information between CDWs. We argue that combination of structural and semantic information within CDWs will be greatly helpful to find more suitable workflows that satisfy both the functional requirements and diverse constraints from users. The whole procedure for retrieving CDWs can be mainly divided into five steps.

Step 1. A repository SCDW of candidate constrained data oriented workflows should be constructed beforehand, denoted by SCDW $= \{\text{CDW}_1, \text{CDW}_2, \text{CDW}_3, \ldots, \text{CDW}_n\}$. Each workflow in the repository can be represented by a formal representation model called CDW, which will be introduced in Section 4. Similarly, the request workflow that users require can be also represented by the CDW representation model, which is denoted by RCDW. Furthermore, we define a counter variable i, which will be used for traversing all the candidate CDWs in SCDW.

Step 2. We traverse each candidate CDW_i in SCDW and compute the semantic similarity between RCDW and CDW_i. The semantic information residing in task nodes is represented by the RDF ontology language. Each task node corresponds to a RDF file. During the semantic similarity computation for RCDW and CDW_i, the semantic similarity between task nodes within RCDW and CDW_i can be reduced to matching their similarity between their RDF files. Meanwhile, the data types of input data and output data between task nodes are also matched. For two task nodes, respectively, from RCDW and CDW_i, both the similarity for matching their RDF files and the similarity for matching their data types will be considered to compute the semantic similarity between them.

Step 3. We need to determine the identicalness between task nodes within RCDW and CDW_i. A similarity threshold value is set, which is 0.7 in this paper. That is, if the semantic similarity obtained in Step 2 between two task nodes is not less than the threshold value (i.e., 0.7), then we regard the two nodes as identical. The reason why we do that is the fact that task nodes in CDWs are very likely to be semantically heterogeneous in an open environment such as polysemy and toponymy.

Step 4. We further compute the similarity between RCDW and CDW_i by comparing their structures. Structural similarity between RCDW and CDW_i is made based on normalized matrices proposed in our previous work [7]. The normalized matrices for RCDW and CDW_i are, respectively, constructed. Then, a distance based metric is used to compute their structural similarity, that is, the distance between their normalized matrices.

Step 5. If $i \leq n$, then go to Step 2 for comparing the next CDW with RCDW. We select the candidate CDW_k as the one that is the most similar to RCDW, where k satisfies the following condition: $\text{sim}(\text{RCDW}, \text{CDW}_k) = \min\{\text{sim}(\text{RCDW}, \text{CDW}_i) \text{ for each } 1 \leq i \leq n\}$.

4. Basic Notations

4.1. Formal Representation for CDWs

Definition 1 (constrained data oriented workflow, CDW). A constrained data oriented workflow (CDW) is denoted by $\text{CDW} = (V, E, \alpha, \beta, \gamma, \lambda, \eta)$, which is a directed labeled graph.

(i) V is a set of nodes representing tasks/services in actual applications.

(ii) $E \subseteq V \times V$ is a set of ordered pairs of nodes, called directed edges.

(iii) $\alpha : V \rightarrow L_V$ is a mapping function which assigns each node a label, where L_V is a set of label names for tasks.

(iv) $\beta : V \rightarrow \text{RF}$ is a mapping function which maps each node to a RDF file, where RF is a set of RDF files that illustrate the semantic information of task nodes.

(v) $\gamma : V \rightarrow 2^{\text{DT}}$ is a mapping function which maps each node to a subset of data types. It represents the input information which can be processed by task nodes, where DT is the set of data types.

(vi) $\lambda : V \rightarrow 2^{\text{DT}}$ is a mapping function which maps each node to a subset of data types. It represents the output information provided by task nodes.

(vii) $\eta : E \rightarrow L_E$ is a mapping function which assigns each edge a constraint label in set L_E.

In the representation of CDW, nodes represent different tasks or functional activities in an actual application. A service can be provided by a specific task node. Some of them integrated together can accomplish a complex and comprehensive service such as scientific computation.

Semantic information is represented using the RDF language and is defined as a property of node, which is a key feature of semantic representation of task nodes. We establish the correspondence relations between nodes and RDF files using a mapping function instead of extending each task node with a RDF graph. This method can largely eliminate the degree of redundancy of a graph and clearly reflect the main workflow execution.

Data oriented workflows are often designed based on open, distributed, and heterogeneous environment to accomplish complex data processing. Therefore, it should have a high degree of modularity. True users care much about the input and output parameters that the task node could process and provide instead of data processing details. The representation for dataflow mainly emphasizes the data types that are used to communicate information between task nodes.

