Workflow discovery: the problem, a case study from e-Science and a graph-based solution

Antoon Goderis
Information Management Group
School of Computer Science
University of Manchester
M13 9PL Manchester, UK
Email: goderisa@cs.man.ac.uk

Peter Li
Bioanalytical Sciences Group
School of Chemistry
University of Manchester
M60 1QD Manchester, UK
Email: peter.li@manchester.ac.uk

Carole Goble
Information Management Group
School of Computer Science
University of Manchester
M13 9PL Manchester, UK
Email: carole@cs.man.ac.uk

Abstract—Much has been written on the promise of Web service discovery and (semi-) automated composition. In this discussion, the value to practitioners of discovering and reusing existing service compositions, captured in workflows, is mostly ignored. This paper presents one solution to workflow discovery. Through a survey with 21 scientists and developers from the m^y Grid workflow environment, workflow discovery requirements are elicited. Through a user experiment with 13 scientists, an attempt is made to build a gold standard for workflow ranking. Through the design and implementation of a workflow discovery tool, a mechanism for ranking workflow fragments is provided based on graph sub-isomorphism matching. The tool evaluation, drawing on a corpus of 89 public workflows from bioinformatics and the results of the user experiment, finds that the average human ranking can largely be reproduced.

I. WEB SERVICES AT WORK IN E-SCIENCE

As more scientific resources become available on the World Wide Web, scientists increasingly rely on Web and Grid services for performing in silico (i.e. computerised) experiments. Bioinformatics for example has seen a spectacular rise in the availability of distributed services – the m^y Grid/Taverna workbench (www.mygrid.org.uk) offers access to over 3000 of these. A popular example of a bioinformatics Web service is BLAST (Basic Local Alignment Search Tool), a service for finding regions of genome sequence similarity (see www.ebi.ac.uk/Tools/webservices).

Distributed service composition is difficult, be it manual or automatic. In this light, and to promote cross-disciplinary scientific collaborations, research councils worldwide are building a supporting infrastructure under the banner of e-Science. Exemplar initiatives include the Open Middleware Infrastructure Institute in the United Kingdom (www.omii.ac.uk), D-Grid in Germany (www.d-grid.de) and the Kepler project in the USA (www.kepler-project.org).

Workflow technology has been widely adopted in e-Science as the mechanism for orchestrating both distributed and local resources from within one environment. It potentially allows the e-Scientist to describe and enact her experimental processes in a structured, repeatable and verifiable way. Fig. 1 displays a bioinformatics workflow loaded up in the m^y Grid/Taverna workbench on the left hand side, while a list of available services and workflows is shown on the right.

II. WORKFLOWS VERSUS WEB SERVICES

Workflows and Web services are different yet similar, and this has an impact on their reuse potential. Both Web services and workflows in essence describe processes. Many different workflow languages exist, such as BPEL (the Business Process Execution Language), MOML (MOdeling Markup Language) [1] or Scufl (the Simple conceptual unified flow language), a high-level language designed for use by end users, not developers, and the one adopted in the m^y Grid/Taverna workbench) [2]. Web services on the other hand are typically standardised over SOAP and WSDL interfaces.

On-line workflows typically orchestrate Web services in conjunction with other kinds of services: local components, different types of distributed services (Taverna for instance accesses 8 different types [3]), even humans can be modelled as part of the process. Workflows are not executable without a workflow enactor. When combined with an enactor, they can be published as a Web service, and in turn possibly incorporated by another workflow.

Given their different nature, there is a difference in the reuse potential of workflows and Web services. Workflow reuse can be seen as the reuse of editable processes, whereas Web service reuse is reuse of encapsulated processes. It is the difference between being able to repurpose other people’s work by editing it versus incorporating other people’s work. Workflows allow to change the process’s data and control flow.

