

А. И. Хобня, О. М. Демиденко

(ГГУ им. Ф. Скорины, Гомель)

ПРИМЕНЕНИЕ КОМПОНЕНТНОЙ АРХИТЕКТУРЫ ПРИ ПРОЕКТИРОВАНИИ WEB-ПРИЛОЖЕНИЙ

Наиболее распространенным паттерном, используемым при построении web-приложений, является MVC (Model-View-Controller). В классических фреймворках, реализующих данный паттерн, каждой «странице» web-приложения соответствует свой контроллер. Однако, при таком подходе в сложных web-приложениях появляются проблемы локализации изменений в сквозной функциональности. Элементы, располагающиеся на нескольких страницах, должны присутствовать в нескольких контроллерах и нескольких представлениях. Эти проблемы могут быть частично решены путем использования аспектно-ориентированного программирования и наследования шаблонов представлений. Но наиболее гибкое и полное решение может быть реализовано с помощью использования компонентной архитектуры при проектировании web-приложений.

При таком подходе компоненты определяют функциональность, а «страницы» определяют расположение и параметры компонент. Компоненты могут в свою очередь объединять в себе другие компоненты. Также различные компоненты могут взаимодействовать друг с другом.

Этот подход разделяет целостное web-приложение на легкоуправляемые фрагменты кода, которые управляют взаимодействием с пользователем, бизнес логикой и состоянием. Компоненты также делятся на базовые (core) и дополнительные (supplemental). Базовые компоненты являются основными строительными блоками приложения. Базовыми являются такие компоненты как валидаторы форм, средства для построения интерфейса или объекты состояния. Дополнительные компоненты определяют, каким образом группа базовых компонент может взаимодействовать с другими базовыми компонентами в группе. Используется связывание между компонентами (объектами) пользовательского интерфейса на web-странице и соответствующими им классами.

В частности, данный подход в некоторой мере реализован в таких фреймворках как Wicket, Tapestry и Google Web Toolkit. В Tapestry шаблоны компонентов представляют собой XHTML файлы. Как правило, связывание между java-классами и XHTML-объектами, которыми они управляют, описывается в XML файле. Взаимодействие между backend'ом и frontend'ом осуществляется с помощью OGNL. В Wicket состояние компонентов пользовательского интерфейса сохраняется, когда пользователь запрашивает новую страницу, и затем восстанавливается, если запрос повторяется. При этом конечная страница может состоять из множества мелких блоков, объединенных в единое целое. Написанный код можно поместить в jar-файл и повторно использовать без изменений.

Сравнительный анализ подходов приводится в докладе.