ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО/МЭК 18384-1— 2017

Информационные технологии

ЭТАЛОННАЯ АРХИТЕКТУРА ДЛЯ СЕРВИС-ОРИЕНТИРОВАННОЙ АРХИТЕКТУРЫ (SOA RA)

Часть 1

Терминология и концепции SOA

(ISO/IEC 18384-1:2016, IDT)

Издание официальное

Москва
Стандартинформ
2017

gost.e-ecolog.ru

Предисловие

1 ПОДГОТОВЛЕН Обществом с ограниченной ответственностью «Информационно-аналитический вычислительный центр» (ООО ИАВЦ) на основе собственного перевода на русский язык англоязычной версии международного стандарта, указанного в пункте 4

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 22 «Информационные технологии»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 5 сентября 2017 г. № 1012-ст

4 Настоящий стандарт идентичен международному стандарту ИСО/МЭК 18384-1:2016 «Информационные технологии. Эталонная архитектура для сервис-ориентированной архитектуры (SOA RA). Часть 1. Терминология и концепции SOA» (ISO/IEC 18384-1:2016 «Information technology — Reference Architecture for Service Oriented Architecture (SOA RA) — Part 1: Terminology and concepts for SOA», IDT).

ИСО/МЭК 18384-1 разработан подкомитетом ПК 38 «Облачные вычисления и распределенные платформы» Совместного технического комитета СТК 1 «Информационные технологии» Международной организации по стандартизации (ИСО) и Международной электротехнической комиссии (МЭК)

5 ВВЕДЕН ВПЕРВЫЕ

6 Некоторые положения международного стандарта, указанного в пункте 4, могут являться объектом патентных прав. ИСО и МЭК не несут ответственности за идентификацию подобных патентных прав

*Правила применения настоящего стандарта установлены в статье 26 Федерального закона от 29 июня 2015 г. № 162-ФЗ «О стандартизации в Российской Федерации». Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (*gost.e-ecolog.ru

© Стандартинформ, 2017

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

Содержание

1 Область применения . . . 1
2 Термины и определения . . . 1
3 Сокращения . . . 6
4 Обозначения . . . 7
4.1 Общие положения . . . 7
4.2 UML . . . 7
4.3 Диаграммы «сущность-связь» . . . 7
4.4 Циклические диаграммы . . . 7
4.5 Диаграммы потоков . . . 7
5 Соглашения . . . 7
6 Соответствие . . . 8
7 Концепции SOA . . . 8
7.1 Введение в SOA . . . 8
7.2 Концепции . . . 9
7.3 Сквозные аспекты . . . 24
8 Архитектурные принципы SOA . . . 29
8.1 Определение архитектурных принципов . . . 29
8.2 Функциональная совместимость — синтаксически и семантически . . . 30
8.3 Описание службы . . . 30
8.4 Повторное использование службы . . . 31
8.5 Обнаружение службы . . . 32
8.6 Позднее связывание . . . 32
8.7 Компонуемость . . . 33
8.8 Автономность . . . 33
8.9 Слабое связывание . . . 33
8.10 Управляемость . . . 34
Приложение А (справочное) Структура управления SOA . . . 35
Приложение В (справочное) Управление и проблемы безопасности . . . 38
Библиография . . . 43

Введение

Комплекс стандартов ИСО/МЭК 18384 под общим наименованием «Эталонная архитектура для сервис-ориентированной архитектуры (SOA RA)» состоит из следующих частей:

- Часть 1. Терминология и концепции SOA;
- Часть 2. Эталонная архитектура для решений SOA;
- Часть 3. Онтология сервис-ориентированной архитектуры.

Сервис-ориентированная архитектура (SOA) представляет собой архитектурный стиль, в котором бизнес-системы и ИТ-системы разрабатываются с точки зрения служб, доступных через интерфейс, и результатов действий этих служб. Служба — это логическое представление набора действий, порождающих заданные результаты; служба является автономной, может состоять из других служб, при этом потребители данной службы не обязаны знать ее внутреннюю структуру.

В рамках SOA «служба» является основным элементом для сборки и интеграции информационных систем, которые способны удовлетворять различным требованиям при решении той или иной задачи. SOA позволяет осуществлять взаимодействие между бизнесами без необходимости выделять характерные особенности любого конкретного бизнес-домена. Использование архитектурного стиля SOA позволяет повысить эффективность разработки информационных систем, интеграции и повторного использования ИТ-ресурсов. Кроме того, использование архитектурного стиля SOA может помочь информационным системам гибко и быстро реагировать на постоянно меняющиеся потребности бизнеса.

Настоящий стандарт устанавливает единый свод технических принципов SOA, определенных норм и стандартов для глобального рынка с целью создания единого понимания SOA, а также улучшения стандартизации и качества решений.

Настоящий стандарт определяет терминологию SOA, ее технические принципы, эталонную архитектуру и онтологию SOA. Целевая аудитория настоящего стандарта включает в себя (но не ограничивается ими): организации по стандартизации, архитекторов, методологов по архитектуре, разработчиков систем и программного обеспечения, деловые круги, поставщиков служб SOA, разработчиков решений SOA и служб, а также потребителей службы SOA, которые заинтересованы в принятии и разработке SOA. Настоящий стандарт может быть использован для определения понятий SOA и руководства разработкой и управлением решениями SOA.

Комплекс стандартов ИСО/МЭК 18384 состоит из трех частей:

a) ИСО/МЭК 18384-1 определяет терминологию, основные принципы и понятия SOA;

b) ИСО/МЭК 18384-2 детализирует уровни эталонной архитектуры SOA, включая метамодель, возможности, стандартные блоки архитектуры, а также типы служб в решениях SOA;

c) ИСО/МЭК 18384-3 определяет базовые понятия SOA и их отношения в онтологии.

Пользователи настоящего стандарта сочтут его полезным для понимания основ SOA. Настоящий стандарт следует изучить, прежде чем приступить к чтению или применению ИСО/МЭК 18384-2. Для тех, кто мало знаком с SOA, следует обратиться к разделу 4 ИСО/МЭК 18384-2:2016 для получения общего понимания эталонной архитектуры для решений SOA. В остальных разделах всесторонне представлены детали стандартных архитектурных блоков и компромиссов, необходимых для решений SOA. ИСО/МЭК 18384-3 содержит онтологию SOA, которая формализует базовые понятия и термины и вводит отображения в UML и OWL. Онтология SOA может использоваться независимо или в сочетании с настоящим стандартом и ИСО/МЭК 18384-2.

В настоящем стандарте представлены и разъясняются основные концепции SOA. В нем даны определения терминов, используемых в комплексе стандартов ИСО/МЭК 18384 со специфическими значениями, которые могут отличаться или быть более точными, чем определения этих терминов, встречающиеся в основных словарях английского языка. Термины, определенные в настоящем стандарте, используются уникальным образом для SOA. Слова английского языка, используемые в обычном значении, не переопределены.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационные технологии

ЭТАЛОННАЯ АРХИТЕКТУРА ДЛЯ СЕРВИС-ОРИЕНТИРОВАННОЙ АРХИТЕКТУРЫ (SOA RA)

Часть 1

Терминология и концепции SOA

Information technology. Reference architecture for service oriented architecture (SOA RA).
Part 1.Terminology and concepts for SOA

Дата введения — 2018—09—01

1 Область применения

В настоящем стандарте представлен словарь, методические рекомендации и общие технические принципы, лежащие в основе сервис-ориентированной архитектуры (SOA), включая принципы, связанные с проектированием функциональности, производительностью, разработкой, развертыванием и управлением.

2 Термины и определения

В настоящем стандарте применены следующие термины с соответствующими определениями:

2.1

агент (actor): Человек или компонент системы, взаимодействующий с системой в целом, который оказывает воздействия, вызывающие последующие действия.

[ИСО/МЭК 16500-8:1999, статья 3.1]

2.2

архитектура (architecture): Основные понятия или свойства системы в окружающей среде, воплощенной в ее элементах, отношениях и конкретных принципах ее проекта и развития.

[ИСО/МЭК/ИИЭР 42010:2011, статья 3.2]

2.3 **хореография** (choreography): Тип *композиции* (2.5), *элементы* (2.8) которой взаимодействуют децентрализованным образом с каждой автономной частью, зная определенный заранее шаблон поведения для всей (глобальной) композиции и следуя ему.

Примечания

1 Хореография не требует полного или точного знания шаблона поведения.

2 См. ИСО/МЭК 18384-3:2016, статья 8.3.

2.4 **коллаборация** (collaboration): Тип *композиции* (2.5), *элементы* (2.8) которой взаимодействуют децентрализованным образом, каждый из них — согласно своему собственному плану и целям без предопределенного шаблона поведения.

Примечание — См. ИСО/МЭК 18384-3:2016, статья 8.3.

2.5 **композиция** (composition): Результат сборки набора *элементов* (2.8) для конкретной цели.

Примечание — См. ИСО/МЭК 18384-3:2016, статья 8.2.

Издание официальное

2.6 **оконечная точка** (end point): Местоположение, в котором принимается информация для вызова и конфигурации взаимодействия.

2.7 **эффект** (effect): Результат взаимодействия со *службой* (2.20).

Примечания

1 Эффект заключается в том, каким образом служба предоставляет результаты своему потребителю посредством *элемента* (2.8), выполняющего ее.

2 См. ИСО/МЭК 18384-3:2016, статья 7.10.

2.8 **элемент** (element): Единица на заданном уровне абстракции, имеющая четко определенные границы.

Примечания

1 Элемент может быть *сущностью* (2.9) любого типа.

2 См. ИСО/МЭК 18384-3:2016, статья 5.1.

2.9 **сущность** (entity): Отдельный идентифицируемый *элемент* (2.8) в системе, который может выступать в качестве *поставщика службы* (2.50) или *потребителя службы* (2.29).

Примечание — Примерами сущностей являются организации, предприятия и физические лица, программное обеспечение и аппаратные средства.

2.10 **событие** (event): Нечто, что происходит и на что *элемент* (2.8) может решить дать ответ.

Примечания

1 Любой элемент может генерировать (порождать) событие или отвечать на событие.

2 См. ИСО/МЭК 18384-3:2016, статья 10.

2.11 **контекст выполнения** (execution context): Набор технических и бизнес *элементов* (2.8), необходимый тем, у кого имеются потребности и возможности разрешить *поставщикам служб* (2.50) и *потребителям служб* (2.29) создание экземпляра службы и взаимодействие со службой.

Примечания

1 Контекст выполнения для *взаимодействия со службой* (2.37) представляет собой набор элементов инфраструктуры, процессов, соглашений и положений политик, которые идентифицируются как часть взаимодействия между экземплярами служб и таким образом формируют связь между тем, кто может предоставить возможности и тем, кому это необходимо.

2 См. [8].

2.12 **интерактивный агент** (human actor): *Агент* (2.1), ограниченный *сущностью* (2.9), представляющей физическое лицо или организацию.

Примечания

1 Данная классификация не является исчерпывающей.

2 См. ИСО/МЭК 18384-3:2016, статья 6.2.

2.13 **интерактивная задача** (human task): Задача (2.60), которая выполняется интерактивным агентом (2.12).

2.14

интерфейс (interface): Совместно используемая граница между двумя функциональными единицами, определяемая различными функциональными характеристиками, параметрами физического соединения, параметрами взаимосвязи при обмене сигналами, а также другими характеристиками в зависимости от задаваемых требований.

[ИСО/МЭК 2382:2015, статья 2121308]

2.15 **слабое связывание** (loose coupling): Принцип, в котором зависимости между *элементами* (2.8) *решения SOA* (2.56) преднамеренно уменьшены.

2.16 **оркестровка** (orchestration): Тип *композиции* (2.5), где один определенный *элемент* (2.8) используется для наблюдения за другими элементами и управления ими.

Примечания

1 Элемент, который управляет оркестровкой, сам не является частью оркестровки (экземпляра композиции).

2 См. ИСО/МЭК 18384-3:2016, статья 8.3.

2.17 **политика** (policy): Положения, которые намеревается выполнять *сущность* (2.9) либо рассчитывает, что их будет выполнять другая сущность.

Примечание — См. ИСО/МЭК 18384-3:2016, статья 9.

2.18 **процесс** (process): Тип *композиции* (2.5), *элементы* (2.8) которой составлены в последовательность или поток действий и взаимодействий с целью выполнения определенной работы.

Примечания

1 Процесс также может быть *коллаборацией* (2.4), *хореографией* (2.3) или *оркестровкой* (2.16).

2 См. ИСО/МЭК 18384-3:2016, статья 8.6.

2.19 **наблюдаемый эффект** (real-world effect): Изменение, относящееся к конкретным заинтересованным сторонам.

Примечание — См. [8].

2.20 **служба** (service): Логическое представление ряда действий, который имеет определенные результаты, является автономным, может быть составлен из других служб и является «черным ящиком» для потребителей данной службы.

Примечания

1 Термин «действие» (activity) в определении «службы» используется в общем значении в обычном языке, а не в специальном значении для процессов [т. е. действие не обязательно является действием *процесса* (2.18)].

2 См. ИСО/МЭК 18384-3:2016, статья 7.2.

2.21 **брокер службы** (service broker): *Элемент* (2.8), который обеспечивает связь со *службами* (2.20) либо на деловом уровне, либо на уровне реализации, т. е. через промежуточное программное обеспечение (ПО).

Примечание — Промежуточное ПО предоставляет *поставщикам службы* (2.50) и *потребителям службы* (2.29) одну или несколько функций, таких как единая *регистрация службы* (2.51) и публикация, *обнаружение службы (2.34)*, маршрутизация, независимый от местоположения доступ к службе.

2.22 **шина служб** (service bus): Шаблон проектирования и времени выполнения для обеспечения взаимодействий *служб* (2.20), таких как передача данных, доступ, потребление, преобразование, промежуточное ПО и маршрутизация сообщений.

Примечание — Шина служб может включать в себя как логический набор таких функций, так и функции, собранные в единый коммерческий продукт. Шина служб широко используется в организационном контексте и часто совпадает с шиной служб предприятия (Enterprise Service Bus, ESB).

2.23 **служба-кандидат** (service candidate): *Служба* (2.20), идентифицированная во время *жизненного цикла решения SOA* (2.58), отвечающая широким требованиям к службе; из набора служб-кандидатов выбирается одна или несколько служб для дальнейшей разработки как части полного *решения SOA* (2.56).

2.24 **каталог службы** (service catalogue), **реестр/репозиторий служб** (service registry/repository, reg/rep): Логический набор *описаний службы* (2.31) и связанных артефактов, который поддерживает публикацию, регистрацию, поиск, управление и извлечение этих артефактов.

2.25 **хореография служб** (service choreography): *Хореография* (2.3), *элементы* (2.8) которой являются *службами* (2.20).

Примечание — См. ИСО/МЭК 18384-3:2016, статья 8.

2.26 **коллаборация служб** (service collaboration): *Коллаборация* (2.4), *элементы* (2.8) которой являются *службами* (2.20).

Примечание — См. ИСО/МЭК 18384-3:2016, статья 8.

2.27 **компонент службы** (service component): *Элемент* (2.8), который реализует *службы* (2.20).

2.28 **композиция службы/сборка службы** (service composition/service assembly): *Композиция (2.5)*, предоставляющая (в функциональном смысле) высокоуровневые *службы* (2.20), которые являются лишь сборкой других *служб* (2.20).

Примечания

1 Композиция может поддерживать различные шаблоны композиции, такие как *коллаборация* (2.4), *хореография* (2.3), *оркестровка* (2.16).

2 См. ИСО/МЭК 18384-3:2016, статья 8.

2.29 **потребитель службы** (service consumer): *Сущность* (2.9), которая использует *службы* (2.20).

Примечания

1 Потребители могут взаимодействовать со службами функционально или по контракту (юридическая ответственность).

2 См. ИСО/МЭК 18384-3:2016, статья 7.4.

2.30 **контракт службы** (service contract): Термины, условия и правила взаимодействия, которые принимают (прямо или косвенно) взаимодействующие *потребители службы* (2.29) и *поставщики службы* (2.50).

Примечания

1 Контракт службы является обязательным для всех участников взаимодействия, включая саму *службу* (2.20) и *элемент* (2.8), который ее предоставляет, для конкретного рассматриваемого взаимодействия.

2 См. ИСО/МЭК 18384-3:2016, статья 7.6.

2.31 **описание службы** (service description): Информация, необходимая для использования *службы* (2.20) или принятия решения об ее использовании.

Примечания

1 Описание службы обычно включает *интерфейсы службы* (2.38), контракты и политики.

2 См. ИСО/МЭК 18384-3:2016, статья 7.

2.32 **развертывание службы** (service deployment): Деятельность, посредством которой реализации *служб* (2.20) становятся доступными в конкретной аппаратно-программной среде и могут использоваться потребителями *службы* (2.29).

2.33 **проектирование службы** (service development): Деятельность по выявлению потребностей и ограничений и проектированию *службы* (2.20) как части *решения SOA* (2.56), для того, чтобы удовлетворять данным потребностям в рамках заданных ограничений.

2.34 **обнаружение службы** (service discovery): Деятельность, посредством которой *потребитель службы* (2.29) может найти *службы* (2.20), удовлетворяющие его определенным функциональным и/или нефункциональным требованиям.

2.35 **руководство службой** (service governance): Стратегия и механизм управления, которые применяются в *жизненном цикле службы* (2.41) и портфеле службы, включающие установление цепочек ответственности, мониторинг за соблюдением политик путем предоставления соответствующих *процессов* (2.18) и измерений в рамках *руководства решением SOA* (2.57).

Примечание — Перечень аспектов жизненного цикла службы, которыми необходимо управлять, включает в себя: рассмотрение модификаций службы, обновлений версии, уведомлений о прекращении функционирования; декомпозиции, способности агентства, возможности декомпозиции и возможности удовлетворения индивидуальным требованиям.

2.36 **реализация службы** (service implementation): Деятельности по технической разработке и физической реализации *службы* (2.20), которые являются частью *жизненного цикла службы* (2.41) и приводят к созданию *компонента службы* (2.27).

2.37 **взаимодействие со службой** (service interaction): Деятельность, связанная с использованием предлагаемой возможности, как правило, через границу владения, для достижения определенного желаемого *наблюдаемого эффекта* (2.19).

Примечание — См. [8].

2.38 **интерфейс службы** (service interface): *Интерфейс* (2.14), посредством которого другие *элементы* (2.8) могут взаимодействовать и обмениваться информацией со службой, где форма запроса и результат запроса находятся в *описании службы* (2.31).