Now, we will give an example of CDWs to intuitively illustrate a data oriented workflow with constraint shown in Figure 1. In this figure, there are three task nodes which provide the services of function analysis, price analysis, and audit, respectively. Task nodes are represented by rectangles. Two control flows are represented by solid arrows as two edges with the constraints $[1, 2]$ and $[0, 3]$. Solid line ovals are

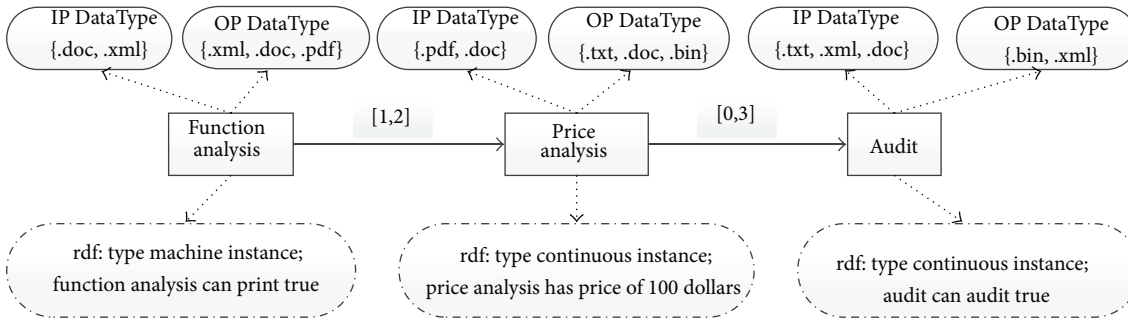

Figure 1: An example of CDW.

used to illustrate the input and output data types for each task node. Dotted lines are used to represent the relations between task nodes and data types properties. Dotted line ovals are used to represent corresponding RDF files of task nodes.

4.2. Definitions Related to Constraints

Definition 2 (constraint). Let CDW $= (V, E, \alpha, \beta, \gamma, \lambda, \eta)$ be a CDW. A constraint of edge $\langle v_i, v_j \rangle$ is represented by an interval of the form $[a, b]$, where $\eta(\langle v_i, v_j \rangle) = [a, b]$, and a and b are real numbers and $a \leq b$.

In actual applications, specific constraints can be determined in the domain of $[a, b]$. It reflects the variable and dynamic features of our representation for constrained data oriented workflows.

Definition 3 (intersection of constraints). Let CDW $= (V, E, \alpha, \beta, \gamma, \lambda, \eta)$ and CDW$' = (V', E', \alpha', \beta', \gamma', \lambda', \eta')$ be two CDWs. Let $\eta(\langle v_1, v_2 \rangle) = [a, b]$ and $\eta'(\langle v_3, v_4 \rangle) = [c, d]$ be two constraints for edges $\langle v_1, v_2 \rangle \in E$ and $\langle v_3, v_4 \rangle \in E'$, respectively. The intersection of constraints between the two edges can be computed by the following:

$$[a, b] \cap [c, d] = \begin{cases} [\max(a, c), \min(b, d)], & \text{if } a \leq d \wedge c \leq b, \\ \emptyset, & \text{otherwise.} \end{cases} \tag{1}$$

Definition 4 (union of constraints). Let CDW $= (V, E, \alpha, \beta, \gamma, \lambda, \eta)$ and CDW$' = (V', E', \alpha', \beta', \gamma', \lambda', \eta')$ be two CDWs. Let $\eta(\langle v_1, v_2 \rangle) = [a, b]$ and $\eta'(\langle v_3, v_4 \rangle) = [c, d]$ be two constraints for edges $\langle v_1, v_2 \rangle \in E$ and $\langle v_3, v_4 \rangle \in E'$, respectively. The union of constraints between the two edges can be computed by the following:

$$[a, b] \cup [c, d] = [\min(a, c), \max(b, d)]. \tag{2}$$

Definition 5 (duration of constraint). Let CDW $= (V, E, \alpha, \beta, \gamma, \lambda, \eta)$ be a constrained data oriented workflow and $\langle v_i, v_j \rangle \in E$. Let $\eta(\langle v_i, v_j \rangle) = [a, b]$ be the constraint for edge $\langle v_i, v_j \rangle$. The duration of this constraint is denoted by $\mathrm{Dur}([a, b]) = b - a$.

Definition 6 (summation of constraints). Let CDW $= (V, E, \alpha, \beta, \gamma, \lambda, \eta)$ and CDW$' = (V', E', \alpha', \beta', \gamma', \lambda', \eta')$ be two CDWs. Let $\eta(\langle v_1, v_2 \rangle) = [a, b]$ and $\eta'(\langle v_3, v_4 \rangle) = [c, d]$ be two constraints for edges $\langle v_1, v_2 \rangle \in E$ and $\langle v_3, v_4 \rangle \in E'$, respectively. The summation between their constraints can be denoted by $[a, b] + [c, d] = [a + c, b + d]$.

5. Semantic Similarity Computation

In our representation of CDW, all semantic information is represented as properties of nodes. Four mapping functions α, β, γ, and λ are to, respectively, associate a node v with its node name, its input set of data types, its output set of data types, and a RDF file RT_v. A RDF file represents the service function of a task node in the semantic level while a name is only a label in actual applications. The similarity between RDF files of two nodes can be used to measure the actual functional similarity in an open and heterogeneous environment. For example, we can measure the similarity between two nodes with different names that have the same function. It is also important for measuring the similarity between two task nodes to consider their data types that includes the semantic information of service data. As a consequence, the semantic similarity includes two closely related parts: the similarity for RDF files and the similarity for data types.

