Execution of Web Service in Internet using Enhydra Shark workflow engine

Krasimir Trichkov

Abstract: *This paper aims to present java open source XPDL workflow product for execution of Web services - Enhydra Shark workflow engine which is an extendable workflow engine framework. The product support XPDL as its 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, essence and functional possibilities of Enhydra Shark to execute these processes at runtime. Proposed is decision for execution of Web Services in Internet (Index for Citations as Web Service).*

Key words: *Information systems, Web service, Workflow execution system, XPDL, WfMC*

Introduction

Enhydra Shark is an Open Source Java workflow solution for integration of Web Service in Internet. It is a powerful and flexible Java XPDL workflow execution engine and it is completely based on WfMC and OMG specification. Enhydra Shark is an extendable workflow engine framework. The potential of this product is to execute the workflow process definitions at runtime. The product includes a standard implementation completely based on WfMC specifications supporting XPDL (XML Process Definition Language) as its native workflow process definition format. 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. In the paper is proposed decision for Web Services in Internet (Index for Citations as Web Service) [1,2].

Enhydra Shark

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 is capable of interpretation in different workflow run-time products. The principles of Process Definition Interchange are based on Meta-Model framework. The Meta-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. The concept of Enhydra Process Definition Interchange is shown on Figure 1 [2,3].

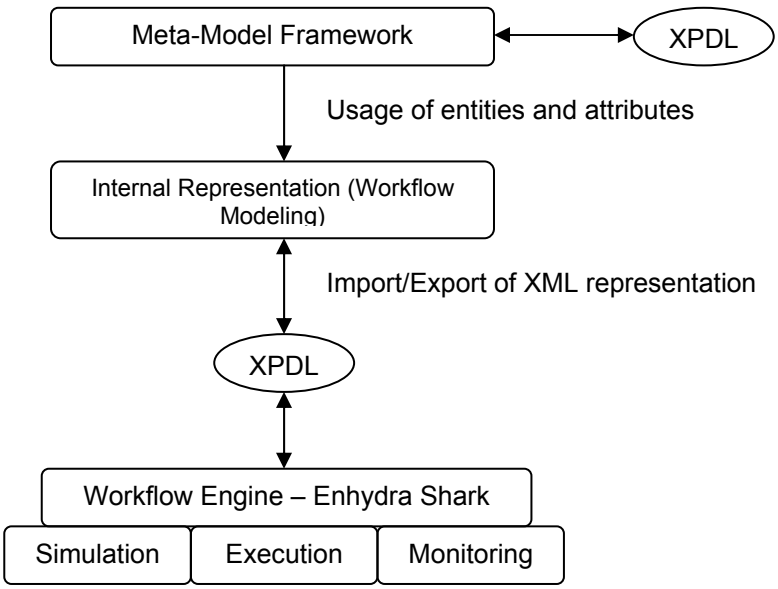

Figure 1. Enhydra Process Definition Interchange

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.

Figure 2 represents process monitor of web service index for citations. The process monitor is divided into four major parts. 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 [2].

