Методы визуального анализа и проектирования систем

Клевцов С.И доцент каф. ВС ИРТСУ

Архитектурные стили

Архитектурный стиль можно определить как семейство систем в терминах шаблона организации структуры.

Точнее, архитектурный стиль определяет:

- номенклатуру компонентов и типов соединительных звеньев,
- а также набор условий, в соответствии с которыми они могут соединяться.

Архитектурный стиль определяется:

- набором типов компонентов, во время счета выполняющих некоторую функцию,
- топологической раскладкой компонентов с указанием их взаимосвязей во время выполнения,
- набором семантических ограничений,
- набором соединителей, служащих средой сообщения, координации и сотрудничества между компонентами.

Архитектурные стили

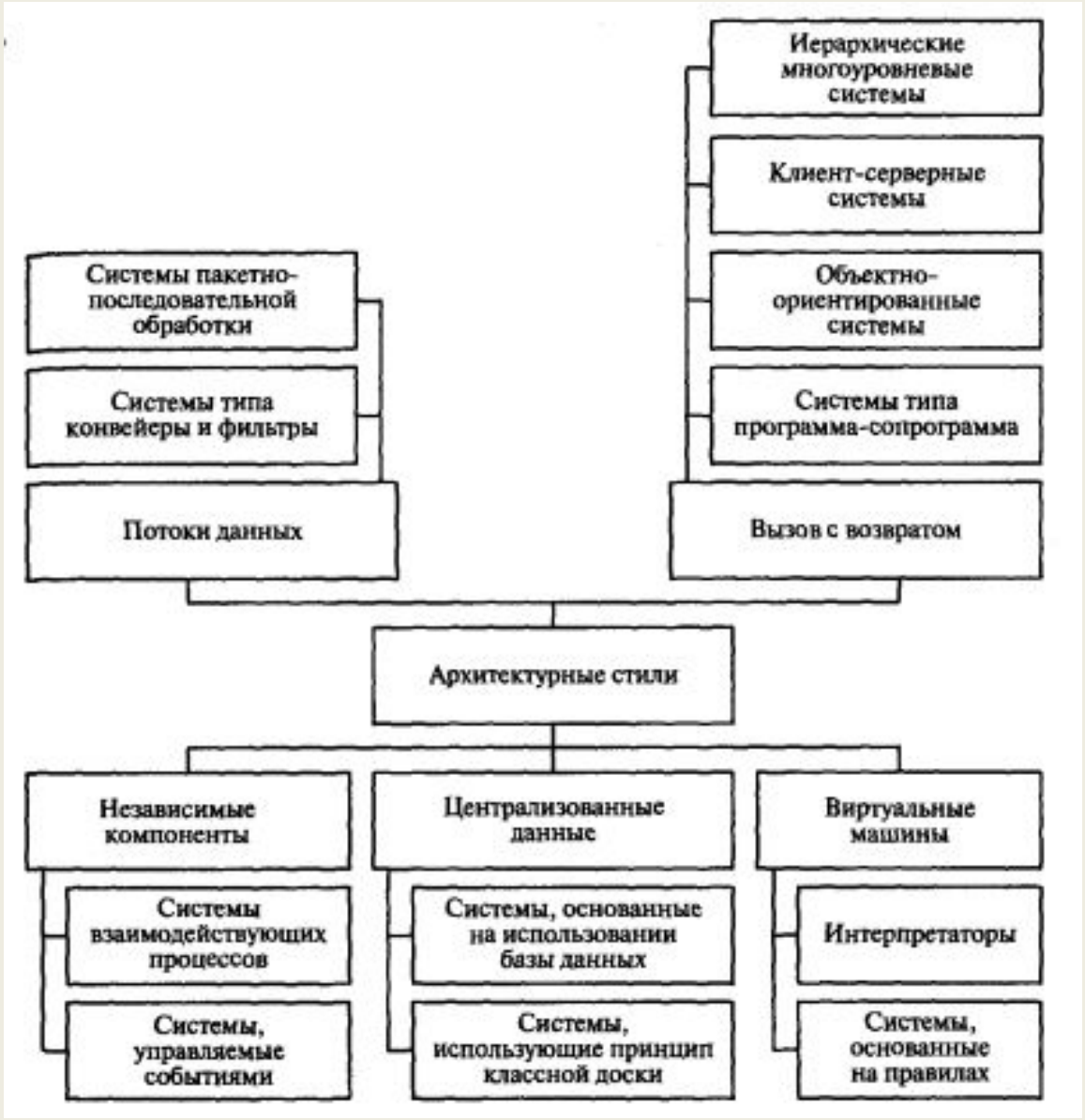
Иногда вместо термина архитектурный стиль используется термин ***архитектурный паттерн (шаблон).***

Есть наличие принципиальных различий между архитектурными стилями и паттернами:

Паттерн — это, по существу, фрагмент кода на конкретном языке программирования

Архитектурный стиль — это подход к проектированию.

