

BPEL Engines

State-of-the-Art Survey for SOAMeS-project

Jenni Kannianen & Jyrki Haajanen



ISBN 978-951-38-6620-4 (URL: <http://www.vtt.fi/publications/index.jsp>)
ISSN 1459-7683 (URL: <http://www.vtt.fi/publications/index.jsp>)

Copyright © VTT 2007

JULKAISIJA – UTGIVARE – PUBLISHER

VTT, Vuorimiehentie 3, PL 2000, 02044 VTT
puh. vaihde 020 722 111, faksi 020 722 4374

VTT, Bergsmansvägen 3, PB 2000, 02044 VTT
tel. växel 020 722 111, fax 020 722 4374

VTT Technical Research Centre of Finland, Vuorimiehentie 5, P.O.Box 1000, FI-02044 VTT, Finland
phone internat. +358 20 722 111, fax +358 20 722 4374

VTT, Vuorimiehentie 3, PL 1000, 02044 VTT
puh. vaihde 020 722 111, faksi 020 722 7024

VTT, Bergsmansvägen 3, PB 1000, 02044 VTT
tel. växel 020 722 111, fax 020 722 7024

VTT Technical Research Centre of Finland, Vuorimiehentie 3, P.O. Box 1000, FI-02044 VTT, Finland
phone internat. +358 20 722 111, fax +358 20 722 7024



Series title, number and
report code of publication

VTT Working Papers 69
VTT-WORK-69

Author(s) Kanniainen, Jenni & Haajanen, Jyrki		
Title BPEL Engines State-of-the-Art Survey for SOAMeS-project		
Abstract This report summarizes the results of a hands-on survey on Business Process Execution Language (BPEL) execution engines. BPEL is one of the most promising new languages intended for automatic business process execution. BPEL supports implementation of both intra- and extra enterprise processes. The form of BPEL most often associated with the Service Oriented Architecture (SOA) is extension called WS-BPEL (Web Services BPEL, formerly aka BPEL4WS). In this research we installed several competing commercial and non-commercial execution platforms and installed and executed a predefined business process on them. Following areas were evaluated: installation, process creation, and process activation. The results showed that the commercial engines are on quite similar level in terms of these criteria. The range of quality and applicability was much wider in the non-commercial engines. However, best of the breed non-commercial engines are a serious option when selecting engine for professional purposes.		
ISBN 978-951-38-6620-4 (URL: http://www.vtt.fi/publications/index.jsp)		
Series title and ISSN VTT Working Papers 1459-7683 (URL: http://www.vtt.fi/publications/index.jsp)		Project number
Date January 2007	Language English	Pages 30 p.
Name of project Service Oriented Architecture in Multichannel e-Services (SOAMeS)	Commissioned by Tekes - Finnish Funding Agency for Technology and Innovation, VTT, Elisa Oyj, Kesko Oyj, Metsäteho Oy, Tietoenator Processing & Network Oy	
Keywords Service Oriented Architecture, electronics services, e-services, multichannel services, automatic business processes, process execution, execution platforms, installation, process creation, process activation	Publisher VTT P.O. Box 1000, FI-02044 VTT, Finland Phone internat. +358 20 722 4404 Fax +358 20 722 4374	

Preface

This research was conducted in SOAMeS (Service Oriented Architecture in Multichannel e-Services). The project is funded by Finnish Funding Agency for Technology and Innovation, VTT, Elisa Oyj, Kesko Oyj, Metsäteho Oy, and TietoEnator Processing & Network Oy.

The project goal is to study the potential and applicability of SOA based business networking both from technical and business perspectives.

This report is one of the technology surveys performed within the project.

Contents

Preface	4
1. Introduction.....	6
2. BPEL.....	7
2.1 What is BPEL?	7
2.2 BPEL Engine.....	8
3. Test Case.....	9
4. Open Source Engines.....	12
4.1 Active Endpoints: ActiveBPEL.....	12
4.1.1 Installation.....	12
4.1.2 Process Development.....	12
4.1.3 Process Deployment.....	12
4.1.4 Process Instantiation.....	14
4.2 Apache: Agila.....	15
4.2.1 Installation.....	15
4.3 Apache: Ode.....	18
4.3.1 Installation.....	18
4.4 Eclipse BPEL Project.....	20
5. Commercial Engines.....	21
5.1 Oracle: BPEL Process Manager 10.1.2.....	21
5.1.1 Installation.....	21
5.1.2 Process Development.....	21
5.1.3 Process Deployment.....	22
5.1.4 Process Instantiation.....	23
5.2 Parasoft: BPEL Maestro 2.2.....	24
5.2.1 Installation.....	25
5.2.2 Process Development.....	25
5.2.3 Process Deployment.....	26
5.2.4 Process Instantiation.....	28
5.3 Other Commercial Engines.....	28
6. Conclusion.....	29
7. Summary.....	30

1. Introduction

There are many BPEL engines in the market these days, both open source and commercial. In this paper we represent the results of our state of the art survey.

We selected three open source and two commercial engines for testing. The selected open source engines were Active Endpoints Active BPEL, Apaches Agila and Apaches Ode. The selected commercial engines were Oracles BPEL Process Manager and Parasofts BPEL Maestro.

First we defined a test case that was used for testing the engines. The defined process was very simple, it only had sequential execution. It was used to test the basic features of the engines.

First we examined how easy it was to install the engines. Then the test process was developed and deployed to the engine. Finally the process was instantiated with a web service client or with the engines own testing environment.

2. BPEL

In April 2003, BEA Systems, IBM, Microsoft, SAP and Siebel Systems submitted BPEL4WS 1.1 (<ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf>) to OASIS for standardization. OASIS WS-BPEL technical committee voted on 14 September 2004 to name their spec WS-BPEL 2.0 (<http://docs.oasis-open.org/wsbpel/2.0/wsbpel-specification-draft.pdf>). This change in name and version number reflects the significant and in many cases incompatible differences between BPEL4WS 1.1 and WS-BPEL 2.0. The WS-BPEL 2.0 Specification entered Public Review on 10 September 2006. Public Review for WS-BPEL 2.0 ends 09 November 2006.

2.1 What is BPEL?

BPEL offers a language for defining how to send XML messages to remote services, manipulate XML data structures, receive XML messages synchronously/asynchronously from remote services, manage events and exceptions, define parallel sequences of execution and undo parts of processes when exceptions occur. A process defines a new Web Service by composing a set of existing services. The interface of the composite service is described with WSDL, just like any other Web Service. There are many implementations in the market, commercial and open source.

BPEL builds on top of WSDL. It uses WSDL for describing services involved in the process, messages being exchanged, operations being invoked and the portTypes these operations belong to.

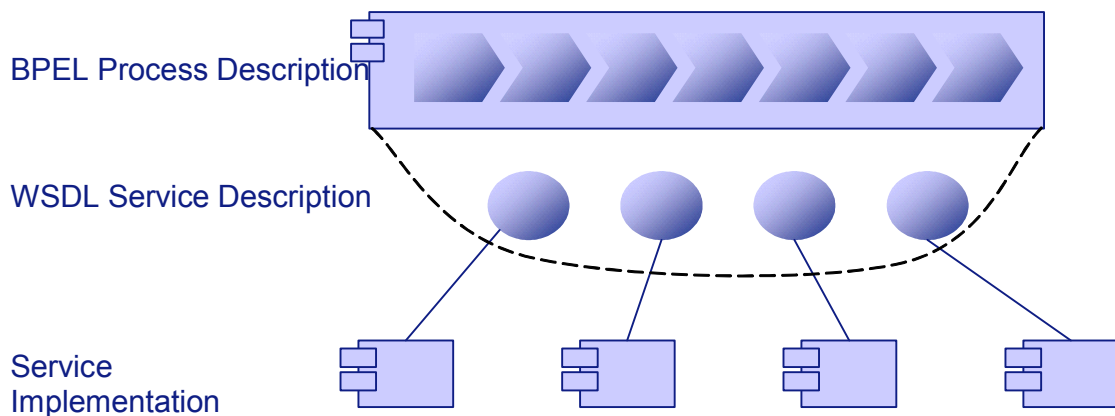


Figure 1. BPEL Process.

