Experiencing Data Grids

Nicolaas Ruberg, Nelson Kotowski, Amanda Mattos, Luciana Matos,
Melissa Machado, Daniel Oliveira,
Rafael Monclar, Cláudio Ferraz, Talitta Sanchotene, Vanessa Braganholo

COPPE/Federal University of Rio de Janeiro, Brazil

{nicolaas, kotowski, amandasm, lrmatos, msm, danielc, rmonclar,
cferraz, talittas. vanessa}@cos.ufrj.br

Abstract. Many scientific experiments deal with data-intensive applications and the orchestration of computational workflow activities. These can benefit from data parallelism exploited in parallel systems to minimize execution time. Due to its complexity, robustness and efficiency to exploit data parallelism, grid infrastructures are widely used in some e-Science areas like bioinformatics. Workflow techniques are very important to *in-silico* bioinformatics experiments, allowing the e-scientist to describe and enact experimental process in a structured, repeatable and verifiable way. The main purpose of this paper is to describe our experience with Tavena Workbench and PeDRo, which are part of myGrid project. Taverna is provided with a workflow toolset and enactor, allowing the specification of processing units, data transfer and execution constraints. As a data entry tool, PeDRo provides a model, a controlled vocabulary and field validations for Web Services descriptions, leveraging the knowledge associated to the workflows. The main contribution of this work is a summary of some considerations drawn by our experience with the use of these tools, emphasizing its advantages and negative aspects, together with proposals for some future improvements.

1. Introduction

The development of computational infra-structures and the mass use of tools to manipulate the bioinformatics data produced by e-scientists have increased the necessity to execute *in-silico* experiments. Such experiments are usually captured by a workflow, and can be enacted using workflow engines. One of such computational strategies is myGrid [19], which exploits Grid technology to efficiently support bioinformatics applications and experiments.

However, the construction of formal data models to represent these experiments and their associate data is characterized by the use of free-text representations or semistructured data. As an example, the experiments are annotated with free-text describing the main aspects of the adopted experimental technique. These annotations are essential for a more complete analysis of the experiment, and also for future experiments.

Traditionally, several formats and formalisms have been used to construct annotation databases. Free-text is still the most common formalism. The main advantage of

this approach is its expressiveness. However, the use of free-text limits search capabilities and automated comparisons. A simple alternative would be to use a controlled vocabulary. Nevertheless, this approach would reduce the expressiveness. The most adequate option would then be to use ontologies together with a tool that allows the construction of data models and their association with the ontologies.

Different areas consider different definitions for ontologies [7,9,10]. In bioinformatics, an ontology is a concise and non-ambiguous description of the relevant entities of the application domain, and of the relationship of such entities [16]. Entities may be objects, processes, functions, predicates and other application-dependant types. An ontology eliminates the uncertainty and misinterpretations of the semantics of data, programs and their relationships. Consequently, it makes it easier to create application systems in the bioinformatics domain.

Towards an engine that could not only provide an effective means of creating and enacting bioinformatics (scientific) workflows, but also deal with ontologies and the benefits that they may provide, in this paper, we describe two tools of the myGrid environment:

- Taverna [20], a workbench for the development and execution of workflows. Taverna allows the integration of Web Services in scientific workflows, which makes it easier to create workflows, and discover ready-to-use Web Services; and
- PeDRo [14], a tool that allows creation, manipulation and maintenance of biological ontologies.

By experiencing these tools, we provide our first contribution: a report on how good Taverna and PeDRo did concerning the aspects just highlighted. With the considerations and aspects shown in this report, we then draw our second and main contribution, which is a set of proposals for future improvements in these tools.

This paper is organized as follows. We briefly describe Taverna and PeDRo in Sections 2 and 3. Section 4 presents a report on our experience in using both of these tools. Finally, Section 5 closes this work with some final remarks and research perspectives.

2. Taverna

An initiative from the collaboration among several institutions (the European Bioinformatics Institute (EBI), IT Innovation, the School of Computer Science, University of Newcastle, Newcastle Centre for Life, School of Computer Science at the University of Manchester and the Nottingham University Mixed Reality Lab), research projects (the Biomoby project [2], Seqhound [17], Biomart [1]) and various individuals in general, Taverna [20] plays the role of a workbench for the development and execution of workflows concerning bionformatics in the myGrid project.