5.1. Similarity for RDF Files.

In this paper, we use RDF to describe the semantic information of a task node, which is regarded as a property in the task node by function β. The similarity computation of RDF files can be reduced to the comparison of their corresponding RDF graphs. The matching for RDF graphs has been studied for decades, and many good methods have been proposed. Each RDF graph is composed by a set of statements (triples). Each statement consists of three elements: subject, property, and object. The similarity of two RDF graphs can be divided into three steps.

(1) We use a three-dimension vector $\overrightarrow{\mathrm{ssv}}$ to represent the similarity between two statements. Let statement $= (s, p, o)$ and statement$' = (s', p', o')$ be any two statements. The elements s, p, o, s', p', and o' are the label strings of elements in statements. Vector $\overrightarrow{\mathrm{ssv}} = (\mathrm{ssv}_1, \mathrm{ssv}_2, \mathrm{ssv}_3)^T$, where $\mathrm{ssv}_1 = \mathrm{SimLD}(s, s')$, $\mathrm{ssv}_2 = \mathrm{SimLD}(p, p')$, and $\mathrm{ssv}_3 = \mathrm{SimLD}(o, o')$.

The notation SimLD(s_1, s_2) is the similarity of label strings s_1 and s_2 by using the Levenshtein distance; that is,

$$\mathrm{SimLD}\left(s_1, s_2\right) = 1 - \frac{\mathrm{LevD}\left(s_1, s_2\right)}{\max\left(\left|s_1\right|, \left|s_2\right|\right)}, \tag{3}$$

where $|s_1|$ and $|s_2|$, respectively, represent the length of strings s_1 and s_2. Obviously, the value of SimLD(s_1, s_2) is between 0 and 1. The notation LevD(s_1, s_2) refers to the Levenshtein distance between strings s_1 and s_2.

(2) We use a matrix ST as an auxiliary matrix for further computation of two RDF graphs: $\mathrm{RDF}_1 = \{st_1, st_2, \ldots, st_m\}$ and $\mathrm{RDF}_2 = \{st_1', st_2', \ldots, st_n'\}$. STU is the union of RDF_1 and RDF_2. All elements in STU are contained in the row and column of ST. For any $1 \le i, j \le m + n$, the matrix $ST_{(\mathrm{RDF}_1, \mathrm{RDF}_2)}(i, j)$ between RDF_1 and RDF_2 is represented as follows:

$$ST_{(\mathrm{RDF}_1, \mathrm{RDF}_2)}\left(i, j\right) = \sqrt{\mathrm{ssv}_{ij}^T \times \mathrm{ssv}_{ij}}, \tag{4}$$

where ssv_{ij} is the statement similarity vector between the $\mathrm{statement}_i$ in row and the $\mathrm{statement}_j$ in column in the matrix ST.

(3) The similarity between RDF_1 and RDF_2 can be computed by the notation SimRG(RDF_1, RDF_2), where

$$\mathrm{SimRG}\left(\mathrm{RDF}_1, \mathrm{RDF}_2\right) = \sqrt{\mathrm{tr}\left[ST_{(\mathrm{RDF}_1, \mathrm{RDF}_2)} \times ST_{(\mathrm{RDF}_1, \mathrm{RDF}_2)}^T\right]}. \tag{5}$$

Definition 7 (similarity for RDF files). Let CDW = $(V, E, \alpha, \beta, \gamma, \lambda, \eta,)$ and CDW′ = $(V', E', \alpha', \beta', \gamma', \lambda', \eta')$ be two CDWs. For any two nodes $v \in V$ and $v' \in V'$, the similarity between their RDF files can be computed according to the following equation:

$$\begin{aligned}\mathrm{SimRF}\left(v, v'\right) &= \mathrm{SimRG}\left(\beta\left(v\right), \beta'\left(v'\right)\right) \\ &= \sqrt{\mathrm{tr}\left(ST_{(\beta(v), \beta'(v'))} \times ST_{(\beta(v), \beta'(v'))}^T\right)},\end{aligned} \tag{6}$$

where $\beta(v)$ and $\beta(v')$ are the RDF graphs for nodes v and v', respectively, according to Definition 1.

5.2. Similarity for Input and Output Data Types. Data types, including input and output data types, indicate that information can be operated by specific task nodes. In this paper, they are represented by file names or information format such as .doc, .pdf, and .bin. Different task nodes may deal with different kinds of data types of services. Therefore, the compatibility of input and output data types of different nodes could be another factor to measure the services similarity provided by task nodes. We use ComIP(v, v') and ComOP(v, v') to represent the input data types compatibility and output compatibility, respectively, for nodes v and v'. SimDT(v, v') is the data types similarity between nodes v and v' considering both input and output data types similarities. The definitions are as follows.

