

Workflow Overview

Introduction

The purpose of workflow is to automate business processes, in Daisy particularly those involving documents. A typical workflow process in Daisy is a document review, whereby a document might need to pass via several persons who need to give their approval.

Daisy uses a generic workflow engine, more specifically [jBPM](#)¹ (version 3.2). From a technical point of view, this workflow engine can be used to manage any sort of workflow process, even if it's completely unrelated to content management.

The structure of a workflow process is defined by a *process definition*. Such a process definition consists of nodes (of various types) with transitions between them. A specific running process (following the structure of a process definition) is called a *process instance*. Much like a normal computer program, a process instance has a pointer to where the execution is currently located (called the *token* in jBPM or the *execution path* in the Daisy API, workflow processes support multiple concurrent execution paths) and has scoped *variables*. An important difference between workflow processes and normal computer programs is that workflow processes can contain *wait states*, whereby the system needs to wait for someone (a user or another process) to perform some task. Since it might take quite some time before this task is performed, the process instances need to be persistable (storeable).

An important concept in workflow is the *task list*: task nodes in the workflow process create tasks that are assigned to users or pools of users, and appear in the task inbox of the users. The user has to perform the task and choose the transition to follow, in order to let the workflow process continue.

Workflow processes can also contain timer-triggered actions, which could e.g. be used for timed publication of documents.

For more generic information on workflow, we refer to the [jBPM website](#)² or [any other resource](#)³ on workflow. The documentation here is mainly focussed on the workflow/jBPM integration in Daisy.

What Daisy does with jBPM

For those already familiar with jBPM, here's a quick summary of what the jBPM integration in Daisy involves:

- Embedded deployment in the repository server
- A workflow API, with a query system
- A custom metadata layer and i18n
- Daisy Wiki-integrated GUI
- Some custom variable types to point to Daisy documents, users and actors
- Some utility classes for use in process definitions: a Javascript action, a mail action, an assignment handler

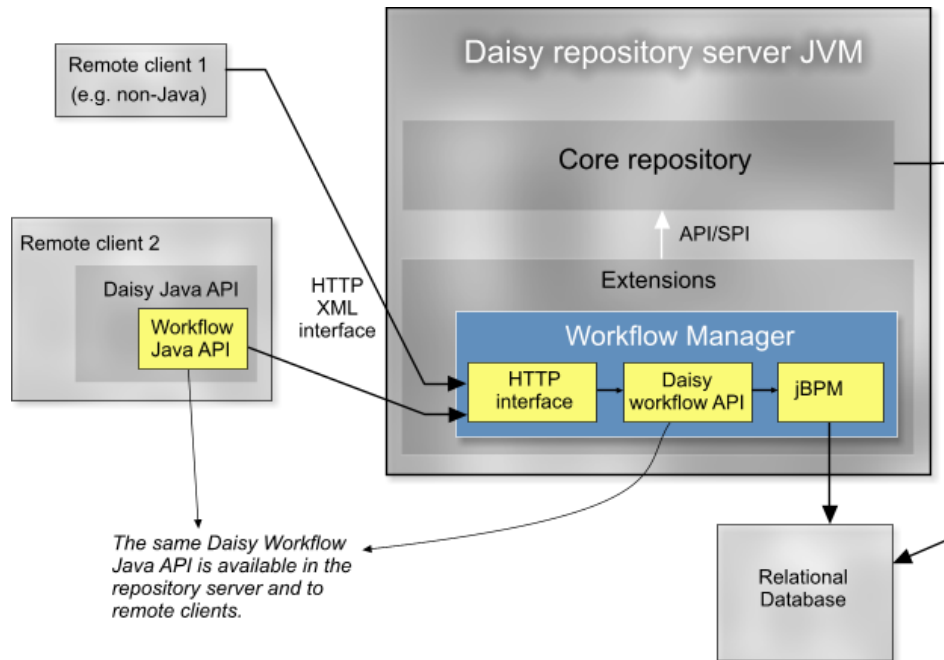
1. <http://www.jboss.com/products/jbpm>
2. <http://www.jboss.com/products/jbpm>
3. <http://www.google.com/search?q=workflow>

Workflow integration in Daisy

Workflow component in the repository server

The picture below illustrates how the workflow is implemented as an extension component in the Daisy repository server. “Extension component” means that the workflow can be disabled without any impact on the core repository server, and its code base is completely separated from the core repository, the workflow only communicates through official APIs/SPIs with the repository.

As discussed in the section on [deployment](#)⁴, we have set up jBPM so that it stores its data in the same relational database as the Daisy repository server.



Daisy workflow API

Daisy defines its own Workflow API. This not only hides the underlying implementation, but allows to layer additional functionality (such as access control), and to provide a remote implementation.

One of the neat features of the Daisy Workflow API is its generic [query system](#)⁵.

Daisy Wiki integration

Next to the workflow extension component in the repository server, the Daisy Wiki provides a complete workflow GUI, including:

- administration screens to deploy process definitions and manage workflow pools
- workflow console (dashboard) showing the tasks assigned to the current user and pooled tasks for the user
- the necessary screens to start workflow processes and perform workflow tasks

4. [daisy:361-cd](#) (Workflow deployment)

5. [daisy:353-cd](#) (Workflow query system)

- search screens for processes, tasks and timers
- and the obvious little things like deleting/suspending/resuming process instances, viewing the complete details of a process instance, (re-)assign tasks, view timer details (including exception if timer failed), ...