2.2 BPEL Engine

BPEL process descriptions are published to a BPEL Engine. Engine executes the processes. Process can be invoked as a Web Service (input-output messages). Some engines have their own testing environment (test client).

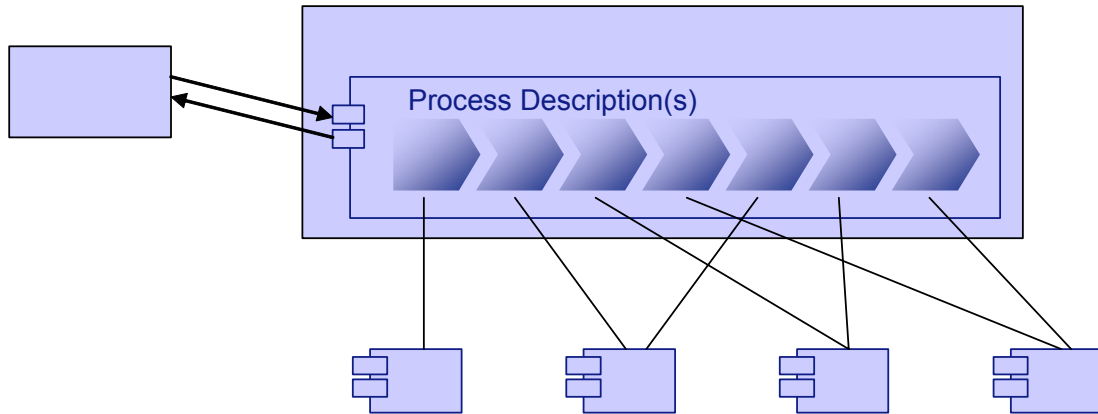


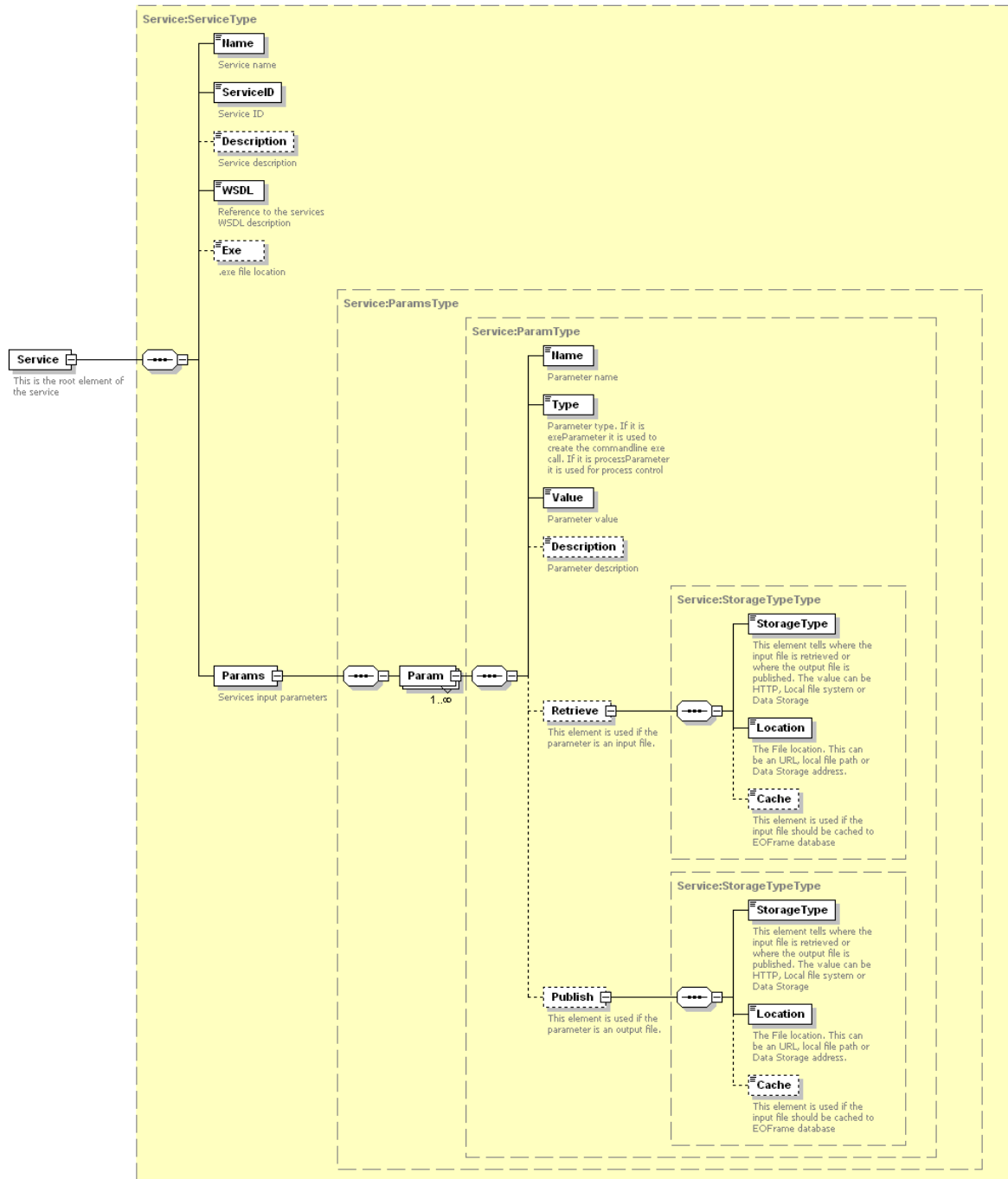
Figure 2. BPEL Engine.

3. Test Case

A test case was defined to test the engines. The defined process was very simple. It only had sequential execution.

GenericService is an asynchrony Web Service that can execute any exe with given parameters. The exe files must be located at local or network drive where the service can access them. A Service Schema (Figure 3) was developed to define the services input message.

The BPEL Process (Figure 4) calls Generic Service to execute four exe files in sequence: Unpack.exe, radiomCalib.exe, thermaCalib.exe and geomCorrect.exe.



Generated with XMLSpy Schema Editor www.xmlspy.com

Figure 3. Service Schema.

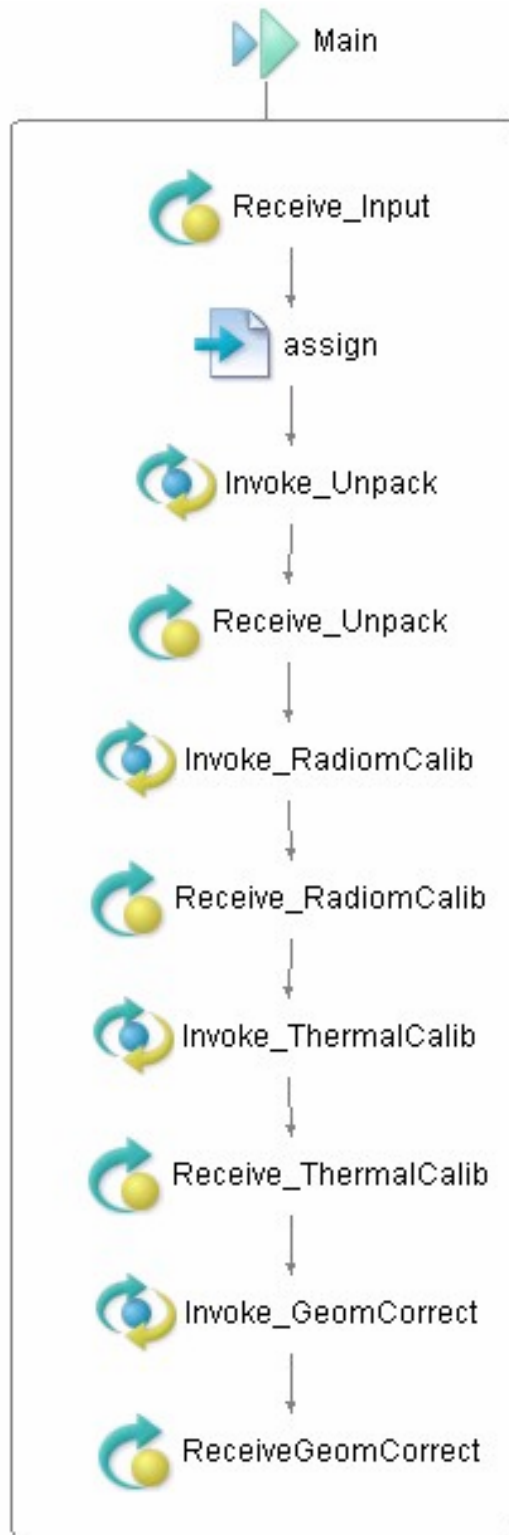


Figure 4. BPEL Process.

