

Composite View

Context

Sophisticated Web pages present content from numerous data sources, using multiple subviews that comprise a single display page. Additionally, a variety of individuals with different skill sets contribute to the development and maintenance of these Web pages.

Problem

Instead of providing a mechanism to combine modular, atomic portions of a view into a composite whole, pages are built by embedding formatting code directly within each view.

Modification to the layout of multiple views is difficult and error prone, due to the duplication of code.

Forces

- Atomic portions of view content change frequently.
- Multiple composite views use similar subviews, such as a customer inventory table. These atomic portions are decorated with different surrounding template text, or they appear in a different location within the page.
- Layout changes are more difficult to manage and code harder to maintain when subviews are directly embedded and duplicated in multiple views.
- Embedding frequently changing portions of template text directly into views also potentially affects the availability and administration of the system. The server may need to be restarted before clients see the modifications or updates to these template components.

Solution

Use composite views that are composed of multiple atomic subviews. Each component of the template may be included dynamically into the whole and the layout of the page may be managed independently of the content.

This solution provides for the creation of a composite view based on the inclusion and substitution of modular dynamic and static template fragments. It promotes the reuse of atomic portions of the view by encouraging modular design. It is appropriate to use a composite view to generate pages containing display components that may be combined in a variety of ways. This scenario occurs, for example, with portal sites that include numerous independent subviews, such as news feeds, weather information, and stock quotes on a single page. The layout of the page is managed and modified independent of the subview content.

Another benefit of this pattern is that Web designers can prototype the layout of a site, plugging static content into each of the template regions. As site development progresses, the actual content is substituted for these placeholders.

Figure 1.1 shows a screen capture of Sun's Java homepage, *java.sun.com*. Four regions are identified: Navigation, Search, Feature Story, and Headlines. While the content for each of these component subviews may originate from different data sources, they are laid out seamlessly to create a single composite page.

Figure 1.1 Screen shot of a modular page, including Search, Navigation, Feature Story, and Headlines regions.



This pattern is not without its drawbacks. There is a runtime overhead associated with it, a tradeoff for the increased flexibility that it provides. Also, the use of a more sophisticated layout mechanism brings with it some manageability and development issues, since there are more artifacts to maintain and a level of implementation indirection to understand.

Structure

Figure 1.2 shows the class diagram that represents the Composite View pattern.

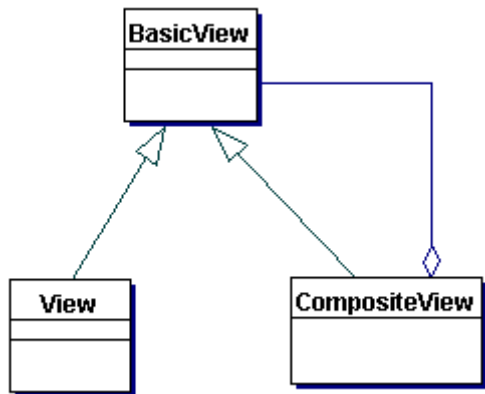


Figure 1.2 Composite View class diagram.

Participants and Responsibilities

Figure 1.3 shows the sequence diagram for the Composite View pattern.

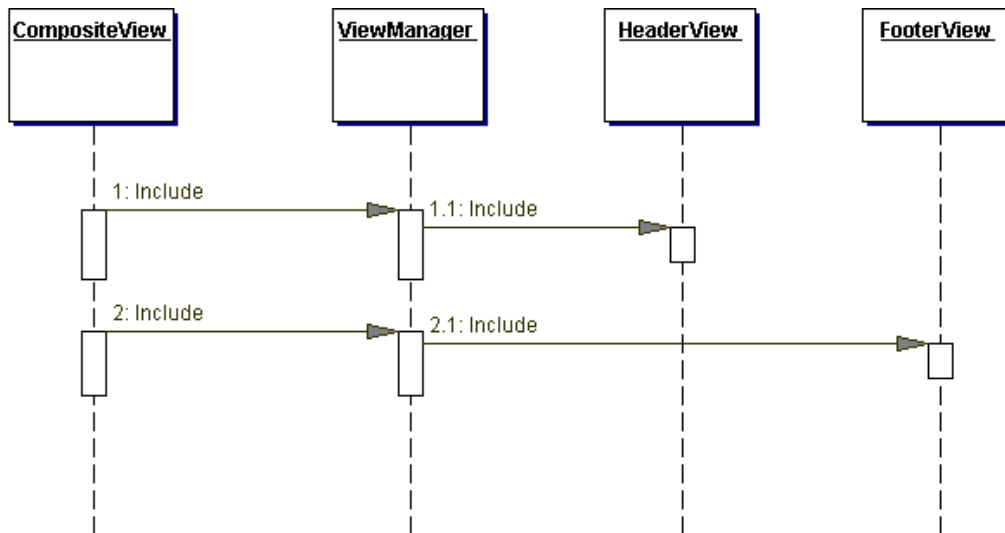


Figure 1.3 Composite View sequence diagram.

Composite View

A composite view is a view that is an aggregate of multiple subviews.

View Manager

The View Manager manages the inclusion of portions of template fragments into the composite view. The View Manager may be part of a standard JSP runtime engine, in the form of the standard JSP include tag (`<jsp:include>`), or it may be encapsulated in a JavaBean helper (JSP 1.0+) or custom tag helper (JSP 1.1+) to provide more robust functionality.

A benefit of using a mechanism other than the standard include tag is that conditional inclusion is easily done. For example, certain template fragments may be included only if the user fulfills a particular role or certain system conditions are satisfied. Furthermore, using a helper component as a View Manager allows for more sophisticated control of the page structure as a whole, which is useful for creating reusable page layouts.

Included View

An included view is a subview that is one atomic piece of a larger whole view. This included view could also potentially be a composite, itself including multiple subviews.