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 - 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. SOA помогает идентифицировать эти функции и собрать их в повторно используемые самодостаточные сервисы, не подверженные влиянию изменений в процессах, их использующих.
* Интеграция с партнерами. SOA способствует применению стандартов, создающих единые критерии для работы всех заинтересованных сторон. Кроме того, обеспечиваемая архитектурой SOA гибкость улучшает процесс интеграции благодаря возможности подключать, изменять и обновлять сервисы практически незаметно для ваших клиентов.
* Наличие работающих старых технологий. Некоторые организации не желают отказываться от проверенных и надежных старых технологий. Вопросы безопасности делают некоторых пользователей, особенно в сфере банковского обслуживания, недоверчивыми к новым программным системам и их неисследованным уязвимостям. В таких ситуациях SOA может помочь облачить старые технологии в новые стандарты, отобразить их в основанной на стандартах среде и сделать пригодными для интеграции и повторного использования.

Из-за неизбежности изменений единственным гарантом обеспечения непрерывности бизнес-деятельности является способность адаптироваться к изменениям и быть готовым к ним (подвижность бизнеса - agility). SOA обеспечивает возможность адаптации к бизнес-требованиям (что имеет решающее значение для будущего любой деятельности), благодаря следующим факторам:

Слабое связывание

1. Устраняет жесткие связи, препятствующие изменениям.

2. Меньше вложений в реализацию и больше в повторное использование.

3. Улучшает возможности удаленного доступа к оригинальным источникам информации, уменьшая задержки и зависимости.

4. Проекты по интеграции управляются бизнес-требованиями (то есть бизнесдеятельность является основной движущей силой).

5. Благодаря отображению и совместному использованию информации, слабое связывание позволяет компаниям извлекать в режиме реального времени больше данных об эффективности бизнес-деятельности.

6. Облегчает партнерам взаимодействие с вашей компанией.

7. Способствует продвижению и публикации ваших сервисов, облегчая клиентам обнаружение их и вашей компании.

8. Облегчает поиск новых партнеров и сервисов, помогая найти более подходящий под ваши требования сервис.

Повторное использование

1. Делает процессы более согласованными, поскольку они базируются на одних и тех же компонентах.

2. Способствует повышению качества благодаря конкуренции между провайдерами сервисов.

3. Позволяет изменять систему независимо от изменений бизнес-деятельности.

4. Уменьшает влияние изменений, поскольку они выполняются централизовано и охватывают все участвующие стороны.

Расширяемость

1. Делает SOA-решения доступными организациям любого размера.

2. Изменяет процесс разработки на более динамичный, более подходящий для ведения бизнес-деятельности.

3. Ускоряет слияния и поглощения.

Определение сервиса в SOA

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. Виртуализация сервиса обеспечивает гибкость, необходимую при внедрении SOA. Эта гибкость основана на факте, что виртуальный сервис разделяет поставщика и потребителя в терминах местоположения, передачи данных и сообщений.

**Независимость местоположения.**

Виртуальный сервис позволяет скрыть действительное местоположение сервиса от потребителей. Это дает свободу перемещать реализацию сервиса без уведомления потребителей. Например, вы можете переместить сервис на сервера большей мощности для увеличения производительности.

**Независимость передачи данных.**

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

**Независимость сообщений.**

Иногда потребители сервиса не синхронизированы с поставщиками в смысле 7 ожидаемых сервисом XML-сообщений. В таких ситуациях виртуализация сервиса предлагает трансформировать сообщения между форматами поставщика и потребителя. Подобный эффект может быть получен, например, при введении в эксплуатацию новой версии 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). Данные требования изменяются много чаще, чем функциональная логика сервисов.