When we use the term "*workflow*" in the myGrid environment, we are referring to the composition of local and remote (Web) services to achieve a biological experiment. This kind of composition is provided by defining the workflow steps using the SCUFL language. We consider a Web Service a software component that is available on the Internet and that uses a standardized XML messaging system. There should be some mechanisms so that the interested parts can easily locate services and their public interfaces.

Especially in bioinformatics, most of these *in-silico* experiments are related to the use of computational tools and databases. Almost all of these computational tools are being made available as Web services. Because of that, those who make use of such tools feel the need to orchestrate these web services in workflows as part of their *in-silico* experiments. Once the workflow is defined in the SCUFL language, each step within a workflow represents one atomic task (a Web service, for example).

One important issue that needs clarification is the main difference between business workflows and scientific workflows, since Taverna is strictly concerned with scientific workflows. According to Santos [21], scientific workflows share many characteristics of business workflows, but present some important items not found in business workflows:

1. Scientific Workflows are normally designed by scientists: Taverna's main target audience (biologists and bioinformaticians) may neither pursue a wide computational background, nor the necessary computing infrastructure or specialized staff to develop or support such workflows. Usually, Taverna users lack the knowledge of scripting or programming languages. In order to allow the ease of workflow development and usage, Taverna Workbench provides a window-based, user-friendly interface. The workflow components are added through the provided examples, and also through the standardized data structures available, which are close to a general workflow creation language. The workflows developed in Taverna are written in the Simplified Conceptual Workflow Language (SCUFL) and enacted using the Freefluo workflow enactment engine [12].
2. Scientific Workflows are designed to prove a Hypothesis or a Theory. This way, the definition of the workflow is always a dynamic process that it is influenced by the obtained results, generating constant changes in the execution flow to achieve a desired result. The workflow will probably be re-executed many times in a day, week or month. Because of that, a mechanism that allows the scientist to save the developed workflow is needed and Taverna provides this kind of mechanism.
3. Scientific Workflows will probably be reused by other scientists: workflows already executed can be reused to reproduce an earlier experiment. The workflow tool must provide a way to recover previous workflows that can be reused or modified as needed.
4. Provenance data must be collected in order to assure high data quality: provenance data like "Responsible for the workflow execution", "Date and Time of the execution", "Annotation Data" are very important to other scientists who will re-execute the workflows in order to compare the results achieved;
5. Controlled Execution of the workflow (Partial execution): we can define a workflow as "a learning process". Because of that, scientists will only be able to decide to continue workflow execution after they have evaluated the partial results already achieved. If the results achieved are unsatisfactory, they can stop the execution and start it again with new parameters or input data. This kind of mechanism is very useful because some services included in the workflow can take a long time to execute. Because of this, the scientist must be able to stop the execution of any workflow and start it again from the point he/she

has stopped it before. This way, it should be provided some "savepoints" in the workflow to mark the points in which execution can be re-started..

6. Fault Tolerance: when an error occurs, there must be a contingency plan. It is important to say that these errors are related to execution problems, like unavailable services. This way, the user must be able to define alternate services that will be executed as needed.

By default, Taverna provides some "standard" Web services that are available to the users after installation (Biomart Data Services, Soaplab Analysis Services at EBI, SOAP Services, and so on) and new Web services can be added as needed.

The Taverna Workbench is composed of four main modules: the *Advanced Model Explorer (Scufl Model Explorer)*, in which the workflow is developed following the above considerations, the *Workflow Diagram (Scufl Diagram)*, a module that presents the workflow graphically to its users, the *Available Services*, where the user is able to select or simply point to which local or remote service to use inside a workflow, and the *Enactor Launch Panel*, that presents the status of the workflow steps execution and its final result to the user [13].

Besides workflow development and execution, Taverna holds the ability to support highly complex data analysis, not only from private or local databases, but from any Web service at hand. For example, one of the workflow examples provided in Taverna Workbench allows its users to track down a gene ontology graphically. With a simple data input (an alpha-numerical code that represents the gene identification), the user submits the workflow execution, which then accesses a Web service and retrieves the gene ontology for the input provided. While the workflow is being processed, GraphViz starts to draw the result tree and associates specific roles within the ontology with colors.