Примечание — См. ИСО/МЭК 18384-3:2016, статья 7.13.

2.39 **функциональная совместимость службы** (service interoperability): Возможность *поставщиков службы* (2.50) и *потребителей службы* (2.29) взаимодействовать и вызывать *службы* (2.20), а также обмениваться информацией на синтаксическом и семантическом уровнях, порождая эффекты, в соответствии с *описанием службы* (2.31).

2.40 **соглашение об уровне обслуживания** (service level agreement, SLA): Тип *контракта службы* (2.30), который определяет измеримые условия взаимодействий между *поставщиком службы* (2.50) и *потребителем службы* (2.29).

Примечание — Соглашение об уровне обслуживания может определять: набор *служб* (2.20), предоставляемых поставщиком службы; конкретное определение каждой службы, обязанности поставщика службы и потребителя службы, набор метрик для определения того, насколько предоставление службы поставщиком службы соответствует заданному в контракте; механизм аудита для мониторинга службы, механизмы компенсаций для потребителей и поставщиков службы на тот случай, если условия SLA не выполняются, и как будет меняться SLA со временем.

2.41 **жизненный цикл службы** (service lifecycle): Последовательность этапов для реализации *службы* (2.20), от концепции и идентификации до создания экземпляра службы и снятия с эксплуатации.

2.42 **управление службой** (service management): Мониторинг, контроль, сопровождение, оптимизация *служб* (2.20) и управление службами.

2.43 **моделирование службы** (service modeling): Набор операций для разработки набора *служб-кандидатов* (2.23) для функций или действий над *решением SOA* (2.56) с использованием процесса *анализа, ориентированного на службы* (2.47).

2.44 **мониторинг службы** (service monitoring): Отслеживание состояния и условий эксплуатации, связанных с выполнением, производительностью и *наблюдаемыми эффектами* (2.19) *служб* (2.20).

2.45 **оркестровка служб** (service orchestration): Оркестровка (2.16), чьи *элементы* (2.8) являются *службами* (2.20).

2.46 **ориентированность на службы** (service orientation): Подход к проектированию систем с точки зрения *служб* (2.20) и разработки, основанной на службах.

2.47 **анализ, ориентированный на службы** (service oriented analysis): Предварительные шаги по сбору информации, которые выполняются в поддержку *моделирования службы* (2.43) — подпроцесса, который приводит к созданию набора *служб* (2.20).

Примечание — Анализ дает информацию для последующих фаз жизненного цикла SOA и может выполняться один раз для каждого бизнес-*процесса* (2.18) или итеративно.

2.48 **сервис-ориентированная архитектура** (service oriented architecture, SOA): Архитектурный стиль, который поддерживает *ориентированность на службы* (2.46) и является парадигмой для создания бизнес-решений.

Примечания

1 *Службы* (2.20), реализованные в этом стиле, используют действия, включающие в себя бизнес-*процессы* (2.18), имеют описания для обеспечения контекста, могут быть реализованы через *композицию служб* (2.28), могут иметь реализации, зависящие от окружения, описанные в контексте, который их ограничивает либо дает им дополнительные возможности, требуют управления и налагают требования на инфраструктуру для достижения функциональной совместимости и независимости от местоположения с использованием стандартов в максимально возможной степени.

2 См. ИСО/МЭК 18384-3:2016, статья 4.

2.49 **политика службы** (service policy): *Политика* (2.17) применительно к *службе* (2.20).

2.50 **поставщик службы** (service provider): *Сущность* (2.9), предоставляющая *службу* (2.20).

Примечания

1 Поставщики служб могут быть ответственными за функционирование служб или за контракт для служб (юридическая ответственность), или за то и другое.

2 См. ИСО/МЭК 18384-3:2016, статья 7.4.

2.51 **публикация службы/регистрация службы** (service publishing/service registration): Каталогизация *описаний служб* (2.31) в доступном месте, например в *реестре/репозитории служб* (2.24), где поддерживаются такие действия, как поиск и извлечение описаний, что позволяет обнаружить информацию потенциальным *потребителям службы* (2.29).

2.52 **реализация SOA** (SOA implementation): Методы, используемые для разработки решений на основе *SOA* (2.48).

2.53 **зрелость SOA** (SOA maturity): Оценка возможности организации по внедрению *SOA* (2.48) и текущего уровня внедрения.

2.54 **модель зрелости SOA** (SOA maturity model): Подход, включающий цели внедрения SOA и методы оценки *зрелости SOA* (2.53) организации в сопоставлении с этими целями.

2.55 **ресурс SOA** (SOA resource): *Элементы* (2.8), которые предоставляют ИТ-ресурсы, используемые *службами* (2.20).

2.56 **решение SOA** (SOA solution): Решения, частично или полностью реализованные с применением принципов, концепций и методов *SOA* (2.48).

Примечание — Решения SOA включают в себя физическое создание экземпляров *реализаций службы* (2.36), в том числе инфраструктуру, другие элементы архитектуры и возможности, необходимые для процессов управления и жизненного цикла, которые в совокупности позволяют получать предметно-ориентированные результаты, представляющие основанное на SOA решение бизнес-задач.

2.57 **руководство решением SOA** (SOA solution governance): Частный случай руководства информационными технологиями, сфокусированный на стратегиях и механизмах управления *решением SOA* (2.56) конечных пользователей.

Примечания

1 Руководство решением SOA управляет всем *жизненным циклом решения SOA* (2.58) посредством назначения персонала, ролей, процедур управления, а также принятия решений. Руководство решением SOA должно внедрить надлежащую методологию и лучшие практики. Руководство решением SOA обычно требует инструменты для настройки и управления стратегией руководства в соответствии с потребностями.

2 Хотя управление означает определенный *процесс* (2.18) руководства и контроля для выполнения политики, в рамках руководства фокусируются на назначении прав по принятию решений и принятии решения о том, какие меры следует предпринимать, а также каким политикам следовать для выполнения этих решений.

2.58 **жизненный цикл решения SOA** (SOA solution lifecycle): Набор деятельностей по технической реализации *решений SOA* (2.56), включая анализ, проектирование, реализацию, развертывание, тестирование и управление.

2.59 **управление решением SOA** (SOA solution management): Измерение, мониторинг и конфигурация всего жизненного цикла *решения SOA* (2.56).

Примечание — Во время эксплуатации управление решением SOA представляет собой набор деятельностей по измерению и управлению реализацией решения SOA в соответствии со стратегиями и механизмами, определенными процессом *руководства решениями SOA* (2.57).

2.60 **задача** (task): Атомарное действие, которое приводит к определенному результату.

Примечания

1 Задачу выполняют люди или организации, в частности *интерактивные агенты* (2.12).

2 См. ИСО/МЭК 18384-3:2016, статья 6.4.

2.61 **веб-службы** (web services): Программная система, разработанная для поддержки сетевого межмашинного взаимодействия.

Примечания

1 Исходное определение: программная система, разработанная для поддержки сетевого межмашинного взаимодействия. Она имеет *интерфейс* (2.14), описанный в формате, который может обрабатываться машиной (в частности, WSDL). Другие системы взаимодействуют с веб-службой тем способом, который определен ее описанием, с использованием сообщений SOAP, обычно передаваемым посредством HTTP с сериализацией XML в сочетании с другими стандартами, имеющими отношение к Всемирной паутине.

2 См. [8].

3 Сокращения

В настоящем стандарте используются следующие сокращения:

ABB — стандартный блок архитектуры (СБА);

BPMN — модель и нотация бизнес-процессов;

COBIT — цели управления для получения информации и связанной технологии;

EA — архитектура предприятия;

ESB — шина службы предприятия;

HTTP — протокол передачи гипертекста;

HTTPS — безопасный протокол передачи гипертекста;

IT — информационные технологии (ИТ);

ITIL — библиотека инфраструктуры информационных технологий;

KPI — ключевой показатель эффективности (КПЭ);

L2TP — протокол туннелирования второго уровня;

MPLS — многопротокольная коммутация по меткам;

OWL — язык онтологии всемирной паутины;

PKI — инфраструктура открытых ключей;

QoS — качество обслуживания;
RA — эталонная архитектура;
REST — передача состояния через представление[1)];
RPC — удаленный вызов процедур;
SLA — соглашение об уровне обслуживания;
SOA — сервис-ориентированная архитектура;
SOAP — стандартный протокол обмена структурированными данными в сообщениях;
SQL — язык структурированных запросов;
UML — унифицированный язык моделирования;
VPN — виртуальная частная сеть;
WSDL — язык описания веб-сервисов;
WSRP — веб-службы для удаленных портлетов;
XML — расширяемый язык разметки.

4 Обозначения

4.1 Общие положения

Ниже приведена инструкция по интерпретации диаграмм.

4.2 UML

Большинство диаграмм не соответствуют обозначению UML. В тексте перед диаграммами, которые соответствуют обозначению UML, указывается тип диаграммы UML, чтобы пользователь знал, как их интерпретировать.

4.3 Диаграммы «сущность-связь»

Диаграммы «сущность-связь» (Entity relationship) (такие, как приведенные на рисунках 1 и 2), состоящие из прямоугольников, линий, стрелок и обведенных кружками номеров, следует интерпретировать по следующим правилам:

- блоки — метамодельные понятия, уровни, стандартные блоки архитектуры, возможности или компоненты;
- стрелки — отношения между понятиями метамодели; односторонние стрелки показывают направление отношения; двойные стрелки указывают на то, что отношения двунаправленны;
- отношения имеют имена и представлены как маркированные строки или стрелки, число элементов, входящих в отношение не подразумевается;
- число элементов, входящих в отношение, обозначается традиционным образом с использованием математической нотации (* = = 0..*; 0..1 = = необязательный и только 1; 1 = = обязательный, в соответствии с ИСО/МЭК 15474-1).

4.4 Циклические диаграммы

Циклические диаграммы с состояниями показывают эволюцию состояния или жизненного цикла в направлении по часовой стрелке. Пример циклической диаграммы приведен на рисунке 10.

4.5 Диаграммы потоков

Диаграммы потоков часто используются в качестве примеров. Диаграммы следует интерпретировать по следующим правилам:

- прямоугольники представляют уровни, стандартные блоки архитектуры или компоненты;
- стрелки показывают направление потока между прямоугольниками;
- обведенные кружками числа на стрелках обозначают последовательность потока, а также используются для ссылок в тексте.

Примеры диаграмм потоков приведены на рисунках 5 и 6.

5 Соглашения

Настоящая эталонная архитектура определена в трех частях комплекса стандартов ИСО/МЭК 18384. Настоящий стандарт определяет терминологию и концепции для сервис-ориентированной архитектуры. Прежде чем читать или использовать ИСО/МЭК 18384-2 и ИСО/МЭК 18384-3, важно понимать термины и концепции, определенные в настоящем стандарте. ИСО/МЭК 18384-2 определяет эталонную архитектуру для решений SOA. Он описывает метамодель, ряд уровней для мно-

[1)] Технология адресации оконечных точек веб-сервисов (примечание переводчика).

гоуровневой архитектуры и ряд общих типов служб. ИСО/МЭК 18384-3 определяет онтологию SOA с более формальным выражением концепций SOA и отношений между ними. Терминология настоящего стандарта соответствует онтологии, приведенной в ИСО/МЭК 18384-3.

Настоящий стандарт можно читать последовательно или использовать в качестве справочника. Стандарт содержит словарь терминов, подробное введение и концепции SOA. В ИСО/МЭК 18384-2 дается краткое введение в SOA. После введения представлен высокоуровневый обзор всех десяти уровней и типов служб, определенных в этом стандарте. Далее следуют определения и объяснение метамодели, используемой в эталонной архитектуре SOA. Метамодель определяет уровень, возможности и концепции СБА, а также другие логические концепции. Каждый СБА и каждая возможность определены единственным образом в каждом уровне. Для выполнения требований архитектуры возможности и СБА могут потребовать возможности и СБА, определенные для других уровней. Уровни, возможности и СБА, определенные в этом стандарте, являются логическими элементами, и любое упоминание логических элементов вида «выполняет», «поддерживает» или «взаимодействует» означает, что при разработке решения SOA физическая реализация возможностей и СБА в действительности «выполняет», «поддерживает» или «взаимодействует».

6 Соответствие

Комплекс стандартов ИСО/МЭК 18384 содержит три части, требования к соответствию для которых различаются:

a) терминология и концепции — соответствие только терминам и точное следование семантике в определениях;

b) эталонная архитектура для решений SOA — соответствие только семантике метамодели и используемых уровней, СБА или возможностей;

c) онтология SOA — соответствие для приложений OWL или иных приложений (не-OWL).

Соответствие данной части ИСО/МЭК 18384 определено следующим образом.

Если некоторый документ, продукт или стандарт заявляет о соответствии, то термины из данной части ИСО/МЭК 18384 должны использоваться с той же семантикой, что и в настоящих определениях.

Принципы и концепции приведены для объяснения и не предназначены для исполнения.

7 Концепции SOA

7.1 Введение в SOA

Сервис-ориентированная архитектура (сокращенно SOA) (см. 2.48) — это архитектурный стиль, который поддерживает ориентированность на службы (см. 2.46) и является парадигмой для бизнеса и ИТ. Данный архитектурный стиль предназначен для разработки систем с точки зрения служб, доступных через интерфейс, и результатов действий этих служб. Как определено в 2.20, служба — это логическое представление ряда действий, который имеет определенные результаты, является автономным, может быть составлен из других служб и является «черным ящиком» для потребителей службы.

Аналогично другим архитектурным стилям SOA:

- предъявляет уникальные требования к инфраструктуре системы;
- имеет реализации, зависящие от окружения, ограниченные или разрешенные контекстом и описываемые в этом контексте;
- рекомендует руководство ИТ, системами и корпоративным лицензированием;
- имеет бизнес-решения, предназначенные для отражения реальной деятельности бизнеса;
- предоставляет критерии, позволяющие потребителям определить, было ли предлагаемое бизнес-решение правильно и полностью выполнено в соответствии с их ожиданиями.

Кроме того, SOA имеет ряд характеристик, которые выделяют его среди других архитектурных стилей, в первую очередь:

a) SOA способствует использованию открытых стандартов и интерфейсов для достижения функциональной совместимости и независимости от местоположения;

b) службы и процессы разработаны явным образом для функционирования как внутри организации, так и между организациями;

c) SOA требует четких описаний предлагаемой службы;

d) службы и процессы разработаны таким образом, чтобы отражать реальную деятельность бизнеса;

e) представление службы использует бизнес-описание для задания контекста (т. е. бизнес-процесс, цель, правило, политика, интерфейс службы и компонент службы);

f) SOA требует надлежащего руководства представлением и реализацией службы;

g) композиция служб используется как средство реализации бизнес-процессов;

h) SOA устанавливает критерии, позволяющие потребителям служб определять, была ли служба правильно и полностью выполнена в соответствии с описанием службы.

В рамках SOA «служба» является основным элементом для сборки и интеграции информационных систем, которые способны удовлетворять различным требованиям при решении той или иной задачи. Служба с точки зрения бизнеса — это представление результатов бизнес-процессов; служба с точки зрения ИТ — реализация этих бизнес-процессов средствами ИТ. Деятельности по разработке решения SOA могут быть внутренними для организации (например, развертывание службы), совместными между рядом предприятий (например, вызовы службы и хореографии службы) или совместными мероприятиями для поддержания жизнеспособности экосистемы службы (например, публикации новых служб).

Преимущества использования SOA заключаются в повышении эффективности разработки информационных систем, эффективности интеграции и повторного использования ресурсов.

Несмотря на то, что интерес к SOA и тем преимуществам, которые она предоставляет, не ослабевает, для глобального рынка пока не установлен единый набор технических принципов, определенных норм и стандартов SOA. Существующие продукты и решения используют различные стандарты, методы и технологии, что снижает эффективность SOA. Чтобы повысить степень стандартизации и потенциально качество решений, а также способствовать эффективному крупномасштабному внедрению SOA, необходимо установить единый набор терминов, принципов и концепций SOA.

Следует отметить, что концепции SOA, определенные в настоящем стандарте, применимы к разработке программного обеспечения и могут также применяться в системном проектировании для формализации систем, основанных на службах [например, сложных систем, федеративных систем, системы систем (system of systems), архитектур уровня предприятия].

7.2 Концепции

7.2.1 Роли

7.2.1.1 Обзор

В целом роль определяется рядом действий, которые служат общей цели. Когда сущности назначается роль или она принимает некоторую роль, она выполняет действия этой роли. Определение роли включает в себя указание, все ли действия являются обязательными и когда они выполняются. Поставщики службы и потребители службы являются фундаментальными ролями в сервис-ориентированной архитектуре. Экосистема SOA включает в себя службы, предоставляющие функциональность, потребителей службы, которые взаимодействуют со службами, и поставщиков службы, которые разрабатывают и размещают службы для потребителя. Однако роли поставщиков службы и потребителей службы охватывают целый спектр действий и могут выполняться различными сторонами, ответственными за различные действия, т. е. в дальнейших обсуждениях сторона, которая может нести ответственность за функционирование, может не нести контрактной ответственности, и наоборот. Кроме того, сторона, исполняющая роль поставщика службы в одном контексте, может исполнять роль потребителя службы в другом. Далее термин «поставщик службы» используется для указания на сущность, выполняющую действия, связанные с ролью поставщика службы; термин «потребитель службы» — для указания на сущность, выполняющую действия, связанные с ролью потребителя службы.

7.2.1.2 Поставщики службы

Для поставщиков службы действия включают в себя разработку, тестирование и развертывание служб, которые удовлетворяют набору функциональных и нефункциональных требований. Эти требования включают в себя эксплуатационные характеристики и договорные обязательства, а также поддержку и ответ на запросы других сущностей в решении SOA или экосистеме. Поставщик службы контролирует службу во время эксплуатации для оценки соответствия ее требованиям и выявления и минимизации нежелательного поведения. Поставщик службы несет ответственность за определение и осуществление управления службой. Кроме того, поставщик службы отвечает за регистрацию службы и поддержание описания службы.

Существуют два контекста, в которых поставщики службы могут нести ответственность, предоставляя услуги: эксплуатация и контракт. Причем сущность, которая является ответственной по контракту (участник контракта или соглашения об уровне обслуживания), может не быть той же сущностью, которая отвечает за эксплуатацию (т. е. обменивается сообщениями с потребителем службы).