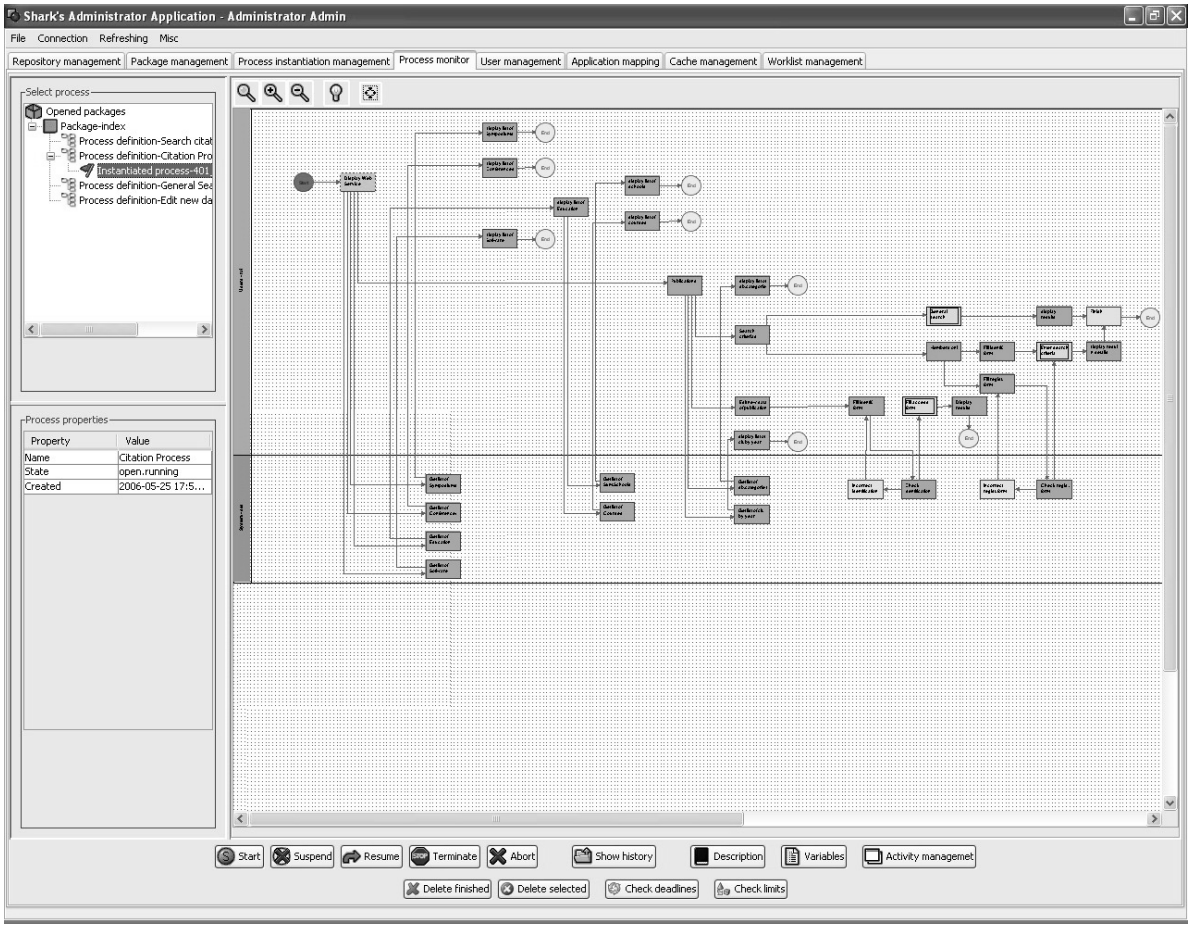

Figure 2. Enhydra Shark process monitor

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 [4].

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 [3,4].

The repository management displays all available files in engine's XPDL repository, Figure 3. This is the place where the user can manage engine's XPDL repository. The user can upload a new XPDL file from his local machine into engine's repository, or delete one from the repository.

