

Архитектура ПО



Литература к курсу

- Л. Басс, П. Клементс, Р. Кацман - Архитектура программного обеспечения на практике.
- 1-е издание, 2006 год, 576 стр., формат 17х23 см (70х100/16), Твердый переплет, ISBN 5-469-00494-5



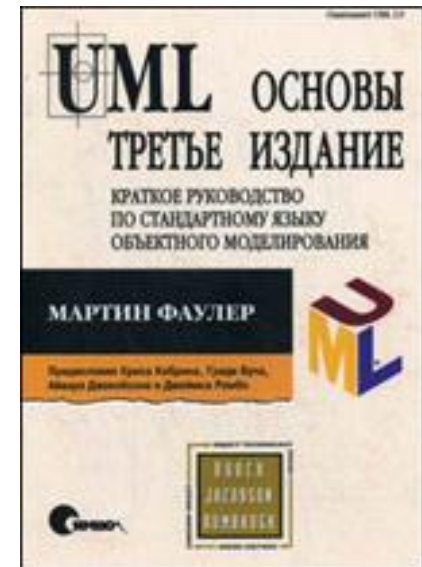
Литература к курсу

- М. Фаулер- Архитектура корпоративных программных приложений – Вильямс,2006



Литература к курсу

- М.Фаулер – UML основы – Символ-Плюс, Питер, 2002.



Литература к курсу

- Лешек Мацяшек - Анализ требований и проектирование систем – Вильямс, 2002



Литература к курсу

- Л. Константи́н, Л. Локвуд – Разработка программного обеспечения - Питер, 2004



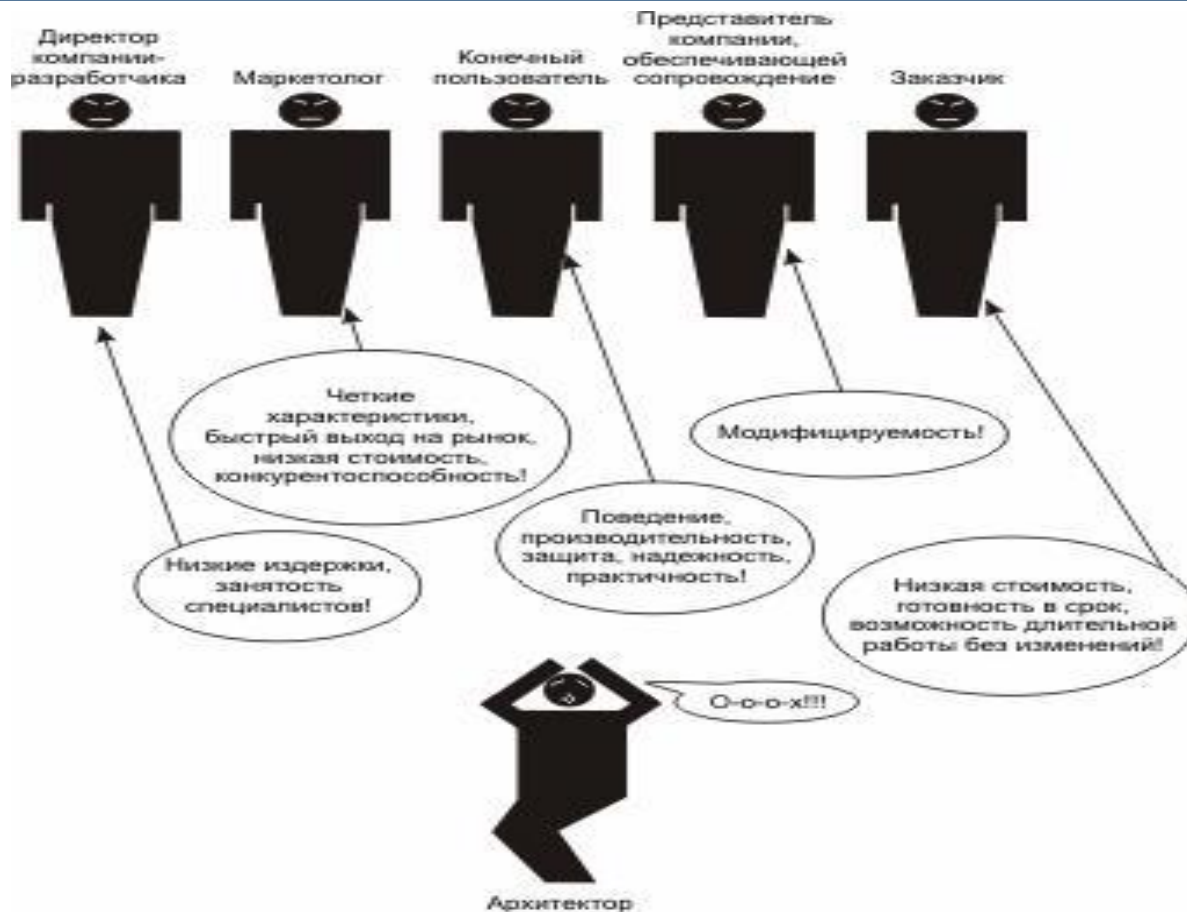
Определения «архитектуры ПО»