Ответственность за эксплуатацию означает, что поставщик службы может участвовать в действиях, которые относятся к обмену сообщениями с потребителем службы, а также к созданию обещанного

эффекта вызова службы. Сущности, принимающие на себя ответственность за эксплуатацию, могут быть вовлечены в следующие действия жизненного цикла:

- разработку служб: процессы жизненного цикла для моделирования и создания реализаций служб, одним из которых является реализация службы (см. концепции жизненного цикла службы в 7.2.8);
- развертывание служб: размещение артефактов службы или изменение окружения служб таким образом, чтобы служба могла обмениваться сообщениями, приводящими к реализации функциональности службы и сопутствующим эффектам;
- предоставление служб: поддержка использования развернутой службы через обмен сообщениями и необходимая поддержка экосистемы SOA для реализации функциональности службы и эффектов службы;
- публикацию служб: создание и поддержка описания службы (включая, при необходимости, категоризацию службы) таким образом, что описание службы возможно обнаружить посредством поиска и получить его всеми участниками службы, включая поставщиков службы, потребителей службы и другие стороны;
- услуги хостинга: предоставление и поддержка использования инфраструктуры, необходимой для разработки, развертывания и предоставления службы;
- руководство службами: предписание условий и ограничений, согласованных с общими целями участников службы, а также структур и процессов, необходимых для определения действий, предпринятых для реализации этих целей, и реакции на эти действия.

Контрактная ответственность означает, что поставщик службы может участвовать в действиях, которые связаны с определением деловых аспектов и юридических обязательств, относящихся к использованию предоставленной службы, и координацией использования и связанных с этим соглашений с потребителем службы в соответствии с контрактом службы. Сущности, принимающие на себя контрактную ответственность, могут быть вовлечены в следующие действия жизненного цикла:

- ценовую политику службы: решение о ценообразовании служб или о целесообразности их использования за иную стоимость;
- определение соглашений о предоставлении услуг: совместно с потребителем службы принятие решений о том, какие формальные соглашения требуются, например такие как соглашение об уровне обслуживания (SLA), соглашение об использовании службы, включая разрешение конфликтов, в предпочтительных политиках службы;
- определение контрактов службы: разработка соглашений о предоставлении службы и применимых аспектов управления однозначным способом, который упрощает их исполнение;
- исполнение контрактов службы: участие в процессах, которые, вероятно, определены в контракте службы, с целью обеспечения выполнения контракта или предписанных действий по исправлению последствий нарушения контрактов;
- управление бизнесом и контрактами: установление бизнес-правил, политики и требований к контрактам, а также обеспечение и контроль оперативного управления службами.

7.2.1.3 Потребители службы

Потребители службы участвуют в действиях по идентификации служб, соответствующих их потребностям, оценивают возможности выполнения предписанных условий использования (например, права доступа, лицензирование и величина запросов), изучают информационный обмен и протоколы, которые обеспечивают связь со службой, и должны быть способны задействовать сетевые возможности для успешного осуществления коммуникации. Потребитель службы должен иметь возможность получить доступ к описанию службы и интерпретировать его как для первоначальной идентификации службы, так и для соответствия изменениям специфики взаимодействия со службы или условий использования.

Кроме того, поставщик службы может одновременно выступать как потребитель службы. В случае композиции служб поставщик службы отвечает на запрос внешнего потребителя службы, но затем становится потребителем службы, входящей в композицию, для создания документированных результатов, которые должны быть произведены композицией.

Существуют два контекста, в которых потребители службы могут нести ответственность при потреблении служб: эксплуатация и контракт. Как и в случае с поставщиками службы, сущность, которая является ответственной по контракту (участник контракта или соглашения об уровне обслуживания), и сущность, которая отвечает за эксплуатацию (т. е. обменивается сообщениями с поставщиком), могут быть разными сущностями.

Примечание — Действия потребителя службы, связанные как с эксплуатацией, так и с договорными обязательствами, часто напрямую дублируют действия поставщика.

Ответственность за эксплуатацию означает, что потребитель службы может участвовать в действиях, которые относятся к обнаружению служб и обмену сообщениями с поставщиком службы. Сущности, принимающие на себя ответственность за эксплуатацию, могут быть вовлечены в следующие действия жизненного цикла:

- обнаружение служб: поиск и анализ доступных описаний служб для идентификации служб, соответствующих потребностям (например, посредством взаимодействия с реестром/репозиторием служб или следуя рекомендациям консультантов);
- вызов службы: обмен сообщениями с предоставленной службой для вызова ее функциональности и реализации ее соответствующих эффектов;
- управление использованием службы: установка и осуществление бизнес-политики использования службы и заключения контрактов на предоставление услуг с целью удовлетворения потребностей бизнеса.

Контрактная ответственность означает, что потребитель службы может участвовать в действиях, связанных с определением деловых аспектов и юридических обязательств, относящихся к использованию предоставленной службы, и координированием связанных соглашений с поставщиком службы согласно положениям контракта службы. Сущности, принимающие на себя контрактную ответственность в жизненном цикле службы, могут быть вовлечены в следующие действия жизненного цикла:

- контракт: заключение и соблюдение контрактов;
- оплата службы: соблюдение соглашений о предоставлении службы, определяющих обмен финансовыми средствами или услугами в обмен на использование предоставленных служб;
- определение соглашений о предоставлении службы: совместно с поставщиком службы принятие решений о том, какие формальные соглашения требуются, такие как соглашение об уровне обслуживания (SLA), соглашение об использовании службы, включая разрешение конфликтов, в предпочтительных политиках службы;
- определение контрактов службы: разработка соглашений о предоставлении службы и применимых аспектов управления однозначным способом, который упрощает их исполнение;
- исполнение контрактов службы: участие в процессах, которые, вероятно, определены в контракте службы, с целью обеспечения выполнения контракта или предписанных действий по исправлению последствий нарушения контрактов;
- управление контрактами: обеспечение соответствия контрактов растущим потребностям бизнеса, а также их выполнение.

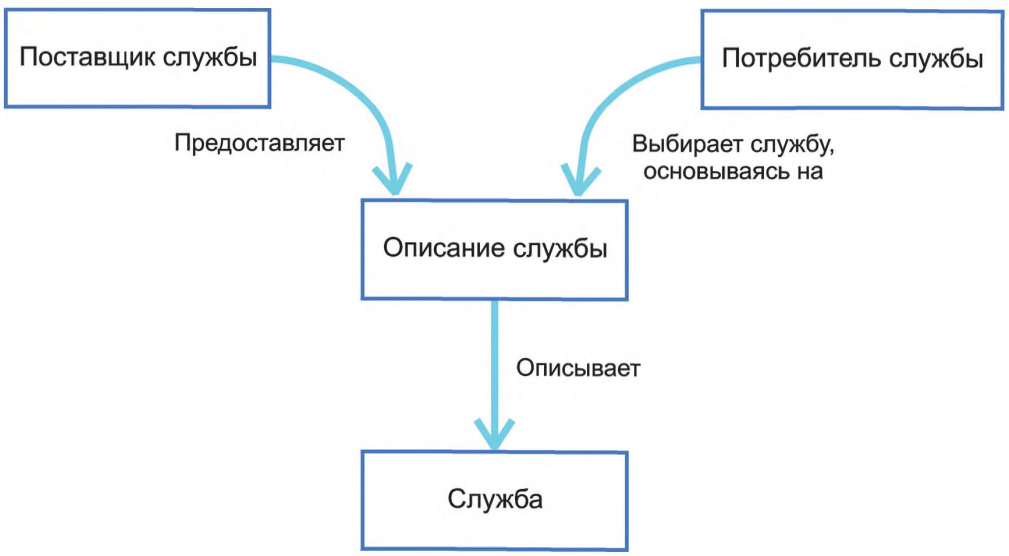

Рисунок 1 — Поставщики службы и потребители службы

7.2.2 Службы

Как определено в настоящем стандарте, служба (см. 2.20) является логическим представлением ряда действий, который имеет определенные результаты, является автономным, может быть составлен из других служб и является «черным ящиком» для потребителей данной службы (см. ИСО/МЭК 18384-3:2016, 7.4). Для службы не имеет значения, применяется ли данная концепция к сфере бизнеса или к сфере ИТ. Служба может иметь одного или нескольких поставщиков службы или потребителей службы и производить указанные результаты.

Потребитель службы может не знать, как реализована служба. Потребители используют описание для выбора службы у поставщика, как приведено на рисунке 1. Если две службы имеют одинаковое описание и производят одинаковые эффекты при одинаковых входных данных, они функционально эквивалентны для потребителя службы и могут быть взаимозаменяемыми. Для поставщика службы служба — это средство предоставления возможностей, и эквивалентность определяется реализацией.

Поскольку сама служба является лишь логическим представлением, любая служба должна иметь нечто, способное выполнять процессы и взаимодействовать, необходимое для порождения результирующих эффектов. Такое исполнение обозначается в тексте как «выполнение службы». Службы могут выполняться элементами любого типа, включая компоненты программного обеспечения, интерактивных агентов и задачи. То, как выполняется служба, скрыто от всех, кто с ней взаимодействует. Службы могут выполняться не системой, а элементами других типов, включая такие элементы, как компоненты программного обеспечения, интерактивные агенты и задачи.

Аналогичным образом служба может использоваться другими элементами, при этом сама служба (как чисто логическое представление) не использует другие элементы. Однако сущность, которая выполняет службу, может использовать другие элементы (и, безусловно, будет использовать в случае композиции службы).

Элемент, взаимодействующий со службой, может выполнять следующие шаги:

- выбрать службу, с которой будет взаимодействовать (данный пункт не определяет, осуществляется ли это динамически во время выполнения или статически при проектировании и/или во время сборки);
- выбрать элемент, который выполняет эту службу [в типичном окружение SOA этот выбор, чаще всего, делается «внутри» шины служб (см. 2.22)];
- взаимодействовать с выбранным элементом, который выполняет выбранную службу (зачастую тоже с помощью шины служб).

Такие понятия, как «промежуточные службы», «прокси-службы», «шины служб» и т. д. используются для описания и реализации аспектов функционирования систем SOA. Все они могут быть зафиксированы как элемент, представляющий сервис. Такое представление обеспечивает уровень абстракции, который критически важен, когда не хотят функционально привязываться к определенной оконечной точке службы, что позволяет сохранять слабое связывание и возможность переключать привязки при необходимости. Понимание того, что именно предоставляет служба и чем она представлена, позволяет проводить инкапсуляцию довольно сложного функционального взаимодействия (выбор службы, выбор элемента, который выполняет службу, и взаимодействие с этим выбранным элементом).

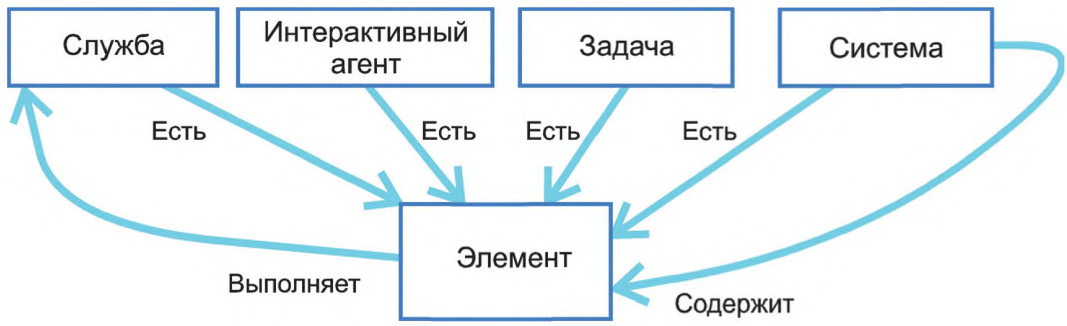

Рисунок 2 — Служба и элементы SOA

Как показано на рисунке 2, элементы SOA могут быть службами, интерактивными агентами, задачами и системами. Службы выполняются любым из этих элементов. Системы состоят из частей, которые могут быть элементами любого вида, включая службы, интерактивные агенты, задачи и сами системы.

Более детальное объяснение задач и систем приведено в следующих разделах (см. подробнее о системах и SOA-системах в ИСО/МЭК 18384-3:2016, раздел 5).

7.2.3 Семантика

Все агенты, включенные в сервис-ориентированное архитектурное решение с участием человека и без его участия, должны уметь правильно анализировать сообщения посредством общего и согласованного понимания синтаксиса и семантики сообщения. Синтаксис включает в себя интерфейсы, протоколы и форматы сообщения. Семантика — это общее понимание намерения, когда агенты должны уметь обрабатывать содержание этих сообщений таким образом, который соответствует намерению отправителя. Другими словами, агенты должны обладать общим семантическим пониманием бизнес-требований, связанных со службой, а не просто стремиться к функциональной совместимости через синтаксическую связь.

Кроме того, семантика важна при описании служб и других ресурсов в решении SOA. Обнаружение служб требует совместного понимания базиса для установления соответствия между потребностями потребителя службы и наблюдаемыми эффектами, возникающими в результате взаимодействия со службой. Это также необходимо для однозначного понимания условий использования, которые должны выполняться для продолжения взаимодействия со службой.

Семантика может быть формально определена посредством использования онтологий, которые обращаются к сообществу понятий, и их приложения к определенным сферам применений.

7.2.4 Задачи и действия

Задача — это атомарное действие, которое достигает определенного результата. Задачи, включая интерактивные задачи, выполняются людьми или организациями, в частности экземплярами интерактивных агентов. Поскольку задачи являются атомарными, они не могут быть приведены к меньшему уровню детализации и выполняются не более чем одним экземпляром интерактивного агента.

Термин «действие» в определении «службы» используется в общем значении в обычном языке, а не в специальном значении для процессов (т. е. действия необязательно соответствуют действиям процесса).

В частности, обозначение моделирования бизнес-процессов (BPMN) 2.0 (см. [14]) определяет задачу следующим образом: «Задача — это атомарное действие в потоке управления. Задача используется, когда работа в процессе не может быть разбита на меньшие уровни детализации. Как правило, для исполнения задачи во время выполнения используются конечный пользователь и/или приложения». Таким образом, понятия «делать» и «выполнять» формально разделяются. Задачи (необязательно) делаются интерактивными агентами, кроме того (как экземпляры элементов), задачи могут использовать службы, которые выполняются технологическими компонентами. Для SOA задачи не ограничиваются потоком управления.

7.2.5 Композиции и процессы

7.2.5.1 Введение в композицию

Композиция — это система, которая является результатом сборки набора сущностей для определенной цели. В этом случае термин «композиция» обозначает набор частей, собранных для достижения цели, а не действие по созданию композиции. Композиции организованы по одному из шаблонов или архитектурных стилей: оркестровка, хореография или коллаборация.

Так же как системы могут быть составлены из других систем, композиции могут быть составлены из других композиций. Никакая композиция не может быть частью себя. Так как композиция является набором элементов, она использует не менее чем один элемент, причем эти элементы могут быть вне ее собственных границ. В SOA такие элементы обычно являются службами, другими композициями, процессами, агентами и задачами.

Композиции, подобно службам, не видны внешним наблюдателям. Однако шаблоны композиции дают представление о внутреннем устройстве композиции и описывают, как создается или используется набор элементов для достижения результата.

7.2.5.2 Образцы композиции

Как показано на рисунке 3, композиции могут быть реализованы с использованием трех общих шаблонов или стилей, которые можно отличить по наличию управляющего элемента и предопределенного шаблона поведения или потока. Стрелки, представленные на рисунке 3, показывают направление потока или вызовов между элементами композиции.

Оркестровка — шаблон, в котором один элемент из композиции наблюдает и направляет другие элементы. Управляющий элемент отличается от оркестровки и может быть внутри или снаружи границ. Управляющий элемент, показанный на рисунке 3, вызывает элементы оркестровки.

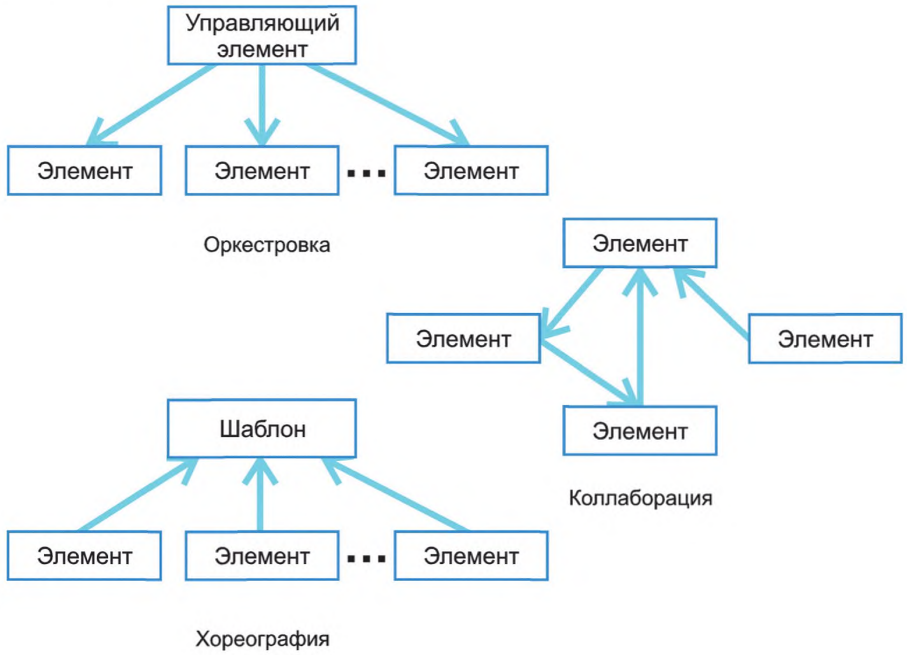

Рисунок 3 — Шаблоны композиции

Хореография — шаблон композиции, в котором элементы композиции взаимодействуют ненаправленным способом (без управляющего элемента), и каждая часть знает шаблон поведения и следует ему. Существует предопределенный, совместно используемый шаблон поведения или потока. Хореография не требует от элементов полного знания шаблона поведения. Элементы взаимодействуют друг с другом в соответствии с шаблоном (см. рисунок 3).

Коллаборация — шаблон композиции, в котором элементы композиции взаимодействуют ненаправленным способом (без управляющего элемента), и каждая часть действует в соответствии со своим собственным планом/целью без предопределенного шаблона поведения или потока. Взаимодействие между частями происходит таким образом, как это необходимо для каждой части. Элементы коллаборации взаимодействуют друг с другом, при этом нет какого-либо шаблона или управляющего элемента, как приведено на рисунке 3.