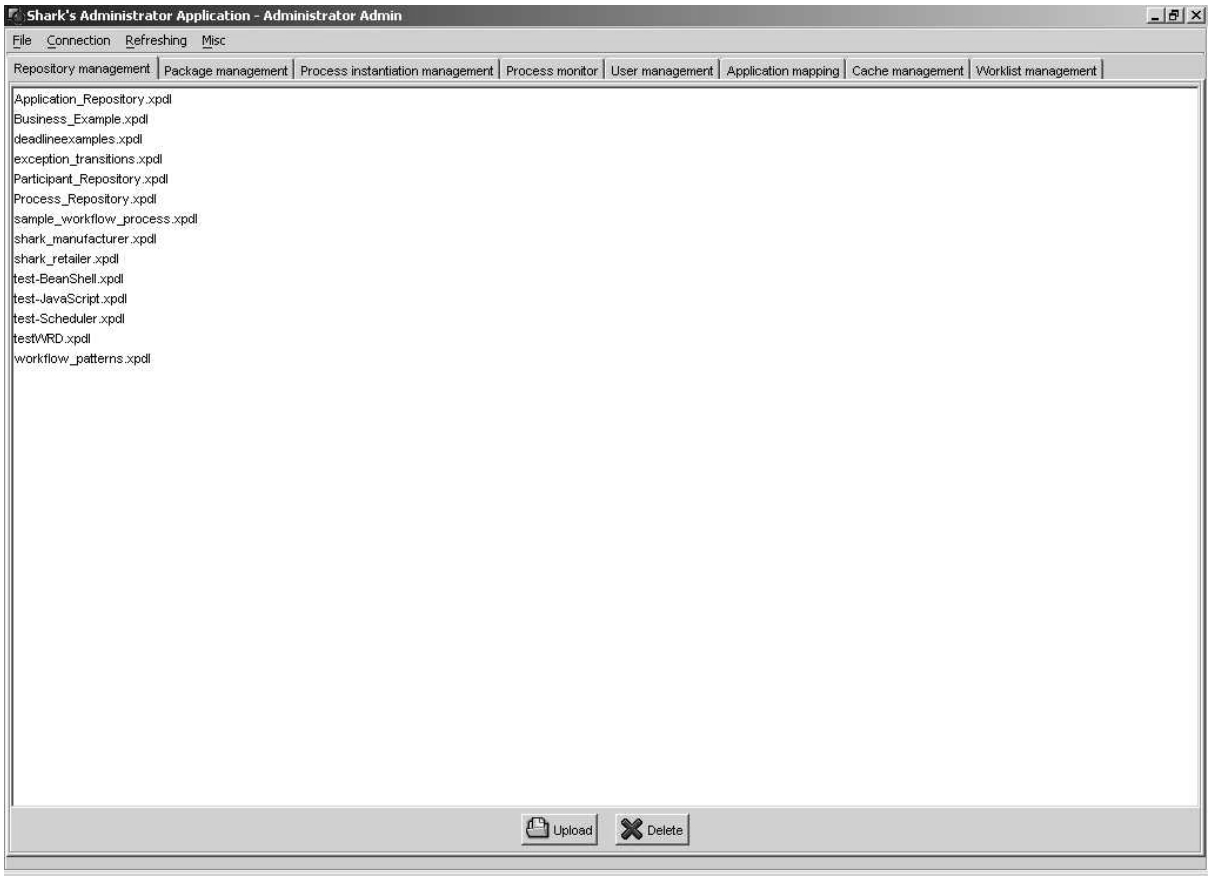

Figure 3. Enhydra Shark repository management

To upload new package the button 'upload' is used. The dialog for choosing XPDL files from the user local file system is shown. When the user choose the package file he can upload, the dialog for entering the file path relative to the engine repository. Here one can enter the directory structure and the file name that the user XPDL will have on the engine. The user can enter something like: test/conformance/test1.xpdl. After the file is uploaded into engine's repository, it can be loaded into engine, and its processes can be started, which will be described in following sections. Also, the user can delete the files from engine's repository by selecting the file and pressing 'delete' button.

