



# Structured Pageflow

In earlier chapters, we explored the core features of the Seam framework, namely its component model (bridging JSF and EJB, among other things) and its extended set of runtime contexts/scopes (providing the concept of conversations, among other things). In this chapter, we begin to look at the ancillary features of the framework. I label these as “ancillary” not to downplay their importance or practical value. Quite the contrary, they are eminently useful. I call them “ancillary” because it is not likely that you will choose to adopt Seam simply to use these features. Under normal circumstances, you’ll adopt the core Seam features (Seam components and conversations), and then pick and choose from among the ancillary features (pageflow, business process modeling, AJAX support) to suit your needs.

This chapter is focused on Seam’s support for managing application pageflow using the jBPM Process Definition Language (jPDL). As the name implies, jPDL is part of jBPM (Java Business Process Management), which is the JBoss tool for defining workflows and orchestrating tasks within a business process. Roughly speaking, jPDL plays the same role as the Business Process Execution Language (BPEL)<sup>1</sup> standard. Both are XML languages for defining process/task flows within a larger BPM environment.

The jBPM support found within Seam can be used in two general ways. Within the scope of a single user session (actually, within the scope of specific conversations) in a web application, jPDL can be used to specify the valid pageflows that a user can take in the user interface. This limited application of jBPM provides a form of pageflow configuration that is richer and more expressive than the standard JSF navigation rules found in `faces-config.xml`.

This in itself can be very useful for managing the view layer of your web application, but jBPM is capable of doing much more. If you generalize the concept of flow between pages to flow between general-purpose tasks, and consider that these tasks could be implemented as web services, web pages, rules in a rules engine, business objects, or whatever, you’ve made the progression from pageflow to workflow. If you further expand

---

1. At the time of this writing, JBoss has released a beta version of jBPM that includes support for BPEL as an alternative to its custom jPDL syntax. You can expect to find this support make its way into a future release of Seam.

the concept to allow for long-lived workflows, involving multiple users acting across multiple user sessions, you've entered the realm of business process management. This is the broader realm of jBPM.

This chapter focuses on the more limited (but still very useful) application of jBPM to manage pageflow (i.e., the use of jPDL to define the navigation of a single user between pages within a single conversation). To do this, I first discuss the basics of jPDL itself, and how it is integrated and extended by the Seam framework. Then we'll return to our Gadget Catalog so you can see how pageflow can be used to define more structured user scenarios in the application.

In Chapter 7, I'll discuss the broader applications of jBPM for full-blown business process management within Seam.

## The Basics of Pageflow with jPDL

Before diving into the details of Seam's pageflow support and its practical use in the Gadget Catalog application, you need to understand the concepts behind jBPM's pageflow definition language, jPDL.

### The Language of jPDL

Description languages, typically based on XML, are at the heart of most current workflow environments, whether they are simple pageflow schemes or full BPM frameworks.

You've already become very familiar with a simple pageflow description language—the navigation rules found in the JSF `faces-config.xml` file. And in these navigation rules, you also saw the basic components of these flow description languages, just geared specifically for pageflow rather than more general process flow. Here's a sample set of JSF page navigation rules for a hypothetical user interface used for editing users in an application:

```

. . .
<navigation-rule>
  <from-view-id>/users/addUser.jsp</from-view-id>
  <navigation-case>
    <from-action>#{userAdmin.saveUser}</from-action>
    <from-outcome>success</from-outcome>
    <to-view-id>/users/userHome.jsp</to-view-id>
    <redirect/>
  </navigation-case>
  <navigation-case>
    <from-action>#{userAdmin.editRoles}</from-action>
    <to-view-id>/users/editUserRoles.jsp</to-view-id>
  </navigation-case>

```