На рисунке 4 показана связь понятий службы, системы и композиции. Элементы систем — это службы, интерактивные агенты, задачи и другие системы. Любой из этих элементов может выполнять службу. У системы может быть любой из этих элементов в качестве части системы. Композиции являются системами и поэтому могут иметь части, которые являются службами, агентами, задачами и системами. Композиции обладают свойством, которое указывает на ее шаблон: оркестровку, хореографию или коллаборацию. Только композиции, использующие шаблон оркестровки, имеют элемент, который управляет оркестровкой частей композиции. Композиции служб и процессов являются композициями; таким образом, они также могут демонстрировать любой из шаблонов композиции.

7.2.5.3 Композиции службы

Композиции службы — это композиции, которые предоставляют (в функциональном смысле) высокоуровневые службы, составленные из других служб. Композиции служб могут демонстрировать один, два или все три шаблона композиции: оркестровку, хореографию и коллаборацию.

7.2.5.4 Процессы

Процессы — это композиции, элементы которых составлены в последовательность или поток действий и взаимодействий с целью выполнения определенной работы. Процессы могут быть составлены из таких элементов, как интерактивные агенты, задачи, службы и процессы (см. рисунок 4). Кроме того, процессы могут представлять любой из шаблонов композиции.

Процесс всегда добавляет логику через шаблон композиции; результат — больше, чем его части. В соответствии с шаблоном композиции могут быть определены следующие процессы:

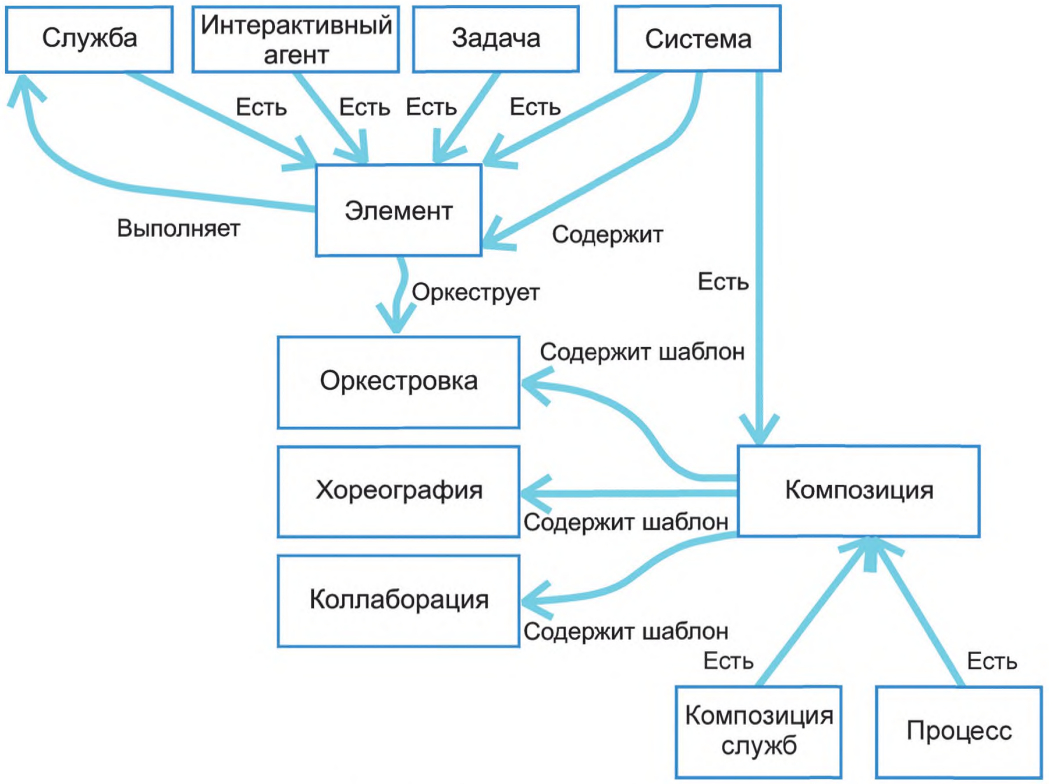

Рисунок 4 — Композиция и ее подклассы

- **оркестрованный:** когда процесс оркестрован в системе управления бизнес-процессами, то получившийся ИТ-артефакт фактически является оркестровкой, т. е. имеет шаблон композиции оркестровки. Этот тип процессов часто называют оркестровкой процесса;

- **хореографированный:** например, модель процесса, представляющая определенный шаблон поведения. Этот тип процесса часто называют хореографией процесса;

- **совместный:** не существует заранее предопределенного шаблона поведения (модели); процесс представляет наблюдаемое (выполняемое) поведение (см. ИСО/МЭК 18384-3:2016, раздел 8).

7.2.5.5 Бизнес-процесс

Бизнес-процесс — это определенный набор операций, которые представляют шаги, требуемые для достижения бизнес-цели. Бизнес-процесс включает в себя поток информации и ресурсов и их использование. Бизнес-процесс может потребовать следующие виды поддержки:

- управление бизнес-процессом — действия, включающие обнаружение бизнес-процессов, их моделирование, выполнение, изменение, управление и обеспечение сквозной видимости бизнес-процессов (см. [14]);

- мониторинг деловой активности — действия, которые осуществляют мониторинг бизнес-операций и процессов, а также связанных с ними SLA и ключевых показателей эффективности (КПЭ).

7.2.6 Регистрация и обнаружение службы

Регистрация службы — это процесс, посредством которого описания услуг становятся доступными потенциальным потребителям службы. Регистрация может осуществляться посредством различных механизмов, включая добавление описания службы к набору описаний, хранение его в реестре/репозитории служб, который обеспечивает возможности управления и поиска или предоставляет доступ к описанию через иной механизм обнаружения службы. Для небольшого набора служб может быть достаточно простой публикации документов, однако реестр/репозиторий служб может ускорить поиск в случае большого числа служб, поскольку это позволяет создавать и сохранять большую, поддерживающую поиск базу данных служб.

При регистрации службы может потребоваться возможность добавлять, удалять, изменять, классифицировать и проверять описания службы, поэтому система регистрации может сопровождаться

поставщиками службы или другими назначенными лицами, отвечающими за сопровождение описания службы.

Термины «реестр», «репозиторий» и «каталог» рассматриваются как синонимы. Единый термин «реестр/репозиторий» используется для указания на такой организованный набор описаний служб и других артефактов. Поддержка регистрации и обнаружения фокусируется на описании службы.

Обнаружение службы — это процесс, в котором потребители службы предоставляют критерии поиска, часто в терминах описания службы, чтобы обнаружить службы в соответствии со своими определенными функциональными или нефункциональными требованиями. Обнаружение службы выполняется как во время проектирования, так и во время выполнения. Во время процесса обнаружения потребитель службы может потребовать предоставить ему возможность запросить и получить описание службы.

Как правило, потребитель службы может использовать реестр/репозиторий служб для обнаружения службы на этапе проектирования или в других случаях, когда потребителю нужно сослаться на информацию, содержащуюся в описании службы, или получать такую информацию, либо необходима ссылка или получение информации об артефакте службы, например нахождение оконечной точки службы для поддержки позднего связывания.

На рисунке 5 показаны шаги, иллюстрирующие использование реестра/репозитория служб потребителями.

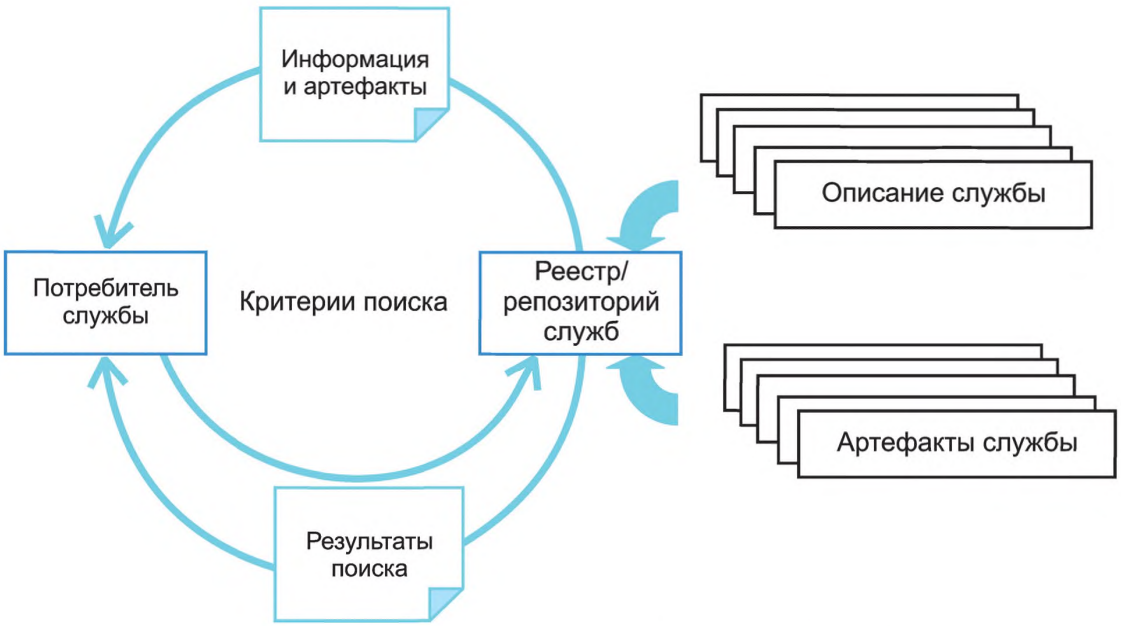

Рисунок 5 — Модель использования реестра/репозитория служб

Как показано в примере, приведенном на рисунке 5, потребитель службы начинает с подачи запроса на поиск в реестр/репозиторий служб; на этот запрос реестр/репозиторий служб отвечает результатами поиска. Результаты поиска содержат информацию о содержимом реестра/репозитория служб, которое соответствует критериям поиска, и, возможно, ссылки или другие механизмы, например служебные вызовы, которые позволяют потребителю службы получить информацию или артефакты от реестра/репозитория служб. В данном примере показано, как потребитель службы использует информацию об артефактах и сами артефакты.

Как показано на рисунке 5, реестры/репозитории не ограничиваются только хранением описаний службы, если они поддерживают другие возможности в решении SOA, такие как информационная архитектура, управление и руководство. С точки зрения архитектуры их можно считать различными видами реестров/репозиториев в решении SOA, но при реализации эти реестры/репозитории могут располагаться в одном и том же реестре/репозитории и пользоваться общими аспектами описания и обнаружения, хранения и извлечения. Реестры/репозитории, поддерживающие руководство, могут предоставлять доступ к политикам руководства и документации процессов коммуникации и соблюдения

политик; описания служб могут ссылаться на политики и процессы, имеющие отношение к конкретной службе.

Реестры/репозитории служб находятся под руководством и часто используются в процессах руководства. Например, процесс руководства для максимизации повторного использования служб может потребовать от архитекторов и разработчиков поиска существующих подходящих служб перед тем, как разрабатывать новые. В этом случае эффективное повторное использование требует четкого описания служб и легкого доступа к ним для разработчиков.

Рекомендуется управлять реестрами/репозиториями и процессами регистрации, что может включать в себя мониторинг, управление хранением, резервное копирование, восстановление и уведомления об исключениях и отказах.

7.2.7 Описание, интерфейсы, политики и контракты службы

7.2.7.1 Описание службы

Описание службы содержит информацию, необходимую для взаимодействия со службой, и может описываться в таких терминах, как интерфейс службы (входные и выходные данные службы и связанная с этим семантика) и политики службы (условия использования службы). Контракты службы могут ссылаться на информацию в описании службы.

Цель объединения интерфейсов службы и политик службы в составном описании службы заключается в том, чтобы упростить взаимодействие и обеспечить видимость, особенно когда участники находятся в различных доменах владения (см. [17] о домене владения). Явные описания службы содержат необходимую информацию о том, что делает служба, условия ее использования, сценарий взаимодействия с пользователем, детали информационного обмена и адрес точки, в которой происходит информационный обмен. Это позволяет потенциальным участникам создавать системы согласно описаниям, а не реализациям.

Описание службы позволяет потенциальным потребителям службы оценить, подходит ли служба для их текущих потребностей, и позволяет поставщику службы установить, удовлетворяет ли потребитель службы требованиям поставщика.

7.2.7.2 Интерфейсы службы

Описания службы могут содержать интерфейсы службы, условия контракта и политики. Поэтому они должны быть определены до определения описания службы. Интерфейсы службы определяют способ взаимодействия и обмена информацией других элементов со службой при обработке запроса к службе (см. ИСО/МЭК 18384-3:2016, 7.13). Интерфейсы включают определение параметров, используемых для передачи информации при вызове службы; это обычно документируется в описании службы.

Характер вызова интерфейса и передачи информации к службе и от нее зависит от области приложения. Интерфейсы службы обычно (но необязательно) основаны на сообщениях (для поддержания слабого связывания). Кроме того, определение интерфейсов службы всегда происходит независимо от любой конкретной реализации службы (чтобы поддерживать слабое связывание и маршрутизацию запросов к службе).

Каждая служба имеет минимум один интерфейс службы. В интерфейсе службы могут накладываться ограничения на разрешенное взаимодействие, такие как ограничения на допустимый диапазон значений определенных параметров. Практически в зависимости от характера службы и рассматриваемого интерфейса службы эти ограничения могут быть определены формально (документированные) или неформально (недокументированные). Ограничения должны быть документированы и формально определены.

Один интерфейс службы может использоваться как интерфейс для многих служб. Это не означает, что данные службы будут одинаковыми и то, что они будут производить одинаковый эффект; это лишь означает возможность взаимодействия со всеми этими службами способом, определенным данным

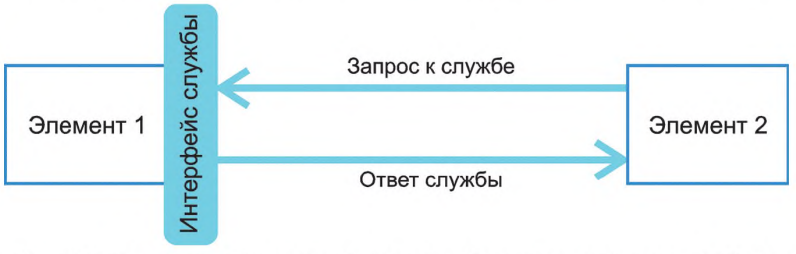

Рисунок 6 — Поставщик службы и потребитель службы, использующие интерфейс службы

интерфейсом службы. Кроме того, требуется, чтобы все службы, использующие интерфейс, соблюдали семантику этого интерфейса.

На рисунке 6 показано, как поставщики службы и потребители службы взаимодействуют через элементы, обменивающиеся данными через интерфейсы службы. Элемент 2 потребителя службы отправляет запрос службы к интерфейсу службы элемента 1 поставщика службы. Элемент 1 выполняет соответствующие функции и отправляет ответ службы (соответствующий ответу, определенному в интерфейсе службы) обратно к элементу 2.

7.2.7.3 Политика

Политика представляет собой ряд юридических, политических, организационных, функциональных или технических утверждений о намерениях или обязательствах о взаимодействии и сотрудничестве. Как показано на рисунке 7, политика внедряется и утверждается интерактивным агентом; интерактивный агент берет на себя обязательство поддерживать и, при желании или необходимости, осуществлять выбранную политику. Интерактивный агент утверждает политику, однако политика может накладывать ограничения на любого агента, интерактивного или нет.

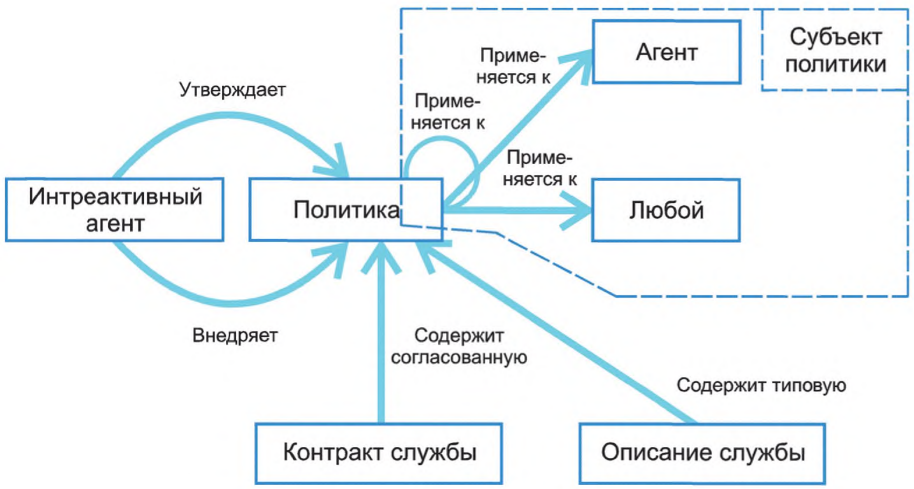

Рисунок 7 — Политика и подмножества политики

Политика является общей концепцией и имеет значение за рамками сферы SOA. Политики могут применяться не только к элементам (субъектам политики); фактически политики могут применяться к чему угодно, включая другие политики.

Политика — это утверждение (заявление о намерениях), которое может быть сформулировано и затем применено к любому количеству субъектов политики. Например, политика может требовать, чтобы все обмены данными со службой SOA были зашифрованы; это единственное требование будет применяться (и можно оценить соответствие) к каждому обмену данными с каждой службой. Агент, утверждающий политику, несет ответственность за ее исполнение посредством прямого действия или делегирования ответственности.

Примечание — Применение политики к различным субъектам не означает, что эти субъекты идентичны, а означает лишь то, что они (частично) регулируются общим намерением. И наоборот, к любому субъекту может быть применена одна или несколько политик. Следует обратить внимание на то, что к одному и тому же субъекту может быть применено несколько политик, что происходит зачастую потому, что различные политики исходят от различных доменов (таких как безопасность и руководство).

Знание политик, которые могут применяться к субъекту, упрощает и делает более прозрачным взаимодействие с таким субъектом. Политики, определяющие их интерактивные агенты и субъекты, к которым они применяются, являются важными аспектами любой системы, особенно систем SOA с различными взаимодействующими элементами.

С точки зрения проектирования политики могут иметь более детализированные части или могут быть выражены и введены в действие с помощью специальных правил.

