Generic Portlets for Bioinformatics in a Grid Infrastructure
Sandra Gesing and Oliver Kohlbacher
Simulation of Biological Systems, Center for Bioinformatics, Eberhard Karls University Tübingen, Germany
gesing@informatik.uni-tuebingen.de

Summary

Bioinformatics tools often require advanced computing skills and specific resources. To provide users without the necessary computational background with an intuitive user interface for different applications, we are developing a workflow-enabled portal for bioinformatics in a grid infrastructure. Our portlets minimize user interaction on technical adjustments and the users are able to concentrate on their specific research.

Motivation

An enormous number of complex and sophisticated algorithms and tools has been developed to aid the research in the area of bioinformatics. These tools often require specific computational resources and advanced computing skills for installation, administration, and use. However, scientists from disciplines like biology, biochemistry, and biomedicine want to focus on their specific research, but do not want to deal with unpleasant details of software installation, usability, and hardware configuration.

To hide complexity from the user we are developing a workflow-enabled portal for bioinformatics in a grid infrastructure. This portal is based on the open source portal software GridSphere[1] and workflows are built with WS-BPEL[2].

Basics

Portlet
A portlet is a Java program that extends the features of a portalservlet, comparable with a servlet and a server.

WSDL[3]
The Web Services Description Language defines an XML standard for the description of webservices for operations and exchange of messages.

SOAP
The Simple Object Access Protocol “is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment”[4]. A SOAP message is also XML-based and can be generated automatically from a WSDL file.

WS-BPEL
The Web Services Business Process Execution Language is a commonly used XML standard for workflow design.

Various tools have been developed for invoking webservices via HTTP using the corresponding WSDL file as basis to generate SOAP, e.g. SOAPUI [5]. By our state of knowledge the existing open-source programs support the user with an XML file for the request and deliver accordingly an XML file with the response. The user has to know how to interpret these files. Our portlets will also use the WSDL files but will generate an intuitive mask for the input of a webservice and an output mask containing the response of a webservice.

BPEL Generator Portlet

To minimize user interaction on deriving a BPEL file from a WSDL file, we will implement a BPEL generator as portlet.

This portlet will re-use the validation of WSDL files from the Generic Mask Portal. Afterwards it will generate a BPEL file exclusively with the informations given in the validated WSDL file. The resulting BPEL file describes a structural very simple workflow. The workflow consists of invoking this one webservice. We decided to embed the open source ActiveBPEL engine[7] as workflow engine in our portal.

Fig.1 Architecture Overview

Fig.2 Functionality of the Generic Mask Portlet

Fig.3 Input mask for the example SVMHCService[6] with the corresponding parts of the WSDL file

Fig.4 Output mask for the example SVMHCService with the corresponding parts of the WSDL file for the sequence “SYFPEITHI” as input

Fig.5 BPEL file for invoking SVMHCService

Before the BPEL and the WSDL file can be deployed into the workflow engine for enactment, ActiveBPEL expects to additionally get a PDD file (Process Deployment Descriptor) and a wsdlCatalog file. Both can be created from the information in the WSDL file and the generated information in the BPEL file.

References

[3] WSDL specification http://www.w3.org/TR/wsdL