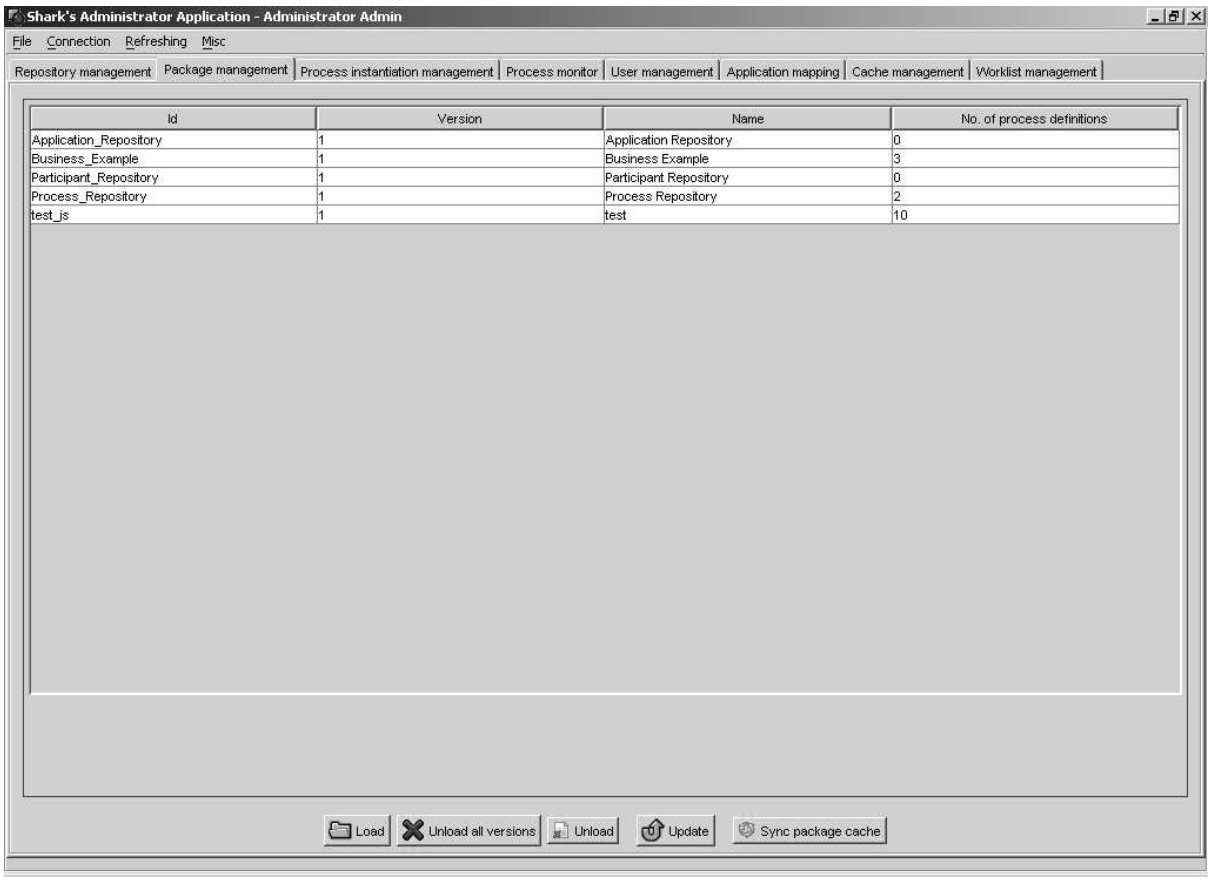

Figure 4. Enhydra Shark package management

The package management displays all packages (XPDLs) that are loaded into engine, Figure 4. It enables to load and unload packages to/from engine, as well as to update some already loaded packages and to synchronize engine's package cache.

Loading packages: To load package(s) into engine, the user have to press the 'load' button, and select the one of the offered packages. The packages which can be loaded are all packages from engine repository, except the ones which are already loaded, and the ones that have the same Id as already loaded packages. When the user selects a package from the list, its file name and Id are displayed in the text box. Then, if the user press 'Load' button, package will be loaded into engine (if it is valid and if there are no problems while package loading), and the user will be able to start processes instances based on the process definitions within that package.

Unloading packages: To unload package from engine, the user have to select the wanted package, and press the 'unload' button. If there are no instantiated processes from that package's process definitions that are still held into DB, and this package is not referenced by any other package, it will be unloaded from the engine. After that, the user will not be able to instantiate the processes from its process definitions.

Updating packages: If the user wants to update package, he can do it by selecting it, and pressing 'update' button. The list of the packages from repository, with the same Ids as the one he wants to update, is shown. The user can select a package from the list, and press 'update' button. The processes that were running based on old package's process definitions remain to run based on them, but the user will be able to create processes based on definitions contained in new package version.