Политики отличаются от контрактов службы (более подробно — см. 7.2.7.4). В то время как политика утверждается одним интерактивным агентом, контракт является соглашением, заключенным двумя или более интерактивными агентами (договаривающимися сторонами) по совокупности условий вместе

с рядом ограничений, которые регулируют их поведение, и/или состоянием в части соблюдения этих условий. Эти условия и ограничения служат основанием для оценки того, соблюдают ли стороны условия контракта.

Политики могут быть утверждены поставщиком службы как условия использования службы по умолчанию. Как показано на рисунке 7, наиболее вероятно, что такие политики фиксируются в описании, однако чаще может быть указана ссылка на внешнюю формулировку политики. Потребитель службы также может задать политики; разрешению конфликтов среди заданных политик служит контракт службы. Политика по умолчанию может быть задана в описании службы либо на нее может быть дана ссылка, т. к. такая политика является ограничением, заданным единственной стороной — поставщиком службы, и применяется к определенному предмету политики — описываемой службе. Контракт службы, с другой стороны, может быть определенным для конкретных поставщика службы и потребителя службы; контракт не может быть применим к службе в общем случае и может являться внутренним документом потребителя и поставщика службы. Таким образом, контракты службы не будут включены в описание службы.

Политики могут применяться к контрактам службы. Например, политики безопасности могут определять, кто имеет право вносить изменения в определенный контракт службы. Также контракты могут ссылаться на политики как часть положений, условий и правил взаимодействия, с которыми согласны участники взаимодействия. Тем не менее, контракты не являются политиками, так как политика описывает намерение минимум одной стороны, а контракт — это соглашение между сторонами.

7.2.7.4 Контракты службы

Контракт службы — набор положений, условий и правил взаимодействия, по которым взаимодействующие участники пришли к соглашению (прямо или косвенно). Он является обязательным для всех участников взаимодействия: логических и физических представлений потребителей службы, поставщиков службы и самой службы. Контракт службы может включать в себя отдельные соглашения, призванные регулировать использование службы или обеспечивать определенную последовательность во взаимодействиях службы.

Все участники, взаимодействующие со службой, должны соблюдать аспекты взаимодействия всех контрактов службы, которые применяются к этому взаимодействию. Эксплуатационный аспект взаимодействия в контракте службы отличается от понятия интерфейса службы, поскольку контракт службы не определяет сами интерфейсы службы, а скорее определяет ограничения на взаимодействие и/или последовательность использования службы через набор интерфейсов. В качестве примера может быть приведена платежная служба, которая может иметь контракт службы, утверждающий, что платеж должен быть создан до того, как его можно будет прослеживать.

Контракты службы явным образом предписывают как эксплуатационные аспекты взаимодействия, например соглашения об уровне обслуживания (SLA), так и юридические аспекты соглашения об использовании служб. В зависимости от потребностей приложения аспекты взаимодействия и юридические аспекты можно разделить на два различных контракта службы. Юридические соглашения могут применяться к определенным интерактивным агентам и к тому, как они используют службы. Фактические юридические обязательства по каждому из таких интерактивных агентов могут быть определены в контракте службы, включая, например, положения о том, кто является поставщиком и потребителем с точки зрения юридических обязательств.

Отдельный интерактивный агент может участвовать в нескольких контрактах службы. Аналогичным образом определенный контракт службы может охватывать несколько интерактивных агентов. Следует иметь в виду, что могут существовать контракты на создание службы, где имеется юридическое соглашение между интерактивным агентом А и интерактивным агентом В (оба участвуют в контракте службы), но интерактивный агент В передает контракт на выполнение службы интерактивному агенту С (другими словами, именно интерактивный агент С выполняет рассматриваемую службу, а не интерактивный агент В).

Примечание — Контракты службы (в их юридической части) могут выражать временные аспекты, такие как «обязан в данный момент», «был обязан» и «в будущем может быть обязан».

Как приведено на рисунке 8, важно различать описание службы и контракт службы. Описание службы фиксирует факты службы, которые не зависят от конкретных механизмов взаимодействия; контракт службы представляет конкретные условия для определенного набора поставщиков службы, потребителей службы и других заинтересованных третьих сторон, таких как регулирующие органы. Несмотря на то, что факты, описывающие службу, могут распределяться среди отдельных артефактов, совокупный набор фактов не должен составлять логически противоречивое описание. Наоборот, контракты службы адаптированы к участникам взаимодействия службы, поэтому со службой может быть связано любое число контрактов службы. Описание службы может предоставлять условия по умолчанию для контракта

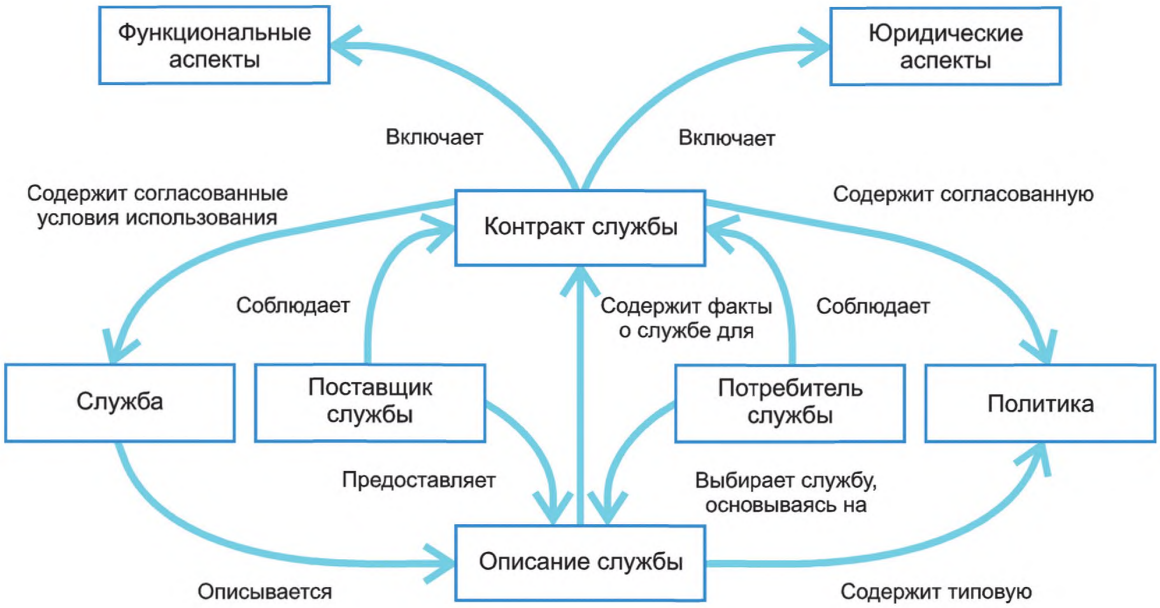

Рисунок 8 — Отношения контракта службы

службы, такие как производительность по умолчанию, тогда как контракт службы может определять более или менее жесткие целевые показатели производительности.

7.2.8 Жизненный цикл службы и решения SOA

7.2.8.1 Жизненный цикл службы

Жизненный цикл службы состоит из действий, ролей и продуктов работы самой службы. Это зачастую является расширением принятого в организации жизненного цикла разработки программного обеспечения. Напротив, жизненный цикл решения SOA относится к значительно широкой области, т. е. ко всему сервис-ориентированному решению, которое создает рыночную стоимость для организации. То, какие службы будут разработаны, определяется портфелем решений SOA во время жизненного цикла решения SOA.

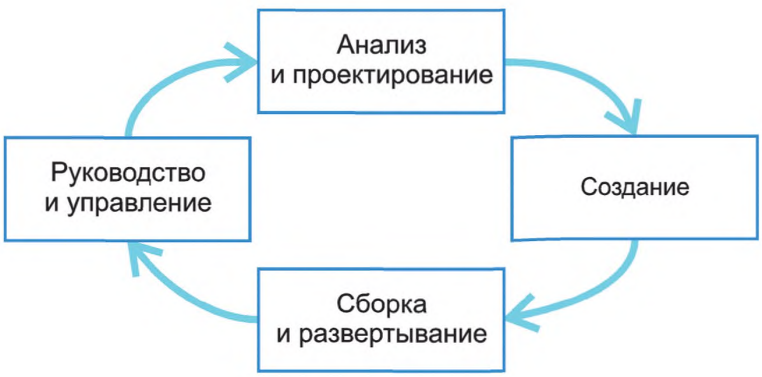

Рисунок 9 — Пример жизненного цикла службы

Как показано на рисунке 9 (движение цикла по часовой стрелке), жизненный цикл службы может включать в себя:

а) анализ и проектирование, которые могут включать в себя управление требованиями, определение службы, планирование реализации службы. Жизненный цикл может начинаться с анализа для опре-

деления того, какие службы должны быть в портфеле служб, и управления рыночной стоимостью. Такой анализ известен также как идентификация служб;

b) создание, которое может включать в себя моделирование службы, реализацию службы и ее тестирование. Сюда входят проектирование реализации службы и разработка кода реализации других компонентов поддержки службы;

c) сборку и развертывание, которые могут включать сборку или приобретение службы, развертывание службы, конфигурацию службы и публикацию службы. Эта фаза включает сборку компонентов, служб и артефактов для перехода к последующему развертыванию. После разработки всех этих элементов может проводиться развертывание службы в операционных системах поставщика службы. В этот момент может быть создан экземпляр службы, чтобы был готов экземпляр службы для взаимодействия с потребителем службы;

d) руководство и управление, которые могут включать управление службой и мониторинг, поддержку службы, снятие службы с эксплуатации, управление политикой.

При передаче службы в эксплуатацию необходимо отслеживать ключевые бизнес-метрики, метрики производительности и доступность. Результаты мониторинга используются на рабочих местах операторов, при осуществлении SLA, тарификации, в процессах управления и т. д. Во время эксплуатации служба может быть сконфигурирована или реконфигурирована, отремонтирована в случае ее неправильной работы или улучшена при изменении бизнес требований.

Процессами жизненного цикла службы необходимо управлять и руководить, в особенности процессами руководства службой, управления политикой, управления требованиями и конфигурацией. Для управления и руководства обычно используются активы и реализации реестра/репозитория служб, так как они обеспечивают доступ к той части портфеля активов, которая необходима для поддержания жизненного цикла и его управления. Данные активы включают в себя реализации службы, процессы, документы и т. д.

7.2.8.2 Жизненный цикл решения SOA

Жизненный цикл решения описывает связь действий, ролей и продуктов работы с предоставлением бизнес-решений на основе SOA.

Жизненный цикл решения SOA может начинаться с моделирования бизнеса (проведения бизнес-проектирования), включая ключевые показатели эффективности для бизнес-целей и задач, сборку и преобразование этой модели в проект информационной системы, развертывание информационной системы, управление этим развертыванием и использование результатов, выходящих из этой среды для определения способов усовершенствования данного бизнес-проекта. Предпосылка жизненного цикла состоит в том, что обратная связь циклически повторяется от фазы к фазе в итеративном процессе уточнения для облегчения выполнения потребностей бизнеса.

При таком подходе жизненный цикл можно разделить на фазы на фоне ряда процессов руководства SOA, которые гарантируют, что решение соблюдает политики соответствия и эксплуатации, а также что изменения происходят управляемым образом, с соответствующими полномочиями, как это предусмотрено бизнес-проектом.

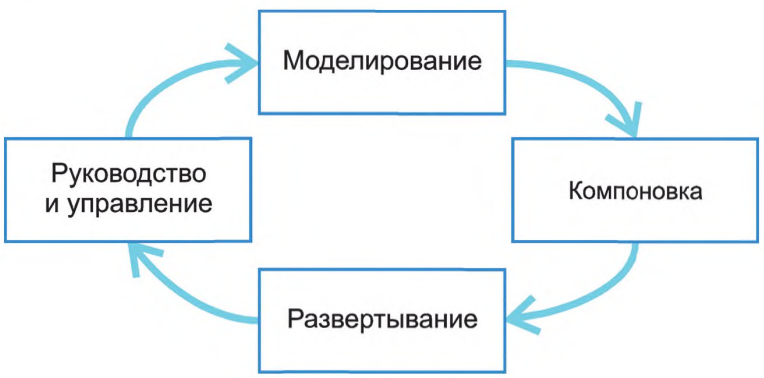

Рисунок 10 — Жизненный цикл решения SOA

Как указано на рисунке 10 (движение цикла по часовой стрелке), можно указать четыре фазы жизненного цикла решения SOA:

- **Моделирование** — это процесс получения бизнес-проекта решения SOA, начинающийся с понимания деловых требований и целей, включающий их преобразования в спецификацию бизнес-процессов, целей и предположений, создание закодированной модели бизнеса, вместе с ключевыми показателями эффективности.

- **Компоновка.** В ходе компоновки проводится сборка артефактов, которые реализуют бизнес-проект решения SOA. Бизнес-проект может быть преобразован в ряд определений и активностей бизнес-процесса, тем самым приводя к созданию необходимых служб из определений активностей.

- **Развертывание.** Эта фаза включает в себя сочетание создания окружения хостинга для решений SOA и фактического развертывания этих решений. Сюда входит определение потребности решения SOA в ресурсах, условий эксплуатации, резервирования мощностей и ограничений целостности и доступа. Кроме того, для решения SOA нужно рассматривать методы обеспечения доступности, надежности, целостности, эффективности и удобства использования службы.

- **Руководство и управление.** Эта фаза определяет то, каким образом поддержать окружение выполнения и политики, выраженные в решениях SOA, развернутых в этом окружении. Сюда относится мониторинг производительности обработки запросов на обслуживание и своевременность ответов

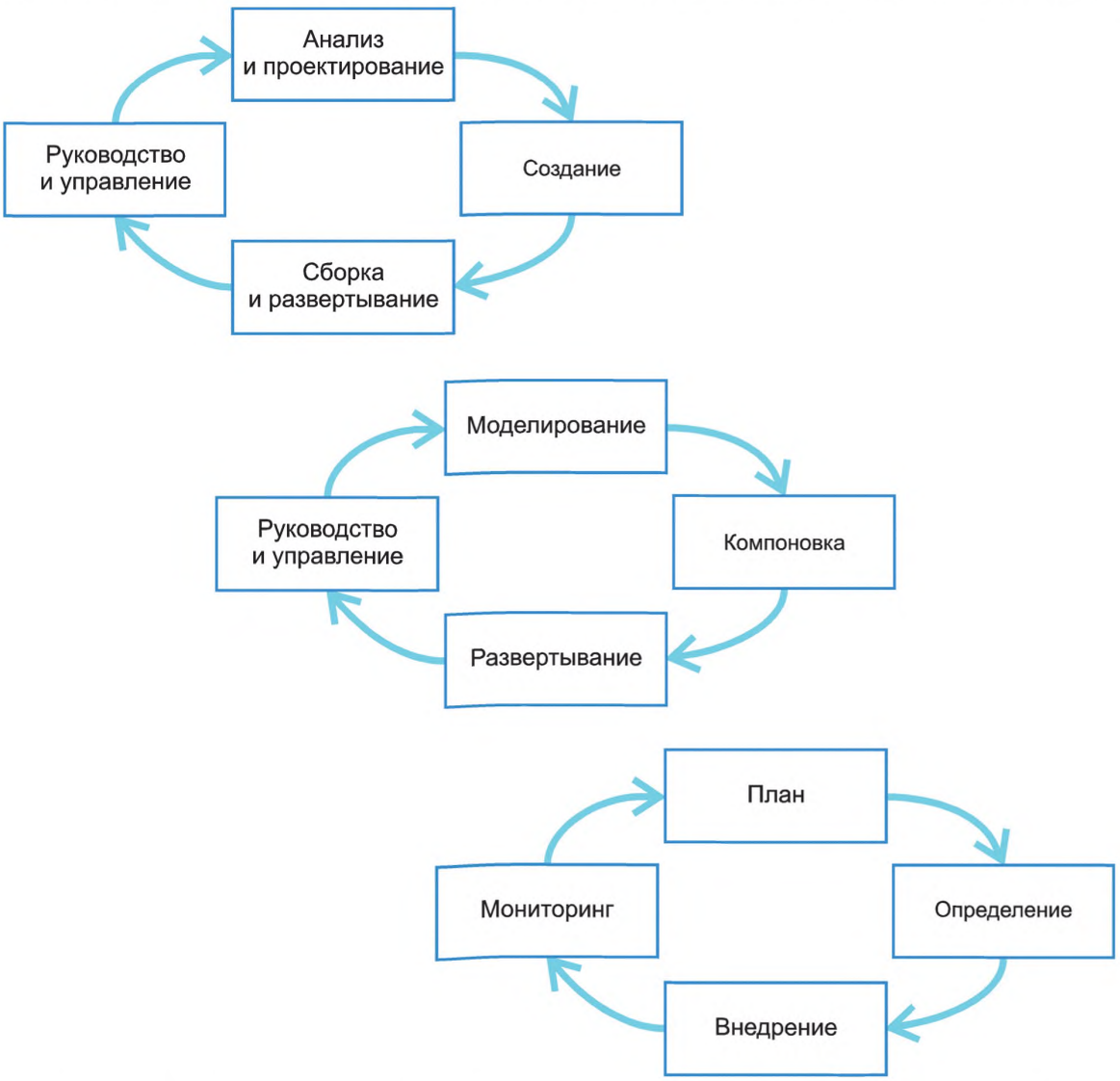

Рисунок 11 — Отношение между жизненными циклами службы, решения SOA и управления решением SOA

службы, ведение журнала отказов для обнаружения отказов в различных системных компонентах, обнаружение и локализация отказов, выстраивание маршрутов обходов отказов, восстановление работы, на которую повлияли отказы, устранение проблем и восстановление рабочего состояния системы.

Продвижение по жизненному циклу не является полностью линейным. Например, изменения ключевых показателей производительности фазы моделирования зачастую должны задаваться непосредственно на фазе управления для обновления окружения эксплуатации. Ограничения фазы развертывания, такие как предположения о расположении ресурсов в системе, могут обусловить некоторые решения фазы компоновки. Ограничения информационных технологий, установленные на фазе сборки, могут ограничить бизнес-проект, созданный на фазе моделирования.

