

УДК 004.05

ПОСТРОЕНИЕ ВЕБ-ПРИЛОЖЕНИЙ С ПОМОЩЬЮ ТРЕХСЛОЙНОЙ АРХИТЕКТУРЫ

*И.В. МИСЕВИЧ**(Представлено: канд. физ.-мат. наук, доц. Ю.Ф. ПАСТУХОВ)*

Рассматривается трехслойная архитектура, при помощи которой ведется успешное построение веб-приложений. Приведен пример многослойной архитектуры – трехслойная архитектура построения приложений, а также приведены основные правила и рекомендации по организации и построению архитектуры приложения.

Многослойная архитектура является одной из архитектурных парадигм разработки программного обеспечения, при которой функциональные области приложения разделяются на группы.

Всю функциональность приложения, так или иначе, можно разделить на группы в зависимости от того, какая задача выполняется этой функциональностью, что приложение посредством ее делает: общается с пользователем или осуществляет проверку данных, или манипулирует данными из базы, или описывает сущности, или описывает логику.

Данная работа дает возможность получить представление о трехслойной архитектуре, понятии архитектурного слоя, правилах определения слоев приложения.

Трехуровневая (трехслойная) архитектура – архитектурная модель программного комплекса, предполагающая наличие в нем трех компонентов: клиента, сервера приложений, к которому подключено клиентское приложение, и сервера баз данных, с которым работает сервер приложений.

Для себя можно представить структуру приложения, построенного по принципу многослойной архитектуры, как пирамиду, в которой каждый следующий уровень расположен поверх предыдущего. Особенностью такой пирамиды является то, что называют перевернутой пирамидой повторного использования, когда каждый слой, расположенный выше, обладает информацией о том, как обратиться к слою, расположенному непосредственно под ним, и ни в коем случае не наоборот.

Рассмотрим классическую 3-слойную модель (UIL – BLL – DAL), но даже в ней могут присутствовать дополнительные прослойки.

Слой – это способ структуризации информации, находящейся в понятии архитектуры и указывающий, какой именно аспект выделяется и рассматривается.

В самом вершине пирамиды находится слой пользовательского интерфейса (UIL – User Interface Layer). Какой технологией он будет представлен – зависит от типа приложения. Это могут быть элементы Swing JSP технологии JavaEE и многое другое. Основная задача этого слоя – предоставлять эргономичный интерфейс пользователю в соответствии с функционалом, описанном в техническом задании. Этот слой просто должен быть красивым, удобным и интуитивно понятным. При этом он должен знать, каким образом взаимодействует со слоем, находящимся на уровень ниже, как он передает туда данные и как оттуда принимает данные для отображения.

Что же касается логики слоя, находящегося под ним, UIL об этом ничего не знает, а именно какие алгоритмы выполняются в тех методах, которые он вызывает по нажатию кнопки или по изменению текста в поле ввода, или при выборе пункта из списка.

Ниже находится слой бизнес-логики. BLL – Business Logic Layer: классы, описывающие сущности и основную логику работы приложения, основные алгоритмы работы приложения, вычисления, обработка. Например, если мы пишем сетевые «крестики-нолики», то на этом слое могут быть описаны классы, описывающие игровое поле, выполненный ход, отправку данных по сети и так далее.

Ниже находится слой доступа к данным (DAL – Data Access Layer). Это работа с источником данных – базами данных разных поставщиков MSSQL, MySql, Oracle, или XML файлами, или бинарными данными, или текстовыми файлами, всем тем, в чем можно данные хранить постоянно. Здесь также описана логика, но направленная именно на взаимодействие с источником и передачу полученных результатов на уровень выше.

Могут быть еще и другие слои, но какие и в какой последовательности они расположены в пирамиде архитектуры зависит от приложения.

Главное, что нужно знать – это набор следующих правил:

- каждый слой занимается конкретной задачей;
- логика разных слоев не повторяется и не пересекается;
- способ обращения к нижестоящему слою четко определен;
- способ поставки информации вышестоящему слою четко определен;

- слои слабо связаны;
- слои расположены вертикально, хотя есть сквозная функциональность, которая может пронизывать пирамиду сверху вниз;
- слои могут размещаться физически на одном компьютере (в пределах одного уровня), а могут быть на разных машинах, например, в распределенных приложениях;
- логика разных слоев инкапсулирована, соответственно, разным слоям не нужно делать никаких предположений о том, как реализован код других слоев приложения;
- возможность использования в других сценариях (за счет слабого связывания и четко определенной задачи).

По сравнению с двухзвенной клиент-серверной архитектурой или файл-серверной архитектурой трехуровневая архитектура обеспечивает, как правило, большую масштабируемость за счет горизонтальной масштабируемости сервера приложений и мультиплексирования соединений, большую конфигурируемость за счет изолированности уровней друг от друга.

Заключение

Представляемая работа дает возможность получить представление о трехслойной архитектуре, понятии архитектурного слоя, правилах определения слоев приложения

ЛИТЕРАТУРА

1. Трехуровневая архитектура : словари и энциклопедии Академик [Электронный ресурс]. – Режим доступа: <http://dic.academic.ru/dic.nsf/ruwiki/198141>. – Дата доступа: 23.09.2017.
2. Разговор о MVC и архитектуре веб-приложений [Электронный ресурс] // Хабрахабр. – Режим доступа: <https://habrahabr.ru/post/283096>. – Дата доступа: 25.09.2017.
3. Электронный университет ВГУ [Электронный ресурс]. Трехзвенная архитектура с сервером приложений. – Режим доступа: <https://edu.vsu.ru/mod/book/view.php?id=12520&chapterid=237>. – Дата доступа: 25.09.2017.