Архитектурные стили



Классификация архитектурных стилей

Архитектурные стили

Принято выделять двенадцать базовых архитектурных стилей, которые делятся на пять групп:

- потоки данных (Data Flow Systems);
- вызов с возвратом (Call-and-Return Systems);
- независимые компоненты (Independent Component Systems);
- централизованные данные (Data-Centric Systems);
- виртуальные машины (Virtual Machines).

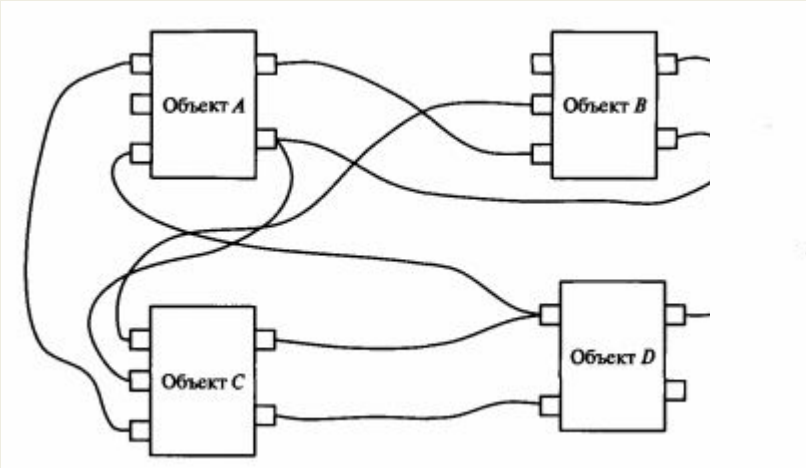
Архитектурные стили

Вызов с возвратом

Двухслойная организация и Трехслойная организация клиент-серверной архитектуры

Архитектурные стили

Вызов с возвратом



Простейшая Объектно-ориентированная система

Типовая структура распределенной объектной системы

Архитектурные стили

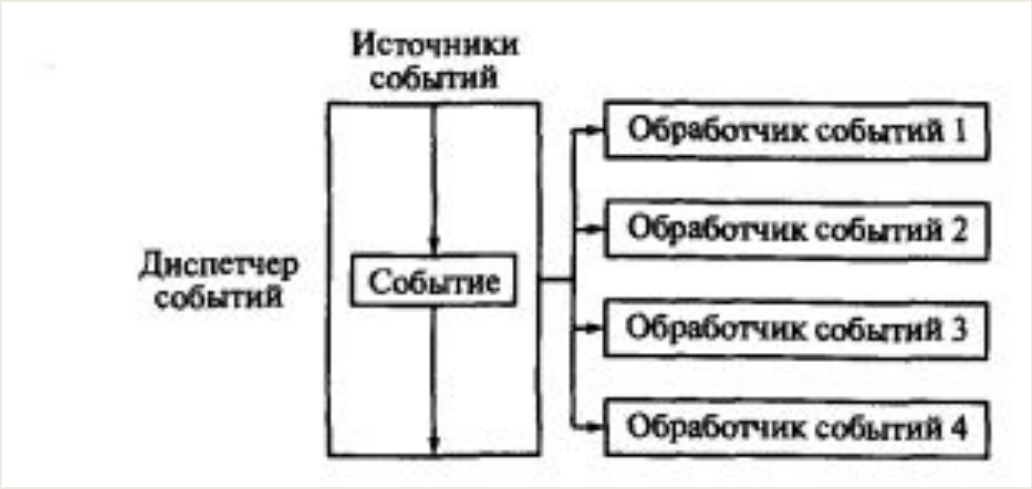
Вызов с возвратом

Операционная система с трехуровневой организацией

Архитектурные стили

Системы, работающие по принципу независимых компонентов.

Имеются системы взаимодействующих процессов и системы, управляемые событиями.



