2.1 Открытые системы. Сервис-ориентированная архитектура

**Открытость**

Одной из главных тенденций современной индустрии информатики является создание **открытых систем**.

Свойство открытости означает:

1) переносимость (мобильность) ПО на различные аппаратные платформы,

2) приспособленность системы к ее модификациям (модифицируемость или собственно открытость)

3) приспособленность системы к комплексированию с другими системами в целях расширения ее функциональных возможностей и (или) придания системе новых качеств (интегрируемость).

Аспекты открытости отражены в стандартизации:

* API {Application Program Interface) - интерфейсов прикладных программ с операционным окружением, в том числе системных вызовов и утилит операционной системы (ОС), т.е. связей с ОС;
* межпрограммного интерфейса, включая языки программирования;
* сетевого взаимодействия;
* пользовательского интерфейса, в том числе средств графического взаимодействия пользователя с ЭВМ;
* средств зашиты информации.

**SOA**

**SOA** - это архитектурный подход к определению, связыванию и интеграции повторно используемых бизнес-сервисов, имеющих четкие границы и самодостаточных по своей функциональности. В рамках такой архитектуры можно организовывать бизнес-сервисы в бизнес-процессы. Внедряя концепцию сервисов (более высокого уровня абстракции, не зависящего от приложений и платформы информационной инфраструктуры, а также от контекста или других сервисов), SOA переносит информационные технологии на следующий уровень, более подходящий для обеспечения функциональной совместимости и реализации в гетерогенных средах.

Существуют определенные ситуации и бизнес-функции, когда следует немедленно обратиться к SOA, поскольку эта архитектура может существенно повысить конкурентоспособность и производительность и четко проявить свои преимущества. К таким ситуациям главным образом относятся:

* Централизованные бизнес-функции, используемые несколькими субъектами.
* Интеграция с партнерами.
* Наличие работающих старых технологий.

**Концепции SOA**

Давайте рассмотрим некоторые концепции архитектуры SOA, чтобы лучше понять, что она собой представляет.

**Определение сервиса в SOA**.   
Существует множество различных определений сервисов, но, на мой взгляд, лучше всего объясняют сущность сервисов определения, приведенные ниже.

Сервис - это функция, являющаяся четко определенной, самодостаточной и не зависящей от контекста или состояния других сервисов.

**Концепция слабого связывания в SOA  
Виртуализация сервиса (Service Mediation)**  
Разделение сервиса происходит с помощью механизма виртуализации. Виртуальный сервис является прокси-объектом для реального сервиса. Прокси-сервис представляет собой желаемый потребителем сервиса интерфейс. Потребители обращаются к проксисервису, передающему сообщения к действительному сервису.

**Независимость местоположения.**Виртуальный сервис позволяет скрыть действительное местоположение сервиса отпотребителей. Это дает свободу перемещать реализацию сервиса без уведомленияпотребителей. Например, вы можете переместить сервис на сервера большей мощностидля увеличения производительности.

**Независимость передачи данных.**Виртуализация сервиса позволяет снабжать сервис несколькими средствами передачиданных. Предположим, вы создали сервис «CreateOrder», доступный через JMS (JavaMessage Service). Сервис стал популярен и некоторые пользователи желают расширитьфункциональность своих приложений данным сервисом. Сложность в том, что они могутиспользовать HTTP-протокол. Обычно требуется создать другую реализацию сервиса“CreateOrder” для поддержки HTTP, но возможности виртуальных сервисов позволяетсоздать виртуальный HTTP-сервис без изменения реализации. Это прозрачно решаетпроблему взаимодействия и позволяет расширять число пользователей сервиса.

**Независимость сообщений.**Иногда потребители сервиса не синхронизированы с поставщиками в смысле ожидаемых сервисом XML-сообщений. В таких ситуациях виртуализация сервисапредлагает трансформировать сообщения между форматами поставщика и потребителя.Подобный эффект может быть получен, например, при введении в эксплуатацию новойверсии сервиса и изменении XML-схем, определяющих параметры сообщений. Предполагается, что потребители сервиса должны всегда соблюдать ожидаемый поставщиком формат. Но, при изменениях, сложно заставить всех потребителей мгновенно приспособиться.

**Типовые функции виртуального сервиса**Виртуальный сервис – наилучшее место реализации некоторых технических условий или обеспечения качества сервиса (QualityOfService):

1. Проверка XML сообщений на корректность формата и соответствие интерфейсу сервиса.

2. Аутентификация и авторизация: идентификация потребителя сервиса и проверка наличия у него прав для вызова сервиса.

3. Расшифровка сообщений и проверка подписи.

4. Балансировка нагрузки и гарантии наличия ресурсов для работысервиса.

5. Маршрутизация сообщений. Передача сообщений различным реализациям сервиса в зависимости от содержимого сообщений или внешних условий.

6. Мониторинг работы сервиса, производительности, а также проверка предоставления поставщикам требуемых услуг (SLA).