Modeling and Execution on Web Services in the Web Information Systems

Krasimir Trichkov, Elisaveta Trichkova

Modeling and Execution on Web Services in the Web Information Systems: *This paper aims to present java open source XPDL workflow products for Web services – Enhydra Java Workflow Editor (JaWE) and Enhydra Shark workflow engine. The both products support XPDL as their native file format. XPDL provide a formal model for expressing executable processes that addresses all aspects of enterprise business processes. This paper examines the potential of Enhydra JaWE to create, manage and review workflow process definitions and Enhydra Shark to execute these processes at runtime.*

Key words: *Information systems; Web service; XPDL; LDAP; WfMC; CORBA.*

INTRODUCTION

A lot of activities in today's dynamic world concern effective information exchange. Enhydra JaWE is an open source graphical Java workflow process editor which allows the user to create, manage and review workflow process definitions using a visual tool. Enhydra Shark is an extendable workflow engine framework. Both products include a standard implementation completely based on WfMC specifications supporting XPDL (XML Process Definition Language) as their native workflow process definition format. XPDL provide a formal model for expressing executable processes that addresses all aspects of enterprise business processes. XPDL is conceived of as a graph-structured language with additional concepts to handle blocks. Scoping issues are relevant at the package and process levels [1].

ENHYDRA JaWE (JAVA WORKFLOW EDITOR)

Enhydra JaWE (Java Workflow Editor) [2] is an open source graphical Java workflow system that allows the user to create, manage and review process definitions (workflows) using a visual tool. JaWE has been developed fully according to WfMC specifications supporting XPDL (XML Process Definition Language) as its native file format and LDAP connections. As a consequence of involving the WfMC proposed package concept, JaWE divides itself into two logical parts: Package Level and Process Level. As will be explained in detail below, the Package level manages entities and attributes within the Package, while Process level manages entities and attributes within Workflow Process Definition.

JaWE is a tool for Process Definition modeling. The final output of this process modeling is a XPDL output file, which can be interpreted at runtime by the workflow engine(s). JaWE accomplished three main goals:

- Graphical representation of process definition
- Export of process definitions to XPDL
- Import of any valid XPDL and its graphical representation

The workflow process definition interface defines a common interchange format, which supports the transfer of workflow process definitions between different products. A workflow process definition, generated by JaWE, is capable of interpretation in different workflow run-time products. The principles of Process Definition Interchange are based on Meta-Model framework. This meta-data model identifies commonly used entities within a process definition, their relationships and attributes. A variety of attributes describe the characteristics of this limited set of entities. Using this Meta-Model, JaWE can transfer models using a XPDL as a common exchange format. Beside this interchange, JaWE is also used for internal representation of process definitions (Figure 1). The Meta-Model (Process Meta-Model) describes the top-level entities contained within a Process Definition, their relationships and attributes. It also defines various conventions for grouping process definitions into related process models and the use of common definition data across a number of different process definitions or models.
Below is presented a short list of the entities:

- Workflow Process Definition
- Workflow Process Activity
- Transition Information
- Workflow Participant Declaration
- Workflow Application Declaration
- Workflow Relevant Data
- System & Environmental Data.
- Resource Repository or Organizational Model
- Data Types and Expressions

All of these entities are maintained by JaWE, except "System & Environmental Data".