- Архитектура - это базовая **организация системы**, воплощенная в ее **компонентах**, их **отношениях** между собой и с **окружением**, а также принципы, определяющие проектирование и развитие системы.
- Архитектура - это **набор значимых решений** по поводу организации системы программного обеспечения, набор **структурных элементов** и их интерфейсов, при помощи которых конструируется система, вместе с их **поведением**, определяемым во взаимодействии между этими элементами, **компоновка** элементов в постепенно укрупняющиеся подсистемы, а также **стиль архитектуры** который направляет эту организацию -- элементы и их интерфейсы, взаимодействия и компоновку.
- Архитектура программы или компьютерной системы - это структура или структуры системы, которые включают элементы программы, видимые извне свойства этих **элементов** и **связи** между ними.

Определения «архитектуры ПО»

- *Архитектура - это структура организации и связанное с ней поведение системы. Архитектуру можно рекурсивно разобрать на части, взаимодействующие посредством интерфейсов, связи, которые соединяют части, и условия сборки частей. Части, которые взаимодействуют через интерфейсы, включают классы, компоненты и подсистемы. [5]*
- *Архитектура программного обеспечения системы или набора систем состоит из всех важных проектных решений по поводу структур программы и взаимодействий между этими структурами, которые составляют системы. Проектные решения обеспечивают желаемый набор свойств, которые должна поддерживать система, чтобы быть успешной. Проектные решения предоставляют концептуальную основу для разработки системы, ее поддержки и обслуживания.*

Влияние на архитектуру оказывают заинтересованные в системе лица



Влияние на архитектуру оказывает компания-разработчик

- Компании делают прямые инвестиции в различные активы — в частности в существующие варианты архитектуры и основанные на них продукты. Каждый последующий проект в таком случае мыслится как продолжение ряда схожих систем, а в его смете заложено активное повторное использование имеющихся средств.
- Следуя своим стратегическим задачам, компании иногда делают долгосрочные инвестиции в инфраструктуру; в таком случае предполагаемая система мыслится как одно из средств финансирования и расширения этой инфраструктуры.
- Определенное влияние на программную архитектуру оказывает организационная структура компании-разработчика.

Влияние на архитектуру оказывают опыт и привычки архитекторов

- Положительный опыт – повторение его в последующих работах.
- Отрицательный опыт – отказ использовать его в последующих работах.
- Архитекторы любят экспериментировать с новыми образцами(pattern) и методиками.

Влияние на архитектуру оказывает техническая база

Подготовка и опыт архитектора проявляется, в частности, в его работе с технической базой (technical environment).