Definition 8 (compatibility for input and output data types). Let CDW = $(V, E, \alpha, \beta, \gamma, \lambda, \eta)$ and CDW′ = $(V', E', \alpha', \beta', \gamma', \lambda', \eta')$ be two CDWs. For any two task nodes $v \in V$ and $v' \in V'$, the compatibility of input data types and output data types between v and v' can be computed by the following formula:

$$\begin{aligned}\mathrm{ComIP}\left(v, v'\right) &= \begin{cases}1, & \text{if } \gamma\left(v\right) \cap \gamma\left(v'\right) \neq \emptyset; \\ 0, & \text{otherwise,}\end{cases} \\ \mathrm{ComOP}\left(v, v'\right) &= \begin{cases}1, & \text{if } \lambda\left(v\right) \cap \lambda\left(v'\right) \neq \emptyset; \\ 0, & \text{otherwise.}\end{cases}\end{aligned} \tag{7}$$

Definition 9 (similarity for input and output data types). Let CDW = $(V, E, \alpha, \beta, \gamma, \lambda, \eta)$ and CDW′ = $(V', E', \alpha', \beta', \gamma', \lambda', \eta')$ be two CDWs. For any two task nodes $v \in V$ and $v' \in V'$, the similarity of data types between v and v' can be computed by the following equation:

$$\mathrm{SimDT}\left(v, v'\right) = \frac{\left[\mathrm{ComIP}\left(v, v'\right) + \mathrm{ComOP}\left(v, v'\right)\right]}{2}. \tag{8}$$

6. Similarity Computation between CDWs Based on Distance Metric

6.1. Identicalness Measure of Two Task Nodes. Semantic information similarity between two task nodes, which is measured by RDF files and data types similarities, can be used to indicate the identicalness of services provided by them. In this paper, we use an index IM(v, v') to measure the identicalness of two task nodes from different constrained data oriented workflows. We set the threshold of IM 0.7. This index will be used to determine whether it is suitable to assign two task nodes the same name from two workflows for comparison as follows:

$$\mathrm{IM}\left(v, v'\right) = \mathrm{SimRF}\left(v, v'\right) \times \mathrm{SimDT}\left(v, v'\right), \tag{9}$$

where SimRF(v, v') and SimDT(v, v') are, respectively, the RDF files similarity and data types similarity according to Definitions 7 and 9.

Identifying the identical task nodes between two CDWs is a preprocessing for comparing their similarity of the future. In this paper, the similarity computation of CDWs depends on their normalized matrices. We need to determine the task nodes related to the normalized matrices. Let CDW = $(V, E, \alpha, \beta, \gamma, \lambda, \eta)$ and CDW′ = $(V', E', \alpha', \beta', \gamma', \lambda', \eta')$ be two CDWs.

Our preprocessing can be formally represented as follows.

(1) For any two nodes $v \in V$and $v' \in V'$, we use the notation Ident(v, v') to represent that v and v' are identical. Ident(v, v') if and only if IM(v, v') > 0.7 and IM(v, v') = max{IM(v, v''), IM(v''', v') for all $v'' \in V'$ and $V''' \in V$}. If Ident(v, v'), then v and v' will be assigned the same name label.

(2) The set of task nodes related to normalized matrices is denoted by STN = $\{v_1, v_2, \ldots, v_k\}$, where $v_i \in V \cup V'$ for all $1 \le i \le k$, and, for any $v_i, v_j \in V \cup V'$ and $i \neq j$, v_i and v_j are not identical.

6.2. Normalized Matrices and Distance Metric

Definition 10 (normalized matrices for two CDWs). Let CDW $= (V, E, \alpha, \beta, \gamma, \lambda, \eta)$ and CDW$'$ $= (V', E', \alpha', \beta', \gamma', \lambda', \eta')$ be two CDWs. NM and NM$'$ are, respectively, the two normalized matrices for CDW and CDW$'$. n is the cardinality of the set STN after preprocessing. Let STN $= \{v_1, v_2, \ldots, v_k\}$, where $v_i \in V \cup V'$ for all $1 \le i \le k$, and, for any $v_i, v_j \in V \cup V'$ and $i \ne j$, v_i and v_j are not identical. The normalized matrices NM and NM$'$ are computed by the formulas as follows:

$$\mathrm{NM}(i,j) = \begin{cases} \dfrac{1}{n}, & \text{if } (v_i, v_j) \in E \setminus E', \\ \dfrac{\mathrm{Dur}(\eta(v_i, v_j)) \cap \mathrm{Dur}(\eta'(v_i, v_j))}{\mathrm{Dur}(\eta(v_i, v_j))} \times \dfrac{1}{n}, & \text{if } (v_i, v_j) \in E \cap E', \\ 0, & \text{otherwise,} \end{cases} \tag{10}$$

