Recommend-As-You-Go: A Novel Approach Supporting Services-Oriented Scientific Workflow Reuse

Jia Zhang*, Wei Tan**, John Alexander*, Ian Foster***, Ravi Madduri***
*Department of Computer Science, Northern Illinois University, USA
**IBM T.J. Watson Research Center, USA
***University of Chicago and Argonne National Laboratory, USA
*jiazhang@cs.niu.edu, **wtan@us.ibm.com, ***foster@mcs.anl.gov, ***madduri@mcs.anl.gov

***Abstract*—**Services computing technology enables scientists to expose data and computational resources wrapped as publicly accessible Web services. However, our study indicates that scientific services are currently poorly reused in an *ad hoc* style. This project aims to help domain scientists find interested services and reuse successful processes to attain their research purposes in the form of workflows. In contrast to existing interface-based services discovery approaches, this paper proposes a novel approach of proactively recommending services in a workflow composition process, based on service usage history. The underpinning is a People-Service-Workflow (PSW) network that models existing scientific artifacts, services and workflows, and their past usage relationships into a social network. Various social network analysis techniques are applied to discover hidden knowledge accrued. A prototyping search engine has been developed as a proof of concept, and is seamlessly integrated as a plug-in into the Taverna workbench, a widely used scientific workflow management tool.

I. Introduction

To accelerate data-intensive scientific exploration, many disciplines including biology and astronomy have adopted workflows [1] as data-pipeline orchestrators to design scientific applications. As illustrated in Fig. 1, a scientific workflow precisely describes a multistep procedure to streamline a composition of tasks (T_1~T_6) and the dataflow among them. Such a workflow may be collaborated among multiple scientists, e.g., scientist S_3 handles tasks T_5 and T_6.

Recently emerged services computing technology enables and encourages scientists to expose data and computational resources wrapped as Web services [2], so that they become publicly available to other researchers through standard interfaces. For example, BioCatalogue [3] has registered over 1,700 life science services. A scientific workflow thus may leverage published Web services as tasks to speed up workflow composition. For example, task T_3 in Fig. 1 calls an external Web service from the Internet. Throughout this paper, we will use two terms interchangeably: *Web service* and *service*.

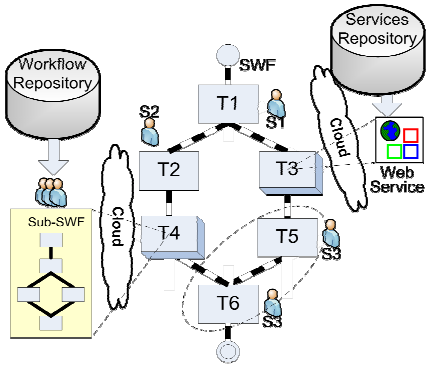

Fig. 1 Services-enabled scientific workflow development.

The scientific world is an open community. Researchers often publish workflows to share experimental routines with colleagues as best practices. These colleagues either use those workflows unchanged or repurpose them to compose new ones. To facilitate domain scientists in finding available workflows, several domain-specific online repositories have evolved in recent years. For example, myExperiment [4] stores over 1,000 life science workflows. As shown in Fig. 1, task T_4 invokes a sub-workflow registered at a repository. As software reuse may occur at any granularity, here we focus on service- and workflow-level reuse. The term *artifact* will refer to either *workflow* or *service*.

However, our recent network analysis study [5] over the workflows stored in myExperiment revealed that, the use of life science services is low (about 7%) and only several utility services are frequently used. In myExperiment, only 280 of the workflows leverage one service or more; only 179 operations from 118 services are ever invoked.

The goal of this research is to study how to facilitate scientific artifact reuse. In contrast to the existing interface-based services discovery approaches, we explore and leverage hidden knowledge implied from historical service usage data. Our hypothesis is that: related researchers' past experiences are carried by the structure of the past use of artifacts. Such information may convince and guide domain scientists in using existing artifacts properly; therefore, we study the published scientific experiments and mine their implied knowledge.

Our key approach is to model available scientific artifacts using social networks and leverage social network analysis to study their relationships and usage patterns. Social network analysis [6] refers to the techniques of mathematical and visual analysis of the nodes; while relationships are measured as various social relationship ties. To our best knowledge, this research is the first attempt to exploit social network analysis techniques to study relationships between artifacts, in addition to people-related relationship analysis, to mine hidden knowledge to facilitate

978-0-7695-4462-5/11 $26.00 © 2011 IEEE
DOI 10.1109/SCC.2011.120

services-oriented scientific artifact reuse.

We proposed a network-based **P**eople, **S**ervices and **W**orkflows (PSW) framework in a services-oriented e-science community, as a foundation to answer services discovery-related queries. We also implemented a prototyping framework named CASE – for information Collection, Annotation, Search and rEcommendation, as a plug-in to Taverna [7], a widely used scientific workflow management environment.

The remainder of the paper is organized as follows. In Section 2, we use a motivating example to explain the technical challenges. In Section 3, we present our PSW network and associated techniques. In Section 4, we present the CASE framework. In Section 5 and 6, we present system implementation and preliminary experimental results, respectively. In Section 7, we discuss related work. In Section 8, we draw conclusions.

