

## Сервис-ориентированная архитектура

### XML и SOA

XML является центром инфраструктуры SOA, потому что:

- XML является фундаментом практически всех стандартов web-сервисов, в том числе XML Schema, SOAP, WSDL (Web Services Description Language) и UDDI (Universal Description, Discovery, and Integration). Эти стандарты опираются на основополагающую концепцию основанных на XML представлений - поддерживаемый во всем мире формат обмена информацией между провайдерами сервисов и инициаторами запросов в SOA.
- 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.

Используемые SOA стандарты:

- SOAP – основанный на XML протокол обмена информацией.
- WSDL – документ, написанный на XML и описывающий web-сервис, определяет месторасположение сервиса и отображаемые им операции. WSDL-файл описывает:
  - Сервисы, доступные через интерфейс web-сервиса, такие как список имен методов и сообщений-атрибутов.
  - Тип данных сообщений.
  - Адрес сервиса, используемый для его вызова.

### Реестры сервисов

Это каталог сервисов, доступных в системе SOA. Он содержит физическое месторасположение сервисов, версии и их срок действия, а также документацию по сервисам.

Роль реестра в архитектуре SOA:

- реализует SOA слабое связывание, облегчает потенциальные сложности замены одной реализации сервиса другой при необходимости.
- позволяет системным аналитикам исследовать корпоративный портфель бизнес-сервисов.
- может выполнять функцию управления сервисами, обязывая подписывающиеся сервисы быть согласованными.

### Что такое 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.

Бизнес-процесс управляет протоколами сервисов и 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. Выходные данные представляют собой бизнес-цели и показатели, необходимые для бизнес-деятельности.

События (events) - уведомления о возникновении чего-либо важного, например, визуальная индикация. Они могут возникать до, во время и после выполнения процесса.

Подпроцесс (subprocess) - более мелкий процесс или этап в рамках процесса. Подпроцесс используется тогда, когда невозможно представить объем работы одним набором действий. Он имеет те же элементы, что и процесс.

Действие (activity) - наименьший элемент работы в процессе.

Показатели производительности (performance metrics) - атрибуты, представляющие эффективность процесса для определения его соответствия необходимой производительности. Эти показатели помогают определить производительность и сравнить ее с требуемыми значениями. Они также выделяют потенциальные области совершенствования процесса, реализуя в конечном итоге цикл улучшений, обещанный архитектурой SOA.

## **Как SOA управляет транзакциями?**

Для решения проблемы транзакций был разработан ряд спецификаций web-сервисов. К ним относятся:

- **WS-Coordination.** Позволяет зарегистрированным процессам принимать участие в создании общего контекста, ответственного за хранение текущих данных и распространяемой между ними информации. Координация обработки существующих транзакций, потоков работ и других систем осуществляется интегрированной средой. Это позволяет скрыть проприетарные протоколы и работать в гетерогенной среде. Этот протокол обеспечивает инфраструктуру для других протоколов, таких как WS-AtomicTransaction или WS-BusinessActivity.
- **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.

## **Основы архитектуры SOA**

### **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".

В этой модели базовый сценарий таков. Сначала провайдер сервиса создает сервис, принимает решение открыть этот сервис и публикует его. Публикация выполняется путем отправки информации о сервисе в каталог сервисов. С другой стороны, инициатор запросов сервиса (service requester), нуждаясь в определенном сервисе, просматривает каталог сервисов в поисках того из них, который удовлетворяет необходимому критерию. После обнаружения такого сервиса и использования доступной в каталоге сервисов информации инициатор запросов сервиса может напрямую обратиться к провайдеру сервисов надлежащим способом для удовлетворения бизнес-потребности.

*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 распределенных корпоративных информационных систем. Обычно включает в себя промежуточное ПО, которое обеспечивает взаимосвязь между различными приложениями по различным протоколам взаимодействия.

Архитектура ESB заключается во взаимодействии всех приложений через единую точку, которая, при необходимости, обеспечивает транзакции, преобразование данных, сохранность обращений. Данный подход обеспечивает большую гибкость, простоту масштабирования и переноса. При замене одного приложения подключенного к шине нет необходимости перенастраивать остальные. Она предоставляет магистральную сеть и инфраструктуру для соединения провайдеров и потребителей сервисов в архитектуре SOA.

Роли ESB в информационной системе:

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

#### Недостатки ESB:

1. Требуется 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. Требуется тщательного продумывания и контроля над версионностью сообщений, в противном случае может увеличить связность систем друг с другом (при недостаточной унификации сообщений).

#### Оркестровка и хореография

Организация OMG определяет оркестровку как «моделирование направленных, внутренних бизнес-процессов», а хореографию как «спецификацию взаимодействий между автономными процессами».

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.

Стандарты WSCI (Web Service Choreography Interface) и BPEL4WS (Business Process Execution Language for Web Services) разработаны для облегчения взаимодействия веб-сервисов.

Оркестровка отличается от хореографии тем, что она описывает процесс, протекающий между сервисами, контролируемый основным участником. В хореографии нет участника, ведущего обмен сообщениями.

Технические требования для оркестровки и хореографии:

- гибкость;
- простые и структурированные действия;
- рекурсивная композиция;
- дополнительные требования:
  - хранение состояний и корреляция запросов;
  - обработка исключений и транзакций.

#### WSCI

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. Один интерфейс описывает только сообщение единственного участника обмена. Хореография включает набор интерфейсов, по одному на каждого участника. Нет контроллера, регулирующего обмен. WSDL описывает входные точки сервиса, WSCI описывает взаимодействие между WSDL-операциями.

### BPEL4WS

BPEL4WS поддерживает как абстрактные бизнес-протоколы, так и выполняемые бизнес-процессы.

- Бизнес-протокол поддерживает публичный обмен сообщениями между участниками обмена. Его нельзя выполнить и он не определяет внутреннее выполнение процесса.
- Выполняемый процесс моделирует выполнение действий. Он обеспечивает оркестровку, в то время как бизнес-протоколы сфокусированы на хореографии.

Спецификация поддерживает простые действия для общения с веб-сервисами. Типичный сценарий заключается в приеме сообщения выполняемым процессом. Процесс запрашивает дополнительную информацию, вызывает внутренние сервисы и возвращает результаты.

BPEL4WS поддерживает структурированные действия для построения бизнес-логики процесса. Переменные и партнеры – важные составляющие BPEL4WS

Процесс, определенный в BPEL4WS, состоит из:

- Действий (activities), которые являются отдельными бизнес-этапами внутри процесса. Действия могут быть простыми или состоять из других действий (структурированными).
- Ссылок на партнеров, которые определяют внешние сущности, взаимодействующие с процессом или, наоборот, использующие WSDL-интерфейсы.
- Переменных, хранящих сообщения, передаваемые между действиями, и, следовательно, представляющих состояние.
- Корреляционных наборов (correlation sets), использующихся для корреляции нескольких сообщений запросов сервиса и ответов с одним экземпляром бизнес-процесса. (К сервису могут обращаться различные бизнес-процессы, нельзя смешивать результаты работы для разных бизнес-процессов).
- Обработчиков неисправностей (fault handlers), занимающихся исключительными ситуациями, которые могут возникнуть во время работы бизнес-процесса.
- Обработчиков событий (event handlers), принимающих и обрабатывающих сообщения параллельно с обычным выполнением процесса.
- Корректирующих обработчиков (compensation handlers), определяющих логику коррекции для отката действия или нескольких действий при возникновении исключительной ситуации.