4. Open Source Engines

Three open source engines were selected for testing. These were Active Endpoints Active BPEL, Apaches Agila and Apaches Ode.

4.1 Active Endpoints: ActiveBPEL

The ActiveBPEL engine is a commercial-grade open source implementation of the BPEL4WS Version 1.1 specification, and is fully compliant with that spec. Active Endpoints also have a commercial engine called Active BPEL Enterprise.

ActiveBPEL can be found at: <http://www.activebpel.org>.

4.1.1 Installation

The ActiveBPEL engine runs within a web/application server. The current version has been tested and runs with Tomcat 5.X.

Download: <http://www.activebpel.org/download/index.html>.

Installation guide: <http://www.activebpel.org/docs/install.html>.

Run the script `install.bat` (Windows) or `install.sh` (Unix). It copies the contents of `lib` into `$CATALINA_HOME/shared/lib` and creates the directory `$CATALINA_HOME/bpr`, where BPEL process `.bpr` archives are deployed.

The Installation was easy and straight forward.

4.1.2 Process Development

Free BPEL Designer:

<http://www.active-endpoints.com/products/activebpeldes/index.html>.

4.1.3 Process Deployment

Deployment guide: <http://www.activebpel.org/docs/deploy.html>.

Deploying a BPEL process involves creating a deployment archive file (a JAR with an extension of ".bpr") and copying that to your servlet container. To create this archive, you need to organize your files into a particular directory structure, create one or two configuration files, and then create an archive from that directory.

The BPEL OnDemand Validation service checks your BPEL process definitions and their associated WSDL files for conformance to the BPEL4WS 1.1 specification. The service validates your BPEL and WSDL files by performing static BPEL analysis.

<http://www.activebpel.org/code/validator/>.

List of deployed services: <http://localhost:8080/active-bpel/services>.

BPEL Administrative Console: <http://localhost:8080/BpelAdmin/>.

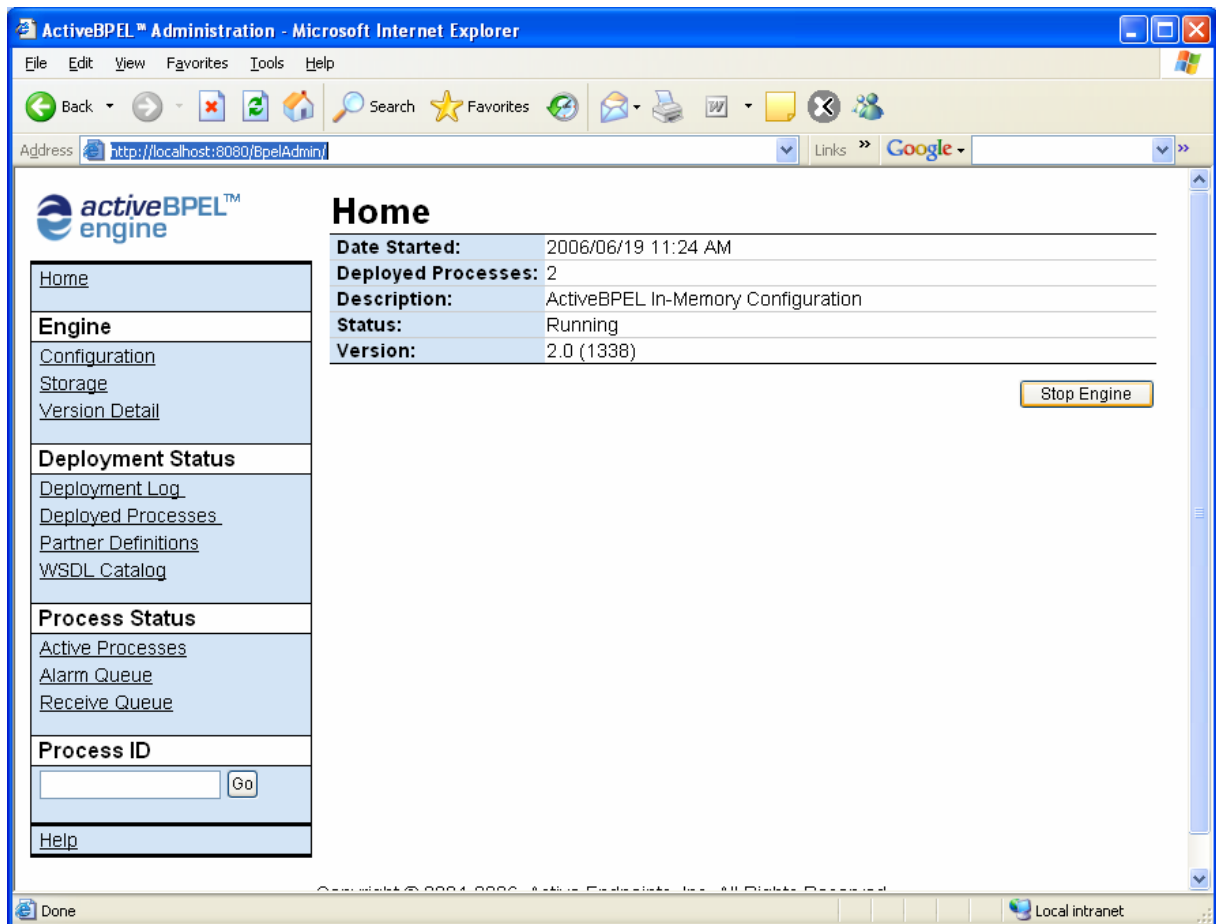


Figure 5. BPEL Administrative Console.

4.1.4 Process Instantiation

SOAP Client: http://localhost:8080/soap_client/index.jsp.

Sends a SOAP message wrapping the text in the “Message Content” field to the specified URL.

You can modify the URL to reflect any available process or service. For example, replace "[service_name]" with the (case-sensitive) name of a deployed process or service.

(Note: The default service name for a process deployed using ActiveBPEL Designer is the processes partner link name appended with “Service”.) Then type or paste a message into the “Message Content” field and click the “Invoke Process /Service” button. Any response appears in the bottom text area.

Process can also be instantiated from a Client as a normal Web Service.