III. BUILDING WORKFLOWS BY EXAMPLE

In specific scientific domains, collections of workflows are now starting to pile up. To give an example, to date in m^y Grid/Taverna some three hundred workflows have been built, of a size ranging from five to fifty distributed services, and covering biological topics like gene annotation [4], protein structure prediction, microarray analysis [5] and systems biology. In biology, users are often not savvy with computers, and therefore have an strong incentive to reuse as much of the existing workflow engineering as possible. As more of these workflows are released, we are indeed witnessing that bioinformaticians start to share, discover, reuse and repurpose stand-alone compositions of services, or workflow fragments. Researchers repurpose an existing workflow or workflow
Workflow reuse is expected to take place in e-Science in the following ways: personal reuse, reuse by collaborators and reuse by third parties who the workflow author never met [6]. We provide new use cases here, specific to workflow discovery and based on a bioinformatics workflow repository. The repository is publicly available at www.myExperiment.org.

1) **Personal discovery**: Building large workflows can be a lengthy process, and in some cases can take over a year. The handling of workflow versioning and evolution are pressing issues in e-Science because the development of scientific workflows yields many versions. Manually keeping track of the relationships is a challenging task as the workflows get more complex, so versioning support is required. Versioning can be seen as a case of “personal reuse”. For example, bioinformatician Peter working on the *my* Grid/Taverna project has been building microarray workflows to research Graves’ disease for over a year now, and managed to produce 66 related workflows. One day he comes to work, only to discover he cannot quite remember how workflow **AffyidToGeneAnnotation2.xml**, shown in Fig. 3, differs from workflow **AffyidToGeneAnnotation4.xml**, shown in Fig. 4, or from the version that lies in between. Unfortunately any documentation is missing. Can one provide support for Peter to quickly discover how his workflows differ?

2) **Discovery by collaborators**: Scientists are typically part of a research group and various research projects, inside of which they exchange knowledge. For example, Paul, a fresh PhD student eager to build microarray workflows, has heard news from his group leader that Peter did a lot of work with microarray services a while back. Unfortunately, Peter has since moved universities and on to another project. All that remains of his work is a public directory of workflows at www.myexperiment.org. How can one best support Paul in making sense of the workflows?

3) **Discovery by third parties**: The scientific community is distributed across the globe, and people get insight and input from experiments done by colleagues they never met. Reuse by parties who the original authors have not met is likely, especially if a packaged workflow is published as an *in silico* experiment alongside an on-line publication. As a result of the e-Science infrastructure getting in place and being used, there now exists the prospect of a large repository of workflows and workflow data products across scientific disciplines, available for further experimentation. Especially between closely related disciplines there is a lot of potential for overlap and collaboration. For example, chemoinformatics workflows producing candidates for drug development can be plugged into bioinformatics workflows which retrieve the drug candidates’ known hazardous interactions within cells from databases worldwide.

### IV. THREE PRACTICAL USE CASES FOR WORKFLOW DISCOVERY

![Fig. 1. The AffyidToBlastxPDB.xml workflow loaded in the *my* Grid/Taverna workbench. The Available services pane provides access to both services and workflows.](image)

The workflow by example approach complements the popular view in the Web services literature that on-line processes will be composed (semi-) automatically from scratch. Workflows hold within them successful examples of working service compositions which can help future compositions. In fact, the amalgamated collection of these workflows *de facto* provides the closest we currently have to a worldwide “Web of Services” (as opposed to a Web of pages) because workflows document which sets of real world services fit well together.

**A. Participants**

During two *my* Grid/Taverna User Day events (5-6 May and 15 November 2005), 21 out of a total of 45 participants...
completed a questionnaire, of which 15 were bioinformaticians and six software developers.

B. Materials

A questionnaire was designed which asked users to indicate how important various criteria for doing workflow discovery were. The questions were based on the capabilities offered by the "myGrid/Taverna" environment, which the participants were familiar with. The questions also probed for users’ general attitude towards workflow reuse. The survey and the survey data are available from www.mygrid.org.uk/wiki/Papers/IcwsPaper2006. This survey complements an earlier one with interviews of core developers of six e-Science middleware projects, which is reported on in [6] and identifies seven bottlenecks for workflow reuse and repurposing.

C. Procedure

Participants were handed out a questionnaire during the User Days. Users were asked to rate the relevance of various search criteria, with values ranging from 1 (unimportant) to 5 (highly relevant).

