Taverna Workflow for Validating BioMart Services

Kasikrit Damkliang1 and Pichaya Tandayya2
Department of Computer Engineering
Faculty of Engineering
Prince of Songkla University
Hat Yai, Songkhla, Thailand
s5010120123@psu.ac.th1, pichaya@coe.psu.ac.th2

Ekawat Pasomsub3 and Wasun Chantaratit4
Department of Clinical Pathology
Faculty of Medicine Ramathibodi Hospital, Mahidol University
Bangkok, Thailand
g4737429@student.mahidol.ac.th3, rawct@mahidol.ac.th4

Surakameth Mahasirimongkol
Center for International Cooperation
Department of Medical Sciences
Ministry of Public Health
Nonthaburi, Thailand
surakameth.m@dmsc.mail.go.th

Abstract—This paper presents a novel Taverna workflow for validating BioMart Services. It saves a significant amount of time by avoiding incorrect workflows caused by out-of-date services. It also reduces debugging time by finding out-of-date processor query fields and output fields. In addition, it provides associated information required for future validation.

Keywords—workflow; BioMart; Taverna; Web Services; validation

I. INTRODUCTION

Workflow is a data flow methodology where output of one component forms the input of others [1][2]. If any service in the workflow produces erroneous results, then the entire data flow in that section will produce incorrect results.

To decide whether a workflow is correct or not, we normally need to run the workflow and analyse its results. If the workflow is complex, with a lot of data to analyse, then it will take a long time to produce results. The results cannot be guaranteed until they are verified with other research work or data [3]. Therefore, validating the workflow before running it will help avoid unnecessary and non-productive processing, and reduce the working time.

Taverna version 1.7.0 (and later) has a workflow existence plug-in, which can check responsiveness of web service endpoints [4][5]. However, it cannot detect upgrade changes to web services, which produce empty result lists, even though the endpoints are correct.

This problem also impacts the search for processor query fields (filters) and output fields (attributes) which may simply be out-of-date [6]. The more complex the workflow, the more time will be lost in running and debugging the workflow.

The test case described here is a complex pharmacogenomic research workflow, requiring BioMart Web services to gather information from databases. The BioMart system is a flexible data warehouse aimed at complex interlinked biological data sets. Taverna’s BioMart query integration provides full search and retrieval functionality over these data sources [5][7].

This paper presents a workflow method based on the Taverna environment which validates the BioMart services before they are utilized in the processor query fields and processor output fields. This detects out-of-date services, filters and attributes, greatly reducing debugging times.

II. BACKGROUND

Many useful bioinformatic databases are accessible via the Internet, but finding the most suitable for a task is difficult since there are a few standards for representing and sharing data. myGrid e-Science provides an analysis method depending on a workflow system which can convert data to comply with web service interfaces and direct the flow of data between resources [1][2][8]. The myGrid e-Science Taverna tool is an open source workflow composer for orchestrating bioinformatic web services and existing applications into workflows. In essence, Taverna helps compose local and distributed resources into a single logical system, by employing Web Service technology [5].

Taverna can access over 30,000 services or processors in the bioinformatics domain, and there are plans to include other fields such as astronomy, chemoinformatics, and health informatics [9].

Taverna can add service endpoints, such as Web Service Definition Language (WSDL) documents [10], for improved portability, and can access other types of resources, such as, local Java services, BioMoby, beanshell scripts, and BioMart services [5][7].

BioMart [7][11] is a widely used query-oriented data management system jointly developed by the Ontario Institute for Cancer Research (OICR) and the European Bioinformatics Institute (EBI). It is open source and freely distributed without restrictions. The system accepts various types of data and also provides a kind of ‘data mining.’ Service configuration is achieved through a graphical user interface based on applications using web services, a dedicated API. A BioMart service configuration is saved into a Simple Conceptual Unified Flow Language (Seufl) [5] document in the workflow. BioMart is available to Taverna as a plug-in, providing good ways to compose workflows together.

Taverna version 1.7.0, and later, has a ‘Taverna 2 preview’ plug-in for performing workflow health checks. The plug-in examines whether the workflow can be translated from the old style Seufl model into the T2 style, and whether the web service endpoints are responding [4]. It cannot detect upgrade changes for BioMart processor filters and attributes, or examine services inside nested workflows.
As a consequence, ’Taverna 2 preview’ is somewhat lacking for validating services and reducing debugging time.

The following running and debugging steps are described in order to show how difficult it is for a user to check and debug a workflow by themselves.

When a user knows a workflow has produced incorrect results, he will check the data flow by verifying the input and output of each service and processor, which should highlight the erroneous resource. If not, then the data type and syntax of the resources must be examined. Even if these seem alright, the workflow may still produce incorrect results.

In that case, the problem may actually lie within the processor. Only then will the out-of-date processor filters and processor attributes be found.

Clearly, the main problem is when the BioMart services are upgraded, causing workflows to produce incorrect data flow.