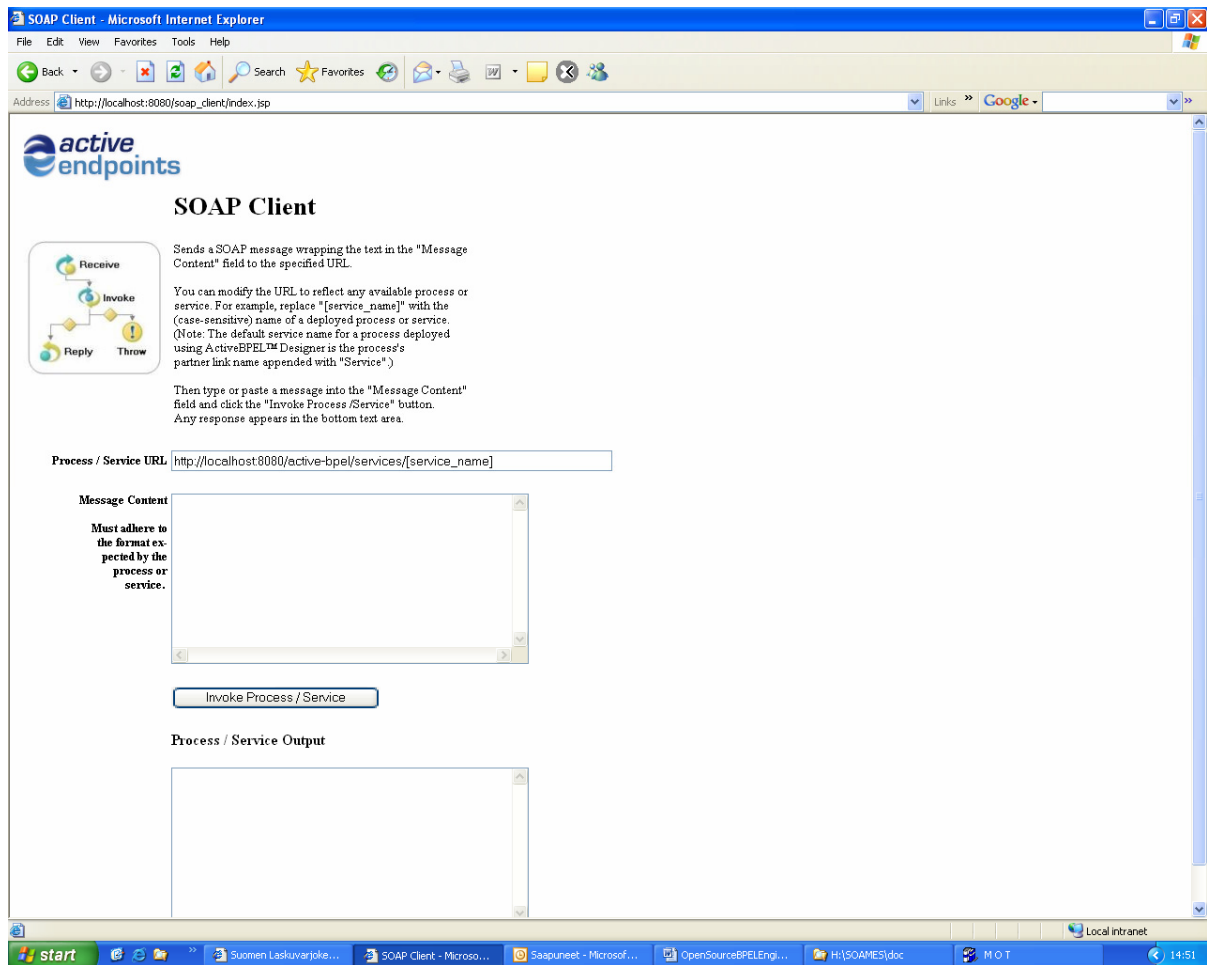


Figure 6. SOAP Client.

We were unable to instantiate the process from our own test client. We had a namespace problem that we could not solve. We could work around this problem by catching the messages with a tcp monitor, correcting the message by hand and then sending it back to the process.

The correct message was:

```
<callback xmlns="http://eoframe.vtt.fi/service">
```

```
.....
```

```
</callback>
```

Our client send message:

```
<callback xmlns:ns1="http://eoframe.vtt.fi/service">
```

```
.....
```

```
</callback>
```

4.2 Apache: Agila

Agila is composed of two modules: Agila BPM and Agila BPEL. Agila BPM is basically handling tasks and users who have to complete these tasks. Agila BPEL is a BPEL-compliant Web Services Orchestration solution.

Agila has been accepted by the Apache Jakarta PMC for inclusion into the Jakarta project. Once the requirements for incubation are complete, the community and code will move to Jakarta.

4.2.1 Installation

Installation guide: <http://wiki.apache.org/agila/AgilaBpelInstallationGuide>.

Agila BPEL is working with the latest versions of Tomcat 4.1 but using Tomcat 5 is recommended. Installing Agila BPEL in another web container shouldn't be a problem either.

The other thing you will need is a relational database installation. For testing and playing around, Agila BPEL is shipped with HSQL-DB, an open source relational database written in Java but even if HSQL is a pretty fast database it doesn't scale very

well so it wouldn't recommend using it in production. Agila BPEL can be integrated with a large variety of databases like MySQL, Oracle, DB2 or Sybase.

Third thing that is needed is an XML database. Agila BPEL is well integrated with Apache Xindice.

Right now Agila BPEL hasn't been released yet, so the only way to obtain a distribution is to build it yourself.

To obtain Agila BPEL, you have to checkout the sources from our Subversion repository. Install a Subversion client:

```
http://subversion.tigris.org/servlets/ProjectDocumentList?folderID=91.
```

Go to subversion/bin and run:

```
svn checkout -r 365970 http://svn.apache.org/repos/asf/incubator/agila.
```

If you get error:

If you are behind a corporate firewall which is blocking http access to the Subversion repository, you can try to access it via HTTPS:

```
svn checkout https://svn.apache.org/repos/asf/incubator/ode/trunk ode.
```

```
Install Maven 1.0.2: http://maven.apache.org/maven-1.x/start/install.html.
```

```
Download Maven 1.0.2: http://maven.apache.org/maven-1.x/start/download.html.
```

```
Install XDoclet: http://xdoclet.sourceforge.net/xdoclet/install.html.
```

Extract XDoclet's distribution and copy maven-xdoclet-plugin-1.2.1.jar located in the lib directory in Maven's plugins directory.

Create a file named build.properties in your user's home directory (like /home/username or C:\Documents and Settings\username). Copy and paste the following line into this file:

```
maven.repo.remote=http://www.ibiblio.org/maven,http://www.smartcomps.org/twister/maven-repo.
```

Add also the following lines if you are behind a firewall:

```
maven.proxy.host=rohto.vtt.fi.  
maven.proxy.port=8000.
```


Using a console go to Agila BPEL source distribution root (should be something like agila/trunk/modules/bpel) and type:

```
maven buildall.
```

If you get error opening zip file while compiling the zip file is corrupted.

The only alternative is to download the JARs yourself, and to place them into your file system.

I had to download and copy:

```
jboss-aop-1.0b2.jar, jboss-common-aop-1.0b2.jar and javassist-aop-1.0b2.jar to  
C:\Documents and Settings\username\.maven\repository\jboss\jars.
```

```
jta-1.0.1.jar to C:\Documents and Settings\username\.maven\repository\jta\jars.
```

```
xindice-1.1b4.jar to C:\Documents and  
Settings\username\.maven\repository\xindice\jars.
```

```
jcs-1.0-dev.jar to C:\Documents and Settings\username\.maven\repository\jcs\jars.
```

This will build all Agila BPEL modules. The results of the build are placed in the target directory of each module (usually a jar or a war). The two necessary Agila BPEL web applications that you deploy in Tomcat are in client/web/target and engine/wsa/target.

Before deploying you'll have to download Xindice 1.1b4 to deploy it in Tomcat as well.

```
http://xml.apache.org/xindice/download.cgi.
```

Copy the two war files obtained after compilation to tomcat/webapps.

Copy hsqldb-x.x.jar to tomcat/common/lib. Note: Use the hsqldb-1.7.2.2.jar (from C:\Documents and Settings\tteman\.maven\repository\hsqldb\jars).

Copy engine/wsa/src/test/resources/conf/hibernate.cfg.xml and agila-configuration.xml to Tomcat/common/classes.

Copy Xindice war file included in Xindice's distribution in tomcat/webapps (make sure the file name is 'xindice.war').

Using HSQL

Agila BPEL configuration is using HSQL by default so there is nothing special to setup. The only thing you need to do is starting HSQL and initialize the database. Two Maven goals have been defined in the dist sub module. The first one starts an HSQL instance and the second initializes the database with the right schema. To use these two goals just go to the modules/bpel/dist directory and run:

```
maven hsql:start  
maven db:create
```

Run Tomcat and go to <http://localhost:8080/agila-web>. You should be able to login with admin/admin.

We had an installation problem with this engine. We could see the login page at <http://localhost:8080/agila-web> but could not log in.