II. Motivating Example and Challenges

In recent years, we have assisted scientists in various domains, including astronomy and life science, in building scientific workflows. From the projects, we have observed a significant reason that may explain the scarcity of scientific artifact reuse, as explained using the following example.

The workflow aims to automate a process for cancer researchers to diagnose tumor type by leveraging the microarray analysis [8] technique. The upper left part of Fig. 2 shows its high-level workflow comprising three sequential tasks: task 1 extracts hybridization data from tumor samples; task 2 pre-processes the obtained hybridization data; task 3 builds a classification model.

In contrast to the original in-house implementation by Shipp et al. [8], the right-hand side of Fig. 2 is our realization featuring artifact reuse. Six tasks pointed by arrows underneath represent invocation of four external services[1,2,3,4] registered at BioCatalogue; the middle sub-workflow is repurposed from a more generic workflow[5] registered at myExperiment.

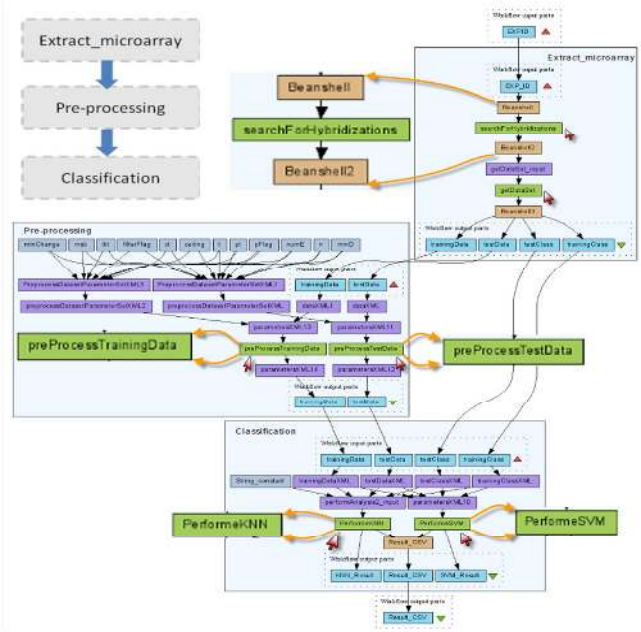

Fig. 2 A motivating example.

Our ability to develop this workflow depends on in-depth knowledge of available artifacts. For example, we selected from the same service provider the *preProcessTrainingData* service in the pre-processing sub-workflow and its subsequent *PerformKNN* service (a machine learning method *K-Nearest Neighbor*) in task 3 of Fig. 1. As another example, we duplicated the procedures of the general pre-processing workflow[5] to handle training data and test data, respectively (task 2 in Fig. 1).

Life scientists, on the other hand, may not possess such experience. As a result, they may be reluctant to reuse existing artifacts from their peer organizations, since scientific artifacts usually carry complex application logic and require careful tuning.

Thus, we strive to tackle two research questions in this project: 1) What implicit knowledge (usage patterns) may be automatically extracted from historical data to help scientists better understand existing artifacts? and 2) How to leverage such data to facilitate scientific artifact reuse?

III. PSW Networks

Our main strategy is to model the published scientific artifacts using networks and leverage social network analysis to study and extract hidden knowledge. Since our aim is services-based artifact reuse, we will focus on the relations between published workflows and services. Meanwhile, human reputation and relationships are usually important for artifact selection. For example, given several candidate services with similar functional and non-functional measurements, a scientist may select one provided by a past collaborating group. Therefore, we view our study space as a triple model:

$PSW = < \vec{P}, \vec{S}, \vec{W} >$, where:

$\vec{P}$ represents people involved in the lifecycle of scientific workflow development and usage. $\vec{S}$ and $\vec{W}$ represent published scientific services and workflows at public repositories, respectively.

Our task then turns into constructing the space, mining the relations among the three comprising elements in the space, and establishing a network-based **P**eople, **S**ervices and **W**orkflows (PSW) framework. Note that although we focus on the interactions between the three dimensions in this project, the data structure is extensible for other dimensions to be included.

[1] caArray service: http://array.nci.nih.gov/wsrf/services/cagrid/CaArraySvc?wsdl
[2] preprocessing service: http://node255.broadinstitute.org:6060/wsrf/services/cagrid/PreprocessDatasetSTATMLService?wsdl
[3] SVM service:http://node255.broadinstitute.org:6060/wsrf/services/cagrid/SVM?wsdl
[4] KNN service:http://node255.broadinstitute.org:6060/wsrf/services/cagrid/KNN?wsdl
[5] http://www.myexperiment.org/workflows/964.html

A. PSW data models

According to social network theory, structural relations between entities are often more important than their individual attributes [6]. Therefore, we study the implicit relations in the PSW space by analyzing the published data. In contrast to typical social networks [6] focusing on uni-modal networks, the PSW space represents a multi-modal, multi-relational, and multi-featured network.

As the high-level ontology diagram shown in Fig. 3, PSW space comprises three high-level node types: *service*, *workflow* and *people*. People may in turn be divided into various sub-categories according to their roles (such as composer, annotator, and user) on artifacts (workflow or service) organized in *groups*. As shown in Fig. 3, we also study finer-grained level of the node type *service*, as a service may expose multiple accessing points as *operations*.