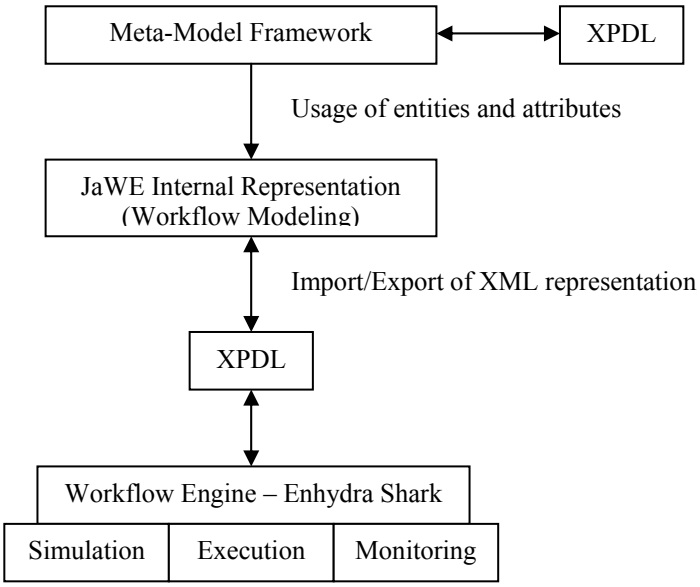

Figure 1. Enhydra Process Definition Interchange

JaWE gives two logical parts of application as was said in the previous text. There are "Package level" and "Process level". Package level manages entities and attributes within Package while Process level manages entities and attributes within Workflow Process Definition.

1. Package level

At the Package level (Figure 2), the user is allowed to choose among several XPDL views such as graphical, text, and xpdl view which represent the XPDL as it will be saved into file. Graphical view of package level is divided into some parts. Left side of the main window shows Package Hierarchy Tree. Root of the tree is main package and branches are imported external packages. If these external packages have their own external packages, they are shown in the tree too. Since all packages have unique IDs, circular references are allowed, though tree expansion will stop on package which is displayed before. Right side of the main window displays workflow processes defined in package which is selected on the hierarchy tree.

2. Process level

The second part of JaWE is Process level (Figure 3), which is some kind of editing window. This part of JaWE is used for graphical representation of process definition and for defining attributes of entities on that level. The Process level view is essentially an editing window that can be used for graphical representation of process definition and for defining attributes of entities on that level. In the visible working area, the user inserts visible objects and adjusts them. The first thing drawn must be a participant, after which the user may insert other elements such as Activities and Transitions.

There are 4 kinds of Activities defined in JaWE:

- Generic Activity
- Sub-Flow Activity
- Block Activity
- Route Activity

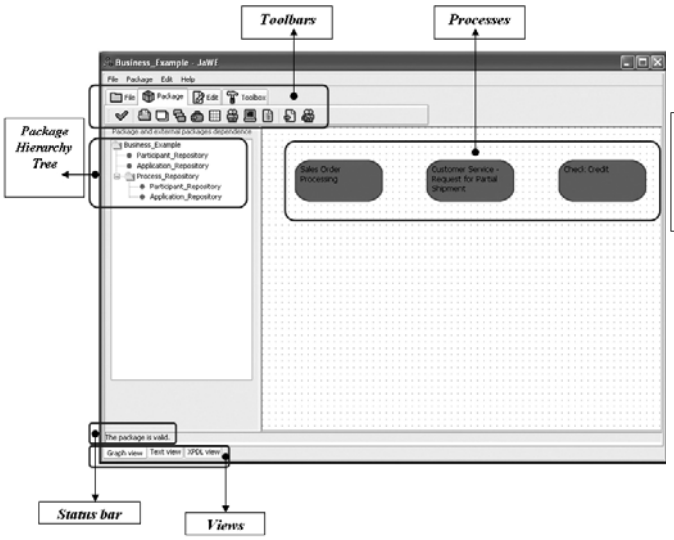

Figure 2. Package level

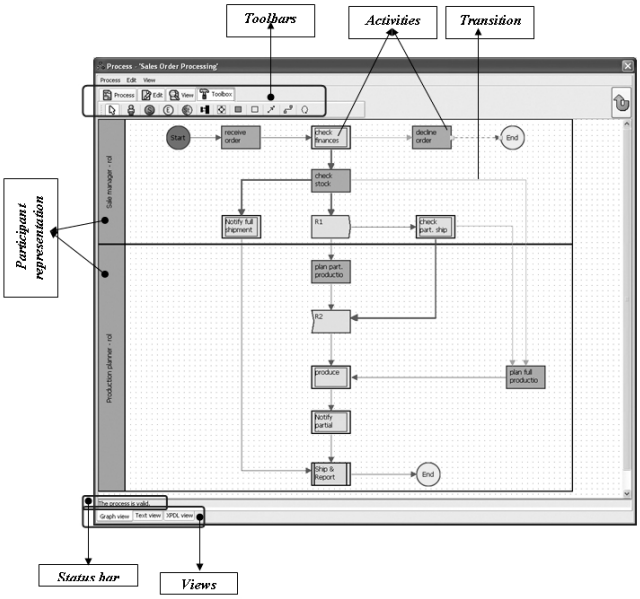

Figure 3. Process level

The link between two activities is established by transitions. Transitions are more than just link between activities. They describe possible transitions between activities and the conditions that enable or disable them during workflow execution. JaWE has three types of transition - simple, self-routed and circular. Simple transition is link between two activities, represented graphically with one straight line. Self-routed transition is link between two activities which is graphically 'broken' in three parts (but essentially, they do not differ in the XPDL logic they are representing), and circular transition is transition from activity to itself, and is graphically represented as a circle with an arrow.

ENHYDRA SHARK

Enhydra Shark [3] is workflow engine completely based on WfMC and OMG specifications.

Shark Admin application is Java swing application meant to be used by administrator to manage Shark engine. There are two kind of admin application, the first one is using shark directly as a library, and the other one communicates with shark deployed as a CORBA service using shark's CORBA wrapper interface. It can be used to handle shark's external repository containing XPDLs (to upload new XPDL files or delete existing ones), to load some XPDL file into shark, unload it, update it, to instantiate and monitor shark's processes, to perform mappings among participant definitions and real users, and among application definitions and Tool agents. It also contains a built-in work list handler application that can be used for performing work items, or for reassigning work items from one user to another.

The process monitor is divided into four major parts, Figure 4. The package-process definition-process instances tree enables to select a running instance of a package's process definition. When the user selects the process instance, other parts graphical data correspond to this process instance. He can see the main properties of the instance (the name, and the current state), the graphical diagram of the process instance with activities that are currently running being marked, to perform different operations on that process instance using the buttons at the bottom.

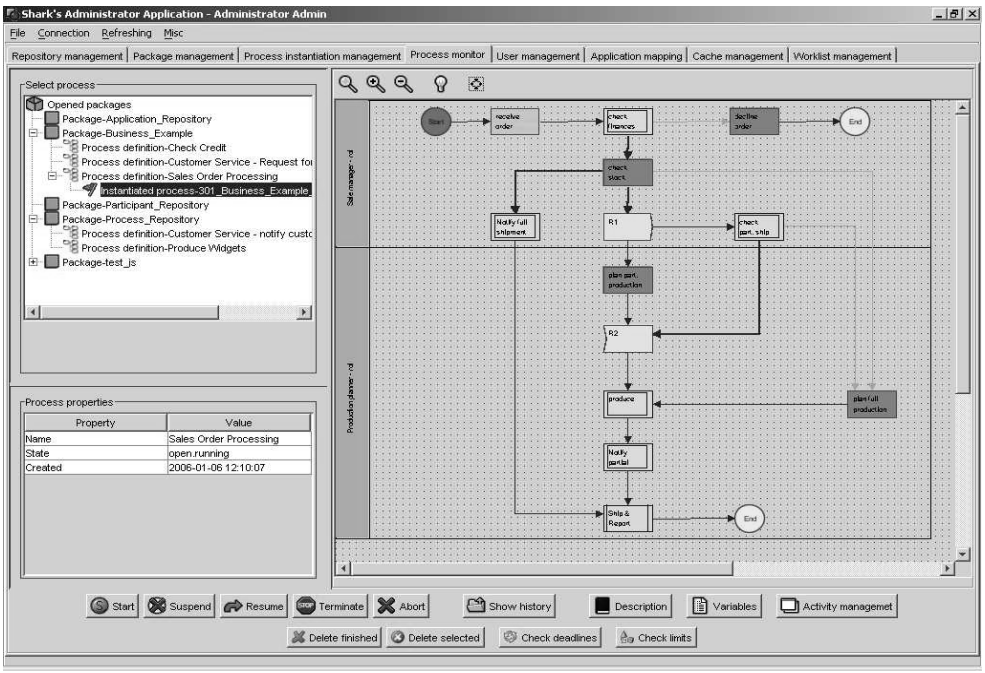

Figure 4. Enhydra Shark process monitor

The operations which can be performed are:

