Distributed Execution of Workflows

Ismael Navas-Delgado¹, Jose F. Aldana-Montes¹, and Oswaldo Trelles²

¹ University of Malaga, Computing Languages and Computer Science Department,
29071, Malaga, Spain
{ismael, jfam}@lcc.uma.es
<http://khaos.uma.es>

² University of Malaga, Computer Architectures Department,
29071, Malaga, Spain
ots@ac.uma.es
<http://www.ac.uma.es>

Abstract. In this work we present a tool for manually connecting services by means of their input and output data, helping bioinformaticians in the automatic evaluation of workflows for the dynamic integration of heterogeneous data sources, services and computational resources. Our workflow platform offers a view of the different tools available as a single and uniform pool of services readily available for enhancing query processing.

1 Introduction

A Web-based service facilitates access to remote resources promoting the development and availability of highly diverse and specific tools. These new resource capabilities are of special interest in the bioinformatics domain where a mixture of databases and services are required to produce a more complete view of biological problems. Unfortunately, the common bioinformatics research field is hard to operate in since it can involve the localization of adequate web services by collecting URLs of the useful ones, selecting the most popular or qualified ones, getting familiar with their specific interfaces (e.g. see the very popular sites like NCBI, EBI, ExPASy, etc.) copying and pasting data, manually selecting and combining partial results, and building by-hand and scheduling workflows. Thus, the main component of a bioinformatician's daily work consists in carrying out a set of simple activities that they usually perform using a diverse set of tools for solving a problem. This implies interacting with different interfaces and storing partial results, in order to use them subsequently in another tool. This manual interaction with services is costly and prone to error.

The use of a workflow-based system is very useful when the tasks that the user needs to solve are usually predefined and the relationships between the tasks are well known. Our proposal provides an execution environment for related tasks, so that users can develop a workflow (making use of an external graphical tool) defining a set of related tasks for solving a specific problem, and can then execute the workflow with specific inputs. This process can be repeated several times with different inputs in order to derive biological conclusions.

In this paper we focus on the description of a workflow management environment that provides a set of applications for storing, executing and monitoring workflows defined by means of XML files in the Scuf1 [1] representation language.

2 System and Methods

The process of defining and executing workflows that we propose is composed of several steps (see Figure 1): (1) To define workflows manually (writing an XML Scuf1 file) or using a graphical tool for this language, such as Taverna Workbench, which allows users to graphically describe the services to be executed and how they are related; (2) To store user or generic workflows in the platform making use of the provided web interface; (3) To specialise workflows by defining their inputs, specific values for simple data objects or links to existent complex BioMOBY objects; (4) To execute workflows, taking the inputs and then executing services as soon as their inputs are available; (5) To monitor the execution, noting the changes in the status of the processors and analysing the obtained data.

Fig. 1. Data and Process Flow for using the workflow management platform. Blue nodes are external applications or services, and green ones are internal processes.

Internal Representation and Storage

Our platform offers an interface for loading, executing and monitoring the execution of workflows (see Section 2.3). This platform offers a persistent database to store workflows in order to improve their use and their analysis (see Figure 2). This database contains a set of tables that can be queried to retrieve workflow information or for executing the workflow (Workflow, Processor, Link, Source and Sink). In the current version we offer the capability of executing BioMOBY services and scripts related with them, like those for getting the inputs and showing the outputs (*Create_moby_data* and *Parse_moby_data*).

Each processor is connected to a workflow (by means of the Id), and those processors are related by means of links, in which one processor acts as a source and the other as a sink. Then, sources of a workflow are linked by means of the link table with a processor (source data can be used in several processors). In the same way, the sinks of a workflow are linked to the output of a processor (usually each sink is related to only one processor).

The database tables designed include all the elements for defining a workflow (composed of BioMOBY services), so that there is no loss of information in the

storage process. Besides, the use of constraints allows verifying the correct insertion in the database, and additionally it is possible to rebuild a workflow (in XML format) from the database. Thus, if a workflow is published and shared between different users, all of them can retrieve the XML description of the workflow in order to make use of it.

When a workflow is loaded, it is parsed and its elements (processors, links, sinks and sources) are stored in the database and a copy of the XML document (describing the workflow with ScufI) is uploaded onto the INB web server. Besides, additional information is added to workflows in order to improve the documentation of loaded workflows: Name, Short Description and an optional Long Description (as documentation of the workflow). This information is essential if we want to share our workflows with other people, because (only) the title of a workflow (information included inside the XML description) is usually insufficient.

Executing and Monitoring Workflows

The platform provides a web interface for searching workflows, which shows a short description of each workflow (so users have an initial description that could be useful for selecting a service). Once a workflow is selected from the list, the next step is to define the inputs of the workflow in order to execute it. When a user inserts or selects the input/s, the system stores information about the workflow that is going to be executed. Thus the inputs are stored for execution and future use. In addition, a set of internal tasks are created (one for each processor), and related processors imply that the corresponding tasks have as input the output of the predecessor. Relationships between the designed tables are quite similar to the relationships between the tables for storing the workflows, though the sources include a field for storing the value inserted by the user.

The created tasks (that are related to the processors by means of the `IdTask` field) are executed by the system scheduler, which does not execute a task until the required input has been created (the objects have a state that indicates whether an object has been created or not). Thus, the synchronization between processors is ensured.

The workflows, which a user has executed, can be monitored by means of a web interface that shows the execution of the workflow with automatic refreshing of the information of the web page. This monitoring tool also includes all the created tasks, their state: Waiting, Finished or Error, and their outputs. Thus, the partial and final results can be examined in order to extract biological conclusions to the executed workflows.

3 Conclusions

The work described in this paper has been developed and implemented in the National Institute for Bioinformatics (INB) in Spain, and the INB platform is available at <http://www.inab.org/MOWServ>. The INB system architecture is organized at three main levels, with the user minimally armed with a Web-browser, and demanding services to process their collection of biological data: (a) a web-interface at the top of the architecture facilitates communication between the user and the platform (b) the architecture core including services' interface though bioMOBY API; and (c) at the

bottom of the scheme the services' providers. A web interface manages user sessions with an authentication mechanism. An automatic web interface builder is able to dynamically build on interfaces for browsing data objects, services and namespaces (associated with data containers).

At internal level, once a service has been requested, the system provides notification about the progress status of services, -including historical records of executed tasks-, together with the relationships between input data, applied services and output data. Frequently output data becomes the input for new services. The GUI provides a specific list of suitable services that can be applied.

In summary we report a client engine, which is based on semantic interconnection concepts. The platform is able to integrate various processing services developed by different users and groups into workflows, through a web-based interface, thereby expanding the functionality of current services and enabling the easy incorporation of new procedures to customize the system for specific concerns.

The support for loading, executing and monitoring workflows is based on a very common and well-defined representation language. Our proposal extends this tool with the capability of using authentication-based systems, in which the confidentiality of the data is ensured. In addition the use of a scheduler which has statistics about the services that are stored in the system enables a better execution process. Another important advantage of this system is that data obtained by means of the execution of a workflow can be used to execute other services or even workflows.

Acknowledgements

This work has been partially supported by grant "GNV5-Bioinformática Integrada" from Genoma-España and the Spanish MEC Grant TIN 2005-09098-C05-01. We would also like to thank all the other nodes in the INB infrastructure for their heavy involvement in the development of services and integration procedures.

Reference

1. Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M. Carver, T., Glover, K., Pocock, M.R., Wipat, A., Li, P. (2004) Taverna: A tool for the composition and enactment of bioinformatics workflows *Bioinformatics Journal* 20(17) pp 3045-3054, 2004