3. PeDRo

The Taverna workflow environment is provided with a tool for data entry of biological data models. This tool, PeDRo [14], allows biologists to enter descriptions and ontological annotations on data sources and biological services. The data input is validated against an XML schema, and data fields are verified against a controlled vocabulary. The idea of an XML schema validation is to provide an intrinsic support to a domain metadata; and the goal of a controlled vocabulary field association is to enable an easy way to support ontologies.

The XML schema provided with Taverna/PeDRo is conceived to describe services and workflows for the purpose of discovery. The actual standards for service descriptions, UDDI, OWL-S and WSDL are not semantically rich enough to provided queries over the Taverna ontologies. Thus, such standards are extended to incorporate the concepts needed for Taverna to search and discover services in the Grid. More details on these issues are described in [14].

In order to build and to support the use of ontologies, PeDRo plays two roles: i) a data entry tool from a predefined XML schema; and ii) a quick modeling tool. As a data entry tool, PeDRo is embedded within Taverna environment as part of the Java

application interface. When activated, it opens a window with a navigation tree and an edition form. On the navigation tree, elements are structured and presented accordingly to the XML schema. On the edition form, data is inserted and ontologies associated to each field. As a modeling tool, PeDRo is available as a standalone application. It provides the same interface as in the bundle application but with more flexibility wrt the construction of XML schema and ontologies -- both data are stored in plain text files. Therefore, to custom the XML schema, it suffices to change the XML schema text file (Figure 1), as well as to custom the ontologies, editing the respective file will incorporate the desirable property (Figure 2).

```
<xs:element name="serviceDescription">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="serviceName" type="xs:string" minOccurs="0"/>
   <xs:element ref="organisation" minOccurs="0"/>
   <xs:element name="serviceType" minOccurs="0">
   <xs:simpleType>
    <xs:restriction base="xs:string">
     <xs:enumeration value="Soaplab service"/>
     <xs:enumeration value="WSDL service"/>
      ...
```

Figure 1: XML Schema sample for Taverna ontology

```
bioinformatics_application
 Basic_Local_Alignment_Search_Tool
  tblastn
  tblastx
  blastn
  blastp
  blastx
 EMBOSS
 primer3
...
```

Figure 2: Taverna field to Ontologies association file

4. Taverna and PeDRo getting together

In this section we describe our experience using the Taverna Workbench and PeDRo tools by means of a practical example. However, before proceeding, we present some difficulties found and positive aspects of Taverna.

4.1. Experiencing Taverna

We have installed and used Taverna Workbench version 1.2 in both Linux and Windows platforms and we found it very useful and easy to use. Despite the simplicity of the installation process and the effectiveness of the installation guide, some difficulties were encountered in this phase.

In order to fully experiment the Workbench we installed ^{my}Grid, and for that installation and experience we point out three minor faults observed: i) the configura-

tion process; ii) security issues; and iii) lack of tools to integrate/create virtual organizations.

The first aspect that called our attention was the amount of configuration files required to run myGrid services, which is not an easy task, susceptible to errors. To uncomment the wrong line or miss a comment in the configuration file is enough to make some tools not to work at all. Moreover, some guidelines found within those files do not match the available information in the user's guide. Thus, for unskilled users such as e-scientists, it is difficult to install the required tools.

While configuring the XML files and properties, we noticed that security aspects are not fully observed. For example, logins and passwords are stored in plain text in XML files, which are edited by the users themselves. Since most partial results in these experiments are confidential, security can be an issue.

The third aspect observed regards the definition, construction and use of a virtual organization in a grid environment. In that sense, there is no documentation in how to setup a custom virtual organization in myGrid. It is not clear how to aggregate services, since Taverna does not easily provide features to publish these services. Finally, there is no authentication in the grid. Therefore there is no restriction in the use of the services provided.

Despite of these difficulties, Taverna gave us a positive impression. The tool has shown us an expressive importance to bioinformatics researchers as it offers a simple and efficient environment. This workbench is intuitive, useful and loads bioinformatics web services in its initialization. The user can verify the workflow status in real time, based on the services selected to compose the workflow activities. Also, services can be added to Taverna from specific sites which contain their definition code.

The workflow definition language (SCUFL) is simple and easy to learn. Although one can find some difficulties to use this language, it is possible to create a workflow connecting operations and filling some properties in a friendly user interface that is provided in Taverna. The user may also define them directly on a XML file. A drawback is that the SCUFL language do not complies to the *defacto* standard for Web Services Workflow BPEL4WS[3].