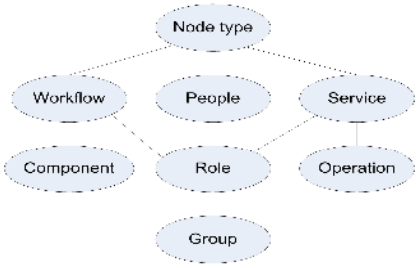

Fig. 3 PSW space.

Using the common social network terminology [6], the nodes (called actors) can be denoted as:

$$A = \{WF, WS, P_R\} = \{WF, WS, P_{r1}, P_{r2}, ..., P_m\}$$

Each node carries a rich set of metadata. For example, a workflow carries five categories of metadata: (1) basic information (workflowId, workflowTitle, workflowURI, currentVersion, description), (2) content information (workflowType, workflowTypeURI, createdOn, lastEditedOn, imageThumbnail, imageSVG, licenseType, licenseTypeURI, contentURI, contentType, contentValue), (3) annotation information stored in the format of arrays (tags, taggings, versions, reviews, comments, ratings, credits), (4) comprising components (inputs, processors, beanshells, outputs, links, coordinations), and (5) references (attributions, citations). We store each category of data in individual relations. Therefore, each node type can be represented as a relation:

$$AT = (A\_id, A\_M_1\_id, A\_M_2\_id, ..., A\_M_m\_id)$$

where $A\_M_i\_id$ (i=1-m) indicates the foreign key of the relation that stores the corresponding category of metadata for the node type.

In other words, various categories of metadata of a node type are maintained separately, while only keys of each relation are maintained in the main workflow relation. Such a design not only ensures flexibility and extensibility of the node type; more importantly, it allows us to apply multiple levels of various *join* operations while ensuring performance and data consistency.

B. Building PSW networks

On top of the PSW space, a variety of networks can be built based on various kinds of relationships that may link different types of nodes together. For example, two different workflow-service networks can be built: one is based on their inclusive relationships (an edge exists if a service is invoked in a workflow), and one is based on their co-ownership relationships (an edge exists if a workflow and a service come from the same research group).

We started from building workflow-service relationships based on inclusive events, by examining the source code of published scientific routines. A structural connection can be dynamically built between a workflow and a service through code analysis. Leveraging XPath to iterate through all workflows, we can find all service invocations in each workflow. To ensure performance, only workflows that invoke at least one service will be included, and only services that are used by at least one workflow will be included. We will thus obtain a matrix Q that describes the inclusive relationships between workflows and services:

$Q = [q_{ij}], 0 \le i \le m, 0 \le j \le m$, where:

$q_{ij} = 1$ if workflow i contains service j.

This matrix can be obtained using relational algebra operations over our PSW space. Given relations WP and WS, representing workflows and services, respectively, the matrix Q can be achieved by applying a series of relational algebra operations as follows:

$$\pi_{WP.wp\text{-}id,WS.ws\text{-}id}((\pi_{WP.wp\text{-}id,WP\text{-}Content.content}(WP \bowtie_{WP.wp\text{-}content\text{-}id=WP\text{-}Content.wp\text{-}content\text{-}id} WP\text{-}Content)) \bowtie_{WS.ws\text{-}id \in WP\text{-}Content.content} (\pi_{ws\_id} WS))$$

Such a relation is equivalent to the matrix Q: the set of all workflows and services represent the nodes; each row between a workflow and a service represents an edge in the graph. Relation Q can be viewed as a projection of the three-dimension space PSW on the $< \vec{S}, \vec{W} >$ plane. Such projections allow users to view interested relationships only.

We can derive two more relations, W and S, from Q as follows:

$W = Q \bullet Q^T = [w_{ij}], 0 \le i, j \le m$, where:

w_{ij} = number of services shared by workflows i and j; w_{ii} = number of services in workflow i;

$S = Q^T \bullet Q = [s_{ij}], 0 \le i, j \le n$, where:

s_{ij} = number of workflows where both services i and j are invoked; s_{ii} = number of workflows where service i is invoked.

The three matrices illustrate workflow-service, workflow-workflow, and service-service relations derived from the service-level workflow usage history. Note that such relations each carry a base entity as a context. For example, the relation W is a workflow-workflow relation based on service usages; the relation S is a service-service relation based on their usages in workflows.

Building indirect relations can be formalized as follows.

We have three types of entities E_1, E_2 and E_3 (an entity could represent people, service or workflow, $E_{1,2,3} \in PSW$), and two relations $R_1 : E_1 \times E_2 \to \{0,1\}$ and $R_2 : E_2 \times E_3 \to \{0,1\}$. An *indirect*, *2-hop* relation can be derived between E_1 and E_3 through a matrix multiplication $R_3 : E_1 \times E_3 \to \{0,1\}$ where: $R_3 = R_1 \bullet R_2$. Assume E_1, E_2 and E_3 refer to workflow, service and workflow, respectively. If R_1 defines a relation "workflows invoke services" (relation Q) and R_2 defines "services are invoked by workflows" (relation Q^T), then $R_3 = R_1 \bullet R_2$ defines an indirect relation between workflow and workflow through services (relation *W*).

