

УДК 004.05

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

В данной статье рассматриваются распространенные архитектурные подходы, при помощи которых ведется построение приложений.

Наиболее распространенными архитектурами для построения приложений являются двух-, трех- и пятислойные архитектуры.

Двухслойная архитектура появилась в первой половине 90-х годов при проектировании приложений типа клиент-сервер. Суть заключается в том, что существует два слоя – клиент и сервер; на клиентской части располагаются средства представления данных, а также предметная логика взаимодействия с источником данных. Такие приложения еще называют толстыми клиентами. На уровне сервера располагается слой источника данных, например, СУБД. Клиент посылает серверу запросы на получение данных и получает ответ в виде данных или сообщений об ошибках. Приложения, состоящие из 2-х слоев, достаточно эффективны если необходимы данные и механизмы взаимодействия изменяются редко, а также нет необходимости в передаче данных на большие расстояния.

Трехслойная архитектура представляет собой эволюционное развитие двухслойной. В этой архитектуре предметная логика и представление разделены, что обеспечивает значительную гибкость разработки приложения, но увеличивает сложность. К преимуществам архитектуры можно отнести то, что слой представления не зависит от источника данных, а слой логики может быть адаптирован к различным видам представлений.

Однако быстрый рост количества пользователей и появившиеся вследствие этого дополнительные издержки стали снижать эффективность применения архитектуры приложения из трех слоев. Проблема состояла в том, что при добавлении дополнительного уровня представления или обновления хранилища данных приходилось вносить значительные изменения в предметную логику приложения. В некоторых случаях это вызывало ряд затруднений, и предметная логика состояла из различных частей, которые плохо взаимодействовали друг с другом. Единственным выходом из сложившейся ситуации было добавление новых уровней абстракции. Так появилась пятислойная архитектура распределенного приложения. В трехслойную архитектуру было введено два дополнительных слоя: контроллер/медиатор и слой сопоставления данных. Предназначение слоя контроллера заключается в том, что уровень представления обращается к слою контроллера за необходимыми порциями данных и полностью изолируется от логики предметной области. Таким образом, предметная логика могла быть сохранена для любого из представлений в неизменном виде. При добавлении нового представления достаточно было реализовать слой представления и реализовать дополнительный контроллер.

Если сравнивать по поддержанию бизнес-логики в актуальном состоянии для множества приложений и написанию дополнительного контроллера, то становится очевидно, что новое решение является более эффективным. Таким образом, слой контроллера надежно вписался в архитектуру распределенного приложения.

С серверами БД ситуация обстояла несколько хуже – промышленные сервера стоят дорого, поэтому с экономической точки зрения покупать для каждого приложения свой сервер БД невыгодно. При этом, используя трехслойную архитектуру, необходимо еще обеспечить соответствие предметной логики принципам хранилища данных, что в большинстве случаев делало приложения непереносимыми между различными серверами БД.

Решение проблемы взаимодействия слоев предметной логики и сервера БД заключается во введении дополнительного слоя – слоя сопоставления. Такой слой включает в свой состав механизмы отображения данных из БД, ориентированные на конкретную реализацию БД с одной стороны, а с другой стороны – обеспечение унифицированного интерфейса доступа со стороны слоя предметной области.

Приложения, построенные с использованием пятиуровневой архитектуры, показали значительно лучшую расширяемость при добавлении новых представлений или миграции на другие сервера БД. Предметная логика оставалась неизменной или дорабатывалась, но все изменения на этом уровне касались только уровней контроллера и сопоставления, но, как правило, эти изменения были достаточно простыми.

К недостаткам пятиуровневой архитектуры можно отнести еще большее увеличение сложности по сравнению с трехуровневой архитектурой.

Таким образом мы рассмотрели многоуровневые архитектуры разработки программ. Их достоинства и недостатки, и причины появления.

ЛИТЕРАТУРА

1. Блог программиста [Электронный ресурс] Многослойная архитектура. Определение и задачи. Режим доступа: <https://pro-prof.com/forums/topic/multilayered-architecture-definition-and-tasks>. Дата доступа: 26.09.2019.
2. Академик [Электронный ресурс] Трехуровневая архитектура. Режим доступа: <https://dic.academic.ru/dic.nsf/ruwiki/198141>. Дата доступа: 26.09.19.
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: 26.09.2019.