D. Results

It took participants 15 minutes on average to complete the questions. The following answers were obtained.

1) Sharing attitude: All subjects indicated they wanted to share. Some were more nuanced and would share workflows but not its data inputs or outputs.

2) Discovery based on workflow signature: Four participants wrote they would search for workflows in the same way as they do for services. The question of workflow discovery then becomes one of discovery of a Web service based on its signature. We asked users how important various service search criteria were to them. All came out as relevant. The criteria, in order of decreasing relevance to participants, were: Task, Input, Output, On-line documentation, Service Provider, Underlying Resource used (e.g. a particular database), and Algorithm used (e.g. a particular clustering algorithm). For details we refer to the on-line survey data. In an optional Other: field, users could enter additional criteria. A few users entered Quality of Service parameters here, in particular performance and reliability measures.

3) Discovery based on workflow structure: Five participants indicated they would not only rely on using a workflow’s signature, during workflow discovery they expected to be using structural information, such as the services contained in a workflow, the specific subtasks addressed by the workflow or to start from existing template workflows. This suggests a type of discovery based more on the shape or structure of a workflow, using more behavioural type of information.

We then asked users to rate the following criteria which also rely on structural information. The criteria are presented here in decreasing order of relevance as assigned by users.

- Data flow Given a set of data points, have these been connected up in an existing base of workflows? Data flow queries came out as very important.
- Service flow Given a set of services, have these been connected up in an existing base of workflows? Service flow queries also came out as very important.
- Workflow similarity The use of similarity to identify relevant workflows came out as important.
- Use of specific control flow constructs Queries based on specific control flow constructs, such as the appearance of looping and conditionals in a workflow, were not considered unimportant. This was against our expectations and indicates that users should be enabled to query specifically for such constructs.

Again, in an optional Other: field, users could also enter additional criteria, but no one did.

VI. REQUIREMENTS ANALYSIS FOR A WORKFLOW DISCOVERY TOOL

Based on the survey, we have established that users consider discovery based on workflow structure to be a valuable part of the workflow discovery process. To support such discovery effectively with an automated tool, we have several questions left to answer:

- A gold standard How do people rank workflows? Can we replicate this behaviour with a tool, based on a gold standard created by people?
- Predictive power of criteria Which criteria are direct predictors of workflow (dis-)similarity?
- The link between task and structure What can structural similarity of workflows tell us about task similarity?

By means of a follow-on survey we aimed specifically to create a gold standard for a workflow discovery tool. In this section we report on the data gathered during this survey and its statistical interpretation. Given the weak results of our statistical analysis afterwards, we are only able to provide a partial answer to the above questions.

A. Participants

During a "myGrid/Taverna" User Day event (6 February 2005), 13 out of a total of 18 users completed an exercise to rate the similarity between an exemplar workflow (shown in Fig. 2) and other workflows. Nine users were bioinformaticians and four were software developers.

B. Materials

The user experiment included the use of an on-line survey, a corpus of similar on-line workflows and the rendition of those workflows. All are made available on-line.

1) An on-line survey: An on-line survey was created through the survey service at www.keysurvey.com. It contained three main sections. The first section gathered basic information on the subject and established whether they understood the biology behind the exemplar workflow.

The second part asked users to rate the workflows and their confidence in doing so. The final part asked which additional information would be helpful in making similarity assessments and whether they found the exercise difficult.
2) A corpus of public workflows: A corpus of 89 workflows was used as the basis for the exercise. The majority of workflows in the corpus (66) were created by one of the authors in support of research on Graves disease [5] and, as a result, are highly related and form a good basis for a workflow similarity experiment. The biological goal of this set of workflows is to discover genes involved in the disease based on microarray data and to prepare the genotyping of single nucleotide polymorphisms (SNPs) which are nucleotide variations that occur in those genes. We made the workflows available on-line and rendered them as icons and diagrams. The workflows can be accessed from within Taverna, as shown in Fig. 1. Five workflows were selected from the corpus as comparison material for the exemplar workflow of Fig. 2. The comparison workflows are shown in Figures 3 - 7. They differ on dimensions such as size, node orderings and differences in the scope of the biological task.

