CHAPTER 4

Contexts and Conversations

In this chapter, we explore Seam's context support, which is an extension of the contexts supported by Java EE web containers. Most of the focus will be on Seam conversations, which is a key new capability provided by the Seam framework. But first, as background, I introduce you to all of the new contexts supported by the Seam component model.

Seam Component Contexts

One of the keys to the Seam component model is its extended set of *scopes*, also referred to as *runtime contexts*. These contexts are familiar to anyone who has developed web applications based on servlets, JSPs, and/or JSF. Runtime data can be assigned to various contexts, and the scope of the data is defined by the duration of the context and the access that is allowed to the context. *Session* contexts, for example, last for the duration of a single user's session with an application, and the data contained in a session is only accessible to a single user. *Application* contexts, on the other hand, last for the duration of the entire application, and the data contained in them is potentially accessible to all users using that application.

Session data is probably the most commonly used context in servlet/JSP/JSF applications. Each user is given a session context when he or she enters an application, and your server-side code can put data associated with the user into his or her session context. The scope of the session is the user's active period with the application, and the access to the session is limited to just the user who owns the session.

Figure 4-1 depicts the contexts that are defined within the standard JSP/JSF environment. A global application context covers all runtime data managed by a particular application. Application data is available during the processing of any user request. Every user is given a session context, as I already mentioned. The session starts when the user first interacts with the application from his or her browser, and lives until it is either automatically or explicitly ended. A session can be ended by several events—the application can explicitly kill off a session because the user hits a "logout" link, or the requests from the user might stop for an extended period of time, causing the session to be expired by the server.

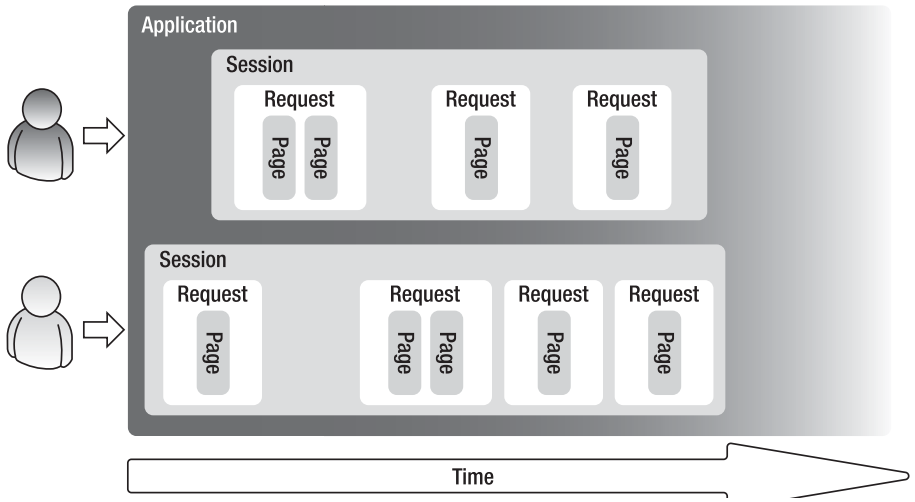

Figure 4-1. *Standard JSP/JSF contexts*

Within each user session are subcontexts that arise as the user interacts with the web application. Every user request has its own context and can hold data that lives only for the duration of that request. Any given request might involve multiple pages (a JSP may include other JSPs in its view, and/or a user request might be forwarded on the server to other pages). Each page can have its own runtime context, with data that only lives within that page invocation.

Seam's component model extends the runtime context structure, as shown in Figure 4-2. The application, session, request, and page scopes are essentially the same as what is supported by standard JSP/JSF containers.[1] Seam adds the concept of a *user conversation*, which segments a user's session into independent data contexts, as well as a context that covers the scope of an entire business process, which is in effect broader in scope than even the application scope, because it can persist across the lifetime of the application.

1. Note that JSF and JSP use the term "request" scope, while Seam uses the terms "request" and "event" intermittently in its documentation to refer to this same context. In this chapter and for the rest of the book, I will try to use the term "request" consistently.