BPEL4Pegasus: Combining Business and Scientific Workflows

Mirko Sonntag[1], Dimka Karastoyanova[1] and Ewa Deelman[2]

[1] Institute of Architecture of Application Systems, University of Stuttgart
Universitaetsstrasse 38
70569 Stuttgart, Germany
{sonntag, karastoyanova}@iaas.uni-stuttgart.de

[2] Information Science Institutes, University of Southern California
Admiralty Way 4676
90292 Marina Del Rey, California, USA
deelman@isi.edu

Abstract. Business and scientific workflow management systems (WfMS) offer different features to their users because they are developed for different application areas with different requirements. Research is currently being done to extend business WfMSs by functionality that meets requirements of scientists and scientific applications. The idea is to bring the strengths of business WfMSs to e-Science. This means great effort in re-implementing features already offered by scientific WfMSs. In our work, we investigated another approach, namely combining business and scientific workflows and thus harnessing the advantages of both. We demonstrate a prototype that implements this idea with BPEL as business workflow language and Pegasus as scientific WfMS. Our motivation is the fact that the manual work to correctly install and configure Pegasus can be supervised by a BPEL workflow to minimize sources of failures and automate the overall process of scientific experimenting.

Keywords: Scientific workflows, business workflows, human tasks, Pegasus, BPEL.

1 System Overview

The main idea of the work is allowing a business workflow to orchestrate and supervise one or more scientific workflows that in combination represent a whole scientific experiment with all its stages. Fig. 1 shows how this concept is implemented in the prototype to demonstrate. Note that the dotted lines represent possible machine borders. Pegasus relies on Condor DAGMan and Condor to execute scientific workflows on Grids. A BPEL engine (we used the open source engine Apache ODE 2.0, `http://ode.apache.org/`) runs a business process that supervises the Pegasus workflow's planning and execution phase. In order to invoke Pegasus by BPEL we created a Web service (WS) that wraps and offers Pegasus' functionality. Apache ODE currently does not implement a direct support for human activities and hence there is no human task (HT) client. That is the reason why we developed a Web application as GUI for the users. It is a HT client and monitoring tool in one, and HTs are called by BPEL in terms of WS calls. Since scientific workflows executed by Pegasus can be long-running, we follow an asynchronous communication model between the GUI, BPEL engine and Pegasus.

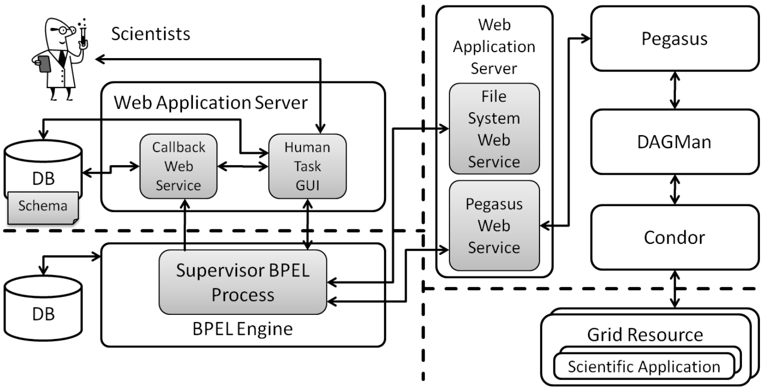

Fig. 1. Overview of the system. Gray ingredients were developed as part of this work.

Contributions of the prototype can be summarized as follows: (1) the correlation between BPEL workflow instances and Pegasus workflows; (2) providing Pegasus' original interface that is a mix of shell commands and file operations as WS; (3) a GUI for preparing and executing Pegasus workflows; and (4) GUI support for users to setup Pegasus' site, transformation and replica catalogs. These contributions enormously simplify the work of scientists since it is more straightforward to use and lowers the learning curve, automates manual tasks and supports decision making.

2 Functions to be Demonstrated

The demonstration has two parts: (1) All aspects of the GUI are shown: The *work list* containing all work items assigned to a user, the *monitor* that shows the current state of workflow instances, and the *view* to select a Pegasus server, and to choose and start a *Pegasus workflow*. (2) Pegasus workflow execution: It is shown how a user can follow the progress of scientific experiments with the help of the monitor, how work items are assigned to users, and how they can handle these work items. Moreover, we present how a user is assisted during the configuration of Pegasus catalogs and how he can react to failures in the Pegasus planning phase and can ensure convergence of result files. A video of the demonstration can be found here: `http://www.iaas.uni-stuttgart.de/institut/mitarbeiter/sonntag/indexE.php`.

Acknowledgments. The authors M.S. and D.K. would like to thank the German Research Foundation (DFG) for financial support of the project within the Cluster of Excellence in Simulation Technology (EXC 310/1) at the University of Stuttgart.