C. Procedure

The survey was presented as an exercise that was an integral part of the training at the User Day. The stated goal of the exercise was to allow a user to study and try to understand some more complex workflow diagrams, while allowing the mVGrid team to understand how similarity between workflows is perceived. Users were first shown an overview of all available workflows, to give them an impression of the complexity involved in the manual discovery task. They were then explained the concept of a gold standard. Five workflows were presented for comparison, pre-selected from the corpus. For each workflow, users were presented with five questions to judge how similar it was to the exemplar workflow. To indicate the similarity between a pair of workflows, users selected a bullet from nine options (see Fig. 8). Each bullet corresponds to a value: 1 corresponds to Identical, 5 to Similar, and 9 to Not similar at all. Users also provided a measure of confidence in their similarity assessment, ranging over: High--Medium--Low, with High having value 1 and Low being equal to 5. Finally, they had to rate how useful they found six factors for estimating similarity, with usefulness defined as: Very useful--Useful--Only a bit useful--Not useful at all, with Very useful
equal to 1 and Not useful equal to 4. The factors users were asked to rate were the following.

1) It makes biological sense to have this workflow as a part of the example workflow
2) It makes biological sense to have this workflow superimposing over the example workflow
3) Workflow shape: number of shared inputs and outputs
4)Workflow shape: service type correspondence
5)Workflow shape: shared service compositions
6)Workflow shape: shared paths between (intermediary) input and output

D. Results

The entire exercise took participants 30 minutes on average. By analysing the generated data, we can partially answer the questions put forward in the beginning of the section. We used the SPSS (www.spss.com) statistical package for analysis. The full results are available on-line.

1) Towards a gold standard: This section analyses whether the rankings users generated in the experiment are suitable as input as a gold standard to a workflow discovery tool. Users ranked the similarity of the exemplar workflow vis-a-vis the comparison workflows as indicated in Table I.

The table shows the mean of values entered by all respondents per workflow. The standard deviation is given inside the brackets (two standard deviations away from the mean account for roughly 95 percent of the people). The respondents reported medium confidence overall in their own judgment. 66.7 percent of respondents found estimating the similarity of biological functionality very difficult to difficult. The estimation of the similarity of shape similarity on the other hand was a difficult task for only 25 percent of respondents.

To establish whether there was consistency in the rankings produced between respondents, we need to know whether a correlation between these rankings exists. To confirm whether this is the case, we performed the following calculations. The user similarity values of Table I were transformed for all participants to reflect the order in which each individual had ranked the workflows (in order of decreasing similarity). We then built a correlation matrix (not shown due to its size; available on-line) based on Spearman’s correlation test and the transformed data. Spearman’s correlation test is a measure of association between rank orders. With respect to the data, for biological functionality similarity, we only used data from people with a biological background (nine respondents and five workflows, no missing values), whilst taking into account data from all 13 respondents for shape similarity.

For rankings based on biological functionality, only for six out of 36 possible participant pairs (e.g. between participant 2 and participant 5) the results were correlated (at a five percent significance level). This means that participants in general disagreed on how to order the workflows according to their biological similarity and therefore lacked consistency in their rankings. A similar result was found in the correlation matrix for rankings based on shape: based on the 13 users, only 16 out of 77 pairs showed correlation (at a five percent significance level). As it stands, the current data set cannot serve as a general gold standard.

<table>
<thead>
<tr>
<th>No.</th>
<th>Biological similarity</th>
<th>Shape similarity</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.5 (2.0)</td>
<td>5.0 (1.5)</td>
<td>3.0 (1.1)</td>
</tr>
<tr>
<td>2</td>
<td>5.8 (1.9)</td>
<td>6.8 (1.4)</td>
<td>2.9 (1.0)</td>
</tr>
<tr>
<td>3</td>
<td>7.0 (1.7)</td>
<td>6.8 (2.0)</td>
<td>3.3 (0.8)</td>
</tr>
<tr>
<td>4</td>
<td>6.2 (1.9)</td>
<td>5.0 (2.4)</td>
<td>2.9 (1.4)</td>
</tr>
<tr>
<td>5</td>
<td>8.0 (1.6)</td>
<td>8.4 (1.3)</td>
<td>3.2 (1.4)</td>
</tr>
</tbody>
</table>
available could be sufficient to reliably rank workflows, as similarity and the shape similarity (see Table III). This sug-
a strong correlation was observed between the biological
i.e. indicated that the total number of services (i.e. the difference
in size of workflows) played a role in the assessment of
paths)

