FUJITSU

Interstage Business Process Manager v11 Architecture Whitepaper

Introduction

The Fujitsu Interstage Business Process Management (BPM) suite helps companies “Sense and Respond” to business change through continuous process visualization and optimization. Research indicates that the use of BPM technology can lead to remarkable cost savings, increased efficiency, and productivity – in other words, big ROI. Enterprises and software developers have come to understand that BPM can be applied to increasingly complex problems. Automating business processes from end-to-end, integrating applications to process steps, and opening up processes for collaboration with partners, customers, and suppliers represent just the beginning of what can be expected from BPM software.

Interstage Business Process Manager includes key components for 1) defining and refining processes; 2) creating rules and rich user interfaces; 3) integrating with systems within or outside an organization; 4) automating the process and supporting dynamic modifications to the course of a process to respond to business changes; 5) analyzing hybrid process patterns – a combination of pre-defined and ad hoc process steps; and 6) continuously optimizing a business process.

Interstage Business Process Manager incorporates all of the building blocks necessary for adding a flexible and sophisticated process layer to business applications. Business processes should not be hard-coded in to application code. Given the pace of change of current business applications, application developers should also utilize BPM to separate process logic from the application logic. The resulting process layer is easy to understand and empowers managers to implement change requests immediately without technical assistance. Interstage Business Process Manager provides an easy-to-use graphical user interface for controlling the process layer of your applications. With the introduction of dynamic BPM capabilities for automating the unplanned, unstructured activities in real work scenarios, business users no longer need to wait in line for scarce IT resources to implement changes to their business processes.

Interstage Business Process Manager can also be easily embedded into other applications and products. It is ideal for supporting large-scale enterprise deployments, Independent Software Vendors (ISVs), SaaS providers and Systems Integrators (SIs) who need comprehensive business process management functionality within their applications.

Today's new business environment requires BPM technology that can be integrated seamlessly into SOA-based and B2B applications and molded easily to fit into changing architectures. Interstage Business Process Manager's architecture is flexible, leaving the door open for future technologies and standards. It fits naturally into the dynamic and evolving business environment.

Suite Components

The Interstage Business Process Management Suite includes the following components:

Studio

Interstage BPM Studio is an easy-to-use integrated development environment to build business process applications. Business and IT professionals are provided with different views of the

same process model to foster better communication and collaboration. All the artifacts required to run the application can be created using the Interstage BPM Studio. Such an application can be developed offline and later be deployed on an Interstage BPM Server that can be accessed by end-users. To facilitate developing and deploying a business process application, Interstage BPM Studio provides the following built-in functions:

Process Designer – Process Designer is a graphical tool to facilitate the creation of sequential, parallel and conditional tasks in a workflow. A process definition consists of activities and their relationships, phases and milestones, criteria to indicate the start and end of the process, and information about the individual activities, such as participants and the data to be handled. The directory services environment determines the user to whom work should be routed to during process execution. It provides designers with the capability to chain processes together at runtime, spawning new processes as required. Process designer provides advanced support for sub-processes for modular based process development, sophisticated dependencies between nodes (the building blocks for constructing process definitions) and parallel threads of activity. In addition to a graphical designer (Figure 1), users can also create processes using a simple outline grid (Figure 2), similar to the way one would define project activities and milestones using Microsoft Project. By switching from the “Outline View” to the “Diagram” view, IT professionals are able to enhance the process generated from a bulleted list of activities.

Process models are represented using the Business Process Modeling Notation (BPMN) and serialized using the XML Process Definition Language (XPDL). For more information on BPMN standards, please refer to www.bpmn.org and on XPDL standards, please refer to http://wfmc.org/xpdl.html .

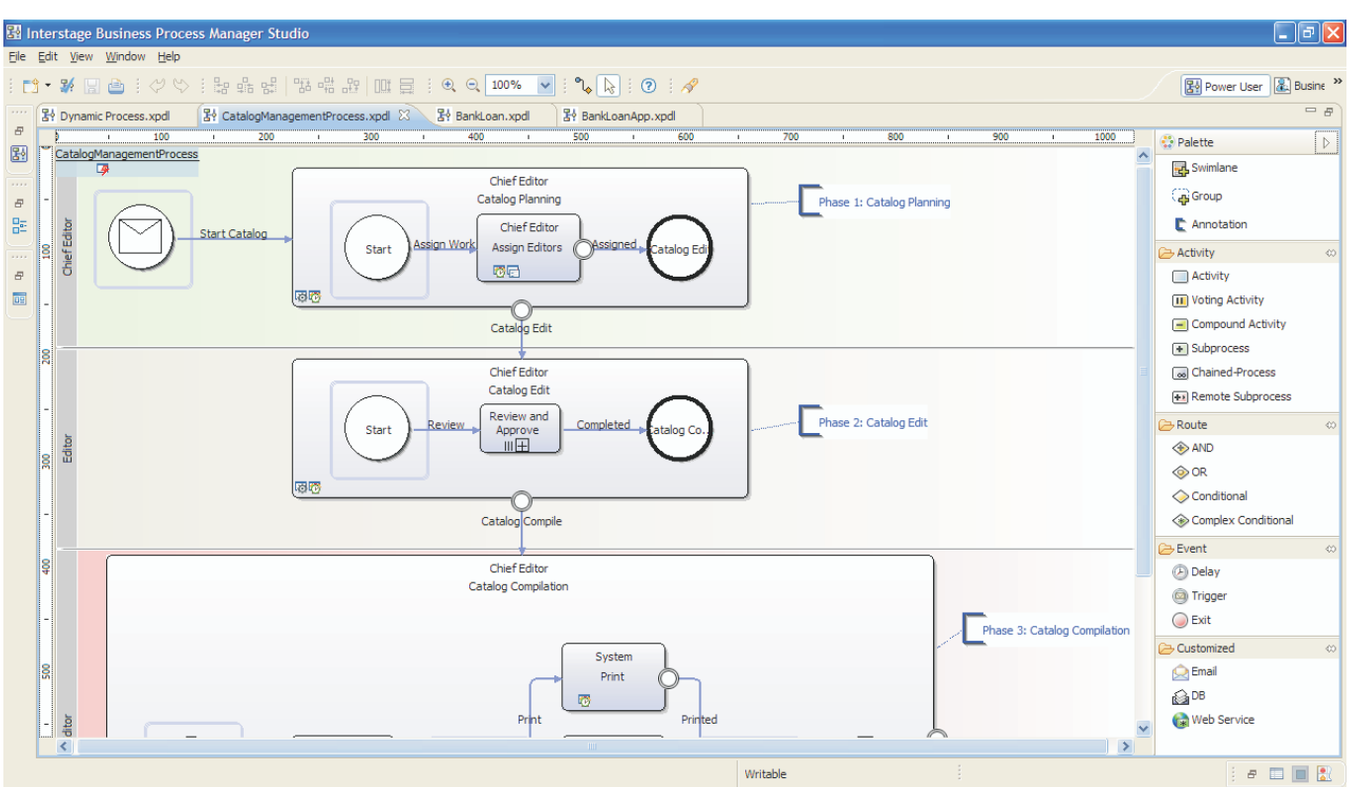

Figure 1: Graphical Process Designer: Sample Catalog Management Process

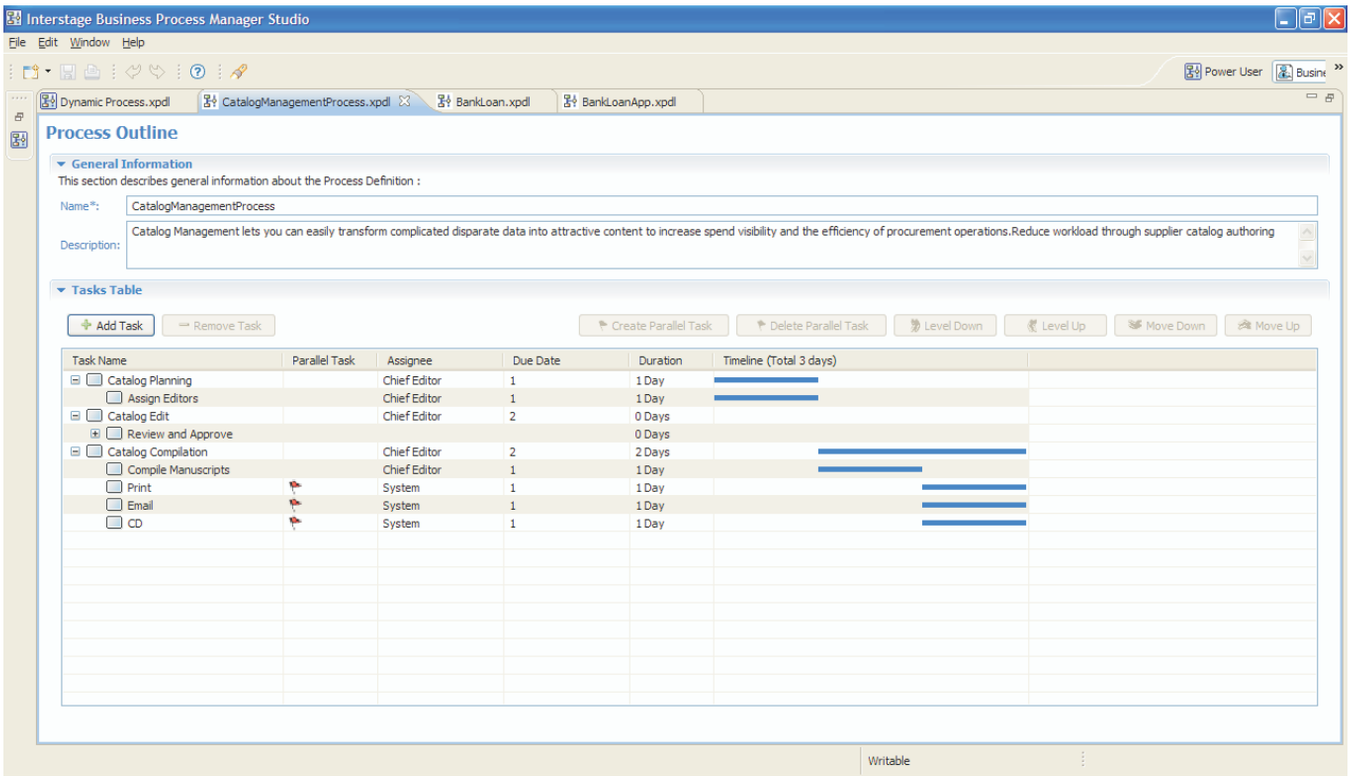

Figure 2: Business User View

WYSIWYG Form Designer – Forms provide a means for end users to interact with the process via a user portal. The Forms Designer provides an AJAX-based WYSIWYG editor to create rich browser-based forms. It provides a Layout Painter which is the central tool for layout development. The Forms Designer provides rich controls (Figure 3) that can be placed within the Layout Painter. Each control has an associated property sheet to configure the presentation and appearance of the form. The created forms can then be attached to any step in the process. The data mapping between the form fields and process attributes can easily be done using the control's property sheets. The end users interact with these forms while performing their tasks. Data entered into these forms are passed to and from the process seamlessly. Each layout created within the designer has a corresponding Java adapter class which can be extended by the developers to include custom business logic.

The Forms Designer also provides tools to include validation rules and formulas within the form. Validation rules can eliminate data entry errors. They can also provide users with useful tooltips to guide them through the completion of a form.