The real time workflow visualization is an interesting aspect observed since it prevents rework. The graphical representation of the workflow can be saved in various image formats. Many kinds of visualization are offered, from the simplest to the more complex ones, in which workflow information is exposed in the graph.

4.2. Using Taverna and PeDRo in practice

Our strategy to test the Taverna workbench and PeDRo attributes is to cover a complete cycle of a biologist interaction with the platform. Our experiment involves: i) constructing and deploying a Web Service in the workbench; ii) describing the Web Service via the PeDRo tool; and iii) constructing and running a workflow with this Web Service.

For didactical purposes, we tested a workflow with the implementation of a simple Web Service. Its WSDL specification is presented in Figure 3. This service receives a string as input, and echoes that string back as an output. We called it *EchoService*. The Web Service is constructed using the AXIS framework, which is the *defacto* standard for Java Web Services implementation.

```
...
<wsdl:message name="serviceMethodResponse">
  <wsdl:part name="serviceMethodReturn" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="serviceMethodRequest">
  <wsdl:part name="inputArg" type="xsd:string" />
</wsdl:message>
<wsdl:portType name="EchoService">
  <wsdl:operation name="serviceMethod" parameterOrder="arg">
     <wsdl:input message="impl:serviceMethodRequest" name="serviceMethodRequest" />
     <wsdl:output message="impl:serviceMethodResponse" name="serviceMethodResponse" />
  </wsdl:operation>
</wsdl:portType>
...
```

Figure 3- EchoService WSDL extract

Constructing and deploying the Web Service. When starting, the Taverna Workbench displays three windows: the Model Explorer, the Workflow Diagram, and the Available Services. In the first interaction, the biologist constructs a scientific workflow by picking up services on the available services window; connecting them on the model explorer window. The graphical visualization of the experiment is shown in the Workflow Diagram window.

In order to make the *EchoService* available to the workbench, we need to include it on the available services window. All the available services are displayed in a tree structure on the interface. To add a new service, we right-click the root of all services, and we select the appropriate service category on the displayed menu, which, in our case, is the "WSDL scavenger". After providing the WSDL description file address or the WSDL description URI, the service is included on the tree of services and is ready for use. In our particular case, the available services window with the *EchoService* is shown in Figure 4.

In this first moment we observed that is necessary to re-include the service every time the workbench is restarted. This can be cumbersome if we have several customized Web Services.

Describing the Web Service via the PeDRo tool. In order to enable a semantic search over the services registered in the Grid, the Feta Engine is provided. This tool relies on an agreed ontology for the services description and an entry tool to input the data required by the ontology. With the perspective of an e-scientist, we used PeDRo to provide the semantic description of the *EchoService*. As mentioned before, PeDRo allows an annotation according to a predefined XML schema, and restricts some fields to a controlled vocabulary.

In order to provide this description, we access the PeDRo tool interface from Taverna's main menu. A form is presented so that one can provide the service description. Most of the input fields are required information to describe the service itself, such as the Web Service WSDL. However, some extra information is also necessary to better describe the service. To illustrate, in Figure 5 we present the service description for the *EchoService*. We observe that the fields Web Service type, author, description text, and organization do not belong to the WSDL specification shown in

Figure 3, though they were included in order to increase the service semantic description.