Структура системы, управляемой событиями

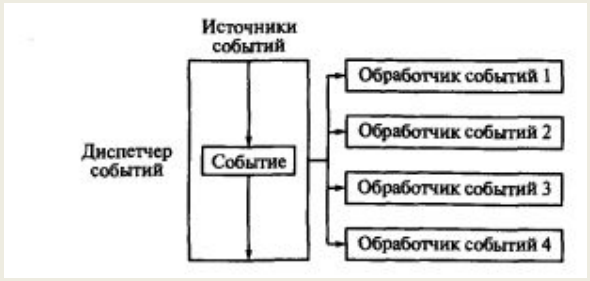
Основными компонентами системы являются:

- источники событий,
- обработчики событий,
- коннекторы (диспетчер).

Источники событий являются генераторами событий.

Архитектурные стили

Системы, работающие по принципу независимых



Структура системы, управляемой событиями

Отличительной особенностью систем, управляемых событиями, является то, что процесс обработки запускается только тогда, когда происходит соответствующее событие.

Отличительная особенность систем, управляемых событиями, состоит в том, что для них обычно невозможно указать поток управления.

Архитектурные стили

Системы, работающие по принципу централизованных данных (репозитария)

Структура системы, работающей по принципу репозитария

Архитектурные стили

Виртуальные машины

Типовая система, ориентированная на работу с правилами

Архитектурные стили

Использование стилей

Лист 1

Название стиля	Условия целесообразности использования
Системы пакетно-последовательной обработки	Задачу можно разделить на четко определенные подзадачи, каждая из которых использует единственную операцию ввода-вывода. Результаты операции либо отправляются пользователю, либо поступают на вход другой подзадачи. Имеются готовые приложения, которые можно использовать для решения отдельных подзадач
Системы типа конвейеры и фильтры	Алгоритм решения задачи можно представить как совокупность повторяющихся преобразований над однотипными и независимыми друг от друга наборами данных. Число ветвлений и обратных связей минимально

Архитектурные стили

Использование стилей

Лист 2

Название стиля	Условия целесообразности использования
Системы типа программа-сопрограммы	Порядок вычислений фиксирован, компоненты не могут делать ничего полезного, пока ждут результатов своих запросов к другим компонентам. Использование механизма наследования не дает существенных выгод
Объектно-ориентированные системы	Можно существенно уменьшить трудозатраты на разработку системы за счет использования в процессе разработки механизма наследования. Объекты находятся на разных хостах

Архитектурные стили

Использование стилей

Лист 3

Название стиля	Условия целесообразности использования
Клиент-серверные системы	Задачу можно сформулировать как совокупность запросов с которыми клиенты обращаются к серверу, т. е. разделить на части, выделив клиентскую часть, отвечающую за формирование задания серверу, и серверную, отвечающую за выполнение запроса. Каждый клиент может обращаться к серверу в любое время. Сервер может только отвечать на запросы. Число клиентов может меняться, и они могут обращаться к серверу с разными запросами
Иерархические многоуровневые системы	Задачу можно представить в виде совокупности слоев с четко определенными интерфейсами, требуется иметь разные варианты бизнес-логики для разных слоев. Важна переносимость приложения между платформами, важна возможность использовать уже существующие реализации слоев

Архитектурные стили

Использование стилей

Лист 4

Название стиля	Условия целесообразности использования
Системы взаимодей ствующих процессов	Передача сообщений — достаточный механизм взаимодействия для процессов. Нет большого объема долгоживущих централизованных данных
Системы, управляемые событиями	Система по своей природе асинхронная. Функционирование системы инициируется асинхронными событиями. Система может быть реализована как совокупность независимых про цессов, функционирующих на разных платформах. Потребители событий отделены от сигнализаторов; важна масштабируемость в форме добавления процессов, переключаемых событиями

Архитектурные стили

Использование стилей

Лист 5

Название стиля	Условия целесообразности использования
Системы, основанные на использовании централизованной базы данных	Задачи можно разделить между выдающими запросы и отрабатывающими запросы или между производителями и потребителями данных Главный вопрос — хранение, представление, управление и поиск больших объемов связан ных долгоживущих данных. Порядок исполнения компонентов определяется потоком входных запросов по доступу/обновлению данных; данные высокоструктурированы; доступна и экономична коммерческая СУБД
Системы, использующие принцип классной доски	Имеется большое число клиентов, которые общаются между собой для решения общей задачи. Важна масштабируемость в форме добавления потребителей данных

Архитектурные стили

Использование стилей

Лист 6

Название стиля	Условия целесообразности использования
Интерпретаторы	Проектирование вычислений, когда нет реальной машины, или ее использование затруднено, требуется скрыть специфику платформы, пользователю необходимо предоставить возможность использовать либо стандартный, либо скрипто-вый язык программирования
Системы, основанные на правилах	Алгоритм решения задачи можно описать в терминах множества правил и способов их применения. Правила могут меняться конечным пользователем