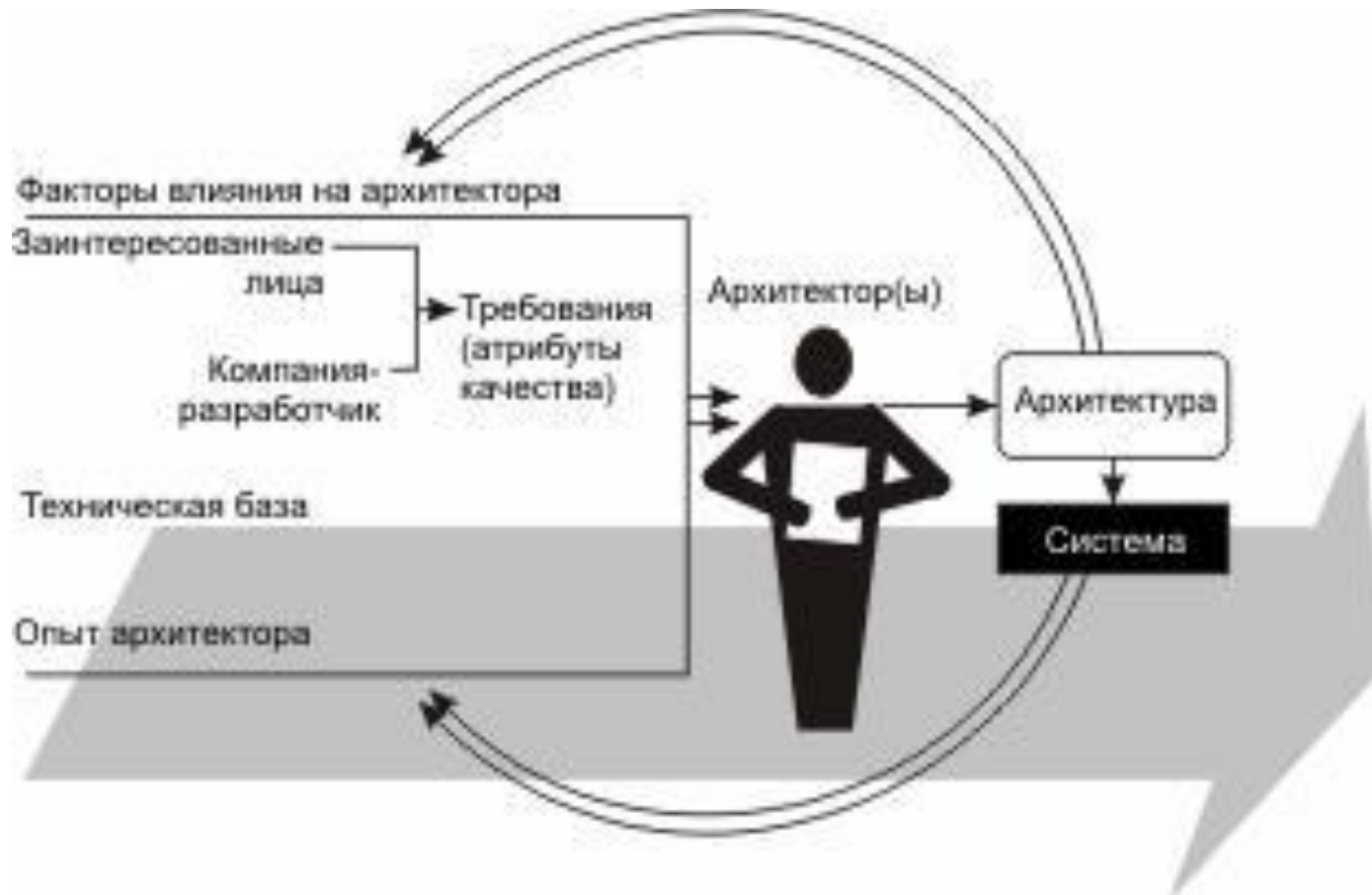
К технической базе можно отнести:

- методы работы, принятые в данной отрасли
- приемы программной инженерии, распространенные в профессиональном сообществе, в которое входит архитектор.