$$\mathrm{NM}'(i,j) = \begin{cases} \dfrac{1}{n}, & \text{if } (v_i, v_j) \in E' \setminus E, \\ \dfrac{\mathrm{Dur}(\eta(v_i, v_j)) \cap \mathrm{Dur}(\eta'(v_i, v_j))}{\mathrm{Dur}(\eta'(v_i, v_j))} \times \dfrac{1}{n}, & \text{if } (v_i, v_j) \in E \cap E', \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Definition 11 (distance metric [28]). Based on Definition 10, one uses a distance metric proposed in [28] for measuring the similarity between two constrained data oriented workflows CDW and CDW$'$ whose normalized matrices are, respectively, NM and NM$'$. Their distance metric between NM and NM$'$ is defined as follows:

$$\mathrm{DM}(\mathrm{NM}, \mathrm{NM}') = \sqrt{\mathrm{tr}\left[(\mathrm{NM} - \mathrm{NM}')^T \times (\mathrm{NM} - \mathrm{NM}')\right]}. \tag{12}$$

The notation tr[•] means the trace of matrix •, which is the sum of elements in the main diagonal of a matrix.

The distance metric DM satisfies all the three properties of distance measure as follows:

(1) DM(NM, NM$'$) ≥ 0 if and only if NM $=$ NM$'$;

(2) DM(NM, NM$'$) $=$ DM(NM$'$, NM);

(3) (DM(NM, NM$''$) $+$ DM(NM$''$, NM$'$)) $\ge$ DM(NM, NM$'$).

What is worth noting is that DM(NM, NM$'$), *de facto*, is the dissimilarity of NM and NM$'$. The larger the DM value between two CDWs is, the more dissimilar they are.

7. Experiments and Evaluation

7.1. Evaluation Indices, Workflow Repository, and Benchmark Methods. In this paper, we will evaluate the effectiveness and efficiency of our method by comparing it with the three methods. We use the two indices: degree of richness of retrieval (DRR) and time complexity. Evaluation of time complexity is mainly done by computing the execution time of workflow retrieval. So this index is shortened as ExT. Another index DRR is defined as follows:

$$\mathrm{DRR} = \frac{\mathrm{SW\_R}}{\mathrm{SW\_C}}, \tag{13}$$

where SW_R is the number of the workflows satisfying requirements of users in the retrieved results. SW_C is the number of the workflows satisfying requirements of users in the workflow repository.

We extended the functionality of workflow modeling tool [7]. In the extended tool, workflows with constraints can be graphically constructed. In addition, it can store ontology based semantic information. Each task node can be also associated with a set of data types describing the input and output parameters of the task node and a RDF file describing the semantic information of the task node. A workflow repository is implemented by a file directory in which many CDWs are stored as .xml files, and semantic information of nodes is stored as .rdf files. We manually constructed ten groups of CDW models with different sizes (i.e., the node number of a CDW) and depths (i.e., the node number of path from the start node to the last node in a CDW) as the testing data set for our experiments and evaluation.

Table 1: List of indices of all methods.

Approach	ExT	DRR
Our method	ExT_CDW	DRR_CDW
Method in [29]	ExT_MCS	DRR_MCS
Method in [7]	ExT_CW	DRR_CW
Method in [28]	ExT_SSW	DRR_SSW

The benchmark methods to compare are as follows. Bunke and Shearer [29] presented a distance metric for comparing the similarity of graphs based on maximal common subgraphs. This approach can be used for measuring the similarity among workflow structures. The second approach fully considers the constraints residing in workflows, such as cost. Ma et al. [7] proposed an approach to compare workflows with time constraints, where a time constraint is represented by an interval. The third approach concentrates on the comparison among semantic workflows, which was proposed by Bergmann and Gil [28]. This method considers comparing the semantic information in workflows.

In the following, we will compare our method with the other three methods mentioned above according to the two indices ExT and DRR. For simplification, Table 1 is the list of indices of all methods.

7.2. Experimental Evaluation

7.2.1. Comparing Degrees of Retrieval Richness for Evaluating Effectiveness. Figure 2 shows the degree of retrieval richness of four methods. From this figure, we can find some facts. Generally speaking, the larger the size of data set becomes, the more suitably the candidate workflows may be contained. Therefore, the value of DRR will rise. From this figure, we can