4.3 Apache: Ode

Ode (Orchestration Director Engine) is an incubation project to develop an open-source, Apache-licensed, implementation of the WS-BPEL 1.1 and WS-BPEL 2.0 specifications.

4.3.1 Installation

Installation guide: <http://wiki.apache.org/ode/Building>

Get source code using subversion:

```
svn checkout http://svn.apache.org/repos/asf/incubator/ode/trunk ode.
```

If you are behind a corporate firewall which is blocking http access to the Subversion repository, you can try to access it via HTTPS:

```
svn checkout https://svn.apache.org/repos/asf/incubator/ode/trunk ode.
```

Install Maven2: <http://maven.apache.org/download.html#Installation>.

If you are working in a corporate network and behind a firewall this simple step of configuring maven to use the corporate proxy will save you a lot of time and aggravation.

Edit your maven configuration file settings.xml under maven-2.0.x/conf

Now we need to configure a proxy. In the settings.xml file find the proxies element and add you proxy information under it:

```
<proxy>

  <id>optional</id>

  <active>true</active>

  <protocol>http</protocol>

  <username/>

  <password/>

  <host>rohto.vtt.fi</host>

  <port>8000</port>

  <nonProxyHosts>*.vtt.fi</nonProxyHosts>

</proxy>
```

Open a command in the source root usually in ode/trunk (I did this in /ode) and type:

```
mvn install
```

After everything has been compiled, you should find a WAR (I had ode-axis2-war-2.0-SNAPSHOT.war) file in the axis2-war/target directory. Rename it to ode.war and copy this file to Tomcat's webapp directory. Start Tomcat and Ode should be up and running. You should get the Axis2 welcome page under <http://localhost:8080/ode>

Ode installation was successful but there was a problem deploying a process. At the moment the engine has very little documentation.

4.4 Eclipse BPEL Project

The goal of the BPEL Project is to add comprehensive support to Eclipse for the definition, authoring, editing, deploying, testing and debugging of WS-BPEL 2.0 processes. Some code has been released to the BPEL project. This code represents the initial state of the BPEL editor and model. This contribution doesn't include validation, samples, docs, nor any runtime framework or debug code.

Installation guide: <http://www.eclipse.org/bpel/install.php>

We did not conduct any tests with Eclipse, but we think this is a project that should be followed in the future.

5. Commercial Engines

Two commercial engines were selected for testing. These were Oracles BPEL Process Manager and Parasofts BPEL Maestro.

5.1 Oracle: BPEL Process Manager 10.1.2

Oracle BPEL Process manager can be found at <http://www.oracle.com/technology/products/ias/bpel/index.html>. Oracle offers 30 day evaluation license.

5.1.1 Installation

Download from <http://www.oracle.com/technology/products/ias/bpel/index.html> BPEL Process Manager 10.1.2 and 10.1.2 Patches.

Install BPEL Process Manager 10.1.2.

When the installer asks *What kind of installation do you want* select BPEL Process Manager for Developers.

Install patches 4369818, 4406640, 4496111 and 4696706).

Installation guide can be found from README.txt files.

Go to `oc4j_home\config "orabpel\system\appserver\oc4j\j2ee\home\config"`, in `server.xml` increase the default: `<transaction-config timeout="60000" />` value to at least 12000000.

5.1.2 Process Development

Oracle offers Oracle JDeveloper 10g for process development.

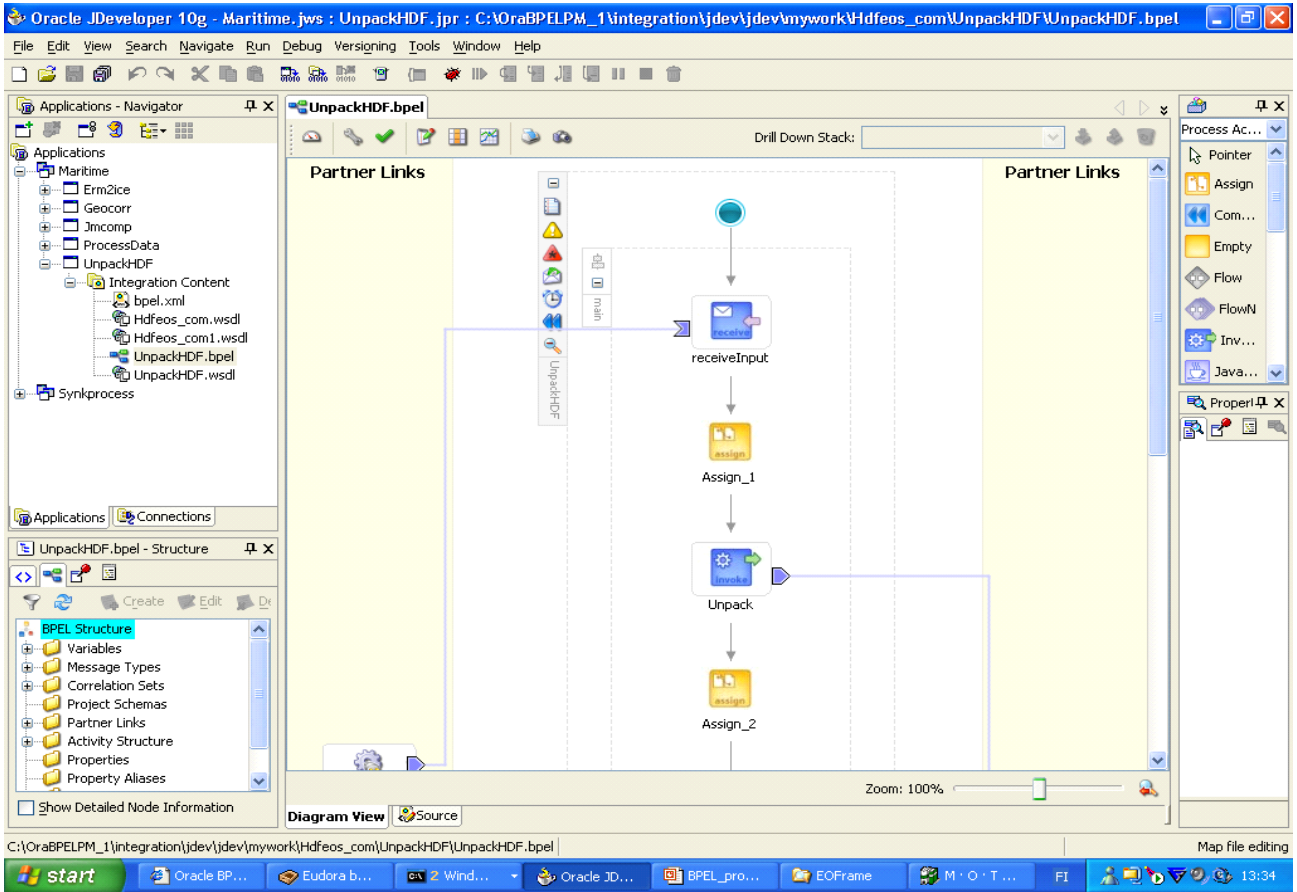


Figure 7. Oracle JDeveloper 10g.

5.1.3 Process Deployment

Go to the Oracle JDeveloper 10g.

Open the process you want to deploy.

Click on the Make icon.

Go to the BPEL Console <http://localhost:9700/BPELConsole>.

Click on the **Deploy New Process** link.