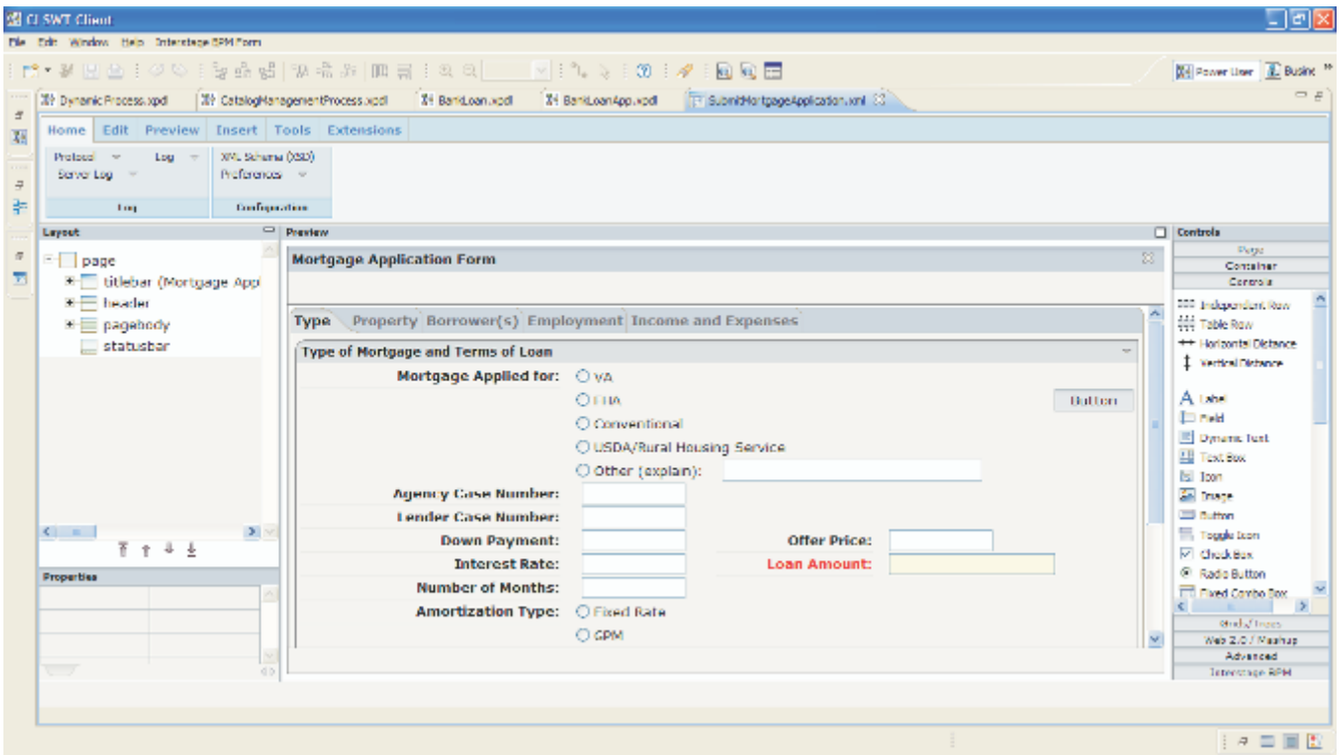

Figure 3: Forms Designer Interface

Decision Tables – It provides users with a business rules modeling and editing environment so that they can create simple process-centric rules in a tabular form. It is recommended that business rules be kept external to a business process. This allows business users to modify the rules that are required to be changed on a periodic basic – e.g., promotional discounts for a product, interest rate tables, loan approval policies, skills matrices (Figure 4). Decision Tables is a powerful, yet easy to use tool to manage these rules. The rules can be invoked from any point in the process to direct the flow of the process, to influence task assignments, or to incorporate calculations.

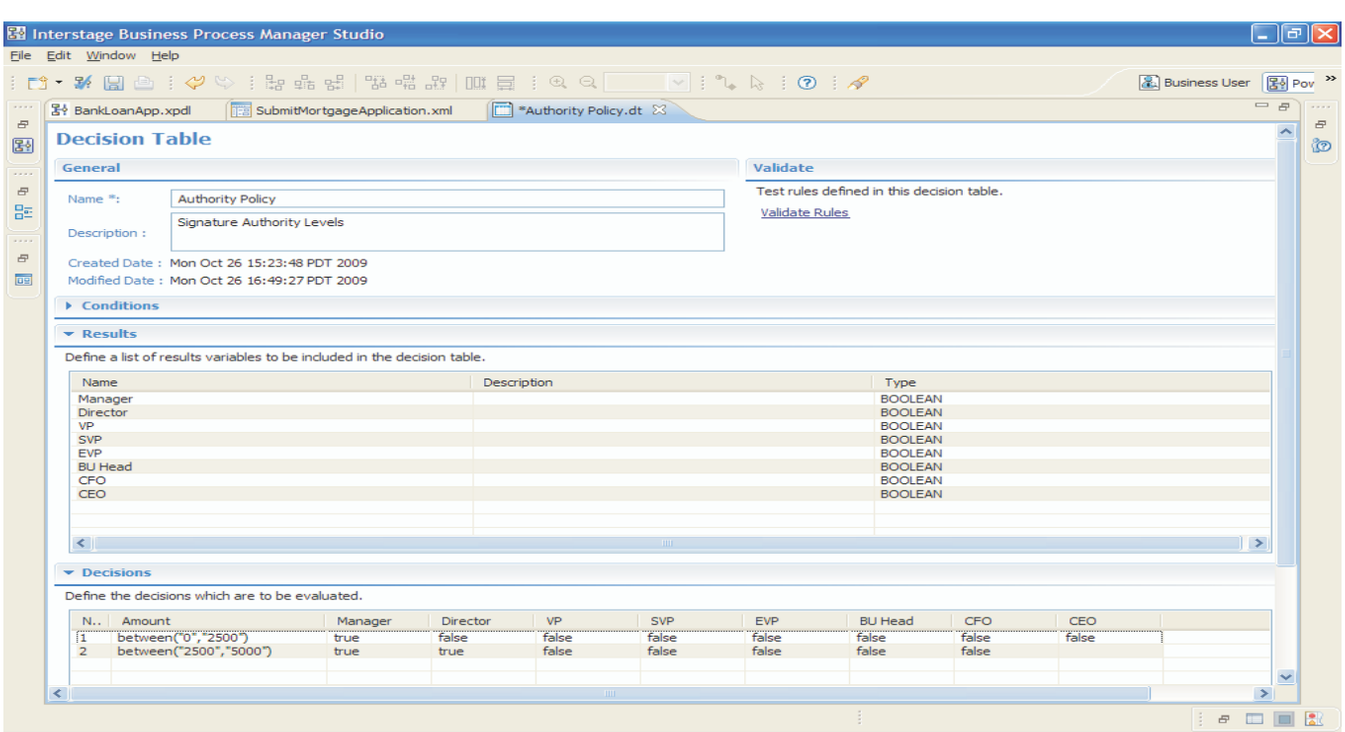

Figure 4: Decision Tables

Simulation – Users can simulate "what if" scenarios within this environment (Figure 5). This feature can help with estimating the cost of resources as well as with determining time-related metrics for specific activities prior to putting a process into production. For a specific activity within the process, users are able to define the working duration, resources required and the probability of a particular path being taken. In cases where the simulation is based on historical process data, the values are automatically computed and set to activities. The results of the simulation are then captured in intuitive graphical charts for further analysis. Simulation reports help users analyze operating costs, process bottlenecks and resource utilization (Figure 6). The report data and charts can be exported and charts in HTML, PDF, and CSV formats. Simulation reports can also be personalized to adhere to corporate standards for sharing. Reports are created to show one process at a time.