Вариативность факторов влияния на архитектуру



Архитектура оказывает обратное воздействие на факторы влияния



Программный процесс и архитектурно-экономический цикл

- создание экономической модели системы;
- выявление требований;
- создание новой или выбор существующей архитектуры;
- документирование и распространение сведений об архитектуре;
- анализ или оценка архитектуры;
- реализация системы на основе архитектуры;
- проверка соответствия реализации архитектуре;

Строительные элементы архитектуры ПО - компонент

Компонент – абстрактная единица инструкций программного обеспечения и внутренних состояний, которая обеспечивает трансформацию данных через свои *интерфейсы*.

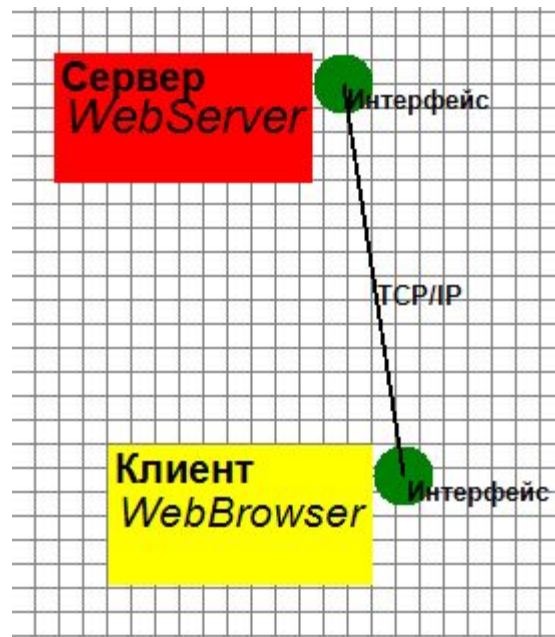
Строительные элементы архитектуры ПО - коннектор

Коннектор – абстрактный механизм, служащий посредником в коммуникации, координации или кооперации между компонентами.

Строительные элементы архитектуры ПО - данные

Данные – элемент информации, передаваемый от одного компонента или получаемый от другого компонента через коннектор.