In summary, our data model allows us to explore various implicit relationships. For example, we also built workflow-workflow relation based on people who develop them, and people who annotate them as users. After using relational algebra operations to obtain basic matrixes, we apply matrix operations to obtain derived matrixes. Table I summarizes the relationships we have built and their usability of providing different views of the PSW space. Throughout the paper, we will use the *PSW networks* as a generic term to refer to PSW networks together with all of its projections and derivations.

Table I. Summary of PSW networks built.

Networks	Descriptions
W-S | S	workflow-service based on service usages
W-W | S	workflow-workflow based on services
S-S | W	service-service based on workflows
W-P | CP	workflow-people based on composers
W-W | CP	workflow-workflow based on composers
S-P | CP	service-people based on composers
S-S | CP	service-service based on composers
W-P | AP	workflow-people based on annotators
W-W | AP	workflow-workflow based on annotators
S-P | AP	service-people based on annotators
S-S | AP	service-service based on annotators

In addition to undirected networks, we also built directed graphs over service operations. A directed link represents an invocation order between the operations on its two ends in some workflow. Such a graph depicts both intra- and inter-workflow invocation sequences.

C. Calculating network metrics and significant patterns

After establishing the PSW networks, we calculated various metrics over them to comprehend the interaction patterns between people, services and workflows.

Our approach builds on the rich tradition of calculation of *centrality and prestige* in social network analysis. For example, our metrics include degree centrality, betweenness centrality, PageRank value [9], and clique. Regarding degree centrality (popularity), in the PSW networks (e.g., W-S|S), we identify the highly used services and workflows that invoke more services based on the popularity of corresponding nodes. Regarding betweenness centrality, in the PSW networks, we examine how information flows through different services and workflows, aiming to identify the hinge services or workflows in the myExperiment. Regarding PageRank value, we study the degree of connection to nodes with high PageRanks.

Through *clique*, we delimit a maximal complete subgraph of three or more nodes, all of which are directly connected to one another. Such metrics represent collaboration relationships at both workflow and service levels. Such collaboration relationships at service level imply association rules among services.

Through these calculations, we aim to answer six categories of direct queries as summarized in Table II: workflow-service, service-service, people-service, people-people, people-workflow, and workflow-workflow. Indirect relations calculated from PSW networks can answer a category of questions regarding *existence of a path or entity*, such as, is there any path between two entities, what is a given entity's counterpart in a relation, and so on.

Table II. Queries that PSW networks can answer.

Category	Example queries
workflow-service	How are different services used together in workflows? What types of workflows in which a service is usually used?
service-service	Are there many services collaborate with each other in workflows, and how? What are the key services in these collaborations?
people-service	How different groups of people use services, do they have any preference?
people-people	Do people share services/workflows, and how?
people-workflow	How different groups of people produce workflows?
workflow-workflow	Do workflows collaborate?

We also conducted cross-validation of network metrics to verify if they are incidental or coherent with each other. The simplest approach we considered is to see if important entities in one type of relation are also important in another. For example, we constantly check whether services frequently reused in myExperiment workflows are also attracting more attention in BioCatalogue.

Besides studying the *global characteristics* of the PSW networks, we also investigated methods to identify *nontrivial patterns* implied. Especially, statistically significant paths are of primary interest to us, which may carry diversified meanings depending on the context. Examples are the paths with more frequency (e.g., what is the most common succeeding service after people use service *foo*), across less organizational boundary (e.g., what is the workflow which can generate a given data, and uses services hosted by least different institutions), or with less length (e.g., what is the shortest path scientist *A* can reach scientist *B* in a collaboration network). Detailed techniques are reported in our another paper [10].

IV. CASE ENGINE

A. CASE Framework

We designed and developed an information Collection, Annotation, Search and rEcommendation (CASE)

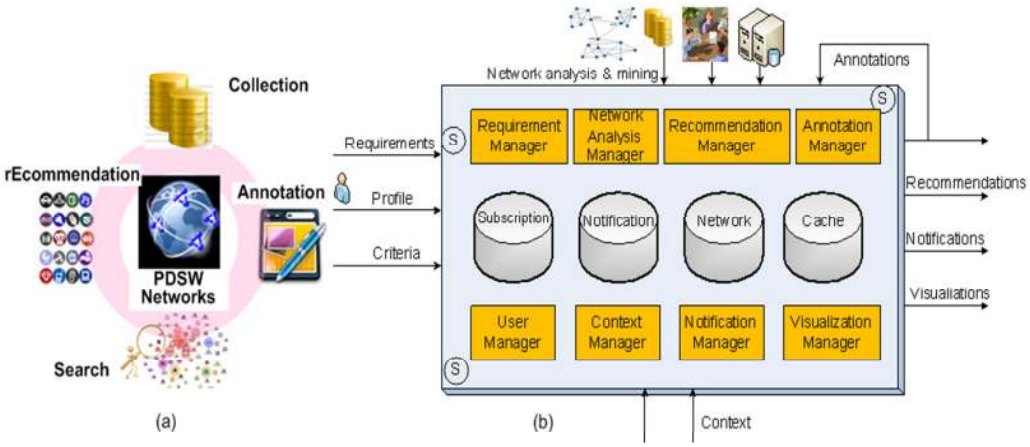

Fig. 4 CASE framework.