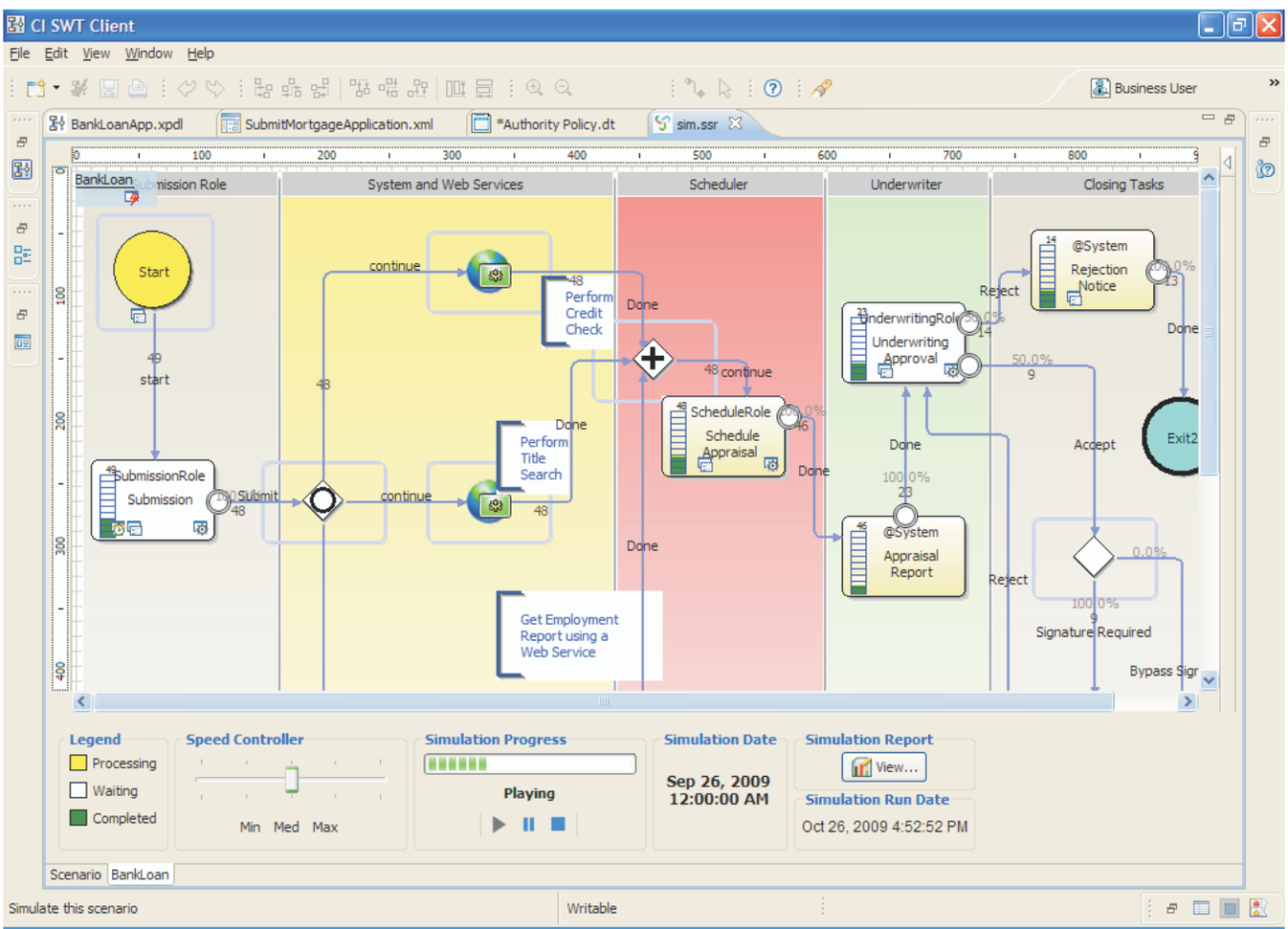

Figure 5: Running a Simulation

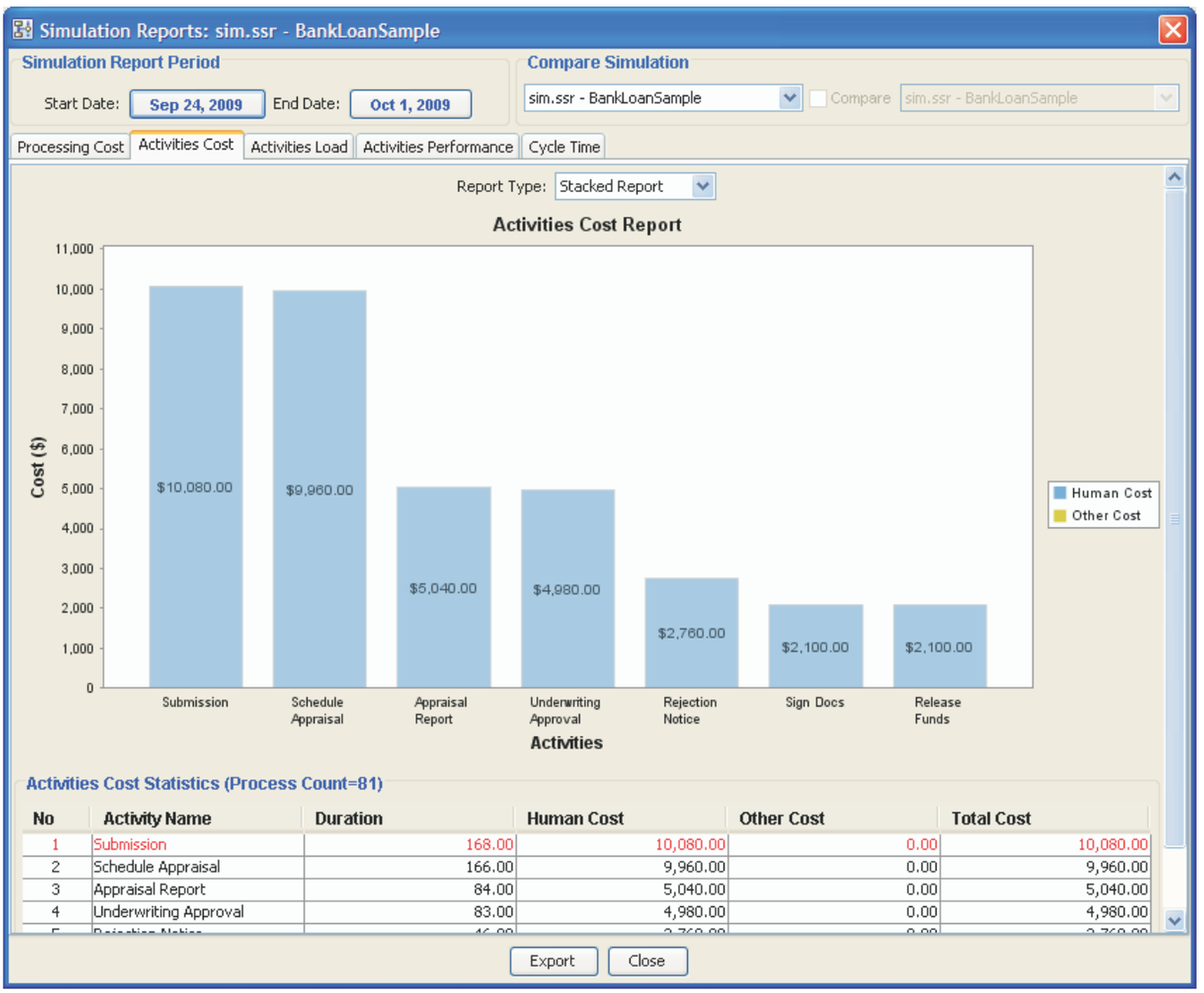

Figure 6: Simulation Reports

Analytics Designer – The Analytics Designer allows users to configure settings for visualizing the state of the business process, generating alerts and handling problems that occur during business activity, identifying exception trends and configuring performance indicators and user dashboards. By specifying the process attributes to be monitored and rules for identifying errors (alert rules), it is possible to identify and display problems that occur during business activity. Troubleshooting measures (such as problem detection, problem confirmation, problem identification and countermeasures) can then be performed more quickly, in addition to automated escalation tasks such as email notifications or executing business rules. Event processing relates to business event management, calculations and alerts. Rule-based calculations can be performed on events that are generated from the business process. For example, if a bank's policy dictates that processing a loan for platinum customer should be completed within two days, alerts will be generated to notify a loan supervisor as soon as that due date is missed. More sophisticated calculation rules can be applied on process attributes to generate alerts (Figure 7).

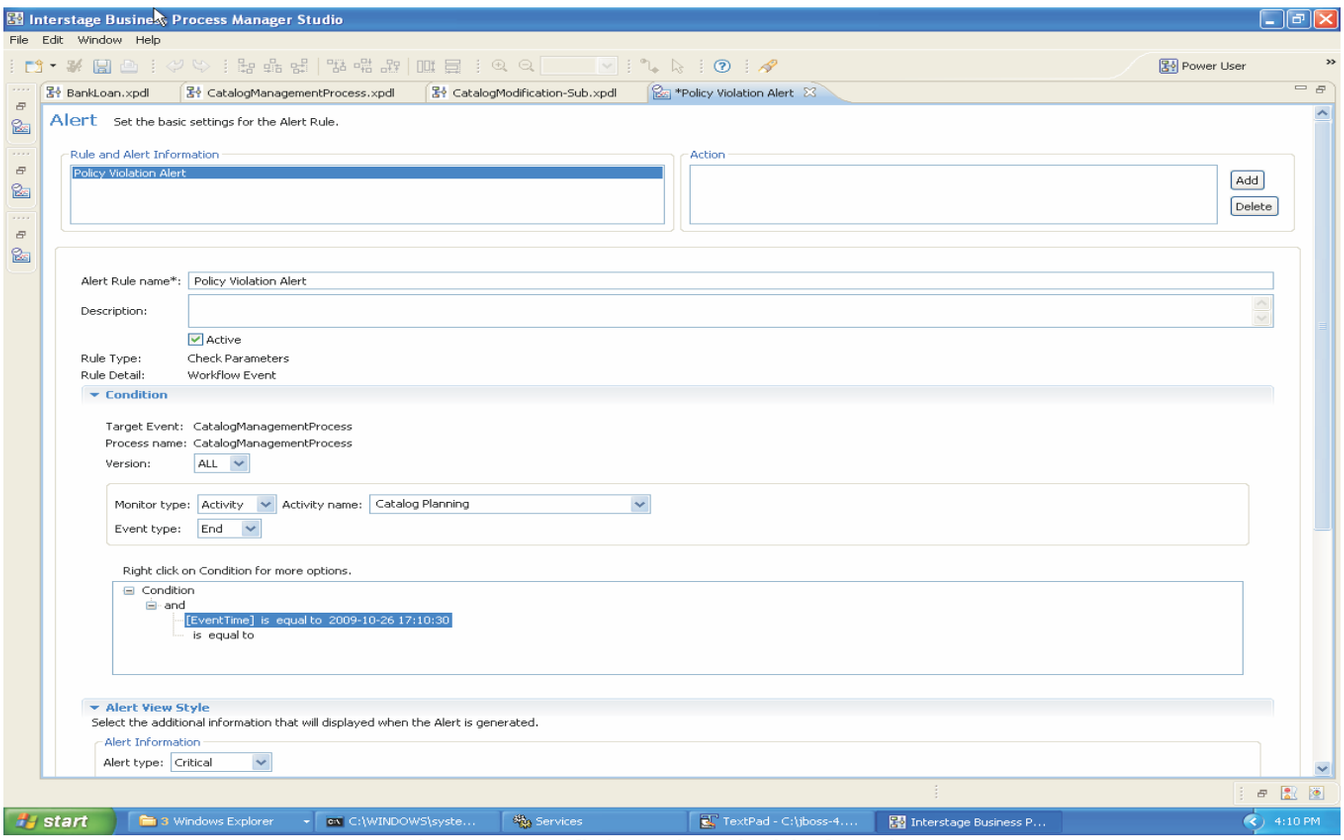

Figure 7: Defining Thresholds for Alerts

The Analytics Designer provides various pre-built charts to choose from to help present process information and metrics to users (Figure 8).

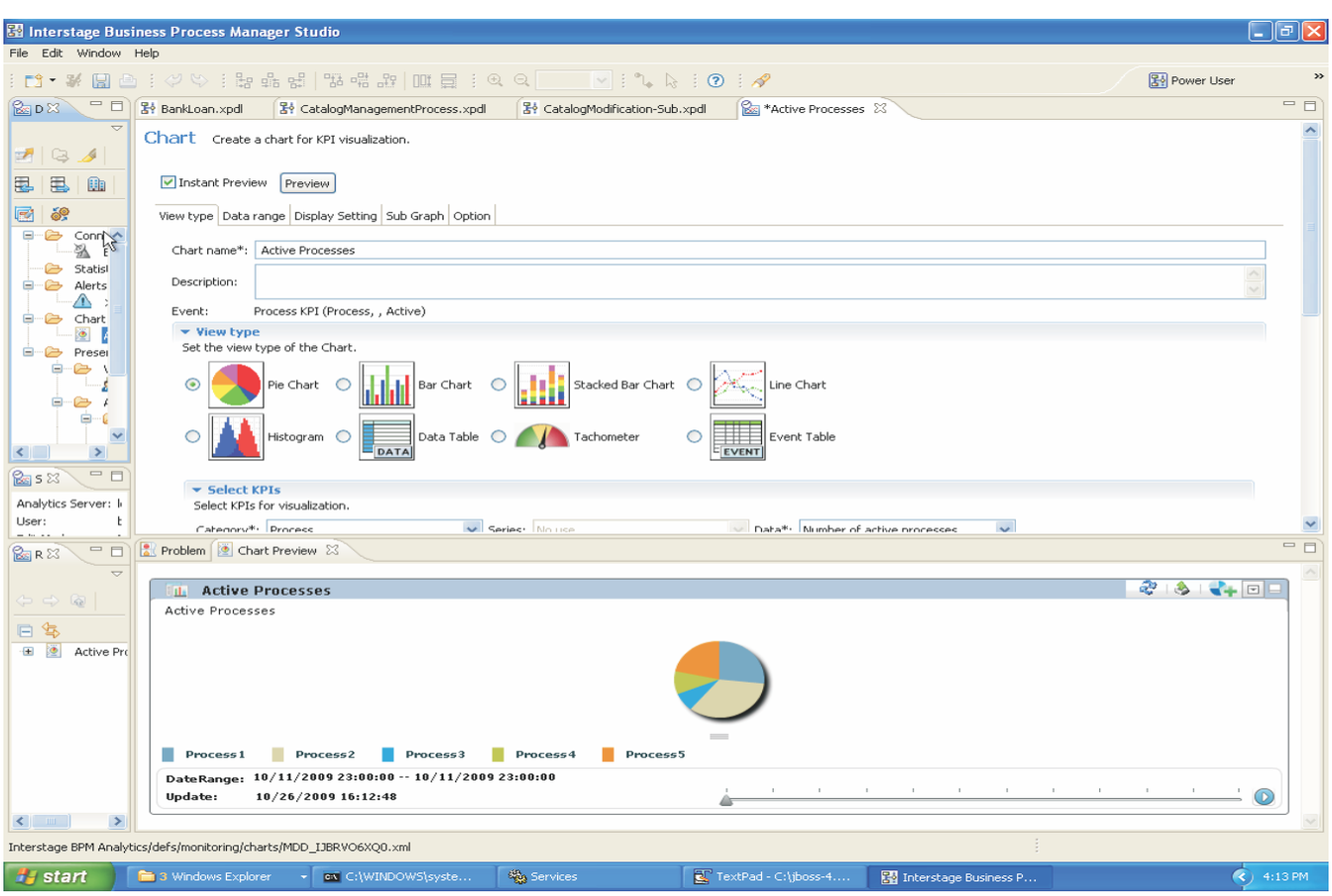

Figure 8: Pre-built Charts

Deployment Wizard – The Deployment Wizard allows users to upload and download applications built within Studio to and from BPM Server. Users can upload process application projects from Studio to the server by packaging all the files that make up the application into a .bar (BPM Archive) file. This .bar file can later be deployed on an Interstage BPM Server using the deployment mechanism provided by the Interstage BPM Console. An upload wizard walks users through the process of uploading a project from a user's workspace to a remote server (Figure 9).