- start the process
- suspend the process
- resume the process
- terminate the process
- abort the process
- view the process history
- see the description of the process
- see and edit the process variables, and that way the user can manage the process flow if needed
- enter activity management dialog. The dialog displays the list of process activities, and when the user selects one of them, its current state is displayed in the text box. From this dialog, the user can perform additional operations on single activities:
    - suspend activity
    - resume activity
    - terminate activity abort activity - process becomes 'stucked'
    - manually start an activity
- delete all finished processes
- delete selected process
- perform a check for activity deadlines for all processes
- perform a check for limits of all processes and activities

The user management console page is divided into three parts. Functionalities:

- Accounts – the administrator can manage the users of the shark server by defining the new ones, deleting the existing ones or changing their properties.
- Logged - displays the list of currently logged users.
- Mapping - enables the administrator to map the package and package's processes participants to the real shark users.

Enhydra Shark allows mapping a package and package's processes applications to the real applications handled by a tool agent. Currently, six tool agents come with the Shark distribution. To map application definition to tool agent application, the user has to go to the application mapping section of admin application, and press the "add" button. The dialog will arise, and one has to select the application definition at the left side of dialog, and the tool agent on the right side of the dialog.

Then the user should enter some mapping parameters for tool agent. When a mapping of the application definition to the tool agent is done, shark will try to connect to the proper tool agent and ask him to execute its application, and will retrieve the results of execution. Here is the brief description of parameters that the user can enter when mapping of the application is performed:

- User name and password - not required for tool agents distributed with Shark. Some other tool agents can use it when calling applications that require login procedure
- Application name - the name of application that should be started by tool agent (i.e. for JavaClassToolAgent that would be the full name of the class, for RuntimeApplicationToolAgent it would be the name of executable file that should be in the path of the machine where tool agent resides. For JavaScriptToolAgent this can be either the name of the java script file, or the java script itself, depending on Application mode attribute.), for SOAPToolAgent it is the location of WEB service and for MailToolAgent it is a class of MailMessageHandler called to actually send/receive mails.
- Application mode - various tool agents use this attribute for different purposes. The RuntimeApplicationToolAgent uses mode 0 to indicate that it should not finish execution until the system application is finished (otherwise it will start system application and return finished status -> activity does not wait for system application to finish, but process proceeds to the next activity), and JavaScriptToolAgent uses mode 0 to indicate that it should search for java script file .

Enhydra Shark project delivers a workflow server with a difference. It is an extendible workflow engine framework including a standard implementation completely based on WfMC specifications using XPDL (without any proprietary extensions) as its native workflow process definition format and the WfMC "ToolAgents" API for serverside execution of system activities. Every single component (persistence layer, transaction manager, scripting engines, process repository,...) can be used with its standard implementation or extended/replaced by project specific modules. This way Enhydra Shark can be used as a simple "Java library" in servlet or swing applications or running in a J2EE container supporting a session beans API, Corba ORB or accessed as a web service.

CONCLUSION

The essence and functional possibilities on Enhydra JaWE and Enhydra Shark were presented in this paper. Special features of modeling and execution on Web Services and their application in the Web Information Systems are outlined. Software components of the products are described. The products are platform and software independent. As a future work is the problem for optimization and developing of Web Services in Internet. Also will be proposed decision for Web Services in the Web Information Systems (Index of Citations as Web Service).

This research is partly supported by the European Commission, project №FP6-027178.

REFERENCES

[1] WfMC, Workflow Standard, Process Definition Interface - XML Process Definition Language, Technical Report Document Number WFMC-TC-1025, Workflow Management Coalition, 2005.

[2] http://www.enhydra.org/workflow/jawe/index.html

[3] http://www.enhydra.org/workflow/shark/index.html

For contacts:

Assist. Prof. Krasimir Trichkov, Institute of Computer and Communication Systems – BAS, 1113 Sofia, Bulgaria, Acad. G. Bonchev str. Bl. 2, e-mail: krasi@hsi.iccs.bas.bg

Assist. Prof. Elisaveta Trichkova, Institute of Computer and Communication Systems – BAS, 1113 Sofia, Bulgaria, Acad. G. Bonchev str. Bl. 2, e-mail: elisaveta@hsi.iccs.bas.bg