The out-of-date filters or attributes can be found manually by looking for suspicious resources on the configuration screen for the BioMart processor node. If certain filters or attributes are absent then, it means they are out-of-date. However, manual validation can take many hours or days.

III. BIOMART WORKFLOW VALIDATION

A. Design

A workflow composed in the Taverna environment is saved into a Scufl document, a standard format from myGrid. The Scufl employs an XML (Extensible Markup Language) [12] so documents can be used with the XML Path Language (XPath) [13]. XPath is employed to extract data elements from the Scufl of a BioMart processor.

The BioMart processor is configured as a ‘data mining’ resource, which consists of a processor, or service name, and a dataset. The dataset is made of information extracted from the configuration databases; for example, the ’Homo sapiens SNPs’ dataset takes data from the dbSNP127, ENSEMBL, TSC1 and HGVbase15 databases [7][8].

The BioMart dataset configuration is presented with the query fields (filters) and output fields (attributes) matching the user’s requirements [6]. The BioMart service structure is shown in Figure 1.

![Figure 1. The BioMart service structure in a Taverna workflow](image)

An example Scufl workflow implementation of a BioMart processor is shown below:

```xml
<s:processor name="hsapiens_gene_GO">
  <biomart:query>
    <biomart:Dataset name="hsapiens_gene_ensembl">
      <biomart:Attribute name="go" />
      <biomart:Attribute name="evidence_code" />/
      <biomart:Attribute name="go_description" />
      <biomart:Attribute name="ensembl_gene_id" />
      <biomart:Filter name="ensembl_gene_id" value="" list="true"/>
    </biomart:Dataset>
  </biomart:Query>
</s:processor>
```

When Taverna opens a workflow, the responsiveness of the endpoints will be checked. If any problematic endpoints are found then Taverna will issue a warning message, and not run the workflow [5]. Since Taverna handles the endpoints, our work concentrates on validating the dataset filters and attributes in the processors. A data flow context diagram for the BioMart validating workflow is shown in Figure 2.

![Figure 2. A data flow context diagram for BioMart debugging workflow](image)

The Scufl workflow document(s) fed into the Validating Workflow process will produce a filter and attribute validation report.

The processor name used in Figure 3 is extracted from the Scufl document, and is vital for obtaining the dataset name. Subsequently, the Scufl document, processor names and dataset names are fed into a loop of validating processes, as shown in Figure 4.

Fig. 4 describes how out-of-date filters and attributes can be detected by checking their availability with the central BioMart dataset registry on the Internet. If any attribute or filter has been updated, the system collects its description, and details of each processor. This validation report will be utilized to cross-check for BioMart upgrades.

B. Implementation

The nested workflow in Figure 4 acts as an iteration module for the main task. It is utilized to check all the BioMart processor filters and attributes in Scufl documents against the central BioMart registry, located on the BioMart web services server (http://www.BioMart.org/BioMart/martservice).

The metadata about service availability is obtained via a GET request using a ’Get web page from URL’ Java local service. A number of metadata queries are available, which can be formulated by appending parameters to the end of the URL [11]. The first parameter is appended using the ? symbol, and subsequent parameters use the & symbol. The format is:

```
../martservice?type=<filters|attributes>&virtualschema=default &dataset=<datasetname>
```

For example, the command for retrieving the attribute registry of the ’hsapiens_gene_ensembl’ BioMart is:
XPath expressions are utilized to extract data elements from the Scufl workflow using the 'Xpath From Text' Java local service, as shown in Table I. The validating processes are coded in Java using the Beanshell local service [4].

Fig. 5 shows the composed workflow for validating the filters and attributes of large BioMart services. The 'Check Filter' and 'Check Attribute' rectangular frames are nested looping workflows.

Taverna manages the data flow of output ports using the cross product ('all against all') relationship. In our work, the data flow management for integrating the nested workflows into the upper processes is carried out by the Taverna 'Configure Iterators' [4] as shown in Figure 6. The processor name and dataset name are used in a dot product ('one against one') relationship, whereas the workflow Scufl employs cross product. The relationship output of the data ports is shown in Table II, showing that a Scufl has many processors, and each processor has only one dataset.

IV. EXPERIMENTAL RESULTS

Our first test utilized an out-of-date pharmacogenomic workflow that no longer produces a correct data flow, and we focused on reducing the validating time and avoiding wasteful executions of workflows that produce no results. The simple testing workflow for Gene Ontology [14] is shown in Figure 7, utilizing a BioMart processor 'hsapiens_gene_ensembl' labelled as 'hsapiens_gene_GO.' The service consists of one filter and four attributes. We ran the workflow using two Ensembl IDs. The response arrived in less than two seconds, but was empty.