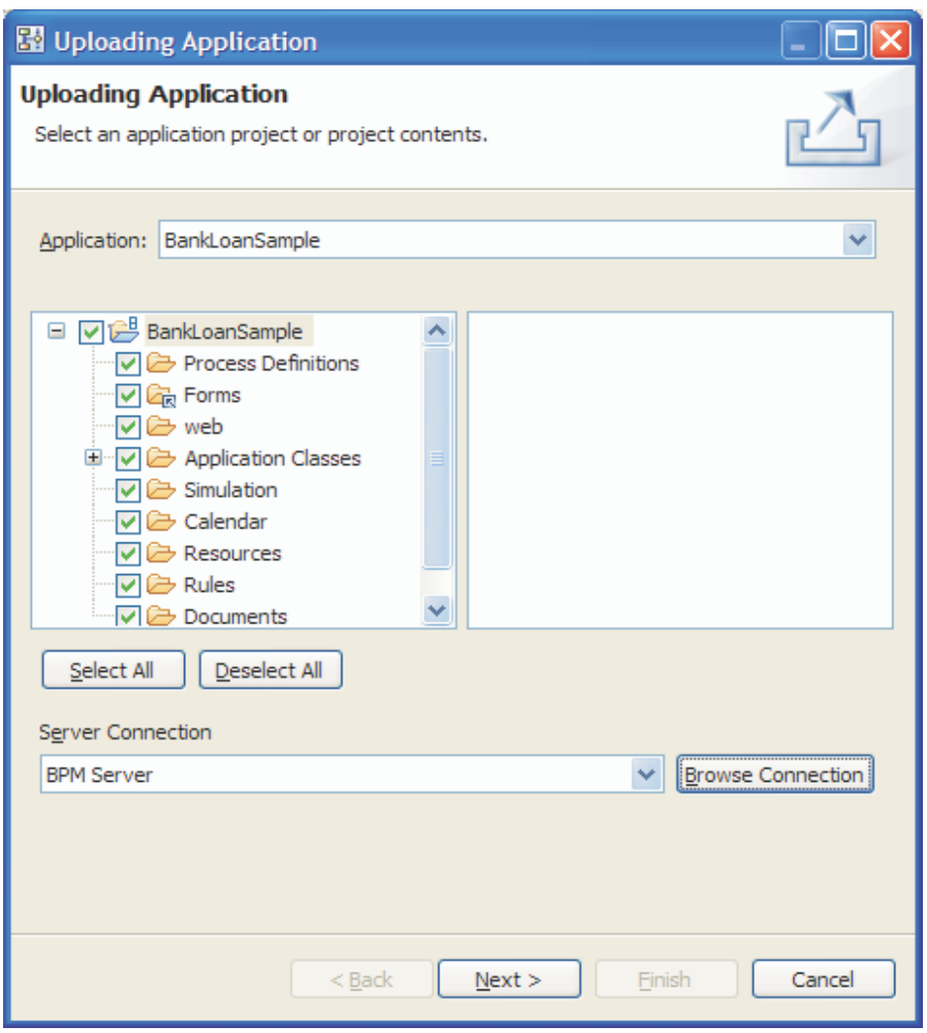

Figure 9: Deployment Wizard

In addition to Studio being available as a stand-alone eclipse application, it is also available as an Eclipse-plugin. By integrating Interstage BPM Studio into the Eclipse software framework, users can leverage Eclipse as the development environment for Java technology. Eclipse has many features that make for quick and painless Java programming and debugging.

Server

The server provides a 100% web-based console for process and task management and an execution engine for process execution. The server engine is J2EE-compliant and runs on popular application servers that support this standard. Interstage BPM can run on multiple application servers thereby leveraging the underlying load balancing and fail-over capabilities

for non-stop operation with 100% reliability. It is certified on the following application servers: Fujitsu Interstage, IBM WebSphere, BEA WebLogic, and JBoss. The server can be deployed in the single tenant mode (ideal for single application or departmental BPM deployments) or in the multi-tenant mode (ideal for Software as a Service (SaaS), enterprise Centers of Excellence/ Shared Services deployments, or Enterprise Cloud deployments). A tenant can host multiple process-based applications. Access to each application can be provided by individual URLs thereby providing the virtualization required at the application level. This makes Interstage BPM the ideal choice for shared services environment within an enterprise and for application service providers (ASP).

The server also offers an Application Programming Interface (API) called the “Model API” and a web services interface to access all the functionality delivered through the server.

Analytics

Analytics, a core component of Interstage BPM, includes an event processing engine that allows users to set rules for business event correlation (correlation can be done between events that occur within the process as well as between events that originate from external systems), filtering, and aggregation. The results of these rules are presented to users as alerts or as automated actions - e.g., kick off a process or invoke a web service. Analytics provides users access to business process performance metrics via dashboards in real time (Figure 10). It lets business users continuously monitor business activity using charts and alerts. It visualizes business process bottlenecks and helps respond to missed SLAs.

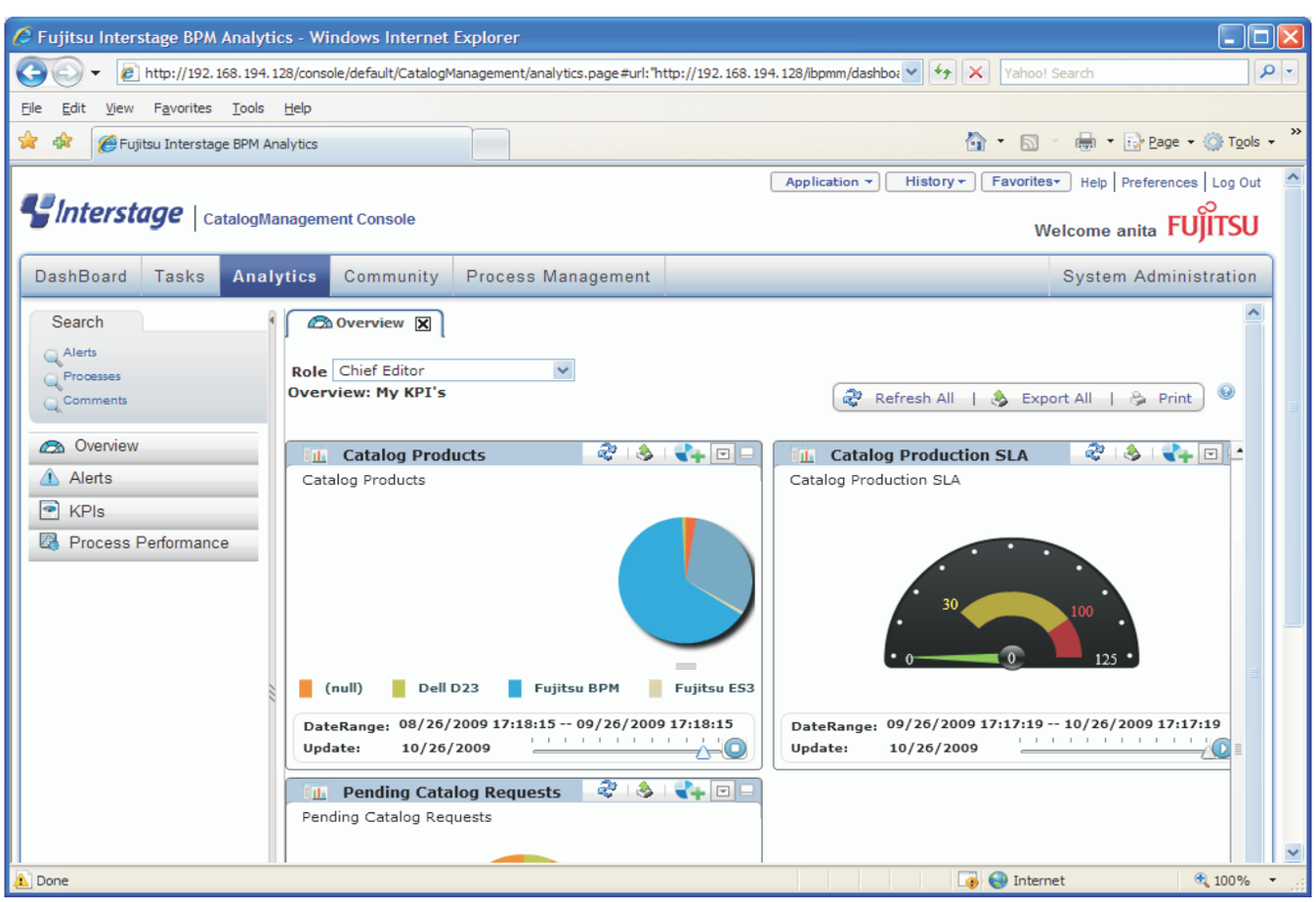

Figure 10: Analytics dashboard

Product Architecture

Flexibility and support for process change have always been key differentiators for Interstage BPM. Interstage Business Process Manager is a J2EE compliant engine that runs on popular application servers. Interstage BPM can run on multiple servers, providing load balancing and fail-over capabilities for non-stop operation with 100% reliability. It is certified on the following application servers: Fujitsu Interstage, IBM WebSphere, BEA WebLogic, and JBoss.

Interstage BPM Version 11 delivers advancements in multi-tenancy and it is easy to install, embed, and extend. We continue to expand our support for the leading BPM standards such as BPMN, XPDL, WebDav, and UDDI. Interstage BPM is ideal for deployments in a service-oriented architecture (SOA) environment as it lets users reuse of business processes as web services.

Independent Software Vendors (ISVs) and solution providers have realized that embedding a business process management engine adds significantly more value to their product offerings. Increasingly, ISVs and solution providers rely on an embedded process management engines to stitch together the different elements and functions within their applications. These developers have learned that homegrown, hard-coded processes limit a company's ability to change or customize them to meet their needs.

The unique multi-tenancy capabilities enable cloud deployments. It allows companies to onboard multiple tenants and then deploy multiple applications for each of these tenants on a single instance of BPM. Each application is hosted as a service that customers can access via the internet. This makes Interstage BPM the ideal choice for enterprises that want to create private Cloud BPM deployments as well as for application service providers (ASP). The Fujitsu Cloud BPM platform (www.interstagebpm.com) also leverages these capabilities to provide enterprises with their own secure BPM space.

The engine offers an Application Programming Interface (API) called the “Model API”. The engine can be extended with Java classes to access the same capabilities outside of the system. The net result is that an application designed to run on one of the engines will run on another engine as well.

The functionality within Interstage BPM can be classified as follows (Figure 11):

- **User Interface Tier** - All the user interfaces (UI) for the system are implemented either on top of the Model API or the Web Services API. The web-based console that is provided with the product also utilizes the Model API layer to interact with the server.

- **Execution Tier** - The main process logic and the event processing engine for analytics reside in the BPM tier.

- **Integration and Connectivity Tier** - The third tier provides the integration hooks to the underlying repositories (database, directory and document management) as well as connectivity to other systems using a variety of mechanisms

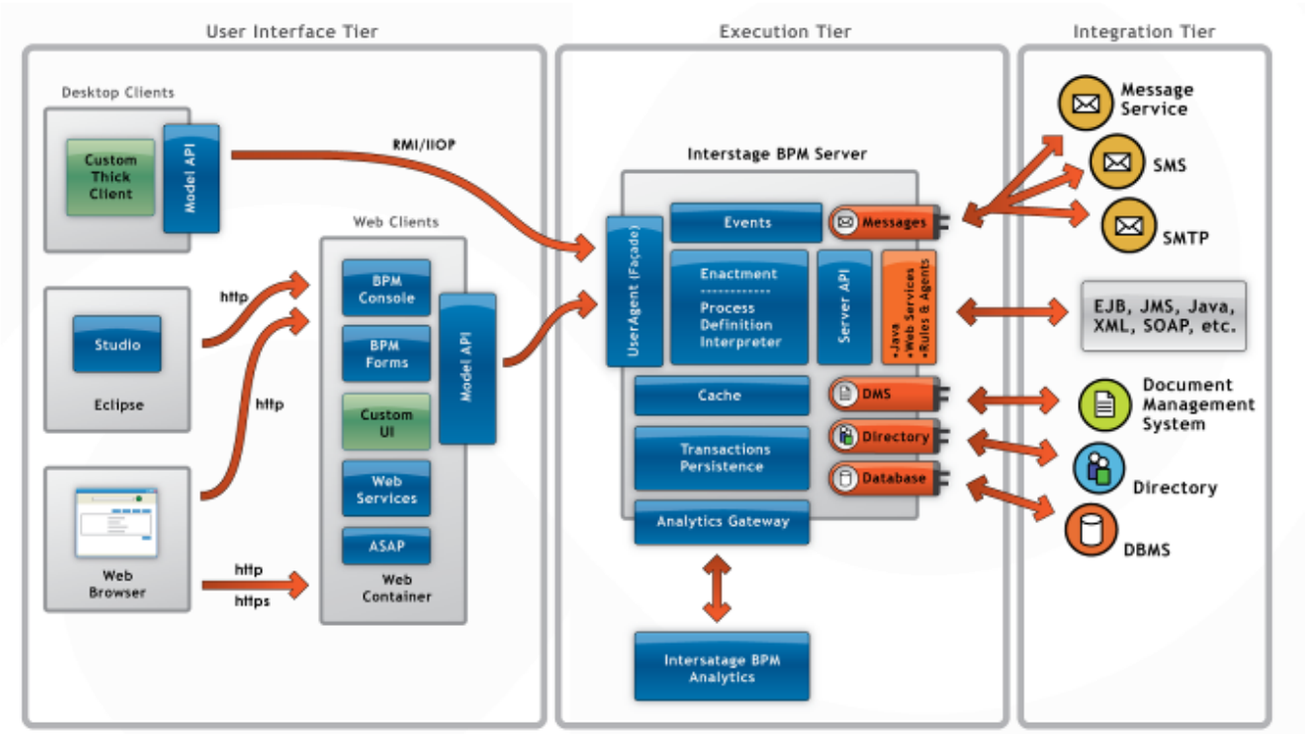

Figure 11: Interstage BPM Architecture

User Interface Tier

The User Interface Tier consists of the following components:

- Studio - An eclipse-based business process application development tool. Studio communicates to the execution tier through web services for application deployment and testing.

