

Архитектура программного обеспечения – это представление, которое дает информацию о компонентах ПО, обязанностях отдельных компонент и правилах организации взаимосвязей между компонентами. Продуманная архитектура облегчает разработку и дальнейшее развитие ПО. Она служит базисом, каркасом создаваемой системы, интегрируя отдельные компоненты и создавая высокоуровневую модель системы. Набор принципов, используемых в архитектуре, формирует *архитектурный стиль*.

«Клиент-сервер»

Главными преимуществами стиля «клиент-сервер» являются:

- **Высокая безопасность.** Все данные хранятся на сервере, обеспечивающем больший уровень безопасности, нежели отдельный клиент.
- **Централизованный доступ к данным.** Так как данные хранятся только на сервере, ими легко управлять (например, обеспечить обновление).
- **Легкость сопровождения.** Роль сервера могут выполнять несколько физических компьютеров, объединенных в сеть. Благодаря этому клиент не замечает сбоев или замены отдельного серверного компьютера.

Многоуровневая архитектура

Многоуровневая архитектура сосредоточена на иерархическом распределении отдельных частей системы при помощи эффективного разделения отношений. Каждая часть соотносится с определенным *уровнем* (layer), для каждого уровня заданы выполняемые им функции, уровни выстроены в стековую структуру (то есть находятся один поверх другого).

Шина сообщений

Архитектура, основанная на шине сообщений, подразумевает наличие общего коммуникационного канала, используя который компоненты обмениваются информацией. Компонент помещает сообщение в коммуникационный канал, после этого сообщение рассылается всем заинтересованным компонентам. Данная архитектура направлена на решение коммуникационных задач.

Принципы архитектуры с использованием шины сообщений следующие:

- Все коммуникации между компонентами выполняются только при помощи информационных сообщений.
- Сообщения имеют стандартный формат. Это позволяет интегрировать компоненты, разработанные на разных платформах.
- Общая логика приложения изменяется путем удаления или добавления компонент, подключенных к шине.
- Как правило, коммуникация происходит в асинхронном режиме.

Преимущества шины сообщений:

- **Расширяемость.** Компоненты, подключённые к шине, добавляются и удаляются без воздействия на другие подключённые компоненты.
- **Уменьшение сложности.** Для взаимодействий с системой компонент должен реализовать только логику работы с общей шиной.
- **Масштабируемость.** При увеличении нагрузки на один из подключенных компонентов достаточно добавить к шине копию этого компонента, которая будет обрабатывать часть сообщений.

Выделенное представление

Выделенное представление – это стиль обработки запросов или действий пользователя, а также манипулирования элементами интерфейса и данными. Стиль подразумевает отделение элементов интерфейса от логики приложения.

Ключевыми принципами данного архитектурного стиля являются:

- Выделение отдельных функций и ролей для задач обработки запросов, изменения данных и представления данных.
- Для интеграции отдельных компонентов может использоваться событийная модель.

1. Модель представляет данные, с которыми работает приложение. Модель также реализовывает логику обработки данных согласно заданным бизнес-правилам и обеспечивает чтение и сохранение данных во внешних источниках.
2. Представление обеспечивает способ отображение данных модели.
3. Контроллер обрабатывает внешние запросы и координирует изменение модели и актуальность представления.

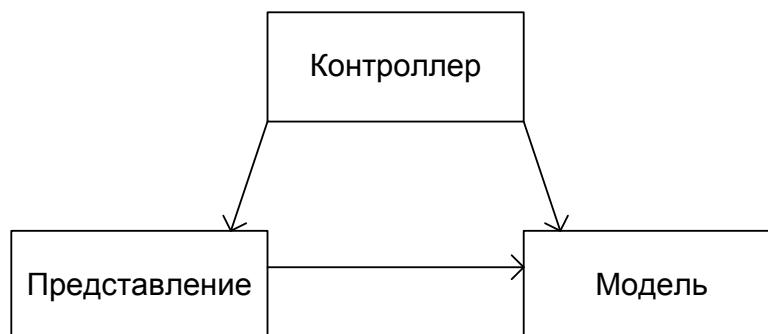


Рис. 26. Модель-представление-контроллер.

Важно отметить, что как представление, так и контроллер зависят от модели. Однако модель не зависит ни от представления, ни от контроллера. Это одно из ключевых достоинств подобного разделения. Оно позволяет строить модель независимо от визуального представления, а также создавать несколько различных представлений и контроллеров для одной модели.

N-звеневая архитектура

Этот архитектурный стиль разворачивания приложений подразумевает разделение компонентов на функциональные группы, подобно тому, как это происходит в многоуровневой архитектуре. Группа (реже – несколько групп) формируют *звено* – часть приложения, которая физически обособлена, то есть выполняется в отдельном процессе или на отдельном физическом компьютере.

Объектно-ориентированная архитектура

При использовании объектно-ориентированной архитектуры система воспринимается как набор взаимодействующих объектов. Каждый такой объект содержит данные и необходимое поведение, а коммуникация между объектами происходит через открытые интерфейсы и путем отправки и приема сообщений.

Архитектура, ориентированная на сервисы

Архитектура, ориентированная на сервисы (Service-Oriented Architecture, SOA), предоставляет требуемые функции в виде набора сервисов. Сервисы используют стандартные протоколы для вызова своих функций, публикации в сети и обнаружения. Отдельный сервис должен рассматриваться как независимое приложение, не как компонент или объект. Основной задачей при использовании SOA является определение интерфейса сервиса и схемы передаваемых при вызове сервиса данных.

Базовые принципы SOA:

- Сервисы автономны и независимы друг от друга.
- Сервисы распределены в локальной или глобальной сети. Местоположение сервиса не важно – главное, чтобы сеть доступа к сервису поддерживала требуемые протоколы.
- Сервисы публикуют контракты использования и схемы для данных обмена, но скрывают внутренние классы своей реализации.

Преимущества архитектуры, ориентированной на сервисы, являются высокая абстракция и возможность использования сервисов в различных приложениях.