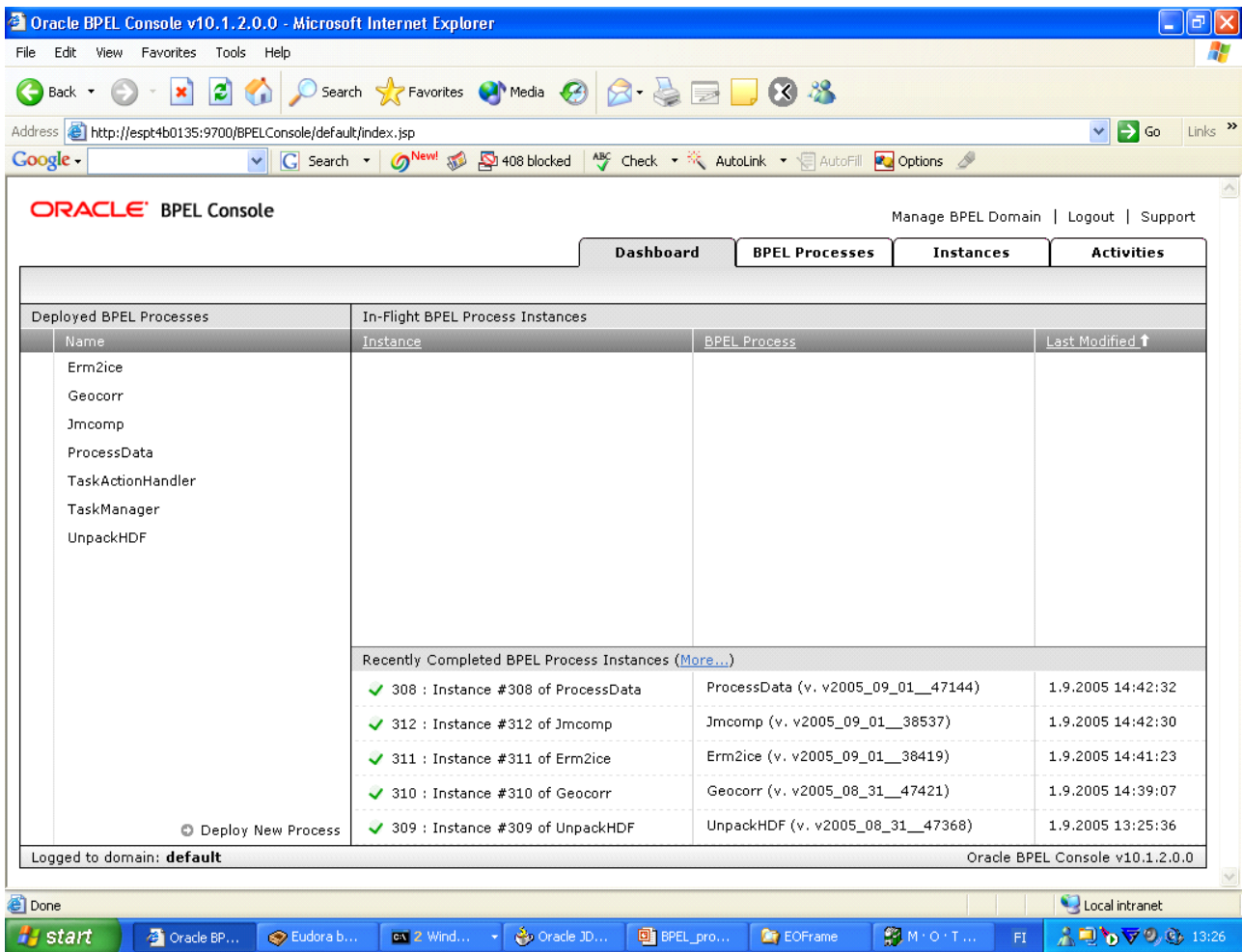


Figure 8. Oracle BPEL Console.

5.1.4 Process Instantiation

Go to the BPEL Console <http://localhost:9700/BPELConsole>.

Click on the process you want to initiate.

Select the Initiate tab.

Fill up the form and click on the **Post XML Message** button.

Active Processes are listed under the Instances tab.

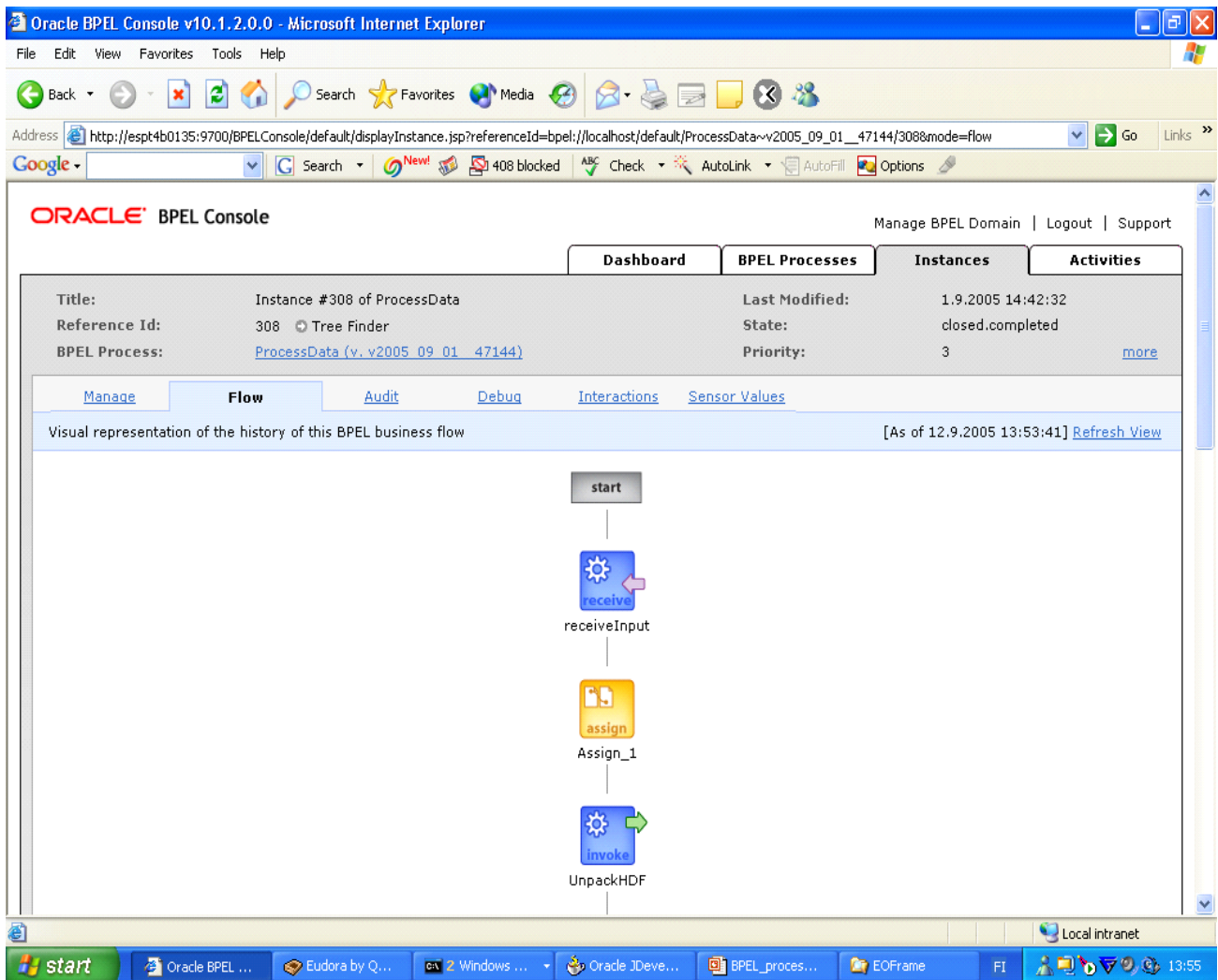


Figure 9. Oracle BPEL Console, Process Flow.

5.2 Parasoft: BPEL Maestro 2.2

Free trial can be found at:

<http://www.parasoft.com/jsp/products/home.jsp?product=BPEL>.

Develop and edit BPEL files and WSDL files.

Validate whether BPEL and WSDL files have valid and well-formed XML.

Validate whether BPEL files comply with the BPEL4WS specification.

Deploy BPEL processes to a test or production BPEL Maestro engine.

Debug deployed BPEL processes.

Test deployed BPEL processes.

Review, update or remove currently-deployed BPEL processes.

BPEL Maestro consists of three main parts. An Eclipse-based editor, a BPEL process engine, and an engine management module. It supports the BPEL4WS 1.1 (May 2003) version of the BPEL4WS specification.