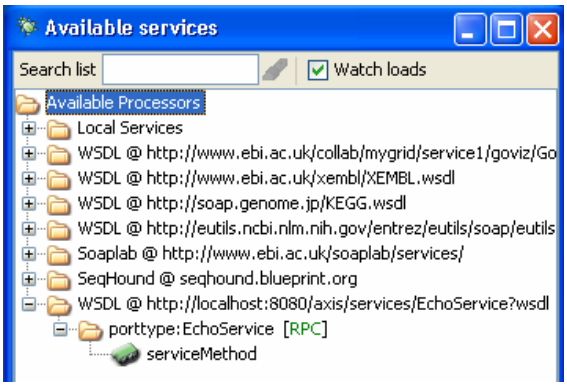

Figure 4 - New service in Taverna Workbench

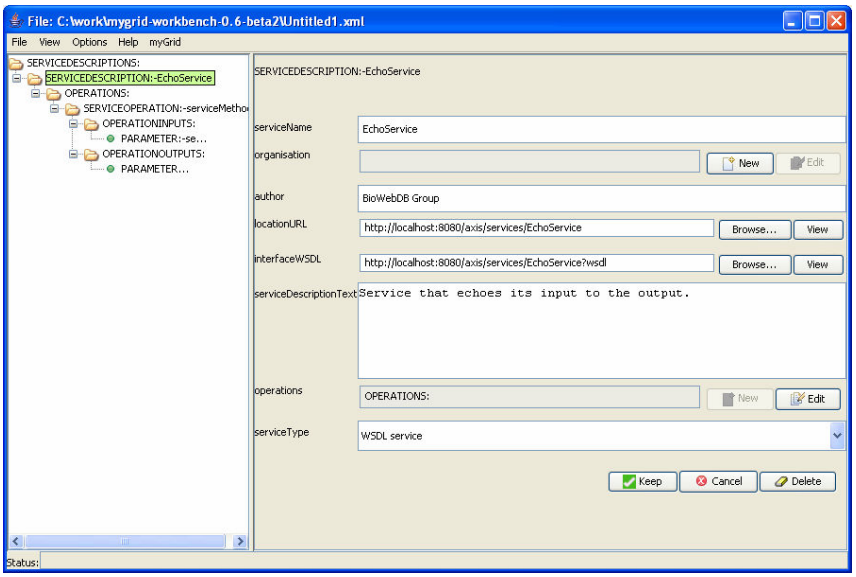

Figure 5 - Describing a Service in PeDRo

Two important aspects were observed in our experiment with PeDRo: i) although the service is already on the workbench, no description information is retrieved automatically by PeDRo; ii) in the interface, the built semantic description is ready to be published in the consortium registry site, but there is no option for a local publishing, at least not in an out of the box manner.

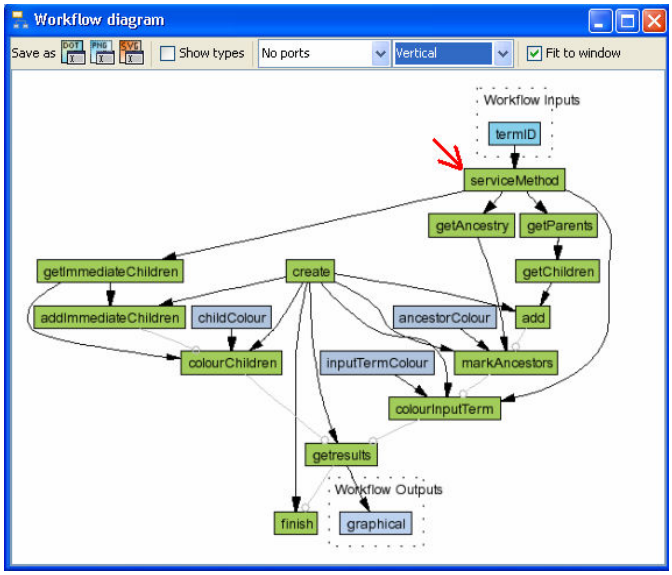

Figure 6 - Workflow with the built Web Service

Constructing and running a workflow with the EchoService. Our third step on the experiment was to include the EchoService in a workflow. We picked up one of the example workflows provided with the workbench, the ShowGeneOntology. The idea of this workflow is to retrieve the Gene ontology tree and display its graphical view for a given Gene Ontology ID. In order to have a running example with our service, we attached the input of our EchoService to the workflow input field, and attached the service output to the corresponding input services in the workflow. The resulting workflow is shown in Figure 6.

5. Suggested Improvements and Final Remarks

In this paper, we have experienced Taverna and PeDRo. Both of these tools are focused on the bioinformatics area. Our University is involved in the BiowebDB Consortium (http://www.biowebdb.org) that aims at supporting genomic workflows to provide interoperability among different analyses tools and more sensitive algorithms for distant homology detection. This evaluation has motivated us to integrate some of Taverna/Pedro tools with current BiowebDB services architecture [4]. With this idea in mind, we would like to point some problems out, and make some improvement suggestions. This is the main contribution of this paper.

Inside the Taverna Workbench we found that one of the most important topics for the development of scientific workflows is the possibility to do a controlled execution. In this tool we can steer the workflow execution by using the breakpoints feature or simply by manually pausing it. This is very useful, as we can partially or com-

pletely execute a given workflow, edit its intermediate values and even simulate a step-by-step execution by placing breakpoints at each activity to be interrupted. Although these assets provide some advantages, they do not have the necessary flexibility to enact scientific workflows completely, as we can not change the activities course at runtime according to the intermediate results..