- Console - A broad selection of out-of-the-box, browser-based user interface modules are provided to give customers a head start.

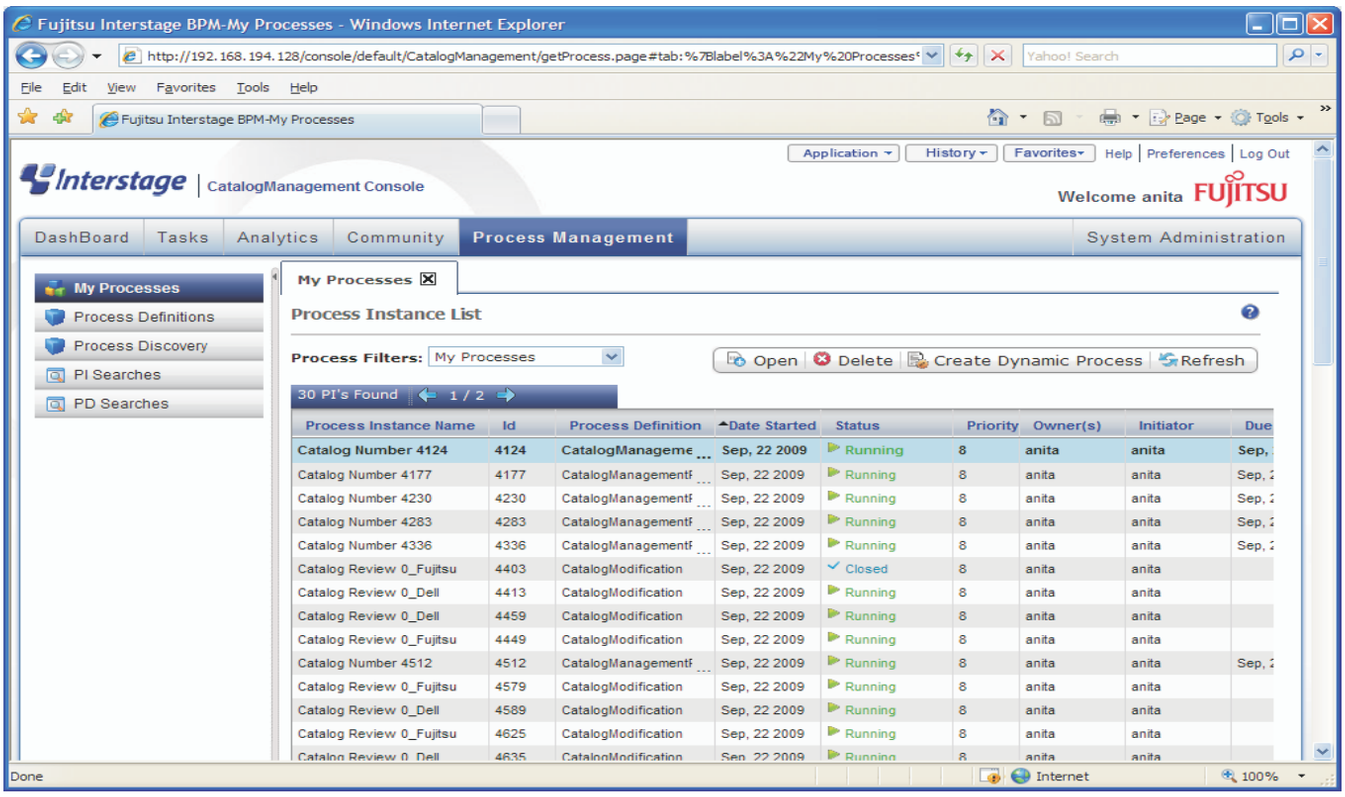

Figure 12: Interstage BPM Console

This console (Figure 12) is designed for use by an end user and provides a single interface for users to design, execute, manage, and monitor processes. It provides access to user tasks, an individual's list of running processes, reports, KPIs, alerts and the list of process definitions via a personalized dashboard (Figure 13).

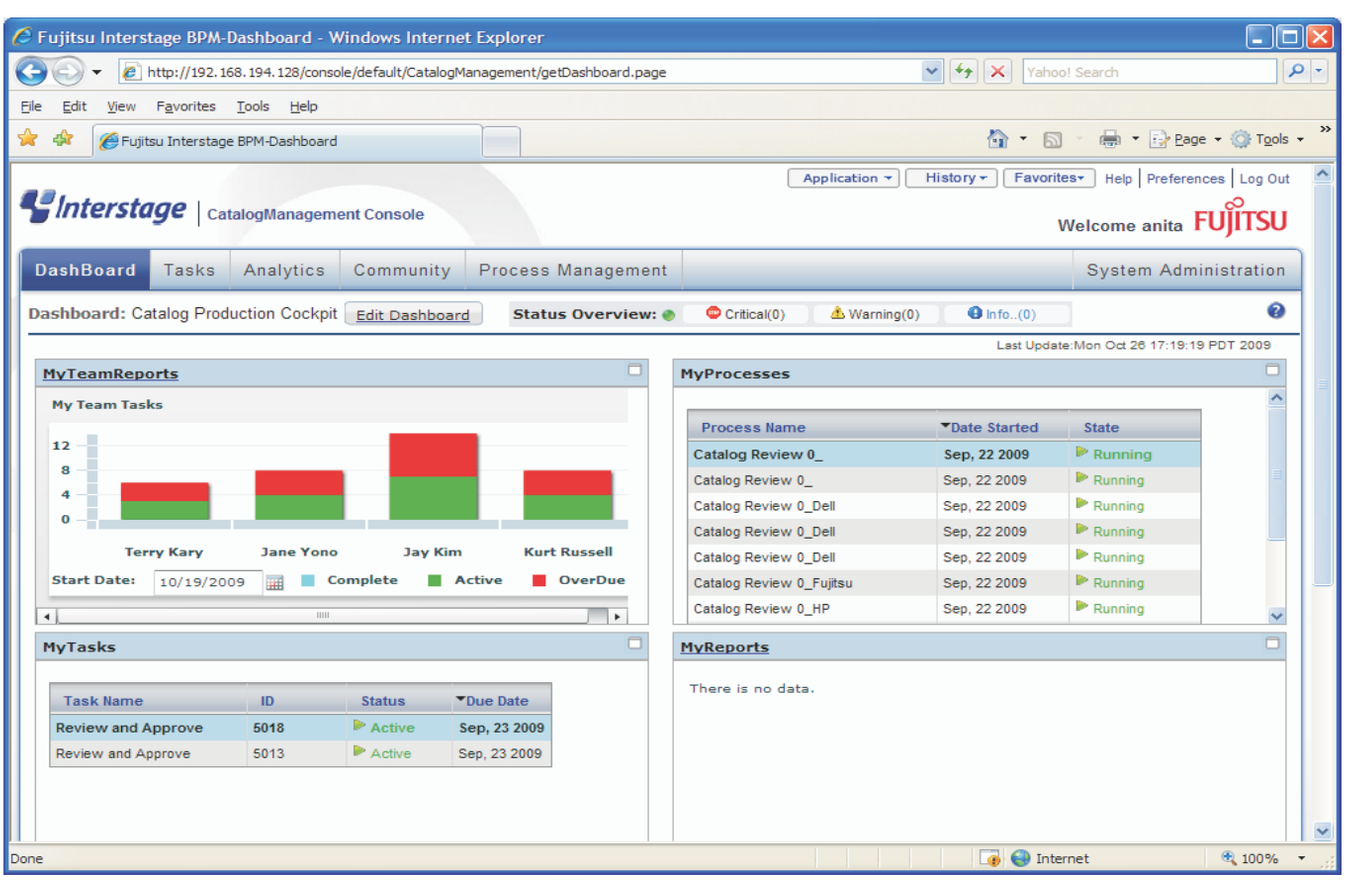

Figure 13: Interstage BPM Dashboards

Server administrators use the console to import, export, archive, and delete templates as well as modify and delete user profiles (Figure 14). It is also used to create new versions of existing process definitions. It is not likely that a customer will want to change or modify these pages, but all the administration functionality is available via the Model API as well. Server administrators can also manage the installed applications and start, stop, or update each application dynamically without having to restart the BPM server.

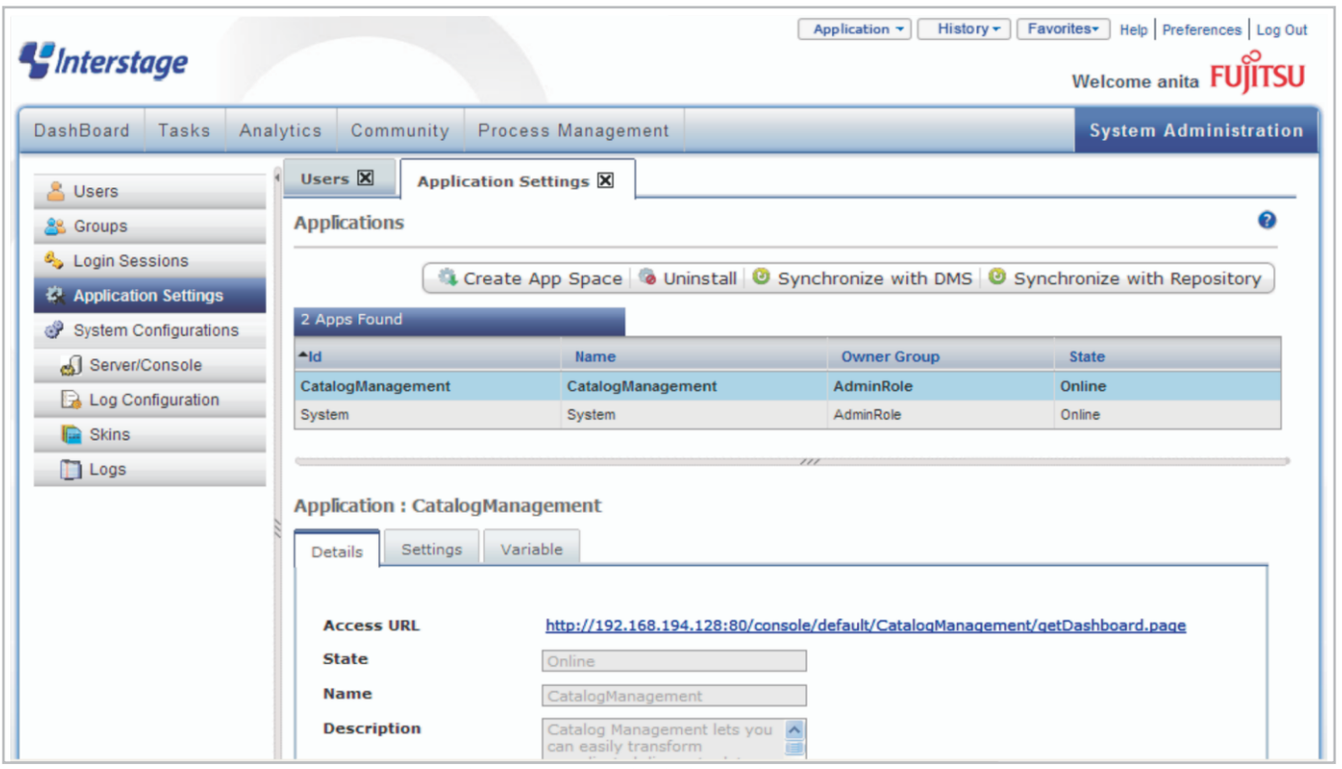

Figure 14: Administration Pages

All interactions between the interface tier and the BPM server are accomplished through the model API interface. The model API provides fine-grained access to process instances, tasks and process definitions. In order to optimize transactions on the server, the model API always retrieves an entire process definition or process instance at once. This is cached in a process instance proxy object (Figure 15).