<table>
<thead>
<tr>
<th>No.</th>
<th>Subtask Supertask</th>
<th>I/O Type</th>
<th>Services</th>
<th>I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.4 2.0</td>
<td>2.5 2.2</td>
<td>2.1 2.1</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>(1.0) (0.9)</td>
<td>(0.9) (0.8)</td>
<td>(0.5) (0.6)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3.0 2.6</td>
<td>2.9 2.1</td>
<td>2.2 2.1</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>(0.8) (0.9)</td>
<td>(1.1) (0.9)</td>
<td>(0.8) (0.9)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3.0 3.3</td>
<td>2.6 2.2</td>
<td>2.5 2.3</td>
<td>2.3</td>
</tr>
<tr>
<td></td>
<td>(0.8) (0.8)</td>
<td>(0.8) (0.8)</td>
<td>(0.5) (0.7)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2.3 3.0</td>
<td>2.4 2.1</td>
<td>2.3 2.3</td>
<td>2.3</td>
</tr>
<tr>
<td></td>
<td>(0.9) (0.8)</td>
<td>(0.9) (0.8)</td>
<td>(0.7) (0.5)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3.6 3.3</td>
<td>2.5 2.5</td>
<td>2.5 2.5</td>
<td>2.5</td>
</tr>
<tr>
<td></td>
<td>(0.7) (1.2)</td>
<td>(1.1) (1.1)</td>
<td>(1.1) (1.1)</td>
<td></td>
</tr>
</tbody>
</table>

2) Predictive power of criteria: One way to explain the
diversity in behaviour during ranking we described above is
to investigate whether different users use different criteria
for establishing workflow (dis-)similarity. Users ranked the
usefulness of the factors for establishing workflow similarity
as indicated in Table II. We are interested to find the effects
of the six factors on the similarity, whilst the factors may be
interacting (e.g. once people looked at how many services are
shared, they might not care any more whether any inputs and
outputs are shared). A common statistical way to establish such
findings is through a Between-Subject Analysis of variance
(ANOVA) (see davidmlane.com/hyperstat) for a good
introduction). Analysis of variance assumes that the groups
come from populations with equal variances. To test this
assumption, we used Levene’s homogeneity-of-variance test,
only to find that the assumption was violated in all cases. As
a result, little can be said on how particular factors impact the
similarity measures based on ANOVA.

One explanation for the inconsistencies is that different
people might be using different metrics, some of which not
included in the list of six factors. One participant for instance
indicated that the total number of services (i.e. the difference
in size of workflows) played a role in the assessment of
shape. One logical step in future experiments would be to
fix the (combination of) criteria people can use, and see what
similarity values across participants this generates.

3) The link between task and structure: From the data,
a strong correlation was observed between the biological
similarity and the shape similarity (see Table III). This sug-
gests that having only the workflow specifications themselves
available could be sufficient to reliably rank workflows, as
opposed to workflows with semantic annotation. This fact does
not remove the need for workflow documentation or annotation
though: a user is primarily interested in those bits of the
retrieved workflows that are different to hers, and without
documentation on what these do, rankings will not be useful.

VII. DESIGN OF A WORKFLOW DISCOVERY TOOL
In the earlier mentioned survey of middleware projects [6],
all projects offer a search mechanism to look for available
services; none however allow for the possibility to discovery
and compare workflow descriptions based on their behaviour
or structure. In Section VI, we confirmed that some users
would indeed find good use for such a mechanism. This paper
reports on the development of such a workflow discovery
component, meant to work specifically over the structure or
shape of a workflow. Such a component should be seen as
complementary to other approaches retrieving a workflow by
its signature. The component is currently limited to supporting
personal discovery (versioning) and service flow queries.