7.2.8.3 Отношение жизненных циклов служб и решения SOA

Идеи, заложенные в службы SOA и решения SOA, тесно взаимосвязаны и дополняют друг друга. Как правило, службы SOA рассматриваются как стандартные блоки решений SOA, но и решения предлагают службы и объединяются в другие решения. Очевидно, что эти два жизненных цикла тесно связаны; жизненный цикл решений SOA служит фоном для жизненного цикла служб SOA. Фазы этих жизненных циклов аналогичны общим фазам жизненных циклов для программного обеспечения, т. е.: анализ требований, проектирование, разработка, развертывание, эксплуатация и снятие с эксплуатации. Однако цели каждой из этих фаз сильно различаются. Жизненный цикл службы фокусируется на самой службе, в то время как решение охватывает деловые аспекты и взаимосвязи между службами, операционными системами и аспектами поддержки. Подобно тому как решения служат фоном для жизненного цикла служб, метод управления жизнеспособностью является фоном для жизненных циклов решений (см. рисунок 11).

Как в любом жизненном цикле, должны быть определены роли, обязанности и результаты. Использование принципов и концепций SOA подразумевает:

- постоянное соотношение между ролями поставщика службы и потребителя службы;
- точные и обновляемые описания службы и решения;
- достаточные метрики для четкого понимания производительности и установка показателей роста и развития использования;
- моделирование экосистемы SOA для прогнозирования и ответа на вызовы роста и развития.

В рамках жизненного цикла следует принимать во внимание, что службы и решения существуют в экосистеме SOA и могут одновременно поддерживать потребности многих потребителей службы. Таким образом, несмотря на то, что вопросы производительности важны и их требуется учитывать, при оптимизации в целом для достижения компромисса может потребоваться, чтобы отдельные службы или решения выполнялись неоптимальным образом. Понимание локальной оптимизации является важным, но в подходе SOA возникают более сложные компромиссы, повышающие важность моделирования.

7.2.9 Слабое связывание

В общем случае слабое связывание достигается уменьшением зависимостей между элементами в системе.

Слабое связывание служб улучшается в случае, если использование службы изолировано от нижележащей реализации. Как правило, это достигается посредством использования интерфейсов службы, политики доступа и обнаружения службы. Кроме того, слабое связывание интерфейса службы может быть в целом улучшено путем обмена полуструктурированных документов. Внесение изменений в рамках такого подхода к информационному обмену оказывается проще, чем в случае информационных обменов посредством удаленного вызова процедур.

Еще большая изоляция достигается, когда потребитель службы полностью полагается на стандартизованные описания службы, которые служат «контрактом» между потребителем и поставщиком. Подобные способы изоляции помогают устранить зависимость от языков программирования, платформ или любых зависящих от поставщика условий, таких как упорядочение во времени, привязка по адресу, язык доступа или протокол доступа.

Архитектурный стиль SOA поддерживает слабое связывание интерфейса используемой службы с предоставляемой реализацией. Обмен сообщениями должен быть четко определенным и при этом обеспечивать принципиальные механизмы для поддержки вариаций во входных данных и расширяемости.

В сочетании с сервис-ориентированностью слабое связывание обеспечивает модифицируемость и гибкость, повторное использование и устойчивость.

7.3 Сквозные аспекты

7.3.1 Определение сквозных аспектов

Сквозные аспекты — это возможности и функциональность, которые справедливы для более чем одной роли или функционального уровня. Например, управление — это сквозной аспект, поскольку оно применимо ко многим областям, включая операционные системы, службы, бизнес-процессы и потребителей службы. Все эти области требуют управления, однако в зависимости от цели управления способы управления различаются. Управление инфраструктурой, например хранилищем и базами данных, сильно отличается от управления бизнес-процессами. Сквозные аспекты могут применяться к другим сквозным аспектам; таким образом, руководство применяется к функциональным уровням, а также к интеграции, информации и управлению.

Ключевые сквозные аспекты SOA приведены далее.

7.3.2 Интеграция

Интеграция — это центральный аспект проектирования и разработки решений SOA, поскольку часто требуется рассматривать службы, взаимодействующие с другими службами или доменами. В сложных реализациях SOA может потребоваться взаимодействие между подразделениями или партнерами. В таких случаях интеграция используется для усиления существующих служб, а также подключения новых служб, находящихся в других доменах. Даже в том случае, когда службы устроены проще и ограничены одним доменом, необходимо идентифицировать аспекты интеграции и соответствующие технологии поддержки.

7.3.3 Взаимодействие через границы доменов

Обнаружение и управление службой может распространяться на несколько различных доменов решений SOA внутри организации или нескольких организаций. Кроме того, обнаружение и управление должны будут осуществляться посредством различных технологий, так как различные домены решений SOA, возможно, выберут различные средства для своей реализации SOA.

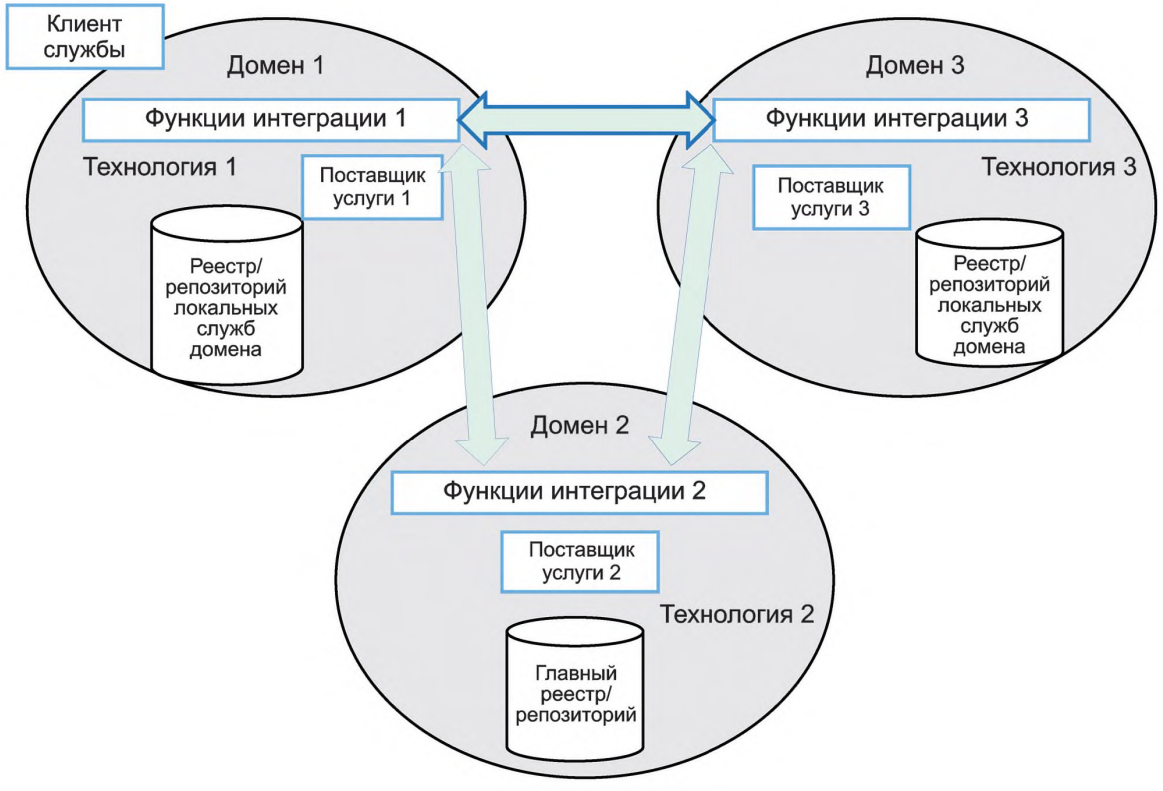

Рисунок 12 — Взаимодействие через границы доменов

На рисунке 12 приведен пример нескольких доменов решений SOA, каждый из которых реализован по своей собственной технологии, где требуется управлять взаимодействием между ними как единым целым.

Управлению сквозными аспектами, особенно обнаружением служб, управлением службами, безопасностью служб и руководством службами, уделяется внимание как в федеративных гетерогенных инфраструктурах SOA, так и инфраструктурах, построенных на единственном домене решения SOA.

В зависимости от технического или делового контекста потребителя службы, одна и та же служба может предоставляться различными поставщиками внутри и вне организации, а также посредством различных технологий. У этих служб имеются конкретные оконечные точки, которые могут находиться в различных доменах служб. Функциональность интеграции и связанный с ней реестр/репозиторий в домене потребителя службы упрощают нахождение и подключение подобных реализаций службы.

7.3.4 Интеграция служб

Интеграция служб крайне важна, поскольку она обеспечивает возможность передачи, преобразования, маршрутизации и доставки запросов от запрашивающей стороны службы до корректной реализации службы. Интеграция служб может включать в себя механизмы интеллектуальной маршрутизации, переключения протоколов, трансформации интерфейса. Такие механизмы часто встречаются в шине служб. Функциональность интеграции (см. рисунок 12) поддерживает слабое связывание между запросом и конкретным поставщиком благодаря функции подбора реализации службы для запроса.

Подобное слабое связывание не ограничивается только техническими вопросами протоколов, адресации или платформ, но может содержать бизнес-семантику для выполнения требуемой адаптации между запрашивающей стороной службы и поставщиком.

Наличие слабого связывания и де-факто неоднородность организаций вызывают потребность в более гибком управлении службами и соответствующими артефактами. Необходимо рассматривать три аспекта интеграции: транспорт, преобразование и посредничество.

Транспорт — сообщение, исходящее от отправителя, должно достигать своего намеченного получателя. Это происходит за счет использования транспортной инфраструктуры, которая обеспечивает протоколы, необходимые для передачи сообщений между этими участниками. И отправитель, и получатель могут иметь общую транспортную инфраструктуру, в противном случае потребуется создание связи между их соответствующими инфраструктурами. Для образования такой связи может потребоваться преобразование протоколов, а также маршрутизация и корреляция сообщений между оконечными точками.

Преобразование — отправитель и получатель должны одинаково интерпретировать содержимое сообщения. Для этого может потребоваться преобразование форматов представления содержимого или словарей для обеспечения одинаковой локальной семантики. Например, если одна из оконечных точек использует версию формата данных с дополнительными полями, то необходимо удалить дополнительные поля для того, чтобы сообщение было приемлемым для другой стороны.

Посредничество — при интегрировании процессов не всегда сообщение одной стороны равнозначно единственному сообщению другой стороны. Эти две стороны могут находиться на разных уровнях абстракции, где одно сообщение может создать веер различных сообщений, причем несколько ответов необходимо объединить в единственный ответ. Посредничество может сочетать транспорт и преобразование с более сложными схемами интеграции процессов.

7.3.5 Управление и безопасность

7.3.5.1 Общие положения

Вопросы управления и безопасности являются важными аспектами SOA. Они обеспечивают способы обработки нефункциональных требований в любом заданном решении. Они предоставляют технические средства, обеспечивающие потребности решений SOA относительно мониторинга, надежности, доступности, управляемости, транзакционности, пригодности для обслуживания, масштабируемости, информационной безопасности, эксплуатационной безопасности, жизненного цикла и т. д. Управление и безопасность относятся к той же области применения, что и традиционные аспекты (отказ, конфигурация, учет, производительность, безопасность) из справочника лучших практик ITIL или три базовых фактора: «надежность, доступность, удобство обслуживания» (RAS — reliability, accessibility, serviceability).

Общими аспектами управления, которые, возможно, потребуется поддерживать и управлять, являются транзакционность, доступность, надежность службы, управление метаданными и службы управления.

Транзакции координируют действия службы над ресурсами, принадлежащими распределенным компонентам, когда эти компоненты требуются для достижения непротиворечивого соглашения относительно результата взаимодействий службы. Менеджер транзакций обычно координирует результат,

скрывая специфические особенности каждого координируемого компонента службы и технологию реализации службы. При таком подходе может быть использована общая и стандартизированная транзакционная семантика для служб. Существует два типа транзакций — атомарные (обычно длятся недолго и отменяют изменения при отказе) и транзакции деловой активности (обычно выполняются долго, при отказе используются механизмы компенсации). Подробная информация о транзакциях в решениях SOA приведена в приложении В.

Доступность обозначает как доступность службы SOA, так и доступность решения SOA. Доступность решения SOA — это доля времени, в течение которой решение выполняет свои бизнес-функции соответствующим образом. Для ее измерения важно идентифицировать и отслеживать ключевые показатели эффективности решения в целом. Измерение доступности отдельной службы не будет отражать доступность и выполнение деловых функций. Доступность служб связана с долей времени, в течение которой служба может быть успешно вызвана.

Доступность служб может быть измерена от границы поставщика, сети и потребителя, тем самым обеспечивается проверка функционирования службы, системы и сети. Доступность службы варьируется в зависимости от того, где проходит граница службы. Зачастую доступность службы можно измерить инфраструктурой и инструментами на стороне службы с использованием обычных метрик, таких как процент времени, когда служба выполнялась, и число недоставленных сообщений. Доступность службы является ключевой метрикой в соглашении об уровне обслуживания, политиках и контрактах.

В то время как доступность связана с функционированием службы, выполнением операций, готовностью к обработке запросов, надежность связана с тем, чтобы сообщения запросов и ответов были доставлены получателям. С другой стороны, надежность рассматривает доставку сообщений от потребителя службы к поставщику службы; она не определяет, обработан ли запрос поставщиком. Транзакционность рассматривает, обработан запрос или нет.

a) Надежность в контексте SOA является качеством службы, направленным на взаимодействие между потребителем службы и поставщиком службы, и используется для указания на различные уровни гарантий посредников и сети, включенных в обмен сообщениями, во время вызовов службы. Надежность — сквозной аспект, означающий, что один и тот же уровень качества должен сохраняться на всем протяжении пути от потребителя службы к поставщику службы. Можно указать четыре качества надежности для систем: «без потери с дублированием», «с потерями без дублирования», «без потерь и без дублирования» и «асинхронность и упорядоченность». Подробнее они описаны в приложении В. Надежность, требуемая для решения в SOA и взаимодействий службы, будет влиять на ключевые архитектурные решения. Например там, где невозможно разработать идемпотентную службу (см. приложение В), можно требовать другие качества надежности, прежде всего «с потерей без дублирования», «без потерь и без дублирования».

b) Службы управления представляют собой набор инструментов управления, используемых для мониторинга потоков службы, состояния нижележащей системы, использования ресурсов, идентификации отключений электричества и узких мест, достижения целей службы, применения административных принципов и восстановления после отказов. Службы управления могут использоваться в моделях службы как часть решения SOA для включения и управления функциональными службами и решениями SOA. Службы управления отличаются от функциональных интерфейсов, и интерфейсы управления могут быть реализованы непосредственно службами для обеспечения управляемости.

c) Руководство SOA и службами использует управление для фактического выполнения и обеспечения соблюдения политики и процессов управления. Эти политики управления вместе с другими политиками в системе (безопасности, надежности, доступности и т. д.) являются теми правилами, которые формируют системы управления.

7.3.5.2 Управление решением SOA

Те виды управления, которые применяются к бизнесу в целом, важны и для управления службами и решениями SOA и, возможно, требуют расширений для управления сервис-ориентированностью и междоменностью многих решений SOA. Управление службами фокусируется только на управлении службами, в то время как управление SOA имеет более широкую область применения и управляет решением SOA целиком или набором решений SOA.

Руководство SOA и службами использует управление для фактического выполнения и обеспечения соблюдения политики и процессов управления. Эти политики управления вместе с другими политиками в системе (безопасности, надежности, доступности и т. д.) являются теми правилами, которые формируют системы управления.

Общими типами управления, которые используются для решений SOA, являются:

- мониторинг и управление службой и решением SOA: обеспечивают мониторинг и управление программным обеспечением служб и приложением, включая возможность собирать данные метрик,

отслеживать и управлять состоянием решения и приложения. Особым вызовом для SOA является ситуация, когда службы или решения, находящиеся под управлением, используют службы из другого домена владения как внутри организации, так и в других организациях. Метрики потребителя службы необходимо согласовать, сбор данных метрик и представление результатов зачастую происходит на инструментальной панели оператора. Для служб из других доменов такая информация может доставляться через отчеты, события или другие информационные каналы в их собственные инструментальные панели;

- мониторинг и управление деловой активностью: обеспечивают мониторинг и управление деловой активностью и бизнес-процессами; это дает возможность анализировать информацию о событиях в режиме реального времени/почти реального времени, а также о сохраненных событиях, рассматривать и проводить оценку деловой активности в форме информации о событии, а также определять ответы и предупреждения/уведомления о проблемах. Это создает баланс в организации между управлением решением и службами SOA и управлением событиями. Зависимости, вызванные наличием нескольких организаций, могут управляться соответствующим образом;

- мониторинг и управление ИТ-системами: обеспечивают мониторинг и управление инфраструктурой ИТ и системами, включая возможность мониторинга и получения метрик и состояния ИТ-систем и инфраструктуры, что включает в себя управление виртуализированными системами. Для решений SOA эти системы сами могут быть представлены и доступны в виде служб, а ресурсы могут находиться в управлении организации или в других доменах. В случае, когда используются несколько доменов, передача отчетов потребителям службы и клиентам может потребовать специальных согласований и разрешений;

- управление безопасностью: управляет и отслеживает безопасность и защищенные решения, что дает возможность управлять ролями и идентификационными данными, правами доступа и наделением правами; защищать неструктурированные и структурированные данные от несанкционированного доступа и потери данных; рассматривать, как разрабатывается и сохраняется программное обеспечение системы и службы на протяжении всего жизненного цикла программного обеспечения; поддерживать состояние безопасности через превентивные изменения, реагирующие на идентифицированные уязвимости и новые угрозы, позволяя ИТ-организации управлять рисками, связанными с ИТ, соответствовать и обеспечивать основание для автоматизации управления безопасностью. Политики безопасности для SOA для согласования между доменами организаций должны быть четко определены в описаниях, а также поддерживаться;

- мониторинг и применение политики: обеспечивают механизм для мониторинга и применения политик и деловых правил к службам и решению SOA; включают в себя обнаружение и доступ к политикам, оценку и применение политик в контрольных точках. Применение политики подразумевает ее применение после того, как получены, доставлены и записаны данные метрик. Применение политики также подразумевает отправку метрики или статуса соответствия (политике), а также уведомлений и ведение журнала нарушений (политики). Должны поддерживаться уведомления о нарушении политик в случае, когда используются несколько доменов организаций. Кроме того, заблаговременно должны быть разработаны и внедрены процедуры рассмотрения тех политик, которые не применяются должным образом;

- управление событиями: обеспечивает возможность управления событиями; включает в себя обработку событий, ведение журнала и аудит. Для SOA использование событий между организациями является общим сценарием уведомлений для рабочих мест операторов, систем мониторинга и управления. Процессы аудита должны быть всегда активированы, однако аудит особенно важен для решений, затрагивающих несколько организаций;

- управление конфигурацией и изменениями: обеспечивает возможность изменения конфигурации, описания решения и службы. В том случае, когда используются службы и решения SOA из нескольких организаций, управление конфигурацией и управление изменениями должны быть скоординированы для отдельных служб и решений;

- управление жизненным циклом службы и решения: обеспечивает механизмы для развертывания, запуска/включения, остановки/отключения и удаления служб и решений SOA. Для организации, в портфеле которой имеются службы, использующие службы, принадлежащие другим доменам или организациям, должен быть скоординирован жизненный цикл служб. Необходимы механизмы коммуникации при обслуживании или обновлении служб либо при прекращении использования служб потребителями;

- управление метаданными службы: обеспечивает управление дополнительными метаданными службы и физическими документами, такими как описания служб и политики, связанные со службами. Метаданные могут потребоваться многим характеристикам SOA: обнаружению, слабому связыванию, интеграции, а также отношениям и свойствам службы.