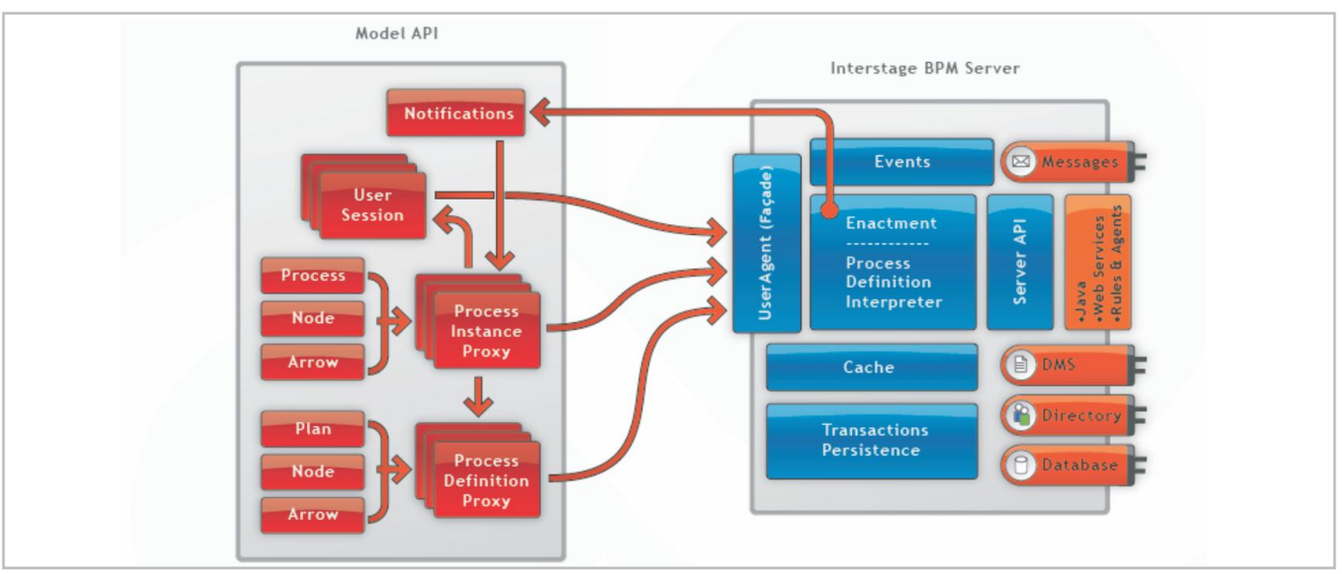

Figure 15: Model API Details

Notification events are sent to keep the caches up to date. If a different user changes a process, an event tells the model API that the process has changed, which invalidates the copy in the cache. The next access from the client will have the most up to date version of the object in question. When the client applications issues an update command, the proxy gets locked for edit by the server, and then maintains an edit version separate from the cache. Finally, upon committing the change, the results of all the edits are sent to the server to update the database in a single transaction and release the lock.

Execution Tier

Figure 11 shows the major components of the execution tier. The code is structured as higher-level business component that manages complex interactions between lower-level decouples lower-level business components from one another, making designs more flexible and comprehensible. It allows them to be deployed into a standard application server that isolates the applications from operating system and database differences. The Interstage BPM engine can run in a clustered environment with full fail-over capabilities which means that if one server goes down, the users will be switched over to the other servers without interruption to their work.

The server is composed of a collection of EJBs that run within an application server, leveraging its functionality. The Interstage BPM EJBs participate in container transactions so that the server and any client application can participate in the same transactions. Container-based transactions ensure a consistent state of the server. A UserAgent instance represents the login session to the server. It holds information for that particular login session. As the client's agent, the UserAgent makes bean requests and method calls to the other Interstage BPM components on behalf of its client - i.e., it acts as a gateway for the model API to access the process definition, process instance, tasks, directory, and other objects. The UserAgent bean is also referred to as the Façade, representing the interface between the server and the model API.

The process definition interpreter or the enactment engine forms the heart of the execution layer. It is responsible for enacting a process defined with Interstage BPM. The server communicates with the database adapter to maintain process state data, process instance and activity-related data, and process history information. Process enactment events are encapsulated in JMS messages that the message-driven beans (MDBs) process. The MDBs realize the flow of information between the server components by means of asynchronous messages.

The analytics gateway is used to send business process events from the BPM server to the Analytics component for analysis and evaluation. The analytics components monitors business activity in real-time and lets users drill-down into and slice and dice data for extensive process analysis. The analytics component has integration framework with Fujitsu's unique automated process discovery capability, that allows users to clearly visualize the structured and unstructured paths in a process thereby providing the real state of the executed business process. The built-in business process monitoring designer lets

users define key performance indicators, alerts and drill down charts so as to be able to track process trends, monitor KPIs and visualize bottlenecks within the process.

Interstage BPM maintains a history of every event that occurs during the execution of a process. A number of attributes are recorded for each event - type, source, target, timestamp, and the user responsible for the event. All events are recorded in the process history (also known as audit data). Users can view this process history from the administration interface.

Integration Tier

Adapters

The server has a number of defined service adapters which allow the server to access critical resources. Fujitsu supplies adapters for most systems required for business process deployment.

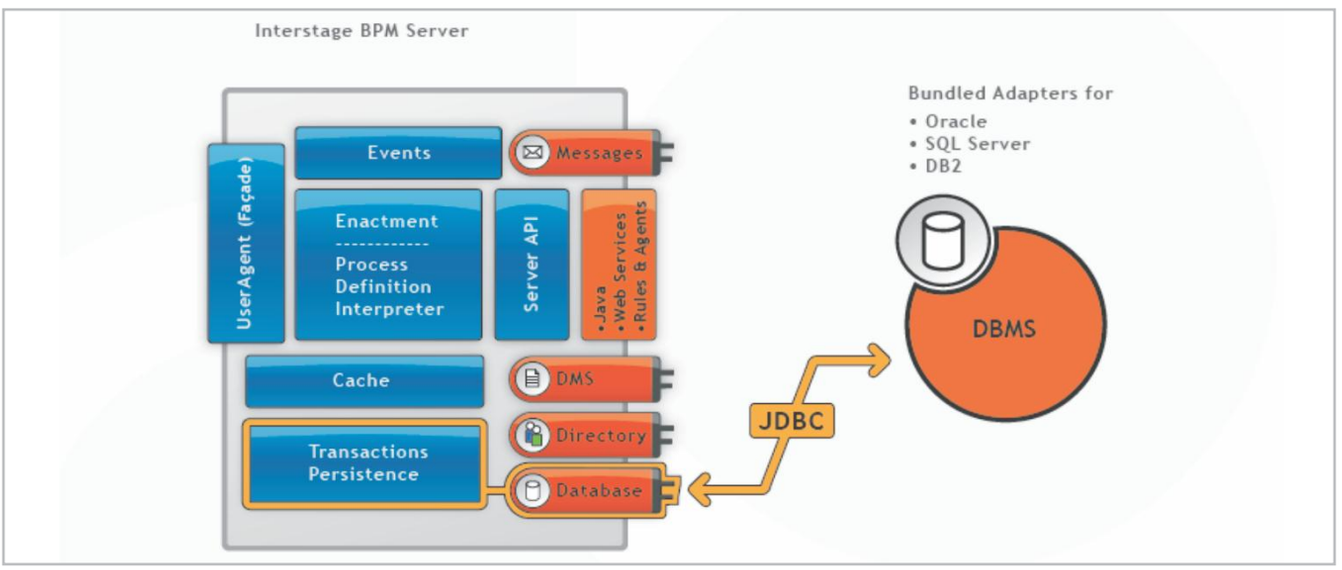

Figure 16: Database Adapters

The Database (DB) Adapter is written in Java, and must implement the special DB Adapter interface (Figure 16). This adapter is responsible for the translation of internal server objects into various persistent formats. Included with the base system is an adapter that persists the structures in a relational database using JDBC and stored procedures. SQL DDL scripts are available to configure Oracle, DB2 and SQL Server database servers. Other adapters are certified on demand.

The Directory Adapter and the Authentication Adapter are actually two separate adapters, but their functions are closely related.

The directory adapter implements an Interstage BPM-specific interface to expand a group into a list of individuals. The enactment engine uses this at runtime to determine work assignments. The ability to read the directory or to expand groups into a list of individuals is exposed by the Model API.

Interstage BPM provides the following directory adapters (Figure 17):

- LDAP connectivity to standard LDAP directory, such as Sun ONE Directory Server
- LDAP connectivity to Microsoft Active Directory
- Built in Local User Management – ideal for departmental solutions

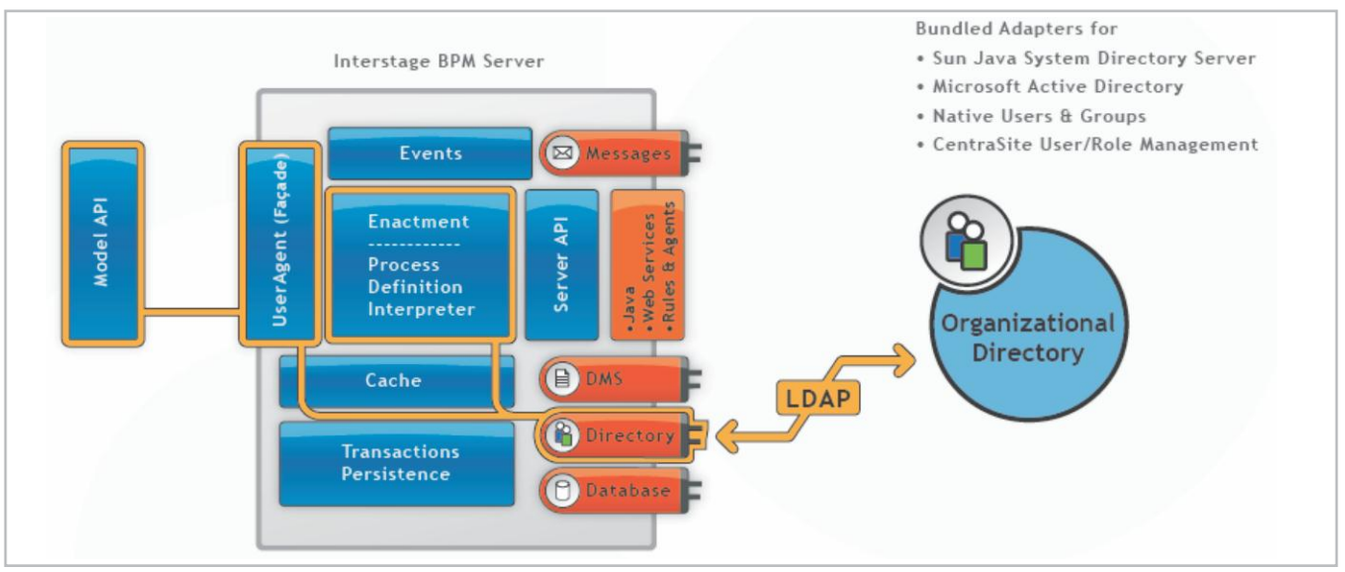

Figure 17: Directory Adapters

Some applications manage their own users in a custom database. In these cases, a custom Directory Adapter can be developed. Implementing a new directory adapter is a straightforward and well documented task. Interstage BPM customers have successfully implemented a number of specialized adapters for proprietary directory systems.