5.2.1 Installation

Submit your contact information, and a Parasoft representative will contact you. http://www.parasoft.com/jsp/customers/customer_login.jsp?caller=%2Fjsp%2Ftrial_request.jsp%3FitemId%3D303.

Download BPEL Maestro and run the .exe file to install it to your computer. You'll need an evaluation license key from Parasoft.

5.2.2 Process Development

BPEL Maestro offers an Eclipse-based editor for process development.

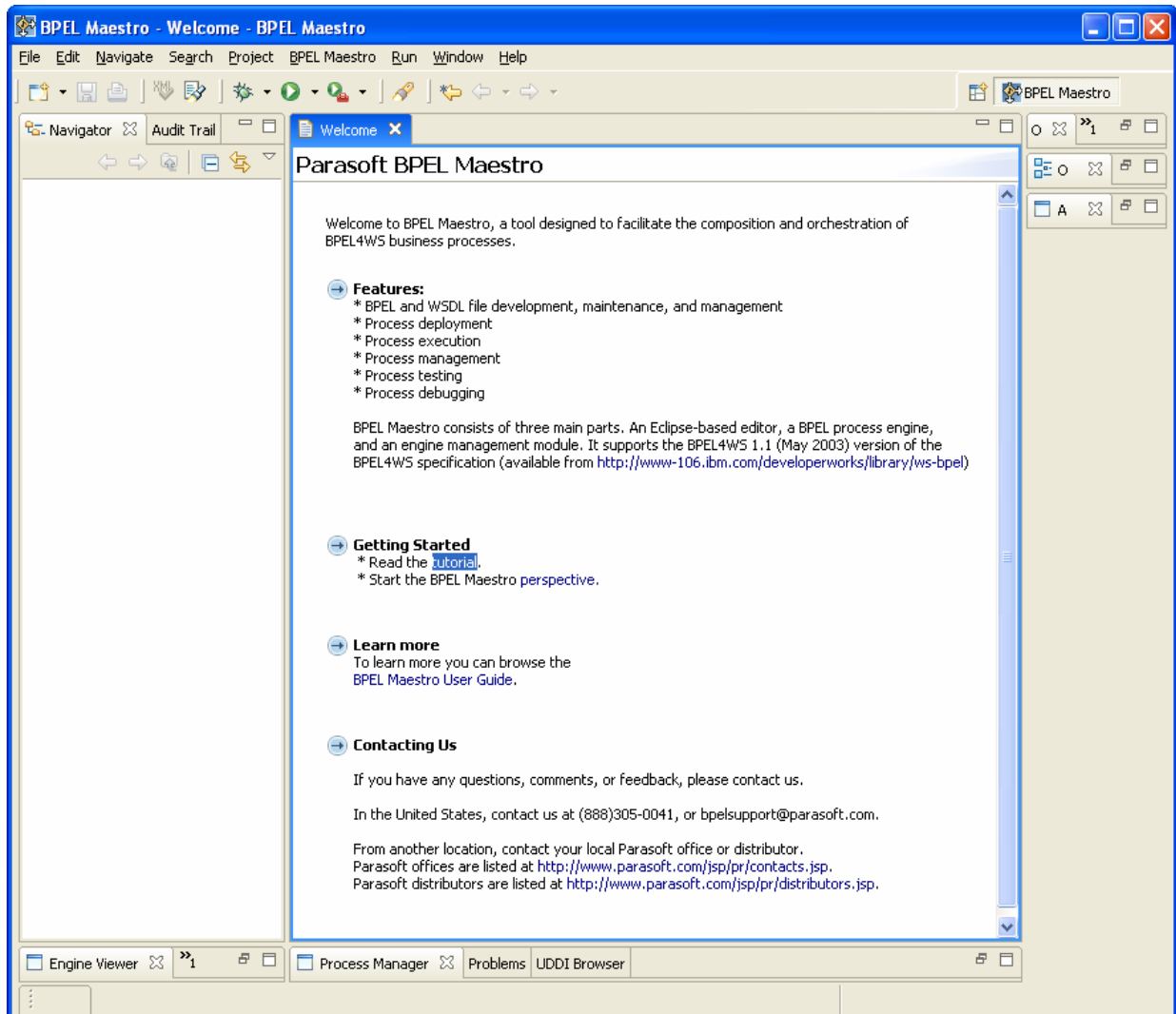


Figure 10. BPEL Maestro.

5.2.3 Process Deployment

To access the general engine entry page (Figure 11), browse to <http://localhost:8080/bpel/>. You can use BPEL Maestro Engine Web interface to check the status of all processes available on an engine and view statistics related to that engine's processes.

The table contains one entry for each process deployed on the engine with a version link. Clicking on an entry's link will display the following information for the related process:

- Process name
- Web Service Description
- Active processes
- Total processes
- Terminal faults.

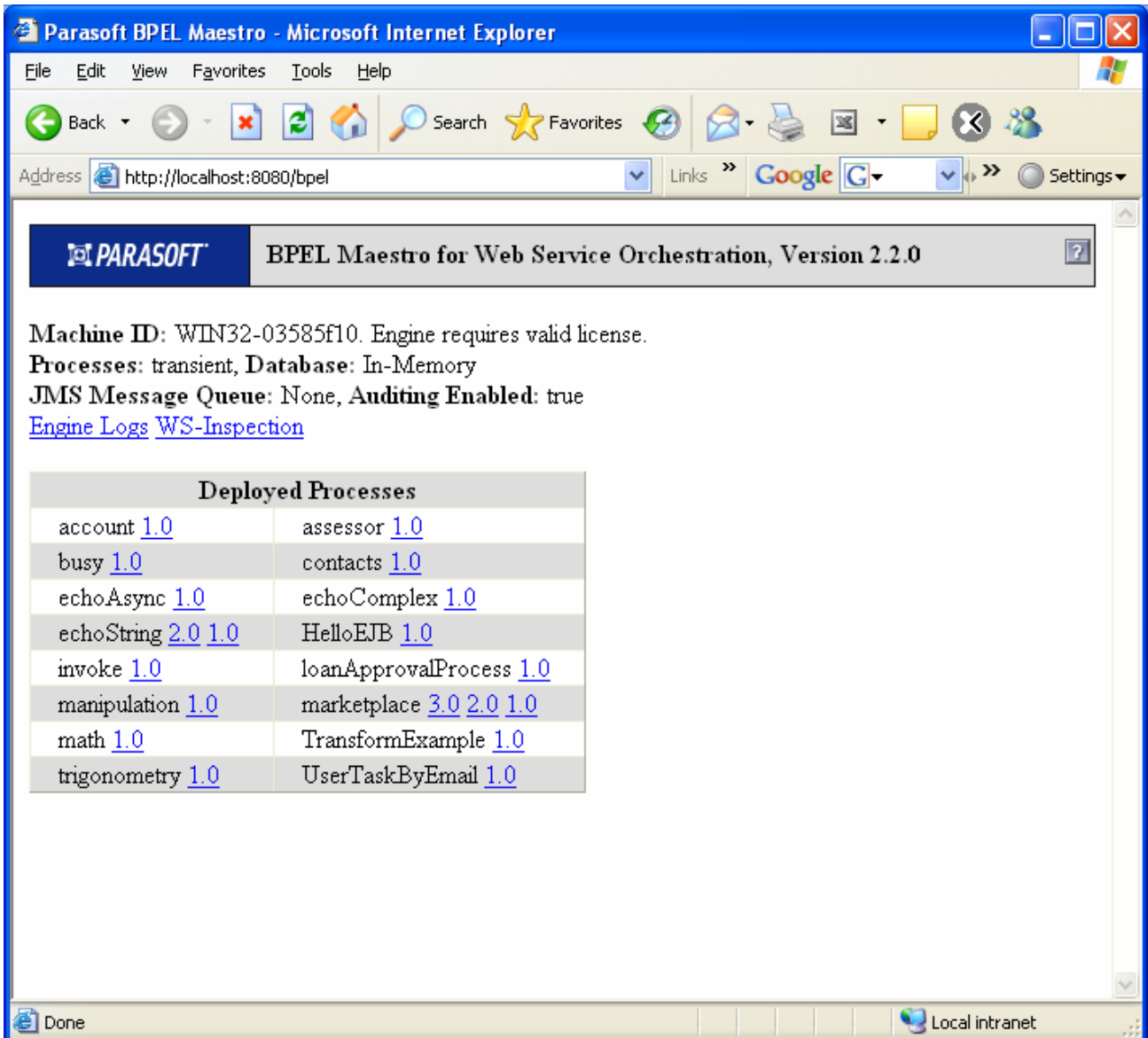


Figure 11. BPEL Maestro General Entry Page.