7.3.5.3 Безопасность SOA

Безопасность SOA рассматривает защиту от угроз в контексте уязвимостей сервис-ориентированной архитектуры; это — защита взаимодействий между потребителями службы и поставщиками службы, а также защита всех элементов, которые участвуют в определении архитектуры.

Подход, используемый для определения безопасности SOA, должен следовать лучшим образцам передового опыта в этой отрасли, таким как рекомендация X.805, разработанная ITU-T, и аспекты, приведенные в комплексе стандартов WS-Security (см. [28]).

Угрозы, которые должны быть учтены при рассмотрении безопасности SOA, — это разрушение, повреждение, удаление, раскрытие и прерывание (информация о данных угрозах приведена в приложении В). Для защиты от перечисленных угроз необходимы восемь аспектов безопасности: управление доступом, аутентификация, неотказуемость, конфиденциальность данных, коммуникационная безопасность, целостность данных, доступность и конфиденциальность (подробная информация об этих аспектах безопасности решений SOA приведена в приложении В).

В рамках SOA необходимо уделять особенное внимание следующим ключевым понятиям:

- аутентификация. Аутентификация должна подтвердить идентификационные данные для использования служб или любых компонентов SOA, таких как общий секрет, инфраструктура открытых ключей или цифровые подписи. Необходимо отметить, что, поскольку SOA определяет взаимодействие между потребителями служб и поставщиками служб, подтверждение идентификационных данных должно выполняться между исходным потребителем и окончательным поставщиком службы. В хореографии служб используются различные агенты, и аутентификация может потребоваться для всех. Когда службы предоставляются двумя или несколькими организациями, то существует риск, что из-за подключения и отключения агентов среда станет трудноуправляемой. Поэтому обычно в SOA используются интегрированные идентификационные данные, основанные на распространении маркеров, которые обеспечивают непротиворечивую аутентификацию начальной запрашивающей стороны посредством доверенного протокола между компонентами и реализуют доверенную модель аутентификации. В качестве примера может служить WS-Federation (см. [29]);

- неотказуемость. Неотказуемость должна предотвращать возможности отказа от факта выполнения действия в компонентах или элементах сервис-ориентированной архитектуры. Примерами могут служить системные журналы и цифровые подписи, защищающие сообщения посредством XML-Signature. В хореографии служб взаимодействуют множество агентов, и может потребоваться поддержка трассировки всех действий от всех агентов;

- доступность. Доступность должна обеспечить доступность элементов, служб и компонентов для законных пользователей. Она должна координироваться с SLA и другими метаданными контракта, которые определяют условия использования службы.

7.3.6 Руководство решением SOA

Руководство — это установление и осуществление сотрудничества между людьми и решениями для достижения целей организаций. Для организации, внедряющей принципы SOA, руководство решениями SOA должно способствовать установлению ожиданий и обеспечению средств по снижению рисков, поддержанию согласованности бизнеса и демонстрации деловой значимости. Как и другие виды руководства, руководство решениями SOA должно создать единый подход для служб, процессов, стандартов, политик и директив, внедряя механизмы контроля и обеспечения соответствия. Фокусирование на внедрении средств управления отличает руководство от ежедневного управления деятельностью (см. [10]).

Ниже приведены некоторые причины общего характера, объясняющие потребность SOA в руководстве, причем иногда отсутствие руководства может привести к неуспеху SOA:

- SOA зачастую вызывается организационными изменениями;

- SOA зачастую приводит к организационным изменениям;

- реализация SOA затрагивает несколько организаций, особенно при совместном использовании служб;

- повторное использование служб часто требует от организаций создания новых требований по разделению затрат;

- SOA в организации способствует появлению новых методов идентификации служб и их совместного использования.

В приложении А приведено краткое изложение ИСО/МЭК 17998, описывающего структуру управления SOA, детали эталонной модели управления SOA, метод жизнеспособности руководства SOA и передовые методы создания режима руководства для служб и решений SOA.

Фактически руководство SOA расширяет возможности руководства информационными технологиями и предприятием, обеспечивая те преимущества, которые приветствуются в SOA. Это подразумевает руководство не только аспектами выполнения SOA, но также и действиями стратегического планирования. Многие организации имеют режим руководства для ИТ-отделов, охватывающий финансирование проекта, разработку и техобслуживание. Их можно определять средствами одной из формализованных стандартизованных структур управления ИТ, таких как COBIT, ITIL и т. д., или средствами внутренней структуры управления, которая создавалась в течение долгого времени.

Режим руководства для решений SOA включает в себя руководство службами, а также руководство решением SOA в целом. Руководство следует применять к:

a) процессам, включая процессы руководства и руководимые процессы;

b) организационным структурам, включая роли и обязанности;

c) возможностям технологий, включая инструменты и инфраструктуру.

Для того чтобы определить изменения, необходимые для внедрения SOA в существующий режим управления, соответствующие действия управления должны быть отображены на деятельности, используемой в существующем режиме, и интегрированы в них. Несмотря на то, что невозможно найти два совершенно одинаковых режима руководства, стандарты руководства определяют наилучшие методы руководства и могут задать отправную точку для настройки решения SOA. Для разработки специализированного режима руководства SOA предприятия может быть определена и использоваться эталонная модель руководства SOA. Она должна включать действия руководства, на которые влияет SOA.

Руководство процессами, соответствием, разрешением и передачей реализует намерения организации. Процессы и действия, управляемые для решений SOA, различаются для аспектов планирования, проектирования и эксплуатации SOA. Как описано в приложении А, руководство решением SOA должно учитывать, что:

- управление портфелем решений SOA определяет, какие из решений SOA будут разрабатываться и сопровождаться;

- управление портфелем служб определяет, какие из служб будут разрабатываться и сопровождаться;

- управление жизненным циклом решения SOA отвечает за разработку, эксплуатацию, модификацию и вывод из эксплуатации решений SOA. Это повышает требования к разработке служб, которые учитываются при управлении портфелем служб, и требования к изменениям служб, которые вызваны управлением жизненным циклом службы;

- управление жизненным циклом службы отвечает за разработку, эксплуатацию, модификацию и вывод из эксплуатации служб.

Для руководства решениями SOA требуется непрерывный процесс поддержания согласованности, нереалистично будет ожидать, что предприятие в единственном проекте будет определять и развертывать режим управления SOA всего предприятия, который соответствует ее долгосрочным стремлениям и целям SOA. Предполагается, что этот процесс будет развиваться с ростом опыта SOA организации. Для поддержания данного процесса в методе жизнеспособности руководства SOA организация проходит через набор поэтапных действий, которые формируют цикл непрерывного совершенствования, определяя дорожную карту для развития управления. Это непрерывный процесс, в котором измеряется прогресс, происходит исправление курса и выполняются соответствующие обновления.

8 Архитектурные принципы SOA

8.1 Определение архитектурных принципов

Архитектурные принципы SOA исходят главным образом из важности трех принципов независимости SOA: независимость от расположения, независимость от реализации и независимость от протокола.

Независимость от расположения: не существует какого-либо предпочтительного местоположения для потребителей службы и поставщиков службы. Они оба могут располагаться в одной и той же системе или в различных организациях и различных географических местах.

Независимость от реализации: не существует каких-либо требований к определенной платформе или технологиям реализации, которые должны внедрить потребители или поставщики службы. Для взаимодействия им не требуется знания технической среды другой стороны или детали реализации.

Независимость от протокола: доступ к службам предоставляется по различным транспортным протоколам и протоколам обмена сообщениями. Допускается использование промежуточного протокола

или компонента для обеспечения совместимости. В случае различных протоколов подключение между разнородными службами могут осуществлять шины служб.

Службы должны проектироваться таким образом, чтобы они были функционально совместимы, их можно было описать, повторно использовать, обнаруживать, включать в композиции, чтобы они были автономными, со слабым связыванием и управляемыми. Каждая из этих характеристик описана в настоящем разделе; в описания включены списки деятельностей, которые должны принимать во внимание архитекторы при проектировании и разработке служб и решений SOA, проявляющих перечисленные характеристики. Некоторые деятельности применимы к нескольким характеристикам. Кроме того, в описании каждой характеристики приведен список характеристик или деловых преимуществ, которые служба или решение SOA могут демонстрировать, если будут обладать указанной характеристикой.

8.2 Функциональная совместимость — синтаксически и семантически

Характеристика «функциональная совместимость» означает, что люди, организации и системы могут передавать информацию, обмениваться ею и эффективно ее использовать для совместной работы.

У функциональной совместимости имеется множество аспектов, и она часто рассматривается с точки зрения системного проектирования, включая технические, синтаксические и семантические факторы. Более широкое представление о функциональной совместимости может включать в себя социальные, организационные, юридические, политические и другие факторы.

Техническая совместимость включает в себя использование общепринятых методов и служб для доступа, передачи, хранения и обработки данных, обычно в определенных доменах платформ приложений и коммуникационной инфраструктуры, на основе стандартов и/или распространенных ИТ-платформ.

Синтаксическая совместимость означает, что две или несколько систем способны передавать и обмениваться данными, используя заданные форматы данных или протоколы связи, например стандарты XML (расширенный язык разметки) или SQL (язык структурированных запросов). Синтаксическая совместимость служит предпосылкой для достижения дальнейшей функциональной совместимости.

Семантическая совместимость может быть достигнута посредством установления общего намерения, которое позволяет получателям оперировать содержанием сообщений таким образом, чтобы соответствовать намерению отправителя. Другими словами, должно присутствовать общее понимание бизнес-требований, предъявляемых к службе, а не только соглашение по общему синтаксису.

Функциональной совместимости способствуют:

- определение понятий протокола, которое включает в себя:
  - описание типов сообщений и типов содержимого, обмен которым будет происходить (синтаксис);
  - описание типов содержимого, включая любые ограничения на значения данных (помогает семантике);
  - определение нижележащих транспортных протоколов, которые могут использоваться (связь);
  - определение шаблонов обмена сообщениями, таких как сообщения запросов и ответов;
  - определение режимов, сообщений и состояний отказов;
  - описание любых диалоговых взаимодействий, например обратного вызова;
  - определение любых ограничений на порядок вызовов, например «open перед write»;
- определение политик, которые влияют на надежность, доступность, безопасность и транзакционность.

Преимущества функциональной совместимости:

a) включает минимальный уровень понимания семантики используемых и предоставляемых служб;

b) способствует обмену данными и совместной работе через границы систем и организаций;

c) позволяет повысить вероятность правильного взаимодействия двух систем;

d) позволяет уменьшить количество ошибочных взаимодействий;

e) позволяет сократить неоднозначность при ошибках и отказах.

8.3 Описание службы

Характеристика «у службы есть описание» означает, что потребитель службы, поставщик службы или разработчик могут найти, получить доступ, интерпретировать и обработать описание службы. Описание службы (см. 7.2.7) является метаданными, которые могут включать в себя детали о контракте службы, интерфейсе службы и политиках для потребителей и поставщиков службы.

Возможности описания службы способствуют:

- представление контракта взаимодействия службы в стандартизированном формате, например XML, WSDL (язык определения веб-сервисов, см. [18]) для реализаций веб-сервисов;

- хранение описаний в реестре/репозитории таким образом, чтобы его можно было обнаружить, получить к нему доступ и повторно использовать;

- изоляция конфигурации в описаниях политики;

- представление в описаниях информации для соглашений об уровне обслуживания;

- представление в описаниях доступного уровня качества обслуживания;

- представление в описаниях бизнес-процессов, процессов руководства и управления, которые используются службами;

- представление места предоставления.

Возможность описания службы предоставляет следующие преимущества:

a) дает возможность слабого и позднего связывания, обеспечивая достаточное описание, чтобы можно было найти и установить связь с экземпляром службы во время выполнения;

b) дает возможность повторного использования, добавляя метаданные таким образом, чтобы соответствующие службы можно было обнаружить и повторно использовать;

c) дает возможность повторного использования путем добавления описания так, чтобы соответствующие службы были доступны для необходимых функций;

d) позволяет осуществлять управление и реконфигурацию служб во время выполнения и при ее повторном использовании для минимизации изменений для реализации службы;

e) позволяет осуществить управление службами во время выполнения путем описания мониторинга управления и процессов;

f) позволяет осуществлять бизнес- и ИТ-руководство решениями SOA и службами;

g) позволяет добавлять функциональность интеграции (см. рисунок 10) и шины служб (см. 2.22).

8.4 Повторное использование службы

Характеристика «возможность повторного использования» означает, что определение или реализация службы либо и то, и другое одновременно используются в нескольких решениях SOA. Наличие возможности повторного использования помогает уменьшить расходы на разработку и обслуживание, увеличивая долгосрочный доход от инвестиций для решения.

Возможности повторного использования службы способствует следующее:

- моделирование и анализ службы — моделирование служб, которые будут разработаны для поддержки решений SOA, поиска в решении общих задач и элементарных служб;

- правильный выбор гранулярности — более мелкие «вспомогательные» службы можно повторно использовать в композициях;

- использование автономных определений;

- соблюдение принципа слабого связывания служб;

- разработка хороших описаний службы, так, чтобы ее можно было найти, оценить применимость и использовать повторно;

- создание служб, поддающихся обнаружению;

- позднее связывание служб, получение адресов для текущей оконечной точки службы во время выполнения, чтобы во время выполнения ее можно было заменить другими службами, не перестраивая реализацию службы;

- управление бизнесом с использованием процессов для определения перепланировки и требующих повторного использования существующих служб;

- понимание того, как будет финансироваться разработка службы при ее использовании в нескольких организациях;

- наличие надлежащего жизненного цикла разработки и сопровождения программного обеспечения для поддержки повторного использования служб в различных решениях (т. е. влияние при обновлении реализации);

- создание возможности управления службами на основании политик;

- реализация служб с изолированной конфигурацией.

Повторное использование служб предоставляет следующие преимущества:

a) уменьшает стоимость разработки, избавляя от необходимости разрабатывать ряд служб;

b) позволяет со временем уменьшить стоимость SOA для организации, т. к. в будущем не потребуется переделывать службы для новых решений;

c) уменьшает время разработки, так как позволяет использовать существующие службы, которые не нужно разрабатывать;

d) уменьшает риски, так как службы прошли качественное тестирование прежде, чем использоваться повторно;

e) повышает гибкость разработки в тех случаях, где службы могут использоваться непредвиденным образом;

f) повышает гибкость разработки, поскольку службы доступны для быстрой разработки.

8.5 Обнаружение службы

Характеристика «возможность обнаружить службу» означает, что имеется возможность узнать о существовании, местоположении и/или описании службы, как правило, в процессе подготовки к взаимодействию со службой.

Возможность определить местоположение или обнаружить службы и их описание на основе определенных критериев может быть реализована реестром/репозиторием или другими видами технологий поиска. Принципиальным является то, чтобы потребители служб могли находить службы и взаимодействовать до того, как могут потребоваться все остальные факторы управления, безопасности и руководства.

Инфраструктура обнаружения служб может варьироваться в широком диапазоне: от предоставления описания службы (нет обнаружения вообще) до поиска в хорошо определенном местоположении или справочнике и до реестра/репозитория со сложным поиском.

Поиск служб может осуществляться на этапе проектирования службы или во время ее выполнения. Поддержка обнаружения во время выполнения службы позволяет реализовать позднее связывание служб, что повышает степень слабого связывания.

Виды соединений также могут значительно варьироваться от статически заданного сильно связанного прямого взаимодействия, где информация о службе может быть получена реализацией службы из априорного знания, до динамического выбора служб-кандидатов на основании метаданных и связывания с правильным сервисом во время выполнения. Промежуточным будет общий сценарий обнаружения службы во время проектирования в реестре/репозитории с использованием на этапах моделирования, сборки и развертывания, а также поиск и привязка адреса службы во время выполнения. Обнаружение оконечных точек для служб во время выполнения может повысить степень слабого связывания.

Возможности обнаруживать службы способствуют:

- хранение артефактов описания службы в реестре/репозитории или другом предсказуемом расположении;
- обеспечение средств/инструментов поиска, которые позволяют искать, находить и получать доступ к описанию службы;
- наличие процессов руководства, диктующих правила регистрации и хранения описаний службы.

Возможность обнаруживать службы предоставляет следующие преимущества:

a) позволяет позднее связывание со службами, что увеличивает устойчивость;

b) позволяет реализовать слабое связывание, обеспечивая обнаружение и доступ к описанию службы;

c) позволяет повторное использование служб, гарантируя возможность обнаружить и использовать существующие повторно используемые службы и описания;

d) позволяет реализовать динамическую компоновку композиций.

8.6 Позднее связывание

Характеристика «позднее связывание» означает доступ к информации во время выполнения для настройки привязки. Данная характеристика может быть независимой от обнаружения службы.

Использование позднего связывания через реестр/репозиторий позволяет потребителю службы изолировать себя от изменений места предоставления услуг. Объединение использования политики с поздним связыванием дополнительно изолирует потребителя услуг от реализации службы, позволяя настраивать взаимодействие службы. Наконец, добавление функции интеграции изолирует потребителя услуг и реализацию услуги и способствует повышению надежности и доступности. Такое сочетание характеристик позволяет изменять количество ресурсов, масштабирование и качество обслуживания в соответствии с потребностями бизнеса без влияния на потребителя службы или реализации службы.

