

Workflow

Motivation

Bringing in a third-party workflow engine would allow us to:

- better support workflow-like activities with a mature standard
- minimize the quantity of siloed Sakai code which seeks to handle workflow-like activities in each particular case

There is not a wealth of complex workflows in Sakai right now, but there has been an expressed need for some time (see

[+ SAK-7854](#) - Ability to configure workflow in tools CLOSED), and a few projects are on the cusp of requiring a workflow engine.

Bringing in a workflow engine or two at this point would not allow the excising of a large quantity of existing code (as the JCR work would), but it would allow us to get ahead of "workflow creep" of tools developing more elaborate and brittle logic in the absence of an engine available to satisfy their use cases.

Kinds of Workflow

There is a need for two kinds of workflow engine:

1. Long-lived workflows with externally driven delays (time or event) controlled by humans in roles, as typefied by workflows conforming to the concepts in WfMC often expressed in XPDL.
2. Automated workflows targeted at high transaction rates with little or no concept of time delays and human roles, as typefied by workflows defined in BPEL.

Projects Affected

- Assignments: see [Assignment Workflows](#)
- Libraries: [CTREP project](#)
- ESB-like integration for sources of group information within the university enterprise
- Copyright Clearance: [Open CourseWork Workflows](#)
- Portfolios: [Workflow Breakout Group](#) and [Multiple Evaluator Workflow - Functional design](#)

Technical Considerations

We would need to integrate through adapters into the core of the Sakai framework so that workflow implemented processes can manipulate the datamodel and services that form Sakai while maintaining an internal record of workflow state.

Ian Boston

For Sakai, IMHO, we need to abstract the workflow engine and its integration points behind some simple interfaces that may be both java interfaces or communication standards (eg data over http, xml documents, etc). Then we need to start expressing our workflows in a standard language, I would strongly suggest BPEL + XPDL, stored as documents in JCR.

There should be 2 workflow engines, one for each type of workflow, but no more than one per type, the BPEL workflow might even come as part of an ESB connector

Once we have this infrastructure in place we should look at migrating existing 'workflow' oriented operations into definitions to reduce the complexity and range of workflow implementations within Sakai.

Requirements

1. A set of functional and technical requirements are identified for short- and medium-term needs.
2. A technical approach is defined, preferring a standards-based approach when feasible
3. If the agreed-upon approach is standards-based, then an existing open source implementation that is compatible from both technical and licensing perspectives is preferred as the default implementation
4. When resources permit, testing against non-default implementations (both open and proprietary) is encouraged to ensure both robust standards compatibility and a wider range of choices for community members

References

- [XPDL](#)
- [BPEL](#)
- [Catalog of OS Java workflow offerings](#)
- [Kuali Enterprise Workflow KEW](#)

- Workflow patterns <http://is.tm.tue.nl/research/patterns/patterns.htm>