Section VI showed that users in general find it hard to
compare workflow diagrams. Part of this difficulty can be
accounted for by the volume and complexity of diagrams. The
visualisation methods which generate the diagrams are an-
other culprit: sometimes they generate different layouts when
only a small number of nodes change, making it harder for
humans to interpret similarity. Automated techniques should
be less susceptible to these issues. Graph matching techniques
in particular seem to offer a particularly good solution for
comparing workflows. Workflows can be seen as graphs,
and the problem of comparing workflows as a problem of
comparing graphs. In addition, graphs provide a theoretical un-
derpinning of the Resource Description Framework (RDF), a
W3C recommendation for describing semantic information on
the Web (www.w3.org/RDF). This means that any technique
we adopt for workflow diagrams based on graphs potentially
extends to RDF graphs describing workflows.

We have resorted to a technique for graph sub-isomorphism
matching (“subgraph matching”) optimised to work over a
repository of graphs, which was developed by Messmer and
Bunke [7]. Standard methods for sub-isomorphism detection
usually work on only two graphs at a time. However, when
comparing workflows, there is more than one graph in the
repository that must be matched with the input graph. Conse-
sequently, it is necessary to apply the subgraph isomorphism
algorithm to each pair of repository graph - input graph,
resulting in a computation time that is linearly dependent on
the size of the repository. Messmer and Bunke’s approach is
based on a compact representation of the repository graphs
that is computed off-line. The representation is created by
decomposing the repository graphs into a set of subgraphs,
where common subgraphs of different graphs are represented
only once. During on-line matching, they are matched exactly
once with the input graph, yielding a technique that is only
sub-linearly dependent on the number of the graphs in the
repository [7]. As explained below, this optimisation is cur-
rently unavailable in our tool, but remains of interest.
The tool was implemented as a plug-in to the Taverna workbench. We interfaced an implementation of the graph matcher by Messmer and Bunke with Taverna through a Parser for workflows in the Scufl language, and graph matching results are rendered as a list of workflows in a HTML document. An architecture diagram is given in Fig. 9.

From the Taverna workbench, a user imports Web directories containing workflows into the Available services pane (see Fig. 1). All workflows in the selected Web directories are translated by the Parser from the Scufl specification into a form and format the Graph Matcher understands. The user’s current working directory and its subdirectories are also scanned for workflows. Put together, these workflows form the corpus against which to match the input workflow. The input workflow equals whatever the status of the workflow is the user is currently working on. Again, the Parser reads in the input workflow by translating it to a form and format the Graph Matcher understands. The Graph Matcher detects which workflows are similar to the input and returns the results to the Formatter. The Formatter renders the results as an HTML page and launches the user’s Web browser.

A. Parser

The Parser translates the Scufl workflow specification of all workflows in a form and format suitable for the Graph Matcher. Messmer and Bunke’s Graph Matcher currently only accepts attribute-less graphs of nodes and (directed or undirected) attribute-less edges. The contents of a graph obviously impacts the outcome of the graph matching process. The workflow’s overall input and output are included as nodes in the graph. The intermediate nodes are instantiated with the names of the services connecting the workflow’s input and output. The graph’s edges are defined as the connections between the services. Other parsing strategies are of course possible (for example based on including intermediary input and output names), and would be chosen based on how well they replicate the gold standard. The information that is captured currently mirrors the information included in the diagram on Fig. 1 (except for the colouring).

B. Graph Matcher

Messmer and Bunke’s prototype implementation supports efficient matching of a large input graph to a collection of smaller graphs in the repository. In the context of workflow retrieval, this corresponds to the case where users would want to retrieve those workflows in the repository which correspond to a fragment in the user’s workflow, perhaps to find out which other authors did the same analysis.

The case where one starts out with a small input graph and matches this to a collection of larger graphs is not implemented by Messmer and Bunke’s prototype. Unfortunately, in the case of workflow matching, the latter case, where one starts out with a small exemplar workflow and one would like to compare it against a repository of large, finished workflows, seems of more practical relevance. Re-implementing the graph matching algorithm to cater for this scenario is non-trivial, however, and beyond the scope of this paper. Instead, we resorted to inverting the matching process by sequentially treating each of the (large) repository graphs as an input graph to the matcher, and treating the (small) exemplar workflow as the whole repository. This work around destroys the graph repository optimisation since it treats the prototype as a standard subgraph matching package which is invoked as many times as there are workflows in the repository.