framework to systematically facilitate scientists in artifact reuse. As shown in Fig. 4(a), CASE is centered on the PSW networks and comprises four major components. (1) data collection: Artifacts will be incrementally collected from centralized repositories as primary data sources. Additional information may be collected from people directories from bioCatalogue, myExperiment, and websites of individual research institutions. (2) annotation: Automatic annotation elicitation, generation and analysis instruments will create incremental knowledge to support services-oriented scientific workflow discovery and composition. (3) search: We adopted Apache Lucene [11], an open-source search library to index the information collection and associated annotations. Besides the full-text search function, we used the PSW networks to support structure-aware cross-artifact search. (4) recommendation: The ultimate goal of CASE is to provide recommendation support in workflow composition. Recommendation can be either passive (requested explicitly by users) or proactive (automatically delivered when CASE perceives such a need).

The overview of the CASE framework is shown in Fig. 4(b) as a feedback system, which possesses internal controls and reacts to surrounding environments. The inputs of CASE are contextual data; the outputs may be represented in the formats of recommendations, notifications, visualizations, or annotations that dynamically integrate analysis data results into existing knowledge. Sensors denote system elements that monitor and detect changes from surrounding contexts, so that the CASE system may react accordingly to provide better services. In summary, a CASE system can be informally defined as a 6-tuple:

CASE = <Inputs, Outputs, Contexts, Transformation, Components, Sensors>, where:

Internally, CASE comprises eight major components: 1) a requirement manager that handles interpretation of user requirements; 2) a user manager that handles user profile and social network analysis; 3) a PSW network analysis manager that handles monitoring and analyzing various data sources; 4) a context manager that handles sensing ever-changing surrounding environments; 5) a recommendation manager that applies systematic analysis over all data and yields recommendations to users; 6) a notification manager that handles user subscriptions and notifies users with relevant updates; 7) an annotation manager that dynamically creates annotations and builds the links to corresponding resources; and 8) a visualization manager that handles generation of requested visual data based on user interests and preferences.

B. CASE visualization

After studying various social network visualization tools (including Pajek [12], Prefuse [13], and JUNG [14]), we decided to adopt JUNG as a foundation to build our visualization framework mainly due to its embedded rich graph mining algorithms. In addition, the JUNG framework offers a good object-oriented programming support, and a rich selection of vertex icons and graph layouts.

We also found some technical issues of applying JUNG into our project: 1) JUNG does not provide native SQL support; 2) nodes have to be added individually as opposed to using an array (i.e., parallelism); 3) its building on top of multiple third-party libraries may limit its reusability. We built another layer on top of JUNG so that users can directly leverage SQL descriptions.

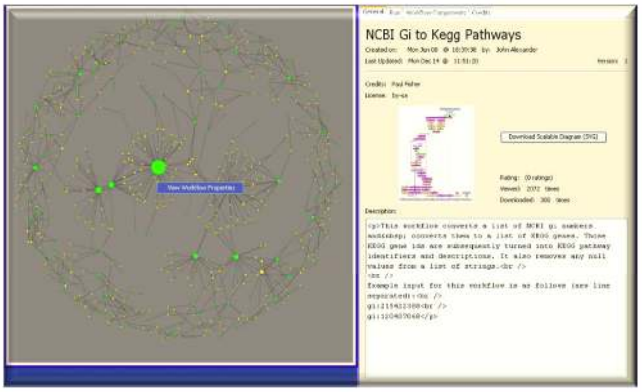

Fig. 5 CASE visualization.

Fig. 5 illustrates our generated workflow popularity graph. Note that upon right-clicking a node, a comment bar is shown as the name of the workflow. As shown on the right, the CASE engine dynamically goes to the workflow repository and retrieves the details of the selected workflow via the REST service invocation technique.

V. System Implementation

The ultimate goal of CASE is to provide recommendation support in workflow composition. We thus developed the CASE framework as an independent software as a service to help scientists design scientific workflows. As a proof of concept, we built CASE as a plug-in to the Taverna [7] workbench, a widely used scientific workflow management system.

The old service recommendation model used in Taverna is shown in Fig. 6 on the left. Every Taverna workbench application connects to the BioCatalogue website and retrieves a list of the links of all services (i.e., a total of 106 services). A user may expand the package and browse the full list and click a link to view the WSDL description of the corresponding service.

In contrast to its preliminary service listing facility, we developed a CASE-enabled fine-grained service recommendation mechanism. As shown in Fig. 6 on the right, a Taverna workbench is embedded with a local CASE engine, which dynamically communicates with the CASE engine at the server side to retrieve only related workflows and services to the user. The CASE engine server spawns several agents, each monitoring a data source (including myExperiment for workflows and BioCatalogue for services). Any specific event happens on these data sources will trigger a recalculation and subsequent changes on the workflow-service networks maintained at the CASE server. Such events include a new workflow publication at myExperiment, a new service publication at BioCatalogue, a new user annotation association to a workflow at myExperiment, a new ranking adjustment at BioCatalogue, and so on. Changes at the workflow-service networks will in turn be propagated to related Taverna users, through the publication-subscription relationships between local CASE engines and the CASE engine server.