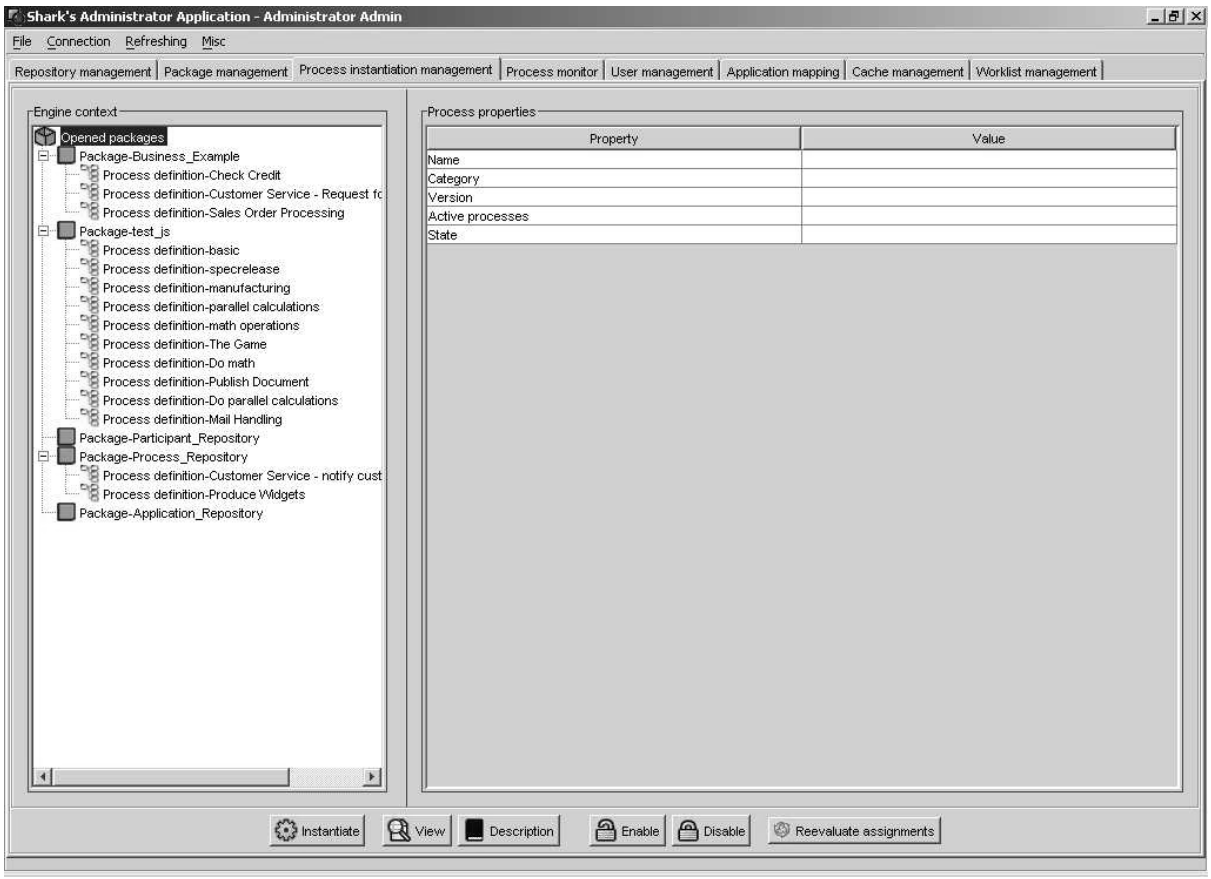

Figure 5. Enhydra Shark process instantiation management

On Figure 5 is presented the package-process definition tree of the loaded packages. If the user selects a package from the tree, and press left-mouse button, he can get the property dialog of the package. If the user selects some of the package's process definition, he can also get the property dialog by right-mouse button which gives additional functionality:

- In the right pane, the user can see some general process properties, along with the number of currently running processes based on this process definition.
- The user can create a new running instance of the process and start it by pressing button 'Instantiate'
- The user can see the graphical presentation of the process by clicking the 'View' button
- The user can see the description taken from process definition by pressing 'Description' button
- The user can enable or disable specific process definition or whole process definitions for package(s).

The user can perform reevaluation of assignments (this can be useful if he changes participant->user mappings, and they are applied on already created assignments).

Enhydra Shark allows mapping a package and package's processes applications to the real applications handled by a tool agent, Figure 6. 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.