C. HTML formatter

The results from the Graph Matcher are rendered into an HTML page, which contains links to the workflows in question (see Fig. 10). We also highlight the differences between the input workflow and the retrieved ones (not shown).
IX. Evaluation

Given the lack of a robust gold standard, no generic claims can be made as to how useful the tool is for end users. As a simple showcase, instead we aim to replicate the average workflow ranking of Table 8 in Section VI. The average ranking, in order of decreasing similarity with respect to the exemplar workflow AffyidToBlastxPDB.xml is: (i) Workflow 1; (ii) Workflow 4; (iii) Workflow 3; (iv) Workflow 2; and (v) Workflow 5.

Running the Parser on the 89 workflows available from www.myexperiment.org takes about ten seconds on a PentiumIV/512MB RAM/WindowsXP machine. The matching process itself takes about five seconds.

Different ranking strategies can be adopted. We show the impact of three different strategies.

1) **Shared nodes, string matching:** The Graph Matcher always returns the biggest subgraph found during matching with the input. We use the size of this subgraph as a measure to rank the collection of matched workflows. Without manipulating the names of the nodes (workflow input, output and services), this matching strategy returns 9 results, and of the list to be ranked contains workflows 1 and 2 (listings are provided on-line).

2) **Shared nodes, lowercase string matching:** When adapting the above strategy to make all node name assignments lower case during the Parser process, another 14 workflows show up in the matching results, including workflow 4. The list now includes workflows 1, 2 and 4, but wrongly ordered.

3) **Shared nodes, lowercase string matching, size:** Introducing a measure that compares the size of the exemplar workflow to the comparison workflow ranks workflows 1, 2 and 4 in the right way. We show the results this strategy in Fig. 10. Workflows are ordered in first instance by the number of nodes they share with the exemplar, and, in those cases where two workflows in the list have the same number, they are ordered by the size of the difference between the two workflows.

4) **Shared nodes, lowercase inexact string matching, size:** The use of strategy 3 still fails to retrieve workflows 3 and 5. Upon closer inspection, it becomes clear that inexact string matching of the service names could offer a solution here. Another solution could do matching based on classes of similar services, which opens up the door for semantic annotation. We plan on exploring both approaches in future.

X. Related work

To our knowledge, no authors have tackled the discovery of on-line workflows through the elicitation of workflow discovery requirements with end users and designed a tool specifically to rank the structural part of workflow descriptions. A vision for reuse of scientific workflows is described in [8] for a closed world system. The paper does not consider the problems associated with building or ranking on-line workflows. Few workflow repositories have been made publicly available, and even fewer have similar workflows in them. The Kepler project [1] is building a platform with workflow reuse in mind and to date offers some 20 workflows covering different sciences.

The authors of [9] use a process ontology and similarity measures to rank business processes from the MIT Process Handbook. These processes are not executable workflows hence no reuse of workflows in a Web services context is envisioned. None of the techniques considered do graph matching. In [10], an extensive ontology is built that allows to look for containing services of workflows. No ordering between the services is encoded, and matching is done through Description Logic subsumption matching without rankings. In [11], the authors consider discovery of BPEL (Business Process Execution Language) workflows based on constraints on the messaging behaviour exhibited by services.

XI. Conclusion

This paper identified the potential of workflow discovery, shown with a case study from e-Science. Requirements were elicited through a survey and a user experiment. A workflow discovery tool was developed based on graph matching to generate workflow rankings and then tested within a workflow environment on a real corpus. The tool largely replicates the average of human rankings but work remains on creating a robust human gold standard and re-evaluating the tool.

Acknowledgment

The authors thank the participants in the survey for their time and helpful comments. Thanks also to Bruno Messmer for making the graph matcher source code available. This work is supported by the UK e-Science programme EPSRC GR/R67743.

References