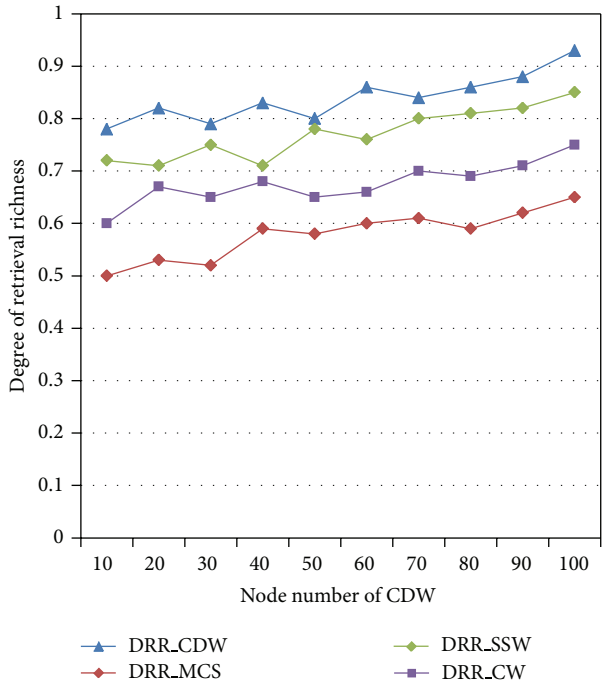

Figure 2: Comparison of degrees of retrieval richness.

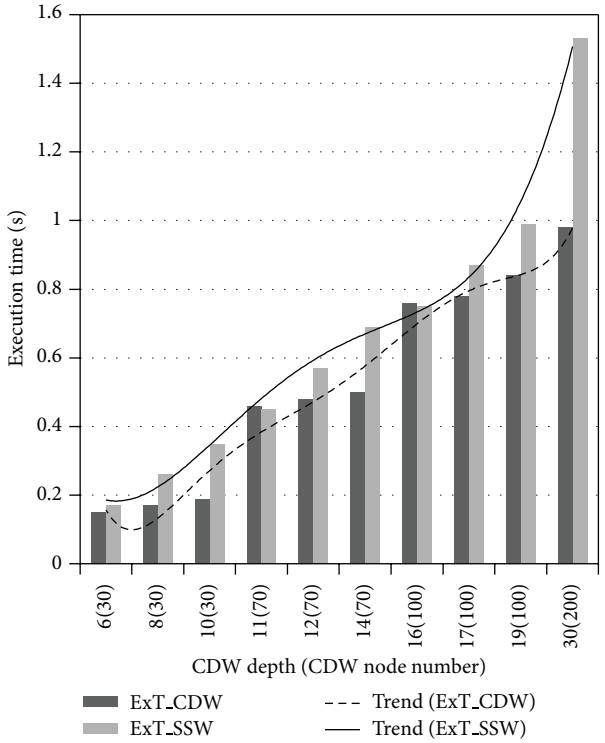

Figure 3: Execution time comparison based on depth.

find that the retrieval richness of our method is larger than others. It indicates that our method will find more suitable workflows with less irrelevant ones. DDR indicates that our approach has a stronger ability to discover information in a deeper level and will have a better retrieval effect obviously.

Higher value of DRR reflects that more suitable workflows have been discovered. One possible reason is that our method considers semantic and structural similarities for constrained data oriented workflows retrieval, which could discover deep relations such as nodes with different names but the same quality of service. This is quite important because users may choose cheaper services that are sufficient for their requirements. From the experimental results, we can conclude that our method with a higher degree of richness has an advantage over others to find more suitable information for specific users.

7.2.2. Comparing Time Complexity Based on Graph Depth for Evaluating Efficiency. Figure 3 is the comparison result of execution time between our method and that proposed in [28]. From the experimental results, we can have the following observation. As the size of graph grows larger, the execution time of both methods increases. For example, execution time of comparison between workflows with 70 nodes is longer than that with 30 nodes. In general, the execution time of our method is lower than that proposed in [28]. Graph depth has a larger effect on the execution of approach in [28], but less on our method. From the experimental results, we can find with the growing depths of the same size of graphs that the execution time increases faster than ours.

There are many factors that will bring about these experimental results. First, both methods adopt graph to represent workflow models. When workflow models became complex and large, more structural and semantic information needed to be dealt with through the whole comparison process which demands more execution time obviously. Second, Bergmann and Gil in [28] used a case reasoning approach to measure the similarity of semantic information of different nodes. This method depends heavily on the depth of graph. Time complexity of this method will grow faster when the depth of graph increases. It straightly influences the whole similarity measure time. In our method, we seamlessly combine semantic and structural similarities into normalized matrices. The whole time complexity largely depends on the matrix computation. Usually time complexity of matrix computation can be affordable in many cases. For example, we have three workflow groups with 100 nodes in graphs but different graph depths: 16, 17, and 19, respectively. From the experimental results, ExT_SSW grows faster than ExT_CDW.

7.2.3. Comparing Total Time Complexity for Evaluating Performance. Figure 4 is the execution time comparison between our method and the other two. From this figure, we can find that all methods cost more time with the growing of graph sizes. The execution time of our method increases a little faster than the other two methods. However, the extra time cost will be tolerant and affordable if we further take a look at Figure 4 because extra time cost will contribute to a better degree of richness of retrieval.