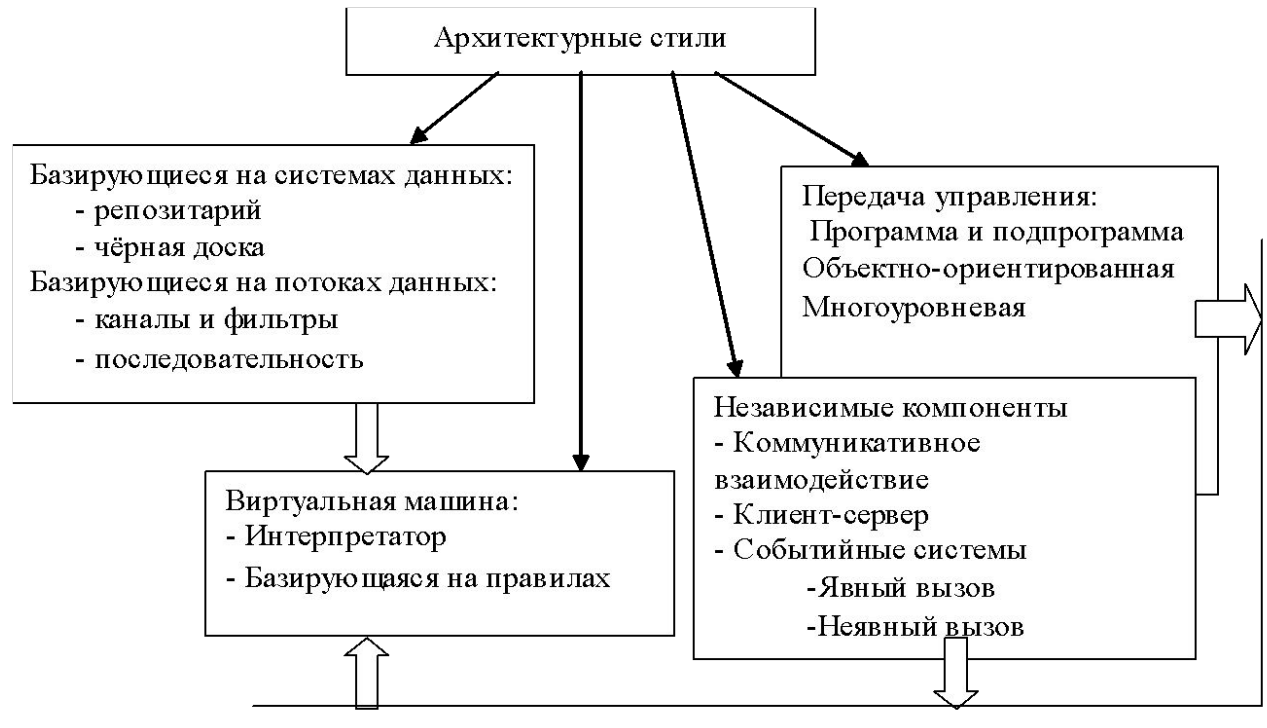
Пример архитектурного описания



Архитектурные образцы, эталонные модели и эталонные варианты архитектуры



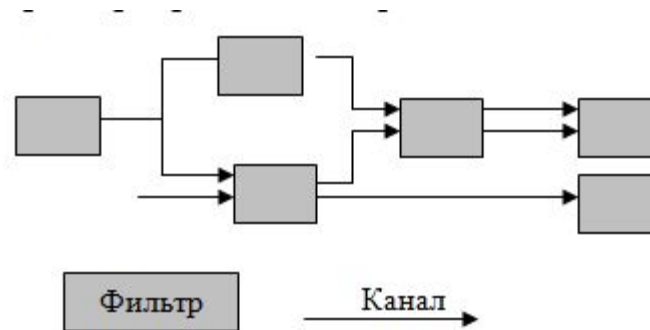
Классификация архитектурных стилей



Каналы и фильтры

Архитектурное описание строится как (или представляет собой) связную совокупность фильтров и каналов , в которой:

- фильтр – независимый блок, получающий на вход поток(и) данных и выдающий поток(и) данных
- возможна достаточно простая переконфигурация системы фильтров (соединение их с помощью pipes)
- фильтры работают параллельно



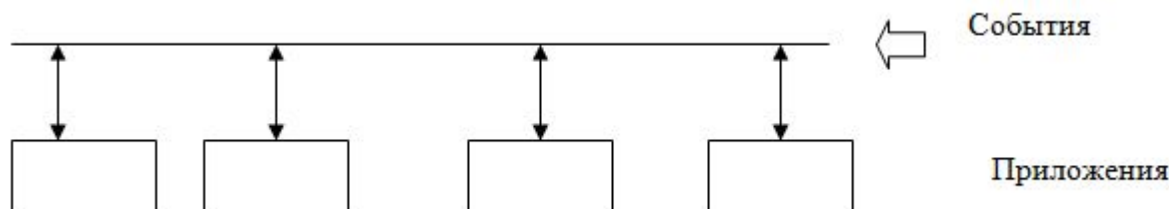
Стиль, основанный на репозитории (Repository-based)

- В рамках стиля, базирующегося на репозитории, общие данные разделяет определённая совокупность приложений, каждое из которых в своих обращениях к данным является независимым.



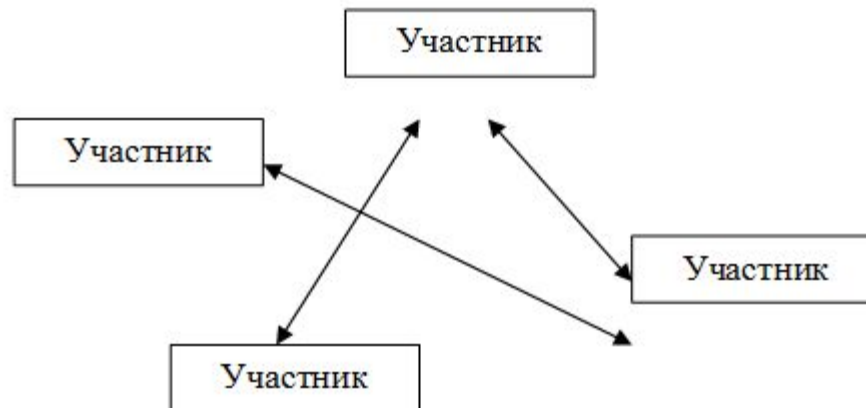
Событийно-ориентированный (Event-based) стиль

