Сервис-ориентированная архитектура

**XML и SOA**Архитектура SOA основывается на открытых стандартах и поддерживает платформенно-независимую бизнес-интеграцию, но она нуждается в общей технологии представления данных, на которой будет базироваться ее инфраструктура.

В центре этой инфраструктуры находится технология XML. Тому есть целый ряд причин:

· XML является фундаментом практически всех стандартов web-сервисов.

· Использование XML решает проблему работы с различными форматами данных в различных приложениях, работающих на разных платформах.

· Преимущество XML заключается в простоте представления, являющегося по своей природе текстовым, гибким и расширяемым.

Примеры стандартов, основанных на XML и используемых в SOA:

**SOAP**. Этот простой основанный на XML протокол позволяет приложениям обмениваться информацией по транспортным протоколам.

**WSDL.** Это документ, написанный на XML и описывающий web-сервис.

**Реестры сервисов**Роль реестра сервисов описывается ниже:

* Реестр сервисов реализует SOA слабое связывание.
* Реестр сервисов позволяет системным аналитикам исследовать корпоративный портфель бизнес-сервисов.
* Реестр сервисов может выполнять функцию управления сервисами, обязывая подписывающиеся сервисы быть согласованными.

**Бизнес-процесс**

Бизнес-процесс может быть определен как набор взаимосвязанных задач, относящихся к деятельности, имеющей функциональные границы. Бизнес-процессы имеют начальные и конечные точки и являются повторяемыми

В парадигме SOA бизнес-процесс управляет потоком сервисов. Бизнес-процесс управляет потоком событий, вызывает и координирует сервисы и создает контекст для их взаимодействия.

**Элементы бизнес-процесса**

* Входные данные (input) - информация, необходимая процессу для формирования результата.
* Выходные данные (output) - все данные и информация, сгенерированные процессом.
* События (events) - уведомления о возникновении чего-либо важного, например, визуальная индикация.
* Подпроцесс (subprocess) - более мелкий процесс или этап в рамках процесса.
* Действие (activity) - наименьший элемент работы в процессе.
* Показатели производительности (performance metrics) - атрибуты, представляющие эффективность процесса для определения его соответствия необходимой производительности.

**Как SOA управляет транзакциями?**  
Для решения проблемы транзакций был разработан ряд спецификаций web-сервисов:

* WS-Coordination. Позволяет зарегистрированным процессам принимать участие в создании общего контекста, ответственного за хранение текущих данных и распространяемой между ними информации.
* WS-AtomicTransaction. Используется в краткосрочных распределенных действиях. Предоставляет три типа протоколов, которые могут использоваться с интегрированной средой WS-Coordination для реализации транзакций с двухфазной фиксацией типа ACID (транзакций, поддерживающих атомарность, согласованность, изоляцию и устойчивость)
* WS-BusinessActivity. - Этот протокол используется с долго работающими транзакциями.

**Составляющие базовой архитектуры SOA**  
Базовая архитектура SOA состоит из провайдера сервисов, сервиса и (необязательного) каталога сервисов. Для обмена информацией используется механизм обмена сообщениями типа "приложение к приложению".



Ниже приведены определения некоторых понятий, используемых в данном разделе:

* Провайдер сервиса. Предоставляет сервисы, контракт по активизации которых и месторасположение опубликованы.
* Потребитель сервиса. Потребляет сервисы, соответствующие его бизнес- потребностям и обнаруженные в каталоге сервисов.
* Каталог сервисов. Служит для публикации и ведения списка сервисов, доступных для потребителей.

**Роль ESB в архитектуре SOA**  
Enterprise Service Bus (сервисная шина предприятия) — подход к построению распределённых корпоративных информационных систем. Обычно включает в себя промежуточное ПО, которое обеспечивает взаимосвязь между различными приложениями по различным протоколам взаимодействия.

Роли ESB в информационной системе:

1. Устанавливает явные независимые от реализации интерфейсы для организации слабого связывания.
2. Использует коммуникационные протоколы, независимые от расположения взаимодействующих сторон.
3. Способствует определению сервисов, инкапсулирующих повторно используемые бизнес-функции.
4. Предоставляет средства для управления инфраструктурой сервисов.
5. Функционирует в распределенной гетерогенной среде через поддержку синхронных и асинхронных взаимодействий, а также использование стандартных интерфейсов.
6. Централизует управление и распределяет обработку.
7. Реализует защиту и обеспечение качества сервиса в проектах SOA.

Недостатки ESB:

1. Требует достаточно больших трудозатрат и специфических знаний для реализации, при этом сама по себе (без дальнейшей реализации SOA) практически не приносит ощутимой пользы для бизнеса;
2. По сравнению с простейшей (точка-точка) интеграцией между системами, вносит задержки, связанные с преобразованием XML сообщений.
3. Требует тщательного продумывания и контроля над версионностью сообщений, в противном случае может увеличить связность систем друг с другом (при недостаточной унификации сообщений);

**Оркестровка и хореография**ИТ организаций должны адаптироваться под требования клиентов и условия рынка.Существующие языки описания бизнес-процессов не поддерживают напрямую веб-сервисы, что заставляет организации разрабатывать собственные проприетарные протоколы компоновки сервисов. Стандарты оркестровки и хореографии позволяютрешать эти задачи.

**Оркестровка** в бизнес-процессах – это серия действий в управляемом потоке работ, обычно имеющем одну линию выполнения.

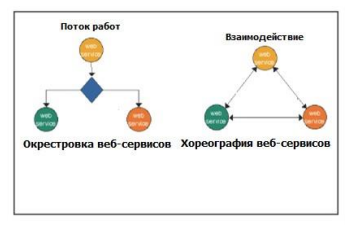
**Хореография** отражает видимый обмен сообщениями, правила взаимодействий и соглашения между двумя и более сервисами.

Ключевые элементы проектирования:

Для **оркестровки**: участник и его роль, переменные и свойства, определяющие взаимодействие участников, обработчики ошибок, события.

Для **хореографии**: структура сообщений, асинхронная и синхронная коммуникация сервисов, служебные сообщения.

Крис Пельтц – «Оркестровка и хореография веб-сервисов»



Технические требования для оркестровки и хореографии.  
Определим требования к оркестровке и хореографии как к инфраструктуре СОА:

1. Гибкость. Достигается разделением между логикой процесса и веб-сервисами..
2. Простые и структурированные действия. Язык оркестровки должен поддерживать действия как для обращения к другим веб-сервисам, так и для описания семантики процесса.
3. Рекурсивная композиция. Отдельный бизнес-процесс может взаимодействовать с множеством веб-сервисов.

WSCI определяет расширение WSDL для взаимодействия сервисов. Первоначально составленный в Sun, SAP, BEA и Intalio, он стал спецификацией W3C. Это язык хореографии, описывающий обмен сообщениями между сервисами и не определяющий выполнение бизнес-процесса BPEL4WS поддерживает как абстрактные бизнес-протоколы, так и выполняемые бизнес-процессы.