<table>
<thead>
<tr>
<th>Processor names</th>
<th>Xpath expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>BioMart processor</td>
<td>//<em>[local-name(.)='processor'] and (@name=&lt;Processor Name&gt;) /</em>[local-name(.)='biomart']/parent::s:processor/@name</td>
</tr>
<tr>
<td>BioMart dataset names</td>
<td>/<em>[local-name(.)='processor'] and (@name=&lt;A BioMart processor name&gt;) /</em>[local-name(.)='biomart']/parent::s:processor/@name</td>
</tr>
<tr>
<td>BioMart dataset filters or attributes</td>
<td>/<em>[local-name(.)='processor'] and (@name=&lt;A BioMart processor name&gt;) /</em>[local-name(.)='biomart']/parent::s:processor/@name</td>
</tr>
</tbody>
</table>

Figure 6. Configure Iterators in Taverna
Our validating workflow helps to identify the out-of-date filters and attributes, as shown below:

Processor Name: hsapiens_gene.GO, Attribute: go is out-of-date
Processor Name: hsapiens_gene.GO, Attribute: evidence_code is out-of-date
Processor Name: hsapiens_gene.GO, Attribute: go is out-of-date
Processor Name: hsapiens_gene.GO, Attribute: go_description is out-of-date
Processor Name: hsapiens_gene.GO, Attribute: ensembl_gene_id is up-to-date
Processor Name: hsapiens_gene.GO, Attribute: ensembl_stable_id is up-to-date
Processor Name: hsapiens_gene.GO, Attribute: feature_page is up-to-date

The validating workflow also collects the up-to-date filters and attributes for later use:

Processor Name: hsapiens_gene.GO, Filter: ensembl_gene_id is up-to-date
Detail: Filter to include genes with supplied list of Ensembl Gene IDs
Processor Name: hsapiens_gene.GO, Attribute: ensembl_gene_id is up-to-date
Detail: Ensembl Stable ID of the Gene, feature_page

The description of the ‘ensembl_gene_id’ attribute, is the Ensembl Stable ID for the Gene, and is available on the feature page of the BioMart service configuration screen.

The second major test of our validating workflow is a pharmacogenomic workflow, available at http://www.myexperiment.org/workflows/610.

The workflow testing results are shown in Table III. It consists of 52 services, including 17 that link to BioMart services, 70 corresponding data links, and three data querying BioMart services with three filters and 22 attributes.

The workflow inputs are 500 to 1,000 Single Nucleotide Polymorphisms (SNPs) are the smallest genetic statistic P values according to the Genome Wide Association Study (GWAS) [3] of 550,000 SNPs.

It took 14 hours to run the workflow with 1,000 SNP IDs. Every service works successfully except the Gene Ontology part which returns an empty list.

As shown in Table III, it took 26.5 seconds to validate the workflow, and identify the out-of-date filters and attributes at the Gene Ontology finding service. This shows that validating a workflow before executing it can significantly reduce work when there are out-of-date services. Debugging time for finding the cause of incomplete results is also reduced. After the validation, we can configure our workflow so that all the units are up-to-date, as shown at http://www.myexperiment.org/workflows/612.

V. RESULT DISCUSSION AND DEBUGGING SOLUTION

The experimental results show that minor workflow re-design is often required: out-of-date services must be removed and new services added.

Even when there is only one out-of-date filter, the workflow will fail to produce correct data flows because query fields must pass all the filter conditions before retrieving information. The user must fix the offending processor node before the workflow will function.

Even a processor with an out-of-date attribute that is not an input for downstream processes may still cause the processor to produce incorrect data.

Out-of-date filters and attributes due to the upgrades are not shown on the BioMart configuration screen, so the user cannot deselect them. Advanced users may manually edit the Scufl document but this tends to be tedious, confusing and error-prone.

Our advice is to add a new service with the same processor name that occurred in the validation report. This is done by reselecting the service from Taverna’s service available panel, and then attaching the processor and its data links to the other processors. When the added processor does not have the necessary filters or attributes, then replacement services in the same domain should be utilized if possible.

VI. CONCLUSIONS

We have developed a Taverna workflow for validating the processor query fields and processor output fields in BioMart services. This validating workflow can help improve workflow processes without requiring the execution of the workflow. It can save significant amounts of computing time when running incorrect workflows, and reduces the debugging time. It also provides the associated information useful for future validations.

ACKNOWLEDGMENTS

This research work is a collaboration between the Computer Engineering Department, the PSU Grid Center at Prince of Sonkla University, and the Thailand Center of Excellence for Life Sciences (TCELS) at Mahidol University. This work is partly supported by the Biogrid Project funded by the Thai National Grid Center (TNGC), Thailand’s Software Industry Promotion Agency (SIPA) and TCELS. The authors are grateful to Prof. Carole Goble, Prof. Andy Brass and Dr. Katy Wolstencroft at the School of...
Computer Science at the University of Manchester, and myGrid team members for Taverna training. The authors are also thankful for Dr. Andrew Davison for proof reading the paper.

REFERENCES


Figure 5. The workflow for validating the BioMart filters and BioMart attributes