- Событийный стиль используется для композиции приложений, которые обращаются (регулярно или случайно) к родственному типу событий (возникающих регулярно или случайно), с которыми взаимодействует система.



Одноранговый (Peer-to-Peer) СТИЛЬ

- Одноранговый стиль применим в условиях, когда разрабатываемая система состоит из приложений (участников взаимодействия), каждое из которых связано с определённым количеством других. В каждом акте взаимодействия оно устанавливается и происходит только между двумя участниками.



Многослойный стиль

- Одним из эффективных подходов к разбиению системы на части является использование многослойных структур. Такой подход применим как к приложениям, развёрнутым на одном компьютере, так и для распределённых систем. Типовые разбиения на слои (для таких вариантов) близки по смыслу, но имеют различия в терминологиях, используемых для их описаний.

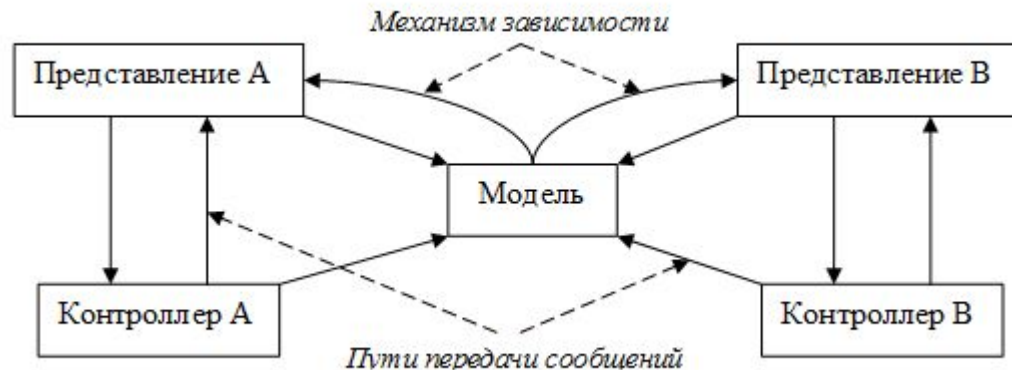
Представление
Интерфейс
Бизнес-логика
Базовые объекты
Данные

Клиент-серверный стиль

- Для распределённых систем типичен случай применения стиля «клиент-сервер», для реализации которого, часть системы с определёнными функциональностями размещают на сервере, а другую часть на клиентских местах пользователей.

Стиль «Модель-вид-контроллер» (Model-View-Control, MVC)

- Стиль MVC согласован с приложениями , в которых основным видом работ является интерактивное взаимодействие пользователя с визуализированными моделями, представляющими определённые сущности.



Примеры использования архитектурных стилей

Стили	Контекст использования	Примеры использования
Каналы и фильтры	Нужно обеспечить преобразование непрерывных потоков данных, преобразования инкрементальны, следующее может быть начато до окончания предыдущего, возможно добавление дополнительных преобразований	Утилиты UNIX, компиляторы
Многоуровневая система	Важны переносимость и возможность многократного использования, имеется естественное расслоение системы на специфичные только для нее функции и функции общего характера, специфичные для платформы	Протоколы (модель OSI и реальные)
Клиент-сервер	Решаемые задачи естественно распределяются между инициаторами и обработчиками запросов, возможно изменение внешнего представления данных и способов их обработки	Основная модель бизнес-приложений