In certain experiments, such as those concerning bioinformatics, it is almost impossible to execute the workflow in its totality, as the processing time of each web service may be enormous [12]. A workflow executed step-by-step could help to visualize errors that may have happened during the execution of a web service that is part of the workflow. Moreover, it makes it possible to cancel the workflow execution, avoiding the execution of all other processes with errors generated by previous web services, saving CPU time and reducing the cost of experiments.

However, in the Taverna development environment, the e-scientist can not find a way to dynamically choose other services to be executed on the next workflow steps depending on the results. Also, it is not possible to re-execute the workflow from a specific previous step, editing the intermediate values. It is only allowed to continue the execution from the paused step or to re-execute the entire workflow.

Another improvement opportunity is related to enabling visual workflow design through the workflow diagram. Currently, the workflow composition task is only available in Workflow Explorer module. The Workflow Diagram provides just the visualization of the created workflow, but not its edition.

We suggest the possibility to create/exclude workflow objects from the workflow project and, moreover, to edit its properties or metadata, working directly on the graph. We consider that with these improvements inside the workflow diagram module, similarly to what is provided in the workflow explorer, the workflow composition would be simplified and faster, especially for complex workflows.

After having the opportunity to analyze the Taverna Scufl Workbench environment, we would like to go further and use it in a more standard grid environment such as Globus [8]. It is not clear for us if the Taverna Team has plans to develop a Globus/myGrid integration module. In our opinion this would broad the Workbench execution possibilities by taking advantage of the Globus Toolkit components, which involves failure management and wider use of grid services. However, the issues here go beyond that, since the use of Web Services in Grids still poses some problems. As defined by Foster (1998) a grid must provide security, unique identification service and quality of service [6]. Nevertheless, the current implementation of the Web Services specification does not provide these features. This is because the HTTP connection between the server and the client uses no cryptography, which means that SOAP messages are exchanged with no security. Besides, there is no user identity guarantee in service calls. The only identity is the IP address, which can be easily forged (through a HTTP proxy, for instance). Another aspect is that the HTTP and SOAP protocols have no mechanism to guarantee the provided service. This way, it is still not possible to apply the current Web Services architecture in Grids, but there are standardization proposals in Web Services involving these features. It is called Web Services Security (WS-Security) [5].

WS-Security proposes an extension to the SOAP protocol by adding message deliver guarantee, confidentiality and a unique authentication mechanism. However, WS-Security is not a standard yet [5]. This way, we can foresee a common path be-

tween Grid Services and Web Services, despite of the deficiencies to achieve the Grid requirements in the current Web Services specification. WS-Security can be the way to get there. Globus GT4 seems to be going in this direction.

Our remarks concerning the PeDRo tool salient its characteristics as a data entry tool as well as a design tool. Pedro allowed Taverna/ myGrid to incorporate ontologies to the description of Web Services and workflows. It favors a simple input of Web Services description and validation against a predefined XML Schema. In addition, the decision to integrate the tool to the workbench was due to some design aspects of PeDRo. It is built in Java and made available as a package with interfaces for other Java applications. Besides defining the data model control, data validation routines can be associated to a data entry. To summarize, the benefits of PeDRo are [14]:

- it can be used for rapid data modeling and for data entry;
- it lets the creation of complete, well-formed data files;
- it supports context-sensitive help that describes the model;
- it supports controlled vocabulary (ontology) service;
- it is free and a supported open-source tool serving a user-base of scientists;
- it is simple to use and has an intuitive interface.

In other words, PeDRo allowed the Taverna development team to easily provide a model, a controlled vocabulary and field validations for Web Services descriptions on the workbench. These characteristics guarantee that the elements described with the tool will respect the requirements of the myGrid ontologies.

The main drawback to PeDRo is the lack of tools to support the data modeling. The PeDRo tool needs several configuration files, as for example, one with the validation schema, another with the contextual help, and other with the controlled vocabulary for an input field. Those configuration files are particular to the tool, which restrains changes in the ontology model; as well as the designer has to rely on other tools, e.g. XML editors, to build the XML configuration and vocabulary text files. Just the verification and integrity validation of these files are done through PeDRo's interface. Those several configuration files bring up another issue that increases the difficulty in modeling with PeDRo, those configuration files are spread in several directories. For example, the XML schema is stored in a different directory from the ontologies. As improvement, the generation of these configuration files should be done automatically through PeDRo's interface or a provided tool.