With the Maestro management system, you can remotely deploy from the client's Maestro management system with a single click. Management system provides BPEL process authoring, testing and debugging, and deployment.

To deploy a new project or update a previously-deployed project:

Right-click the Navigator node that represents the project you want to deploy, then choose BPEL Maestro> Deploy Project from the shortcut menu.

BPEL Maestro will then analyze the project files and check for problems (such as a missing partner link, a missing WSDL, etc.).

If no problems are found, the deployment will be performed automatically. You do not need to log into the server machine and manually edit deployment descriptors. By default, BPEL process files will be deployed to `http://localhost:8080/bpel/[your_process_name]` and public WSDLs will be deployed to `http://localhost:8080/bpel/[your_process_name]?WSDL`.

If problems are found, errors will be listed in the Tasks view at the bottom of the GUI workbench. You must fix these errors before you can successfully deploy the project.

5.2.4 Process Instantiation

One way to run a process is to send it a message through the Message Manager view. Another way to test processes is to have Parasoft's SOAtest tool automatically generate and execute a test suite for each BPEL process deployed on a Maestro engine. Another way to run a process is to run it through the debugger.

Processes can also be instantiated with a web service client. We didn't have time to create a client for testing, but we believe it would have been possible.

5.3 Other Commercial Engines

Table 1. Other commercial engines.

Name	Provider	URL
ActiveBPEL Enterprise	Active Endpoints	http://www.active-endpoints.com/products/activebpelent/index.html
Cape Clear Orchestrator	Cape Clear	http://www.capeclear.com/products/orchestrator.shtml
OpenStorm ChoreoServer	OpenStorm	http://www.openstorm.com/
BizTalk Server	Microsoft	http://www.microsoft.com/biztalk
WebShere Process Server	IBM	http://www-306.ibm.com/software/integration/wps/
SeeBeyond	Sun	http://www.seebeyond.com/
Ensemble	Intersystems	http://www.intersystems.com/

6. Conclusion

Active BPEL is the only open source engine advanced enough to be used in real life business case. It supports fully supports BPEL4WS Version 1.1 specification. There was no indication if Active BPEL will support BPEL 2.0 in the future. This may be considered as a potential huge downsize.

Apache has two different BPEL projects going on; Ode and Agila. Both need a lot of work before they are ready to be used in real life. It will be interesting to see if both projects will continue or is one going to be terminated. And if so which one?

Eclipse BPEL Project this is a project that should be followed in the future to see what they come up with.

All the commercial engines are able to perform the basic features of BPEL. We didn't find big differences between them. The basic features are the same: installing, developing and deploying the process is easy. With more advanced processes there might be more differences.

All engines need to have the process and description files to be packed to their own file structure for the process to be deployed. BPEL descriptions can be reused, but how the process is packed for the deployment (WSDL, Process description, other description files) differs a lot. If a BPEL process is reused in another engine it has to repacked to a correct file structure with the engines own editor or by hand. Usually The BPEL process and WSDL files can be reused but the description files are always engine specific.

7. Summary

The reported survey performed testing and evaluation of various commercial and non-commercial BPEL engines. Tests and evaluation targetted engine, installation, process creation and execution.

We found that generally the non-commercial solutions were not in very high-level, however, some environments were comparable to commercial solutions. in the commercial solutions there were no major differences in terms of the evaluation targets.

VTT Working papers

- 49 Saarinen, Jani, Rilla, Nina, Loikkanen, Torsti, Oksanen, Juha & Alasaarela, Jaakko. Innovation environment today and tomorrow. 2006. 32 p.
- 50 Törnqvist, Jouko & Talja, Asko. Suositus liikennetärinän arvioimiseksi maankäytön suunnittelussa. 2006. 46 s. + liitt. 33 s.
- 51 Aikio, Sanna, Grönqvist, Stina, Hakola, Liisa, Hurme, Eero, Jussila, Salme, Kaukoniemi, Otto-Ville, Kopola, Harri, Käsäkoski, Markku, Leinonen, Marika, Lippo, Sari, Mahlberg, Riitta, Peltonen, Soili, Qvintus-Leino, Pia, Rajamäki, Tiina, Ritschkoff, Anne-Christine, Smolander, Maria, Vartiainen, Jari, Viikari, Liisa & Vilkmann, Marja. Bioactive paper and fibre products. Patent and literary survey. 2006. 83 p.
- 52 Alanen, Raili & Hätönen, Hannu. Sähkön laadun ja jakelun luotettavuuden hallinta. State of art -selvitys. 2006. 84 s.
- 53 Pasonen, Markku & Hakkarainen, Toni. Kaukolämpölinjojen elinikä ja NDT. 2006. 27 s.
- 54 Hietaniemi, Jukka, Toratti, Tomi, Schnabl, Simon & Turk, Goran. Application of reliability analysis and fire simulation to probabilistic assessment of fire endurance of wooden structures. 2006. 97 p. + app. 23 p.
- 55 Holtinen, Hannele. Tuulivoiman tuotantotilastot. Vuosiraportti 2005. 2006. 38 s. + liitt. 7 s.
- 56 Häkkinen, Kai, Hemilä, Jukka, Salmela, Erno & Happonen, Ari. Logistiikka Belgiassa. Vierailukokemuksia keväältä 2006. 2006. 33 s.
- 57 Kulmala, Risto. Tiehallinto ja liikenteen tietopalvelut. Selvitysmiehen muistio. 2006. 29 s. + liitt. 3 s.
- 59 Graczykowski, Cezary & Heinonen, Jaakko. Adaptive Inflatable Structures for protecting wind turbines against ship collisions. 2006. 86 p. + app. 39 p.
- 60 Käsäkoski, Markku, Kurkinen, Marika, von Weymarn, Niklas, Niemelä, Pentti, Neubauer, Peter, Juuso, Esko, Eerikäinen, Tero, Turunen, Seppo, Aho, Sirkka & Suhonen, Pirkko. Process analytical technology (PAT) needs and applications in the bioprocess industry. Review. 2006. 99 p.
- 61 Välisalo, Tero, Räikkönen, Minna & Lehtinen, Erkki. Asset Management vesihuollossa. Kirjallisuustutkimus. 2006. 79 s. + liitt. 8 s.
- 62 Holt, Erika. Current trends in USA building research. 2006. 26 p. + app. 7 p.
- 64 Forsström, Juha. Ydinjätehuollon kustannusriskianalyysi. Esitutkimus. 2006. 51 p.
- 65 Talja, Asko, Törnqvist, Jouko, Kivikoski, Harri, Carpén, Leena & Nippala, Eero. Ruostumaton teräs maa- ja vesirakentamisessa. 2006. 32 s. + liitt. 5 s.
- 66 Rinne, Tuomo, Hietaniemi, Jukka & Hostikka, Simo. Experimental Validation of the FDS Simulations of Smoke and Toxic Gas Concentrations. 2007. 37 p. + app. 9 p.
- 67 Fox, Stephen. Three Original Methods for Minimizing Ambiguities in Global Project Communications. 2007. 50 p.
- 68 Eckhardt, Jenni, Öörni, Risto, Hautala, Raine, Lehtonen, Mikko & Leviäkangas, Pekka. Tietopalvelumalli. Yleinen malli tietopalvelujärjestelmien kuvaamiseen ja arviointiin. 2007. 22 s. + liitt. 6 s.
- 69 Kannianen, Jenni & Haajanen, Jyrki. BPEL Engines. State-of-the-Art Survey for SOAMeS-project. 2007. 30 p.