Added by Jacques-Alexandre Gerber, last edited by Shao Fang on Sep 07, 2008

Architecture

Tempo is highly componentized to offer developers maximum flexibility in replacing and changing any of these components according to their needs.
At a high level, Tempo can be seen as a three-tier architecture with, from a top-down perspective:

- The user interface where end users manage their task lists. Tempo has two components to provide the user interface: the User Interface Framework and the XForms Manager.
- The workflow logic where the tasks lifecycle is managed. This is implemented by a set of WS-BPEL processes called the Task Management Processes accessible through a Web Service interface.
- The persistence layer that is responsible for persisting task properties in a secure manner. This is implemented by the Task Management Service accessible through a Web Service interface.

The other components provide additional/optional functionality to facilitate the integration and administration of Tempo in the overall BPEL environment:

- The Security Framework is a Role Based Access Control framework for security (authorization, authentication, single-sign-on).
- The Task Attachment Service is an interface to store attachments in a data storage or Content Management System.
- The Form Dispatcher Service acts as a proxy between the Task Management Processes and the User Interface Framework.
- The Workflow Deployment Service provides an interface to deploy workflow artifacts in the workflow database and provide other components access to these artifacts.

Finally, the Task Object Model defines the task properties in a common package that is reused in other components.

The following pages provide information about how this works together for:

- Creating and Completing a Task

XForms Manager (XFM)

The XForms Manager (XFM) is responsible for rendering XForms code and providing workflow actions for such forms. It is invoked by the the User Interface Framework when the user clicks on the task in the task list for which the form is an XForms document. XFM then invokes the Task Management Service to retrieve the specific task data and gets the XForms form through the Workflow Deployment Service. This renders the form when the user selects a task in the task list. XFM also adds tools for the workflow actions such as a button to submit/complete a task, a tool for managing attachments and more. This mechanism allows new actions to be added without impacting the code of the forms themselves. Indeed the workflow actions are code in the Form Manager, not in the forms. Form developers can focus on developing forms without worrying about the workflow actions. XFM however only deals with Xforms forms and workflow actions on these forms. In particular, it does not provide the task list nor support other types of forms.

XFM uses Orbeon Presentation Server for supporting XForms. It also uses Orbeon XPL language for implementing workflow actions and invoking the Task Management Service and BPEL processes exposed as web services.

XFM deploys as a WAR file on virtually any J2EE Application Server or Servlet Container such as Apache Geronimo.

User Interface Framework (UIFW)

The User Interface Framework (UIFW) is the web application that give users access to workflow functionality. It provides the login screen and task list. Is it responsible for serving the appropriate Form Manager when the user selects a task. At this point UIFW only supports the XForms Manager that is responsible for XForms forms, but other Form Manager for other types of forms could be plugged in. Form Managers are automatically selected based on the Form URL, which is a meta-data property of each task. Depending on the Form URL, different Form Managers could be invoked to render the form that supports the selected task.

UIFW is implemented using the Spring framework and deploys as a WAR file on virtually any J2EE Application Server or Servlet Container such as Apache Geronimo.

Task Management Processes (TMP)

The Task Management Processes (TMP) supports the lifecycle of workflow tasks from the moment a task is created till it's completed. It's responsible for changing task states according to rules and user interactions as defined in these processes.
The TMP invokes the Task Management Service to change task states in a reliable and secure manner. It provides services to let users perform workflow actions. It also interacts with BPEL processes where workflow activities are used, through the Form Dispatcher Service.

TMP is implemented in WS-BPEL 2.0 and deploys on virtually any WS-BPEL 2.0 compliant engine such as Apache Ode.

Task Management Service (TMS)

The Task Management Service (TMS) is a service that persists task data in the underlying database and provides services for client applications to access and change task data in a secure manner. TMS is used by the User Interface Framework to retrieve the task list, the XForms Manager to retrieve task data and the Task Management Processes to change task states

Task Management Processes to change task state.

TMS is implemented in Java as an Axis2 service and as such provides a Web Service interface. It deploys on Apache Axis2.

Security Framework (SFW)

The Security Framework (SFW) provides a Role Based Access Control interface to security systems, essentially for authorization, authentication and single sign-on. It is used by the User Interface Framework for authenticating users at login and by the Task Management Service for authorizing any TMS call. Different implementation of this interface can be plugged in Tempo in order to integrate with various security systems. Tempo provides a default implementation that uses an XML file to define users and roles. The framework also supports retrieving user and role hierarchies from a LDAP directory server.

SFW is implemented in Java and deploys as a JAR file.

Task Attachment Service (TAS)

The Task Attachment Service (TAS) is a service that persists attachments linked to tasks. The API supports adding and removing attachments (binary files) along with some description and content-type.

TAS is implemented as a Java servlet and deploys as a WAR file.

Form Dispatcher Service (FDS)

The Form Dispatcher Service (FDS) converts between user-process message representation and task-management process message representation. This is necessary to provide specific task creation/completion schemas for the user processes while providing the ability for the task management processes generically handle any task type.

FDS is implemented as a Java servlet and deploys as a WAR file.

Workflow Deployment Service (WDS)

The Workflow Deployment Service (WDS) allows design-time and automation tools to remotely deploy task descriptions and form content. WDS registers existing and available task definitions into the Task Object Model persistent store.

WDS is implemented as a Java servlet and deploys as a WAR file.

Task Object Model (TOM)

The Task Object Model is a data-access layer to create, query and manage task definitions and task instances.

TOM is implemented as a Java library (.jar)

Children Hide Children | View in hierarchy

Creating and Completing a Task (Tempo)

Comments