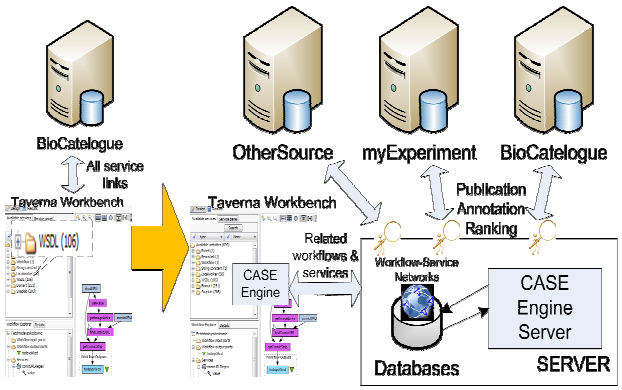

Fig. 6 Change of ways of recommendation.

VI. Experiments

A. Testbed Establishment and Analysis

Our preliminary work shows that studying workflow and service usage is feasible in the open science community. Our study shows that 11 out of 857 registered bioinformatics workflows in myExperiment are available to protected groups only, which means 98.72% of the applications are open to public (data obtained on 19 August 2010).

We examined all workflows published on the myExperiment repository, on October 6, 2010. Altogether, we have found 880 Taverna workflows. Among them, 800 are available to the public and 79 are set to be private to corresponding groups. 798 out of 800 were downloaded successfully through their REST interfaces. The rest two have to be manually downloaded because of non-UTF-8 characters existing in their file names. From our experience, every time we tried to build the testbed, several workflows may have to be manually downloaded to get over the "Connect exception" (i.e., time-out exception).

Studying the content of all workflows in our testbed, we found that 407 workflows invoke at least one external Web service (comparing to 280 such workflows on March 20, 2010 [5]). All these workflows are shown with their current versions. Among the 407 workflows, as shown in Table III, 74.7% of the workflows invoke only one Web service; only 2.7% of them invoke more than three services.

For each workflow, we examined its version number. If a version number is greater than one, it means that the workflow has historical versions. We thus tracked down to fetch all historical versions of such workflows. For each historical workflow version, we again examined the number of services it invoked.

As shown in Table III, 110 workflows have more than one version. The record is that one workflow (id: 1360) has 10 versions. Altogether, we found 178 historical workflows. Most of the workflows invoke more or the same number of services in newer versions. The exceptions are 16 workflows that have a fluctuating number of services in their various versions. Two workflows invoke less number of services in their more recent version (90, 746).

B. Service-workflow relationship analysis

Based on the workflow-service invocation network, we

Table III. Summary of myExperiment workflows.

New version		Old version	
Services invoked	Number	Services invoked	Number
0	0	0	9
1	304	1	100
2	65	2	29
3	27	3	8
4	9	4	10
5	1	5	9
10	1	6	3
Total	407	Total	178

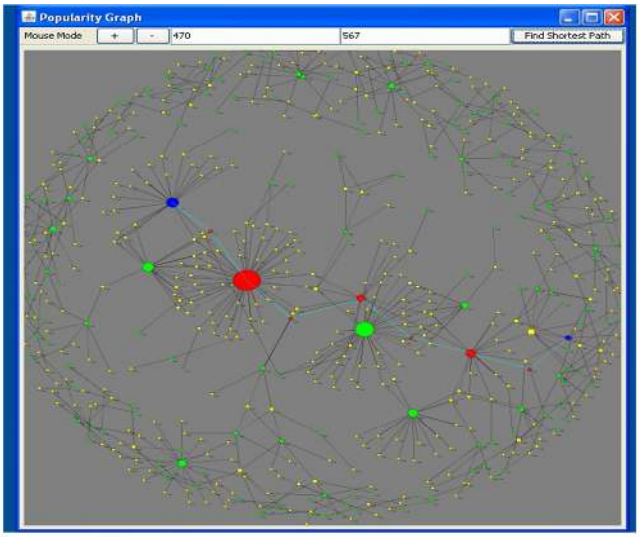

Fig. 7 Shortest path between a pair of workflows.

have studied the communication channels between pairs of services and workflows. The idea is to find out the relationship ties (connections) between the artifacts. If two workflows invoke the same service, there is a path between them with a length of 2 (i.e., the shared service as the intermediate node between them on the path). As shown in Fig. 7, two workflows at the two ends (in blue) have a path of length 8 between them, comprising intermediate services and workflows shown alternatively in the path.

We leveraged the Dijkstra's algorithm [15] to study the shortest path between each pair of workflows and services, respectively. We used the workflow-service invocation network instead of workflow-workflow network or service-service network due to two reasons. First, we aim to force the shortest paths to go through workflow-service invocation paths (i.e., workflows and services appear alternatively). Second, we would like to not only evaluate the shortest paths between each pair of artifacts, but also evaluate and visualize the details of actual paths (i.e., the intermediate nodes).

As shown in Table IV, over the total of 407 workflows in the test bed that invoke at least one service, we found 19,184 paths. It means that 91 workflows do not have paths to other workflows. Over the total of 179 services in the test bed that are invoked in at least workflow, we found 1,250 paths. It means that 91 services do not have paths to any other peer services. Since our major goal is to study how to leverage the past connections between workflows and services to facilitate artifact reuse, we focus on the connected workflows and services.

Table IV. Summary of service-workflow closeness.