The Authentication Adapter is used by the User Agent at login time to authenticate users. Interstage BPM ships with a default adapter which uses the Directory Adapter to authenticate users. This addresses most situations. Special cases can be addressed as follows:

- Single Sign-on: If the organization is using an SSO facility (e.g., Netegrity, WebSSO) the central sign-on facility hands a token instead of a user password. In this case, a custom Authentication Adapter will be used to verify the token.
- Custom Authentication: Some applications will offer their own authentication schemes or password encryption mechanisms. A custom authentication adapter is required to support this scenario. Implementing a custom authentication adapter is a simple task, generally requiring no more than a few hours of work.
- No Authentication: Some applications will authenticate users before they have any access to the BPM system. In this case, authentication by the server can be disabled by using a custom authentication adapter which does not read the password field.

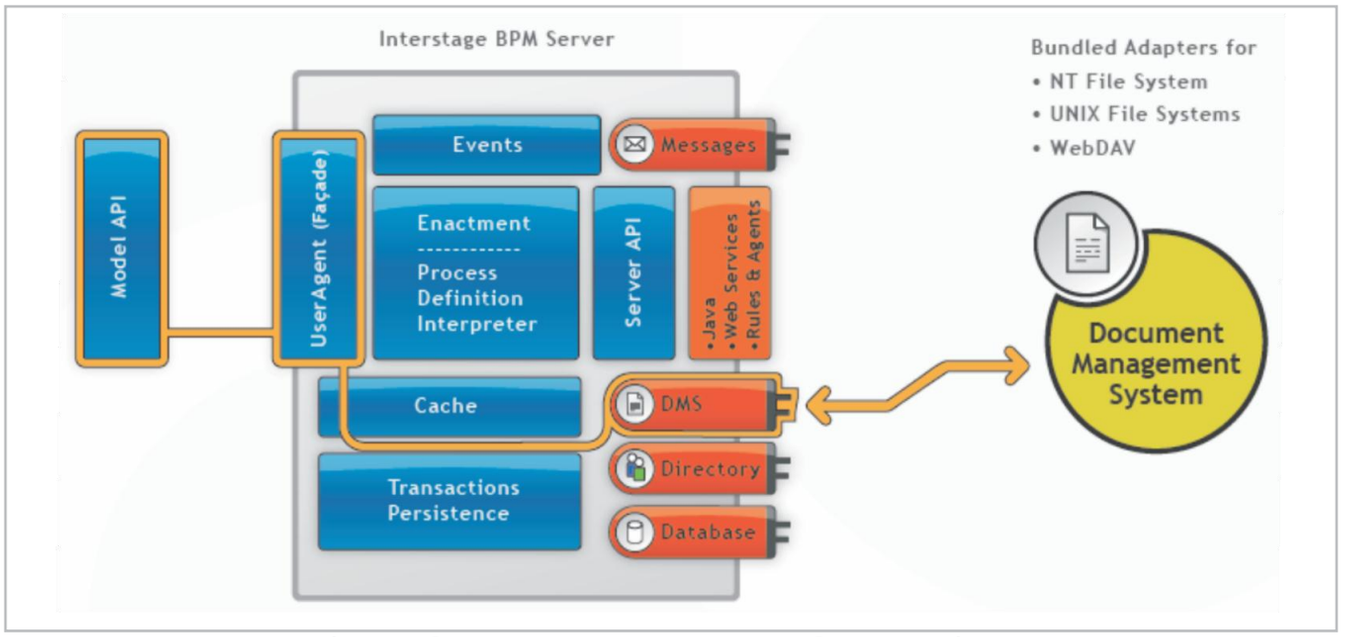

Figure 18: Document Management System Adapter

The Document Management System (DMS) Adapter is used to connect the Interstage BPM system to a document management system (Figure 18). Documents are generally stored outside of Interstage BPM and a locator for the document is stored in the attachment attributes of a process. DMS operations are provided on the model API to give access to documents in a consistent way, regardless of the document management system being used. Interstage BPM provides a DMS adapter that accesses documents stored on a Windows 2000, Unix file systems and WebDav enabled DMSs.

Interstage BPM provides an SMTP adapter for sending email notifications. Emails can be sent automatically as a result of assigning a task to a user. Emails can also be sent programmatically either from a Java Action or as an action associated with a timer. Email listeners are also provided that allow email messages to trigger a process or a complete a step in the process.

Integration within the process

Interstage BPM offers rich and powerful integration with Java which can, in turn, make any type of request to external systems (Figure 19).

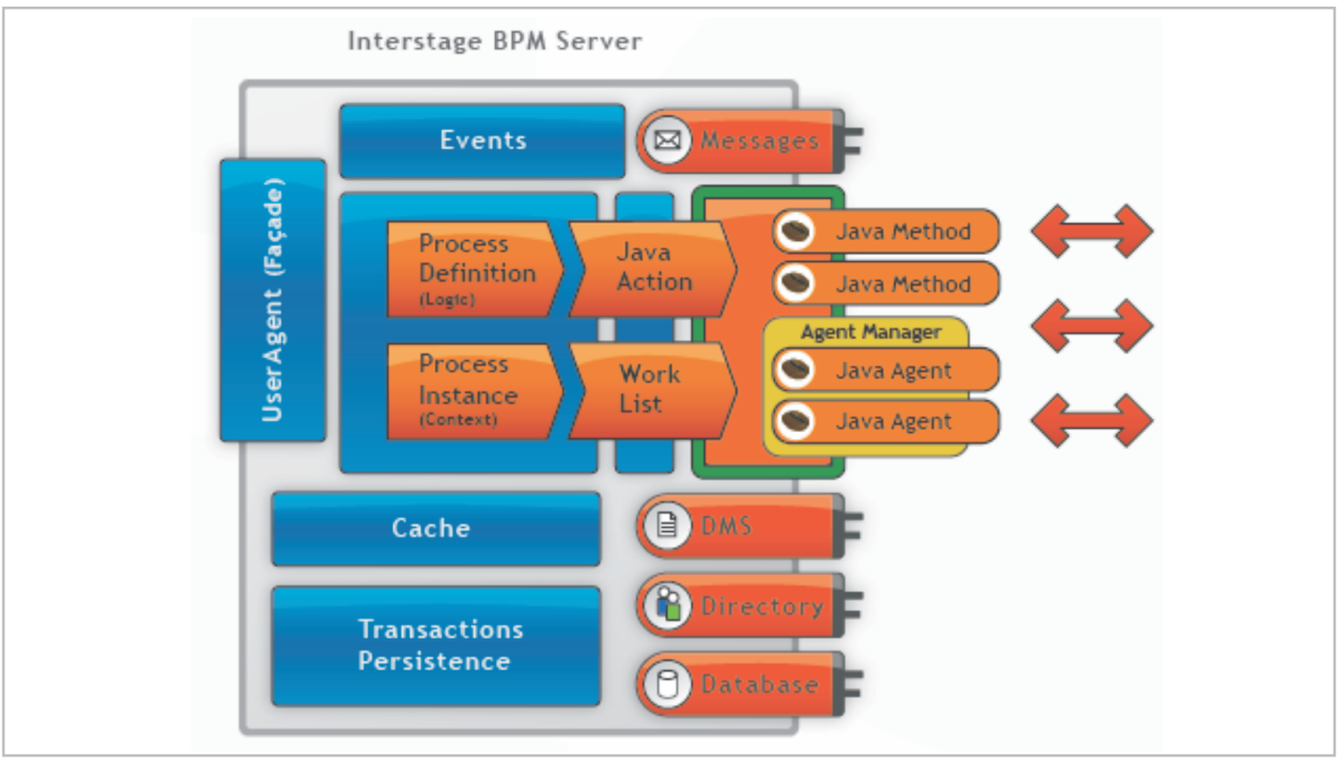

Figure 19: Integration Capabilities

Enactment is always triggered by an event, either a user indicating a choice to complete an activity, or by an event scheduled in the previous phase of enactment. Once triggered, the execution engine brings into memory the process definition and process context which includes all of the current User Defined Attribute (UDA) or process variable values as well as the status of each activity. During the transition from one process state to another, the process logic can evaluate any number of Java Actions.

Java Actions are data structures in the process definition that tell the BPM server how to call a particular Java method during execution. These method calls customize process enactment and allow execution of Java business methods outside the scope of the Interstage BPM Server. Java Actions make application integration easier and calls to external applications and adapters faster.

The Java Action has access to the process context and can manipulate UDA values directly. A Java Action can also call any Java class in the Java class path. Because a Java Action can hold and manipulate Java objects, a sequence of calls can be made by passing Java objects around. Every Java Action can have one or more Error Actions or Compensation Actions. Error Actions provide the mechanism to handle exceptions that arise from a particular integration call. Compensation actions can be defined to take care of a regular Java Action – e.g., for cleaning up the system and ensuring a consistent state of external systems involved in a transaction. In addition to pre-defined sets of Java Actions, a Generic Java Action is a call to a static method on a specified Java class and includes an extension to the class path setting to allow the class

to be found anywhere on the system. The Java Action is configured to pass UDA values as parameters, and the result of the method is assigned to a UDA. The Java Action can be passed the process context as a parameter, allowing the method to manipulate the UDA values in any way.

Agents work differently as they are not called by a Java Action. Instead, an activity is assigned to them by name. The Agent Manager maintains an association between these names and a Java class, along with a configuration file. When an activity that is assigned to an agent becomes active, a work item appears for this agent. The Agent Manager passes the work item, along with a reference to the process context, to the Java object. That Java object can then read and manipulate the UDA values as necessary. Typically, it will also access external systems. It can either complete the activity (by specifying a choice) or not. If it does not complete the activity, the Agent Manager will continue to prompt it at defined intervals of time until the activity is completed. This mode of integrating Java is particularly convenient when multiple retries to the underlying service may be required.

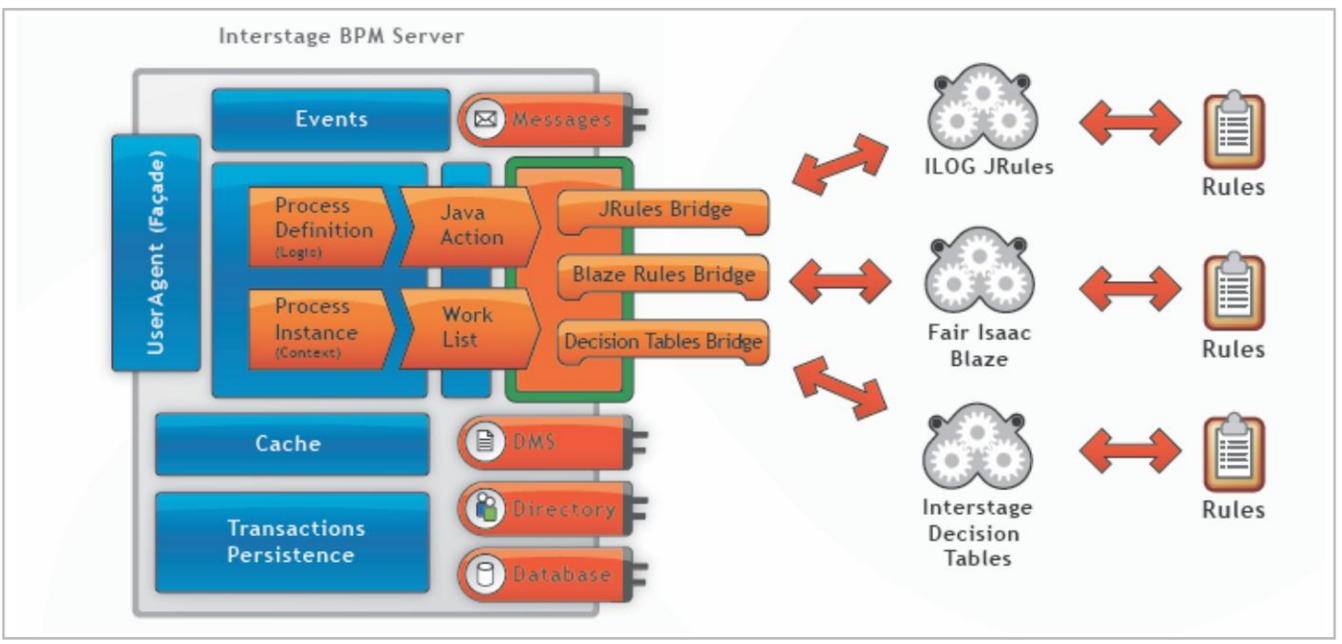

Figure 20: Rule Management Capabilities

The Rules Engine Bridge is a Java Action that invokes the Interstage BPM Decision Tables (built-in rule engine), iLOG JRules Engine or Fair Isaac Blaze Rule Engine (Figure 20). The user only needs to provide the name of the rules file, the set of rules to invoke, and give the result a name for reference elsewhere. When a process invokes the Rules Engine Bridge, the rules file is read and the rules are evaluated. The rules engine has been extended with a set of functions that allow UDA values to be read and tested. UDA values can also be set as the result of a rule. The rules have all of the same capabilities that are available from JavaScript.