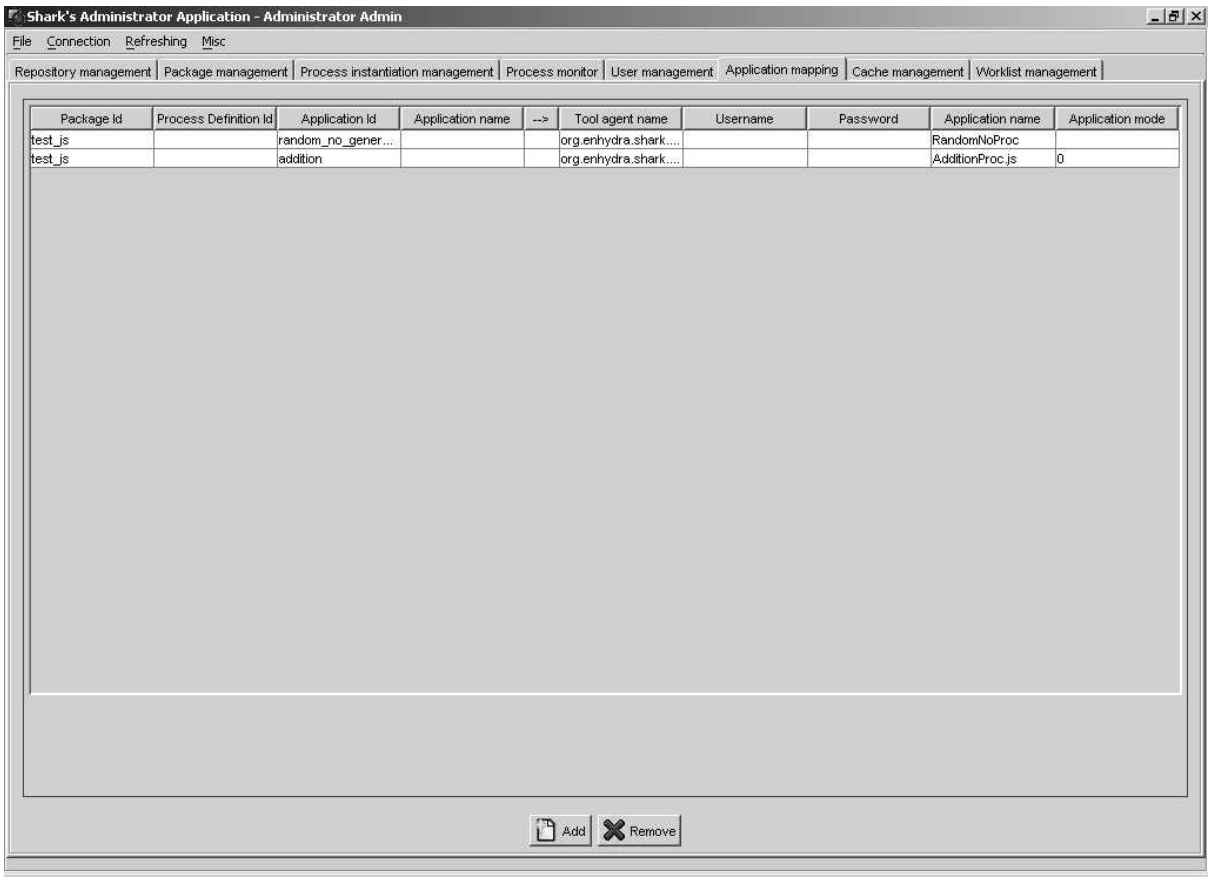

Figure 6 - Enhydra Shark application mapping

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.

Conclusions and future work

This paper defines the essence and functional possibilities of Enhydra Shark Workflow Engine. Definite are special futures of execution on Web Services and their application in Internet. Proposed is a decision for Integration and Execution of Web Service - Index for Citations in Internet. Definitely are software components of the product. The product is platform and software independent. As a future work is the problem for optimization and developing of Index for Citation as Web Service in Internet.

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

References

[1] Stoilov T., K. Stoilova. Integration of Web Services in Internet. 18th International Conference on Systems for Automation of Engineering and Research "SAER-2004", 24-26 September, 2004, St. Konstantin resort, Varna, Bulgaria.
[2] Trichkov K., E. Trichkova, Modeling and Execution on Web Services in the Web information Systems, Научно-приложна конференция по математика, информатика и компютърни науки, ВТУ „Св. св. Кирил и Методий”, 12-13.05.2006г., Велико Търново.
[3] WfMC, Workflow Standard, Process Definition Interface - XML Process Definition Language, Technical Report Document Number WFMC-TC-1025, Workflow Management Coalition, 2005.
[4] http://www.enhydra.org/

About the author

Assist. Prof. Krasimir Trichkov, Department Hierarchical Systems, Institute of Computer and Communication Systems - BAS, phone: (359 2) 979 2774, e-mail: krasi@hsi.iccs.bas.bg