In this paper, we provided a summary on our experience using Taverna and PeDRo, both part of myGrid project, considering their importance to the e-science scenario. Based on these experiences, we proposed some improvements to these tools. Such suggestions aims at making easier the tasks of scientific workflow design and enaction .

References

1. BioMart Project. 2006. Available at http://www.biomart.org/.
2. BioMOBY. Available at http://biomoby.open-bio.org/.

3. Bussiness Process Execution Language for Web Service version 1.1. In http://www-128.ibm.com/developerworks/library/specification/ws-bpel/, Feb 2005.
4. Davila, A.; Lorenzini, D.; Mender, P.; Satake, T.; Sousa, G.; Campos, L.; Mazzoni, C.; Wagner, G.; Pires, P.; Grisard, E. GARSA: Genomic Analysis Resources for Sequence Annotation. Bioinformatics, 2005.
5. Foster, I. A Globus Primer. Available at http://www.globus.org/toolkit/docs/4.0/key/. 2005.
6. Foster, I.; Kesselman, C. The Grid: Blueprint for a new computing infrastructure. Morgan Kaufmann. 1998.
7. van Heijst, G.; Schreiber, A.; Wielinga, B. Using explicit ontologies in KBS development. International Journal of Human-Computer Studies. 46. pp. 183-292. 1996.
8. Globus Toolkit. Available at http://www.globus.org/toolkit/.
9. Gruber, T. A translation approach to portable ontologies. Knowledge Acquisition, 5(2), pp. 199-220. 1993.
10. Guarino, N. Formal Ontology and Information Systems. In: International Conference on Formal Ontologies in Information Systems (FOIS). Trento, Italy, June 1998. pp. 3-15.
11. Kaler, C. et. al. Web Services Security (WS-Security). Available at http://www-128.ibm.com/developerworks/webservices/library/ws-secure/. 2002.
12. Oinn, Tom; Greenwood, Mark; Addis, Matthew; Alpdemir, M. Nedim; Ferris, Justin; Glover, Kevin; Goble, Carole; Goderis, Antoon; Hull, Duncan; Marvin, Darren; Li, Peter; Lord, Phillip; Pocock, Matthew; Senger, Martin; Stevens, Robert; Wipat, Anil; Wroe, Chris. Taverna: Lessons in creating a workflow environment for the life sciences. In: Concurrency and Computation: Practice and Experience, pp.2. 2002.
13. Oinn, Tom; Addis, Matthew; Ferris, Justin; Marvin, Darren; Senger, Martin; Greenwood, Mark; Carver, Tim; Glover, Kevin; Pocock, Matthew R.; Wipat, Anil; Li, Peter. Taverna: a tool for the composition and enactment of bioinformatics workflows. Bioinformatics Journal, 20(17), pp. 3045-3054. 2004.PeDRo, dynamic form generation, XML Schema, data validation, controlled vocabulary services…; Manchester University; 2004; Available at http://pedrodownload.man.ac.uk/main.html.
15. Santos, Rafael – "O Ambiente 10+C para a definição e execução de workflows in silico através de serviços web" – Master Thesis, COPPE/UFRJ, 2004. In Portuguese.
16. Schulze-Kremer, S. Ontologies for Molecular Biology. In: Pacific Symposium on Biocomputing. 1998. pp. 693-704.
17. SeqHound. Available at http://www.blueprint.org/seqhound/.
18. Silva, F.; Cavalcanti, M. Intermediate Data Management for In-Silico Workflows using Web Services. In: Workshop de Teses e Dissertações em Banco de Dados, 2005. Uberlândia, MG, Brazil.
19. Stevens, R.; Robinson, A.; Goble, C. myGrid: Personalized bioinformatics on the information grid. Bioinformatics, 19(1), pp. 302-304. 2003.
20. Taverna Project Website. 2006. Available at http://taverna.sourceforge.net/.
21. Wroe, C.; Lord, P.; Miles, S., Papay, J., Moreau, L.; Goble, C. Recycling Services and Workflows through Discovery and Reuse. Proc UK e-Science All Hands Meeting 2004, pp. 622-629. 2004.