Java Actions can be invoked at the following points of process execution:

- at process initialization (Init Actions and Process Owner Actions)
- on process completion (Commit Actions)

- before an activity starts (Prologue Actions and Role Actions)
- on activity completion (Epilogue Actions)
- when a timer expires (Timer Action)
- when a process instance is aborted, suspended, or resumed (onAbort, onSuspend, or onResume Action)
- when an error occurs during process execution (Error Actions)
- when an error occurs during the execution of a Java Action (Error Actions and Compensate Actions)
- when the starting of a remote subprocess fails (Error Action for a Remote Subprocess Node)

Web services

When integration to other/external systems is required, there are easy and powerful ways to invoke web services so that business users can subsequently use them in a drag-and-drop fashion.

Web services can easily be integrated into the process. They can be invoked from any step in the process. In addition, there is a purpose-built Web Service node that can be added to the process model to graphically represent a web service call.

It is very easy to incorporate existing web services into a process via our web service node. Simply drag the web service node onto your workspace, connect the arrows and configure the property sheet. The Web Services integration wizard in Interstage BPM Studio allows users to find and consume web services from the UDDI compliant registry. Interstage BPM can browse UDDI registries to select a WSDL document and introspect its operations. In the Web Service Action Editor, the Input tab displays the input and output parameters either as a grid or in XML. Process data is mapped to the input parameters or the web service.

On top of these features for consuming / invoking web services, Interstage BPM exposes much of its core functionality as web services as well, enabling it to be driven/invoked by other web service clients - this is especially important as organizations embrace SOA. Every process created within Interstage BPM is automatically a web service and can be published into a SOA registry that supports UDDI. (Figure 21)

There are various services exposed for the corresponding process to initiate, update, query process information's and states. WSDL (to support kicking off a process as a web service) and ASAP (to support kicking off a process from another process) representations of the process are provided out-of-the box via a link within the Interstage BPM console registry which acts as a service component catalog contains IT assets, it holds the pointer/metadata to an IT asset; it is a catalogue comparable to a Yellow Pages. A SOA registry does exactly the same for IT-services. It lists IT-services and describes how these services can be invoked/ accessed, with a brief description.

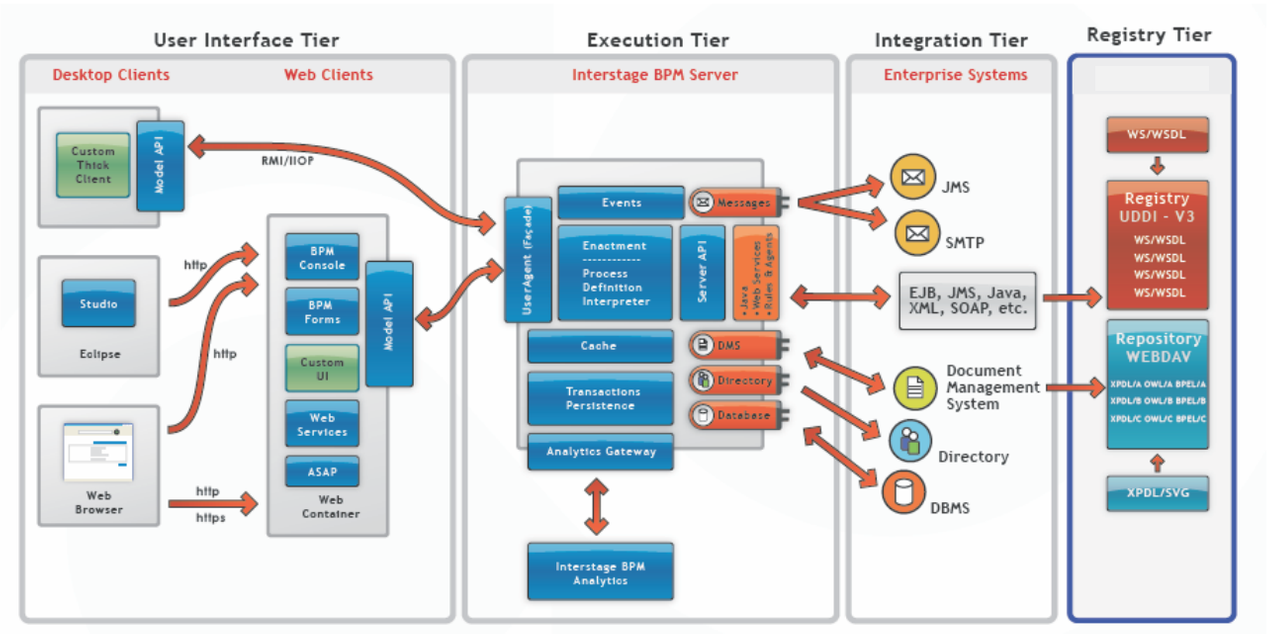

Figure 21 – Interstage BPM SOA Architecture

UDDI is an industry standard and serves the known registry functionality such as publishing, discovering and retrieving Web Services. WebDAV is another industry standard that is leveraged to help with storing and retrieving development artifacts. Interstage BPM provides the capability of publishing metadata into a WebDAV-supported repository.

Listeners

Interstage Business Process Manager comes with pre-built File, JMS, Email and SOAP listeners for performing the most critical BPM functions. All listeners are built using a trigger framework which provides the capability to configure a trigger source to initiate a process or complete a step in the process. Each trigger source is required to send the payload in XML format.

The File Listener can be configured to poll any directory. When a file appears in that directory, a process is kicked off with the file attached. This becomes an easy way to integrate with external systems. The process that is started can be programmed to manipulate the data within the file further. The File Listener can be used in conjunction with an FTP server. The Listener watches a directory managed by the FTP server. Whenever a new file is transmitted via FTP to the server, it is picked up and a process is started.

The SOAP Listener is configured to receive a particular SOAP request. After a message has arrived, a process will be started and the XML message in the request is attached. Further processing can be performed within the Interstage BPM process, and a SOAP response can be generated when necessary.

Email Listeners monitor incoming email messages containing XML payload as email attachments using the POP interface. When they detect messages containing these attachments, functions like starting process instances or making choices on activities can be performed.

Load Balancing and Fail-Over

Interstage BPM can be run on a set of servers configured through the use of an application server into a cluster. The application server provides load balancing capabilities so that each client instance gets connected to a different server based on which server has the lightest load or based on load balancing algorithms configured in the application server. If a server fails for any reason, running process instances are re-routed to other running servers transparently, without

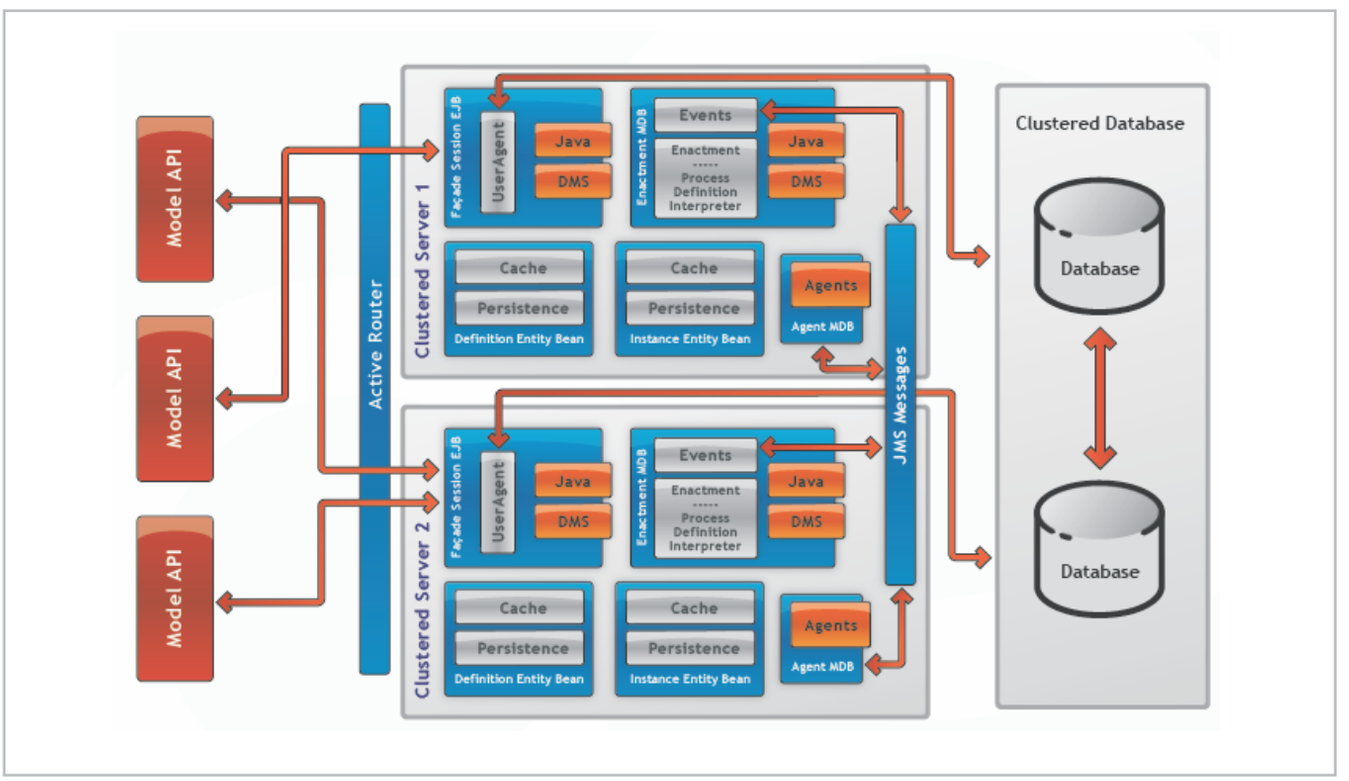

Figure 22: Load Balancing and Fail-over

- User sessions are recorded in the database. If a server fails, the model data traffic can be re-directed to a different running server, and that same user session is re-established to ensure that the communication continues unhindered.
- The cache listens for version change events sent from the other servers. If the cache identifies an object that has changed, it invalidates the cache, forcing a reload the next time that object is required.
- Edit sessions are recorded in the database to ensure that only one server locks in any particular process at any given time.

Conclusion

Interstage BPM's superior architecture lets business users and IT professionals collaborate on defining and refining business processes. It empowers knowledge workers to stretch processes to fit to the business needs. It provides developers with the ability to easily create and deploy high performance process-based applications. Interstage BPM is easily embedded into existing applications and systems and delivers a broad range of functionality along with a set of integration adapters that let organizations leverage their existing IT investments. The core elements of the Interstage BPM architecture include:

- A web-centric design that provides strong support for both human interactions and application integration while enabling process change anywhere, anytime.
- Unique multi-tenancy capabilities to enable cloud deployments. It allows companies to onboard multiple tenants and then deploy multiple applications for each of these tenants on a single instance of BPM. Each application is hosted as a service that customers can access via the internet. This makes Interstage BPM the ideal choice for enterprises that want to create private Cloud BPM deployments as well as for application service providers (ASPs). We also offer a public BPM cloud through www.interstagebpm.com.
- Comprehensive Java and Web service interfaces to utilize the complete functionality of the engine.
- Process optimization based on visualizations of real business process patterns – of processes that include both structured and unstructured activities.

This broad array of features makes Interstage BPM the ideal platform for developing enterprise-wide process-based applications, as well as for embedding BPM functionality into third-party applications and systems.