Workflow-Workflow		**Service-Service**	
Shortest path length	Number of occurrences	Shortest path length	Number of occurrences
2	3,637	2	168
4	6,019	4	254
6	5,523	6	265
8	3,114	8	340
10	946	10	213
12	139	12	13
Total	19,378	Total	1,253
Average	5.19	Average	6.34
Mean	6	Mean	6

Table IV summarizes the statistical shortest path information between each pair of workflows and services, respectively. Our study exposes an interesting phenomenon. The average shortest path between a pair of workflows is 5.18; and the average shortest path between a pair of services is 6.34. Their mean values are both 6. This means that even though the scientists independently work on their own experimental research, their work can become connected through a small number of other colleagues' works in the field. The phenomenon again proves the famous "small world theory" that is widely acknowledged in social networks, which says that any two person in the social world can become connected through 6 people they know.

The shortest path between each pair of artifacts can be considered as their social tie. The closer they are, the tighter their tie is. Therefore, if an artifact is selected, we can rank other artifacts sorted by the length of the shortest path between each artifact and the selected artifact. Such knowledge can become complementary to artifact selection, in addition to other selection criteria such as functional and non-functional requirements. This experiment also reveals a way to support service composition. Connected artifacts can be grouped together as an encapsulated artifact to support composition of larger-scale artifacts.

VIII. Related Work

Artifact reuse and recommendation is well studied in software engineering [16], e.g., recommendation for debugging [17], inter-team collaboration [18], and auto completion of mashups [19]. It is gaining more attention in the area of scientific workflow. VisComplete [20] provides auto-complete suggestions for VisTrails system by mining frequent patterns in existing pipelines. Leake et al. [21] propose a case-based approach to suggest the possible next step(s) aiding re-use of portions of prior workflows. Xiao et al. propose a layered workflow structure that allows users to specify workflows at different levels of abstraction, and a graph matching method to find similar ones [22]. Compared to these approaches, our method can provide suggestions from cross-boundary relations (e.g., suggest a workflow which combining relations from multiple workflows), and more flexible by using full-text search in such an open environment where similar entities are not easily identified by their full names only.

Harrer et al. [23] transform structured data (e-mail, discussion boards, and bibliography sources) into social network data formats to visualize dynamic communities. Duong et al. [24] study automatic generation and visualization of personal ontology from personal and organizational websites, blogs, and publications. Viégas et

al. [25] propose history flow visualization, as an exploratory data analysis tool, to analyze cooperation and conflict of authorships in the wiki context. Vizster system [26] offers an online visualization of social networking, allowing users to discover communities, people and connections. Invenio [27] is a tool for visualizing multi-modal, multi-relational social networks. To visualize workflow-service networks supporting workflow composition, our work focuses on visualization framework and performance issue of dynamic visualization generation.

Our research differentiates with the current literature of interface-based services discovery in two significant ways. First, we focus on the more open scientific world where more workflow and service usage data are available. Second, we develop a technique that can be seamlessly integrated into existing scientific workflow tools to facilitate service and workflow discovery.

VIV. Conclusions and Future Work

In this paper we reported our efforts of building a discovery engine, for scientists to find appropriate artifacts (workflows and services) and obtain advice on their use, more rapidly than at present. We have answered the two research questions laid in the introduction section: 1) much knowledge can be extracted (table II); 2) by mining historical artifact usage patterns, we show how to answer various queries (table III). We also reported a prototyping system as a proof of concept.

In future research we plan to enhance our history-based recommendation techniques with semantics-based discovery ones. We also plan to accumulate practice data to create benchmarks for the presented approach in this paper.

X. Acknowledgement

We thank Daniel Kuc for his development support for this project. This work is partially supported by Google Summer of Code 2010; National Science Foundation, under grant NSF IIS-0959215; and the National Cancer Institute, the National Institutes of Health under contract N01-CO-12400.

XI. References

[1] C. Goble and D.D. Roure, "The Impact of Workflow Tools on Data-centric Research", in *T. Hey, S. Tansley, and K. Tolle, eds., The Fourth Paradigm: Data-Intensive Scientific Discovery, Microsoft Research, Oct.*, 2009, pp. 137-145.

[2] L.-J. Zhang, J. Zhang, and H. Cai, *Services Computing*, 2007: Springer.

[3] J. Bhagat, F. Tanoh, E. Nzuobontane, T. Laurent, J. Orlowski, M. Roos, K. Wolstencroft, S. Aleksejevs, R. Stevens, S. Pettifer, R. Lopez, and C.A. Goble, "BioCatalogue: A Universal Catalogue of Web Services for the Life Sciences", *Nucleic Acids Research*, May 19, 2010, 38: pp. W689-W694.

[4] C.A. Goble, J. Bhagat, S. Aleksejevs, D. Cruickshank, D. Michaelides, D. Newman, M. Borkum, S. Bechhofer, M. Roos, P. Li, and D.D. Roure, "myExperiment: A Repository and Social Network for the Sharing of Bioinformatics Workflows", *Nucleic Acids Research*, 2010, 38: pp. W677-W682.

[5] W. Tan, J. Zhang, and I. Foster, "Network Analysis of Scientific Workflows: A Gateway to Reuse", *IEEE Computer*, Sep., 2010: pp. 54-61.