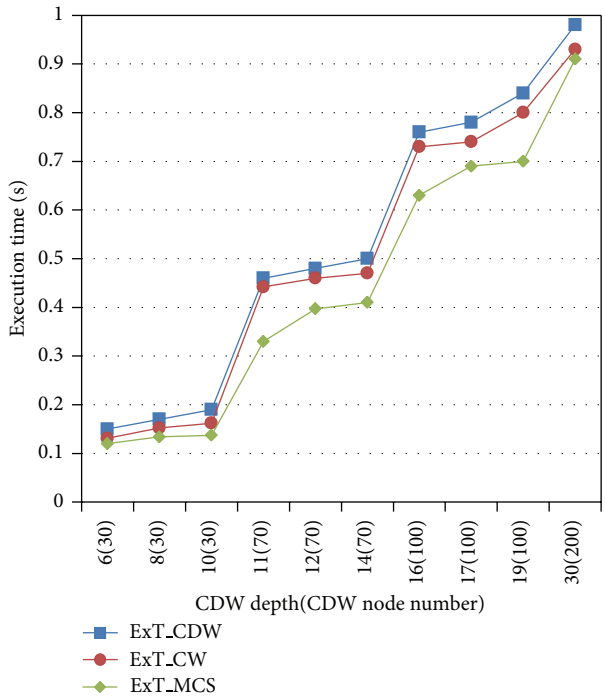

Figure 4: Execution time among the three methods.

Task nodes and relations between them are represented in a graph by Bunke and Shearer in [29]. The authors proposed a distance measure by maximal common subgraph. The formula is as follows: $\text{D\_MCS}(W_1, W_2) = 1 - (|V| / \max\{|V_{W1}|, |V_{W2}|\})$. $\text{D\_MCS}(W_1, W_2)$ computes the distance of W_1 and W_2, where $|V|$ is the number of maximal common subgraph between workflows W_1 and W_2. Time complexity mainly depends on the algorithm of subgraph isomorphism. The time complexity of subgraph isomorphism with unlabeled nodes in a graph has been proven to be an NP-complete problem. However, if each node has a unique label, the time complexity will decrease. Semantic information was not considered in the workflow models which would decrease the execution time.

In brief, we evaluate the effectiveness and efficiency of our approach by comparing it with the existing approaches of workflow retrieval. The experimental results show that (1) our method has more degree of semantic richness, which means that our approach can find out more accurate candidate workflows and has a higher retrieval precision; (2) the execution time that our method needs is no more than the other methods and is even lower than them. So our method outperforms the existing approaches of workflow retrieval and has higher performance.

8. Conclusion

In this paper, we proposed a graph based representation to describe constrained data oriented workflows, which seamlessly combines semantic and structure similarities to gain a better retrieval effect. It can mine out more and more deep information that is crucial to reflect the quality of services provided by workflows. It is convenient for workflows reuse for different domain scientists with specific requirements. The similarity comparison is based on matrix norm to measure the distance of two normalized matrices for corresponding workflow models. The experimental evaluation shows that our method outperforms the existing approaches of workflow retrieval. The method proposed in this paper can be widely used in workflows retrieval, reuse, matching, and so on.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (61001197, 61372182) and the Fundamental Research Funds for the Central Universities.

References

[1] C. Berkley, S. Bowers, M. B. Jones et al., "Incorporating semantics in scientific workflow authoring," in *Proceedings of the 17th International Conference on Scientific and Statistical Database Management (SSDBM '05)*, Santa Barbara, Calif, USA, June 2005.

[2] Y. Gil, E. Deelman, M. Ellisman et al., "Examining the challenges of scientific workflows," *Computer*, vol. 40, no. 12, pp. 24–32, 2007.

[3] M. Zohrevandi and R. A. Bazzi, "The bounded data reuse problem in scientific workflows," in *Proceedings of the IEEE 27th International Symposium on Parallel & Distributed Processing (IPDPS '13)*, pp. 1051–1062, 2013.

[4] Y. Gil, V. Ratnakar, and C. Fritz, "Assisting scientists with complex data analysis tasks through semantic workflows," in *Proceedings of the AAAI Fall Symposium on Proactive Assistant Agents*, pp. 14–19, November 2010.

[5] K. K. Castillo-Villar, N. R. Smith, and J. F. Herbert-Acero, "Design and optimization of capacitated supply chain networks including quality measures," *Mathematical Problems in Engineering*, vol. 2014, Article ID 218913, 17 pages, 2014.

[6] L. Longyi and Z. Yansheng, "Study of supply chain workflow based on grid," in *Proceedings of the International Conference on Management and Service Science (MASS '09)*, September 2009.

[7] Y. Ma, X. Zhang, and K. Lu, "A graph distance based metric for data oriented workflow retrieval with variable time constraints," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1377–1388, 2014.

[8] R. Bergmann and Y. Gil, "Retrieval of semantic workflows with knowledge intensive similarity measures," in *Case-Based Reasoning Research and Development*, vol. 6880 of *Lecture Notes in Computer Science*, pp. 17–31, 2011.

