

Workflow

Ideas and opinions

March 2004 Johan posted '[Experience with workflow](#)', to the Cocoon developer list. The response to this post was overwhelming. The interest in workflow applications is enormous. Below is a summary of the information in the post and its responses.

Scope

In the discussion there were a lot of different views of what workflow encompasses. The requirements will differ depending on the scope which is chosen. This must be kept in mind while evaluating and comparing workflow solutions. Below is a summary of different areas for which the scope levels differed. Although the summary presents two scope levels for each area, it's not black and white and more scope levels are possible.

One of the first things in software engineering is to determine the end users of the software. In the discussion people clearly had different views on this subject. One group would like the developers to implement the workflow specified by the users, and another group would like a tool for end users with which they can define the workflow.

People were also applying workflow to different-sized problems. There is interest in a generalized workflow engine capable of building complete processes, and interest in remembering state for a single object only (object lifecycle).

What is clear from the discussion is that a workflow must not be attached to a particular type of object (documents for example), but that the workflow framework must be flexible enough to be able to attach a workflow to any object.

Requirements

Before workflow can be implemented a number of requirements must be met:

- understandable role assignment (from a user's perspective) and simple role retrieval;
- typed object repository. This is necessary to enable different object types having different workflows attached to them.

A workflow solution must be able:

- to specify under what conditions an action can be invoked. Authorization is considered a specific type of condition; (used for ensuring proper execution of business processes)
- to retrieve the actions which can be invoked by a particular user at this moment; (used for generating menus)
- to query for objects which are in a specific state and are relevant to the current user (i.e. the user can execute an action on them); (used for building task lists)
- to migrate workflow instances of existing objects to a new workflow definition. (evolution of business processes)

If workflow is implemented using state machines, the state machines must support (depends on final set of valid use cases):

- nested states;
- concurrent states.

Implementation

A number of implementations were suggested. And each implementation had its share of proponents. To come to a good evaluation of the implementations it was suggested to build one or more [WorkflowUseCases](#) in each implementation, and compare the results. Some of the criteria to consider are:

- simple workflows must be simple to implement;
- maintainability.

Existing frameworks

Existing frameworks, preferably based on standards, must be evaluated (licensing must be considered also). It is useless to reinvent the wheel. OSWorkflow and Lenya are frameworks with which people already have experience. These two are not based on standards however.

Finite state machines

OSWorkflow and Lenya are based on finite state machines (FSM). If their implementation would turn out not to be powerful enough, an implementation on the same principal of FSMs could be build. There are a number of ways in which FSMs can be implemented:

- using the State pattern as described in the GoF book;
- a code generator which uses (annotated) UML state diagrams as its input;
- an interpreter which uses XML files describing the state machine as its input.

Flow with continuations

Another implementation a lot of people were interested in was using flow with continuations. This can be implemented in a number of ways too:

- manual coding;
- a code generator which uses (annotated) UML state diagrams as its input.

Rule based

The last implementation uses rule evaluation (could somebody with a better understanding than I add some more information? Johan Stuyts). Possible ways to implement this:

- use XML for the data and XSLT for the rules (please verify Johan Stuyts).

Design issues

In the thread a number of design issues were pointed out.

Authorization

Authorization when using workflow is usually very fine grained: it is only possible to create a document in this folder if you are a member of department X. These authorization rules can be seen as a special type of conditions which are used to whether an action on a workflow is valid. But this mixes concerns: authorization and logic. There must be locations at which:

- authorization rules are defined;
- authorization rules are evaluated.

Workflow interface

Workflow uses multiple aspects of objects, its properties, its current state and authorization information, at the same time. All these are needed to be able to use queries like: find all news items for which user X is the editor, and which are waiting for review for at least an hour.

The properties of an object, the state of an object, and the authorization information are stored in different locations which should not query each other directly most of the time. It is undesirable to let the client of these different locations collect and merge this information. A separate component (using an existing interface) must handle this task.

Next steps

The first thing to do is to determine the scope of the needed workflow. If the scope is chosen too wide, the workflow implementation will be so big that a project separate from Cocoon is probably justifiable.

After the scope is clear existing workflow implementations can be used to implement a number of [WorkflowUseCases](#). After the evaluation of the results a decision can be made about whether an existing implementation suffices (or it is doable to modify it to make it suffice), or that a new product must be developed.

See also

- [WorkflowImplementationComparison](#)
- [JSR 207 - Process Definition for Java™](#)

Standards

- [WfMC XPD](#)
- [BPML](#)
- [OMG Workflow Management](#)
- [WSCI](#)
- [BPEL4WS](#)
- [XLang](#)
- [ebXML Business Process](#)
- [Glossary of ISO Standards](#)