[6] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*, 1994: Cambridge University Press, Cambridge.

[7] T. Oinn, M. Greenwood, M. Addis, M.N. Alpdemir, J. Ferris, K. Glover, C. Goble, A. Goderis, D. Hull, D. Marvin, P. Li, P. Lord, M.R. Pocock, M. Senger, R. Stevens, A. Wipat, and C. Wroe, "Taverna: Lessons in Creating a Workflow Environment for the Life Sciences", *Concurrency and Computation: Practice & Experience*, 2006, 18(10): pp. 1067–1100.

[8] M. Shipp, K. Ross, P. Tamayo, A. Weng, J. Kutok, R. Aguiar, M. Gaasenbeek, M. Angelo, M. Reich, G. Pinkus, T. Ray, M. Koval, K. Last, A. Norton, T. Lister, J. Mesirov, D. Neuberg, E. Lander, J. Aster, and T. Golub, "Diffuse Large B-Cell Lymphoma Outcome Prediction by Gene-Expression Profiling and Supervised Machine Learning", *Nature Medicine*, Jan., 2002, 8(1): pp. 68-74.

[9] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web", 1999, http://ilpubs.stanford.edu:8090/422/.

[10] W. Tan, J. Zhang, R. Madduri, I. Foster, and D.D. Roure4, "ServiceMap: Providing Map and GPS Assist to Service Composition in Bioinformatics", in Proceedings of *IEEE International Conference on Services Computing (SCC)*, 2011, Washington DC, USA.

[11] Y. Simmhan, B. Plale, and D. Gannon, "A Survey of Data Provenance in e-Science", *SIGMOD Record*, 2005, 34(3): pp. 31–36.

[12] W.d. Nooy, A. Mrvar, and V. Batagelj, *Exploratory Social Network Analysis with Pajek (Structural Analysis in the Social Sciences)* 2005, New York, NY, USA: Cambridge University Express.

[13] "Prefuge", Available from: http://prefuse.org/.

[14] JUNG, Available from: http://jung.sourceforge.net/.

[15] E.W. Dijkstra, "A Note on Two Problems in Connexion with Graphs", *Numerische Mathematik*, 1959, 1: pp. 269-271.

[16] R. Martin, W. Robert, and Z. Thomas, "Recommendation Systems for Software Engineering", *IEEE Software*, 2010, 27: pp. 80-86.

[17] B. Ashok, J. Joy, H. Liang, S. Rajamani, G. Srinivasa, and V. Vangala, "Debugadvisor: A recommender system for debugging", in Proceedings of *7th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on The Foundations of Software Engineering*. 2009, ACM. pp. 373-382.

[18] A. Begel, K.Y. Phang, and T. Zimmermann, "Codebook: Discovering and Exploiting Relationships in Software Repositories", in Proceedings of *32nd ACM/IEEE International Conference on Software Engineering (ICSE)*, 2010, Cape Town, South Africa, pp. 125-134.

[19] O. Greenshpan, T. Milo, and N. Polyzotis, "Autocompletion for mashups", *Proc. VLDB Endow.*, 2009, 2(1): pp. 538-549.

[20] D. Koop, C.E. Scheidegger, S.P. Callahan, J. Freire, and C.T. Silva, "VisComplete: Automating Suggestions for Visualization Pipelines", *IEEE Transactions on Visualization and Computer Graphics*, 2008, 14: pp. 1691-1698.

[21] E. Chinthaka, J. Ekanayake, D. Leake, and B. Plale, "CBR Based Workflow Composition Assistant", in Proceedings of *2009 World Congress on Services (SERVICES-I)*, IEEE Computer Society. pp. 352-355.

[22] X. Xiang and G. Madey, "Improving the Reuse of ScientificWorkflows and Their By-products", in Proceedings of *IEEE International Conference on Web Services (ICWS),* 2007. pp. 792-799.

[23] A. Harrer, S. Zeini, and S. Ziebarth, "Integrated Representation and Visualisation of the Dynamics in Computer-mediated Social Networks", in Proceedings of *2009 International Conference on Advances in Social Network Analysis and Mining*, 2009, Jul., pp. 261-266.

[24] T.H. Duong, M.N. Uddin, and G.S. Jo, "Collaborative Web for Personal Ontology Generation and Visualization for a Social Network", in Proceedings of *2009 International Conference on Knowledge and Systems Engineering*, 2009, pp. 237-242.

[25] F.B. Viégas, M. Wattenberg, and K. Dave, "Studying Cooperation and Conflict between Authors with History Flow Visualizations", in Proceedings of *ACM Computer-Human Interaction (CHI)*, 2004, Vienna, Austria, Apr. 24-29, pp. 575–582.

[26] J. Heer and D. Boyd, "Vizster: Visualizing Online Social Networks", in Proceedings of *IEEE Symposium on Information Visualization (InfoVis)*, 2005, Minneapolis, MI, UA, Oct. 23-25, pp.

[27] L. Singh, M. Beard, L. Getoor, and M.B. Blake, "Visual Mining of Multi-Modal Social Networks at Different Abstraction Levels", in Proceedings of *the 11th International Conference Information Visualization*, 2007, Washington DC, USA, Jul. 4-6, pp. 672-679.