Workflow Management Component

The Workflow Manager component is responsible for description, execution, and monitoring of Workflows, using a client-server system. Workflows are typically considered to be sequences of tasks, joined together by control flow and data flow, which must execute in some ordered fashion. Workflows typically generate output data, perform routine management tasks (such as sending emails, etc.), and/or describe a business's internal routine practices. The Workflow Manager is an extensible software component that provides an XML-RPC external interface, and a fully tailorable (i.e. adaptable) Java-based API for workflow management.

Architecture

This section describes the Workflow Manager architecture, including its constituent components, object model, and extension points.

Architectural Components

The major architectural components of Workflow Manager are the Client and Server, the Workflow Repository, the Workflow Engine, and the Workflow Instance Repository. The relationship between all of these components are shown in Figure 1.

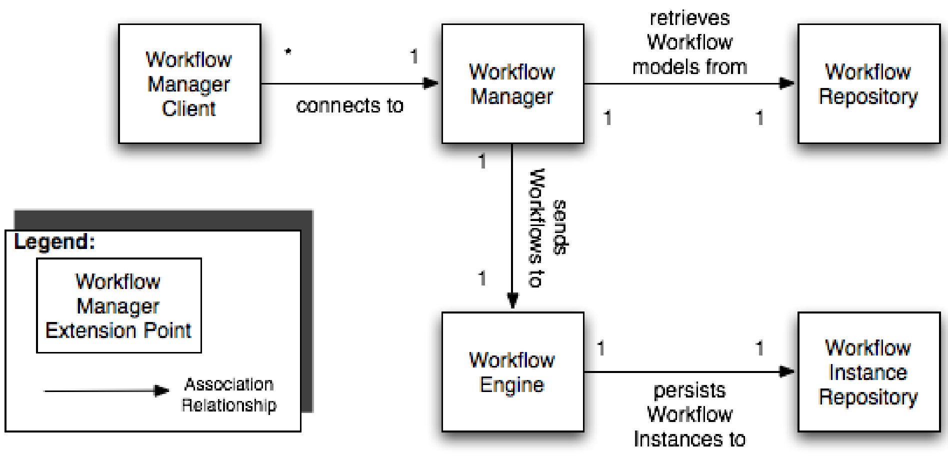

Figure 1. Architectural components of Workflow Manager

The Workflow Manager Server contains both a Workflow Repository that manages workflow models, and Workflow Engine that processes workflow instances. The Workflow Engine also has a persistence layer, called a Workflow Instance Repository, which is responsible for saving workflow instance metadata and state.

Object Model

The critical objects managed by Workflow Manager include:

- **Events** - are the stimuli that trigger Workflows to be executed. Events are named and contain dynamic Metadata information passed in by the user.
- **Metadata** - a dynamic set of properties and values provided to a *WorkflowInstance* via an user-triggered Event.

- **Workflow** - a description of both the control flow and data flow of a sequence of tasks (or stages) that must be executed in some order.
- **Workflow Instance** - an instance of a *Workflow*, typically containing additional runtime descriptive information, such as start time, end time, task clock time, etc. A *WorkflowInstance* also contains a shared *Metadata* context passed in by the user who triggered the *Workflow*. This context can be read/written to by the underlying *WorkflowTasks* present in a *Workflow*.
- **Workflow Tasks** - descriptions of data flow, and an underlying process, or stage, that is part of a *Workflow*.
- **Workflow Task Instances** - the actual executing code, or process, that performs the work in the *WorkflowTask*.
- **Workflow Task Configuration** - static configuration properties that configure a *WorkflowTask*.
- **Workflow Conditions** - any pre (or post) conditions on the execution of a *WorkflowTask*.
- **Workflow Condition Instances** - the actual executing code, or process, that performs the work in the *WorkflowCondition*.

Each Event kicks off one or more *Workflow* Instances, providing a *Metadata* context (submitted by an external user). Each *WorkflowInstance* is a run-time execution model of a *Workflow*. Each *Workflow* contains one or more *WorkflowTasks*. Each *WorkflowTask* contains a single *WorkflowTaskConfiguration*, and one or more *WorkflowConditions*. Each *WorkflowTask* has a corresponding *WorkflowTaskInstance* (that it models), as well as it does each *WorkflowCondition* has a corresponding *WorkflowConditionInstance*. These relationships are shown in Figure 2.

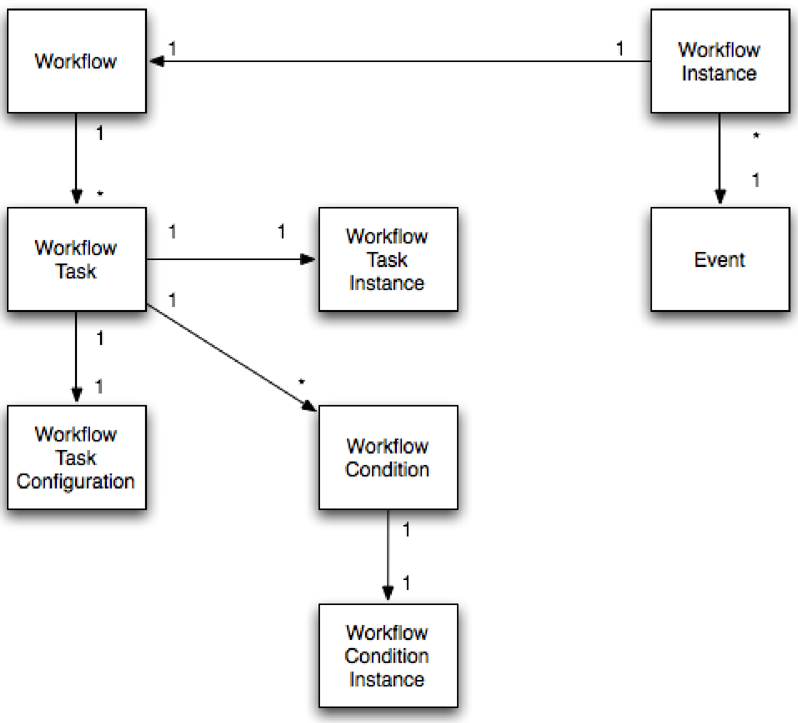

Figure 2. Object Model for Workflow Manager

Extension Points

The Workflow Manager was built with the Factory Method pattern to provide multiple extension points for the Workflow Manager. An extension point is an interface within the Workflow Manager that can have many implementations. This is particularly useful when it comes to software component configuration because it allows different implementations of an existing interface to be selected at deployment time. Each of the core extension points for Workflow Manager is described in TABLE I.

TABLE I. Extension Points of Workflow Manager

Extension Point	**Description**
Workflow Instance Repository	The Workflow Instance Repository extension point is responsible for storing all the instance data for Workflow Instances, including shared context metadata, and runtime properties, such as start date time and end date time.
Workflow Repository	The Workflow Repository extension point is responsible for managing Workflow models, storing control flows and data flows. The Workflow Repository also stores Workflow Condition information, and Workflow Task Configuration. In essence, the Workflow Repository is a repository of abstract Workflow models, which get mapped to Workflow Instances by the Engine extension point.
Workflow Engine	The Workflow Engine's responsibility is to map abstract Workflow models to executing Workflow Instances. The Workflow Engine tracks and monitors execution of Workflow Instances, and provides the ability to start, stop and pause executing Workflow Instances.
System	The extension point that provides the external interface to the Workflow Manager services. This includes the Workflow Manager server interface, as well as the associated Workflow Manager client interface, which communicates with the server.