[9] D. Chiu, T. Hall, F. Kabir, and G. Agrawal, "An approach towards automatic workflow composition through information retrieval," in *Proceedings of the 15th Symposium on International Database Engineering & Applications (IDEAS '11)*, pp. 170–178, 2011.

[10] P.-F. Tsai, "A label correcting algorithm for partial disassembly sequences in the production planning for end-of-life products," *Mathematical Problems in Engineering*, vol. 2012, Article ID 569429, 13 pages, 2012.

[11] B. Cantalupo, L. Giammarino, N. Matskanis et al., "Semantic workflow representation and samples," 2005, http://eprints.soton.ac.uk/268554/1/2005_-_18554.pdf.

[12] N. Russell, A. H. M. Ter Hofstede, D. Edmond, and W. M. P. van der Aalst, "Workflow data patterns: identification, representation and tool support," in *Conceptual Modeling—ER 2005: 24th International Conference on Conceptual Modeling, Klagenfurt, Austria, October 24-28, 2005. Proceedings*, vol. 3716 of *Lecture Notes in Computer Science*, pp. 353–368, 2005.

[13] M. Chadli, H. R. Karimi, and P. Shi, "On stability and stabilization of singular uncertain Takagi-Sugeno fuzzy systems," *Journal of the Franklin Institute*, vol. 351, no. 3, pp. 1453–1463, 2014.

[14] M. Chadli and T. M. Guerra, "LMI solution for robust static output feedback control of discrete takagi-sugeno fuzzy models," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 6, pp. 1160–1165, 2012.

[15] S. Aouaouda1, M. Chadli, P. Shi, and H. R. Karimi, "Discrete-time H- / H∞sensor fault detection observer design for nonlinear systems with parameter uncertainty," *International Journal of Robust and Nonlinear Control*, 2014.

[16] M. Chadli, A. Abdo, and S. X. Ding, "H_-/H_∞ fault detection filter design for discrete-time Takagi-Sugeno fuzzy system," *Automatica A*, vol. 49, no. 7, pp. 1996–2005, 2013.

[17] M. Chadli and H. R. Karimi, "Robust observer design for unknown inputs Takagi-Sugeno models," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 1, pp. 158–164, 2013.

[18] M. Chadli, S. Aouaouda, H. R. Karimi, and P. Shi, "Robust fault tolerant tracking controller design for a VTOL aircraft," *Journal of the Franklin Institute*, vol. 350, no. 9, pp. 2627–2645, 2013.

[19] R. Ikeda, S. Salihoglu, and J. Widom, "Provenance-based refresh in data-oriented workflows," in *Proceedings of the 20th ACM Conference on Information and Knowledge Management (CIKM '11)*, pp. 1659–1668, October 2011.

[20] M. Hutchins, H. Foster, T. Goradia, and T. Ostrand, "Experiments on the effectiveness of dataflow- and controlflow-based test adequacy criteria," in *Proceedings of the 16th International Conference on Software Engineering*, pp. 191–200, May 1994.

[21] E. Deelman, G. Singh, M.-H. Su et al., "Pegasus: a framework for mapping complex scientific workflows onto distributed systems," *Scientific Programming*, vol. 13, no. 3, pp. 219–237, 2005.

[22] I. Foster, J. Voeckler, M. Wilde, and Y. Zhao, "Chimera: a virtual data system for representing, querying and automating data derivation," in *Proceedings of the 14th International Conference on Scientific and Statistical Database Management (SSDBM '02)*, pp. 37–46, 2002.

[23] Y. L. Simmhan, B. D. Plale, Gannon, and S. Marru, "Performance evaluation of the karma provenance framework for scientific workflows," in *Provenance and Annotation of Data: International Provenance and Annotation Workshop, IPAW 2006, Chicago, IL, USA, May 3-5, 2006, Revised Selected Papers*, L. Moreau and I. T. Foster, Eds., vol. 4145, pp. 222–236, Springer, 2006.

[24] "VDS—The GriPhyN Virtual Data System," http://www.ci.uchicago.edu/wiki/bin/view/VDS/VDSWeb/WebMain.

[25] Y. Gil, E. Deelman, M. Ellisman et al., "Examining the challenges of scientific workflows," *Computer*, vol. 40, no. 12, pp. 24–32, 2007.

[26] S. Fortin, "The graph isomorphism problem," Tech. Rep. 96-20, University of Alberta, Edomonton, Alberta, Canada, 1996.

[27] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh, *Feature Extraction: Foundations and Applications*, Springer, 2006.

[28] R. Bergmann, G. Müller, and D. Wittkowsky, "Workflow clustering using semantic similarity measures," in *KI 2013: Advances in Artificial Intelligence*, vol. 8077 of *Lecture Notes in Computer Science*, pp. 13–24, 2013.

[29] H. Bunke and K. Shearer, "A graph distance metric based on the maximal common subgraph," *Pattern Recognition Letters*, vol. 19, no. 3-4, pp. 255–259, 1998.