Обнаружение службы и позднее связывание могут способствовать динамической компоновке композиций.

Позднему связыванию способствуют:

- предоставление средств/инструментов поиска для поиска, нахождения и доступа к описанию службы;

- процессы руководства, поддерживающие использование привязки с оконечной точкой фактической реализации службы во время выполнения.

Преимущества позднего связывания:

a) обеспечивает повышенную устойчивость;

b) обеспечивает возможность слабого связывания, предоставляя возможность обнаружения и доступа к описанию службы;

c) позволяет повысить надежность и гибкость решения SOA;

d) позволяет повторно использовать службы, гарантируя, что существующие повторно используемые службы и описания службы могут быть обнаружены и использованы;

e) обеспечивает динамическую компонуемость (если данная характеристика поддерживается) с использованием реестра/хранилища служб.

8.7 Компонуемость

Характеристика «компонуемость» означает, что служба может быть элементом композиции без изменения реализации службы.

Возможность компонуемости службы обеспечивается надлежащим образом описанными службами, которые слабо связаны и автономны.

Компонуемость не зависит от типов создаваемых композиций, таких как процессы, хореографии и оркестровки.

Компонуемости способствуют:

- соблюдение принципа повторного использования;
- разработка автономных служб;
- обнаружение служб через артефакты в реестре/хранилище данных.

Компонуемость предоставляет следующие преимущества:

- обеспечивает гибкость разработки путем повторного использования существующих служб для разработки новых служб, без ожидания новых разработок;
- позволяет разрабатывать композиции, процессы, хореографии и оркестровки.

8.8 Автономность

Характеристика «замкнутость» или «автономность» означает, что для вызова службы достаточно информации, доступной в описании служб. Потребитель службы должен быть изолирован от деталей реализации службы. Автономные службы должны инкапсулироваться, и их состояние не должно зависеть от других служб либо они не имеют состояния. Службы, которые являются композициями других служб, могут быть автономными, если они не нарушают инкапсуляцию и не позволяют потребителям службы видеть или понимать, как реализованы эти службы.

Автономности способствуют:

- изоляция потребителя службы от реализации (непрозрачность);
- разработка автономных служб;
- разработка полных описаний,
- предоставление независимого от реализации интерфейса.

Преимущества автономности:

a) повышение степени повторного использования;

b) повышение степени компонуемости.

8.9 Слабое связывание

Характеристика «слабое связывание» означает, что потребление службы изолировано от нижележащей реализации.

В большинстве отраслей повторное использование дает возможность некоторой изменчивости и гибкости при соединении с другими элементами. В отрасли информационных технологий подобная изменчивость и гибкость называется слабым связыванием.

Архитектурный стиль SOA поддерживает слабое связывание интерфейса службы с предоставляемой реализацией. Обмен сообщениями должен быть четко определен, но тем не менее должен обеспечивать принципиальные механизмы для изменений во входных данных и расширяемости.

Слабому связыванию способствуют:

- требование, чтобы потребители службы использовали интерфейсы и/или контракты для взаимодействия со службами; это гарантирует, что функциональность службы может быть выполнена любой доступной реализацией, а не каким-то определенным экземпляром реализации;
- отделение реализации от привязки и интерфейса оконечной точки службы;
- выбор интерфейсов в стиле обмена документами вместо удаленного вызова процедур;

- использование потребителями службы позднего связывания (во время выполнения) с экземпляром службы посредством обнаружения службы, которое может быть предоставлено реестром, репозиторием, промежуточным ПО или иным механизмом;

- перенос конфигурации и контрактов времени выполнения в политики;

- использование функциональности интеграции (см. рисунок 10) с целью обеспечения поддержки соединений, протоколов связи, безопасности и других качеств службы. Это освобождает потребителя службы и реализацию службы от необходимости непосредственного обеспечения этих качеств или непосредственно «подбирать» уровень качества обслуживания.

Преимущества слабого связывания:

a) позволяет минимизировать изменения для потребителей служб с течением времени, даже если изменения версий или изменения функциональности необходимы для качества обслуживания или поддержки протокола, при условии, что в остальном случае служба будет приемлема;

b) дает возможность замены альтернативной службой, при условии, что эффекты, семантика и политики остаются теми же;

c) позволяет минимизировать изменения служб поставщиком для изменения версий, протоколов, набора ресурсов и качества обслуживания;

d) позволяет минимизировать изменения в реализации службы для поддержки повторного использования служб в новых возможностях, повышая гибкость бизнеса;

e) повышает доступность и надежность службы для потребителей службы, позволяя выбирать надлежащую службу с поздним связыванием во время выполнения;

f) позволяет снизить стоимость (благодаря повторному использованию служб);

g) обеспечивает дополнительные возможности и преимущества функциональности интеграции;

h) обеспечивает объединение доменов SOA — совместное использование служб из одного домена SOA-решения в другом;

i) позволяет управлять услугами на основе политики;

j) дает возможность повторного использования служб путем изоляции конфигурации в политику.

Для достижения этих преимуществ слабого связывания потребуется дополнительное управление метаданными службы.

8.10 Управляемость

Характеристика «управляемость» означает, что службы и решения могут разрабатываться, контролироваться, отслеживаться, конфигурироваться и управляться таким образом, чтобы соблюдались политики, контракты и соглашения.

Управляемости способствуют:

- определение и мониторинг ключевых метрик доступности, надежности, производительности и руководства;

- предоставление интерфейсов для возможностей управления вместе с интерфейсами для функциональных возможностей;

- возможность управлять службами на основе политики;

- изолирование конфигурации в политику;

- включение конфигурации служб во время проектирования и выполнения;

- обеспечение контроля за соблюдением контрактов;

- разработка политики и процессов руководства;

- возможность управления жизненным циклом службы (включение, отключение).

Преимущества управляемости:

a) обеспечение осведомленности о доступности службы;

b) снижение риска отказа службы;

c) обеспечение исполнения контракта;

d) выполнение и автоматизация процесса управления;

e) согласование бизнес-требований и требований к ИТ для решения SOA;

f) автоматизация управления сервисами и решениями SOA.

Приложение А
(справочное)

Структура управления SOA

В настоящем приложении приведено краткое изложение ИСО/МЭК 17998, определяющего структуру управления SOA. Данное приложение задает контекст тем читателям, которым требуется более глубокое понимание руководства SOA, руководства службой и режима руководства SOA.

Руководство SOA должно рассматриваться как приложение руководства бизнесом, руководства информационными технологиями и руководства предприятием к SOA. Фактически руководство SOA расширяет возможности руководства информационными технологиями и предприятием, обеспечивая те преимущества, которые приветствуются в SOA. Это подразумевает руководство не только аспектами выполнения SOA, но также и действиями стратегического планирования. Многие организации имеют режим руководства для своих ИТ-отделов, охватывающий финансирование проекта, разработку и техобслуживание. Их можно определять средствами одной из формализованных стандартизованных структур управления ИТ, таких как COBIT, ITIL и т. д., или средствами внутренней структуры управления, которая создавалась в течение долгого времени.

Руководство архитектурой предприятия (руководство EA) является практикой и парадигмой, посредством которой управляются архитектура предприятия и другие архитектуры на уровне предприятия (см. [16]).

Руководство информационными технологиями (руководство ИТ) — это система, с использованием которой осуществляется руководство и контроль за существующим и будущим использованием ИТ. Корпоративное руководство ИТ включает оценку и задание направления для использования ИТ, для поддержания работы организации и мониторинга этого использования с целью достижения поставленных задач. Это включает стратегию и политики использования ИТ в организации (см. ИСО/МЭК 38500:2008, пункт 1.6.3).

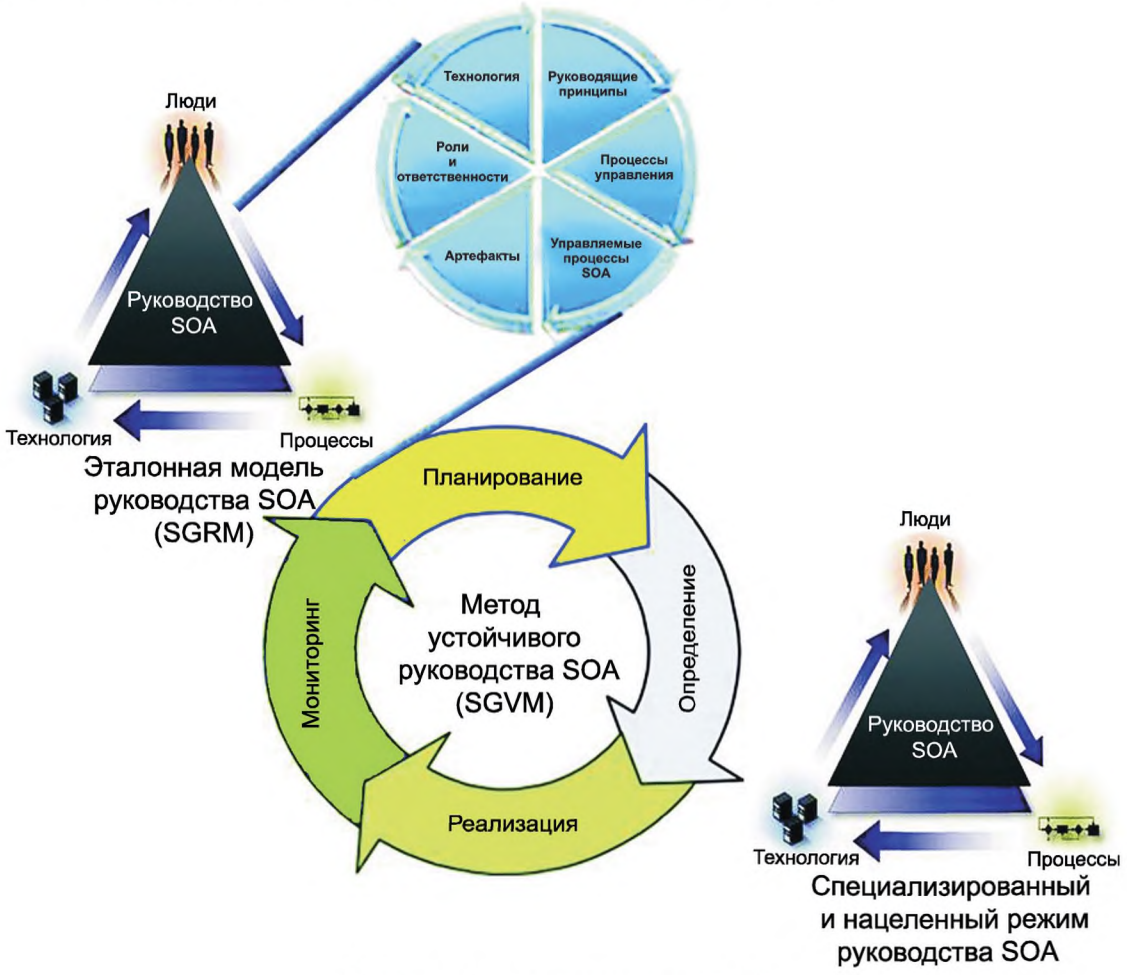

Рисунок А.1 — Режим управления SOA

Управление бизнесом (корпоративное руководство) является набором процессов, традиций, политик, законов и институтов, влияющих на способ управления, администрирования и контроля организации.

Режим руководства для решений SOA включает руководство службами, а также руководство решением SOA в целом. Руководство применяется к:

a) **процессам,** включая руководство и руководимые процессы;

b) **организационным структурам,** включая роли и обязанности;

c) **возможностям технологий,** включая инструменты и инфраструктуру.

Структура управления SOA предоставляет возможность предприятиям определять и развертывать свои собственные сфокусированные и специализированные режимы руководства SOA на основе специфичных для предприятия факторов, таких как бизнес-модель, зрелость SOA и размер компании. Она описывает, какие решения должны быть приняты, кто должен их выполнять, каким образом они будут выполняться и отслеживаться, какая организация и инструменты будут их поддерживать. Здесь используется инкрементный подход для того, чтобы предприятия могли продолжать реализацию своих текущих требований SOA, выполняя свои долгосрочные стремления и цели SOA.

Чтобы определить изменения, необходимые для внедрения SOA в существующий режим управления, соответствующие действия управления должны быть отображены на деятельности, используемые в существующем режиме, и интегрированы в них. Несмотря на то, что невозможно найти два совершенно одинаковых режима руководства, стандарты руководства определяют наилучшие методы руководства и могут задать отправную точку для настройки решения SOA. Для разработки специализированного режима руководства SOA предприятия может быть определена и использована эталонная модель руководства SOA. Она должна включать в себя действия руководства, на которые влияет SOA, а именно:

- **руководящие принципы** — ряд руководящих принципов для поддержки процессов принятия решений при проектировании, развертывании и эксплуатации решения SOA и режима руководства SOA;

- **процессы** — объекты руководства, т. е. описания действий и процессов, которые являются объектами руководства SOA. Для аспектов планирования, проектирования и эксплуатации SOA определены четыре группы действий:

  - управление портфелем решений SOA определяет, какие из решений SOA будут разрабатываться и сопровождаться;
  - управление портфелем определяет, какие из решений SOA будут разрабатываться и сопровождаться;
  - управление жизненным циклом решения SOA отвечает за разработку, эксплуатацию, модификацию и вывод из эксплуатации решений SOA. Это повышает требования к разработке служб, которые учитываются при управлении портфелем служб, и требования к изменениям служб, которые вызваны управлением жизненным циклом службы;
  - управление жизненным циклом службы отвечает за разработку, эксплуатацию, модификацию и вывод из эксплуатации служб;

- **руководство процессами** — описания руководства процессами, которые управляют указанными действиями. Руководство процессами реализует намерения руководства организации. В целом данные процессы будут интегрированы в существующие процессы руководства предприятием без образования отдельного специального набора процессов SOA. Существуют следующие три типа управления процессами:

  процесс **соответствия** гарантирует, что управляемые действия выполняются корректно. Для этого анализируются результаты руководства в контрольных точках;

  процесс **разрешений** позволяет проекту или команде разработчиков решения получать исключения из требований процесса соответствия, в частности, в тех случаях, когда применение общей политики будет неуместным;

  процесс **коммуникаций** нацелен на обучение людей, которые принимают участие в руководимых действиях, и передачу им знания архитектуры системы и режима руководства. Этот процесс включает в себя установку сред и инструментов для обеспечения легкого доступа к информации руководства и ее использованию;

- **роли и обязанности** — определяются через анализ категорий людей, вовлеченных в управляемые действия, и степени их ответственности за эти действия;

- **артефакты** — ряд артефактов, которые будут использоваться процессами руководства и которые должны своевременно обновляться и быть доступны для заинтересованных лиц;

- **включение технологии** — технология для разрешения и реализации процессов руководства. Рамки технологических возможностей могут варьироваться по своим возможностям от ручных процессов до сложного программного обеспечения. Иногда технология, используемая для реализации решений SOA, может быть использована для поддержания руководства им.

Для руководства решениями SOA требуется непрерывный процесс поддержания согласованности, нереалистично будет ожидать, что предприятие в единственном проекте станет определять и развертывать режим управления SOA всего предприятия, который соответствует ее долгосрочным стремлениям и целям SOA. Поэтому метод

жизнеспособного управления SOA должен определять и развертывать режимы управления SOA постепенно, инкрементально. В данном методе за основу берется эталонная модель управления SOA, которая постепенно адаптируется для соответствия конкретному предприятию посредством множества поэтапных действий, что формирует цикл непрерывного совершенствования, определяя дорожную карту для развертывания управления. Это не одноразовое действие, а непрерывный процесс, в котором постоянно происходит измерение прогресса, периодически происходит коррекция курса и выполняются обновления. Такой метод должен включать следующие фазы:

a) **планирование:** идентифицирует требования для руководства SOA, рассматривает существующий режим руководства и определяет видение, область применения и стратегию предполагаемых будущих изменений;

b) **определение:** создает адаптированный режим руководства SOA на базе эталонной модели руководства SOA[1)], используя результаты фазы планирования. Затем анализируются различия между данным режимом и существующим режимом управления для создания планов перехода;

c) **реализация:** реализует адаптированный режим управления SOA, созданный на фазе определения, чтобы построить режим руководства SOA для предприятия; по существу это выполнение планов перехода, созданных на фазе определения;

d) **мониторинг:** отслеживание эффективности режима управления SOA, построенного на фазе реализации, и проверка, соответствует ли данный режим намеченной цели. Мониторинг формирует требования для изменений, которые могут быть рассмотрены на следующем цикле метода жизнеспособности управления SOA.

[1)] См. gost.e-ecolog.ru

Приложение В
(справочное)

Управление и проблемы безопасности

В.1 Общие положения

В настоящем приложении подробно рассмотрены вопросы сквозного управления и безопасности и предоставляется контекст и дополнительная информация относительно управления транзакциями, надежности службы, безопасности SOA и руководства безопасностью SOA.

Управление и безопасность поддерживают нефункциональные требования, являющиеся ключевой особенностью/задачей SOA, и указывают те точки, на которых должно фокусироваться каждое решение. Это дает гарантию того, что SOA будет соответствовать требованиям мониторинга, надежности, доступности, управляемости, транзакционности, пригодности для обслуживания, масштабируемости, информационной безопасности, эксплуатационной безопасности, жизненного цикла и т. д. Управление и безопасность относятся к той же области применения, что и традиционные аспекты (отказ, конфигурация, учет, производительность, безопасность) из справочника лучших практик ITIL или три базовых фактора «надежность, доступность, удобство обслуживания» (RAS — reliability, accessibility, serviceability). Необходимо поддерживать три важных аспекта управления и безопасности — управление транзакциями, доступность и надежность службы.

В.2 Управление транзакциями

Транзакции могут координировать действия службы над ресурсами, принадлежащими распределенным компонентам, когда эти компоненты требуются для достижения непротиворечивого соглашения относительно результата взаимодействий службы.

Менеджер транзакций обычно координирует результат, скрывая специфические особенности каждого координируемого компонента службы и технологию реализации службы. При таком подходе может использоваться общая и стандартизированная транзакционная семантика для служб.

Транзакции состоят из двух типов транзакций, созданных по стандартам Веб-служб, а именно стандарт WS-Coordination (см. [25]), который является основой для WS-AtomicTransaction (см. [26]), и WS-Business Activity (см. [27]):

- **атомарные транзакции.** Атомарные транзакции происходят при построении компонентов служб, требующих непротиворечивого соглашения о результатах недолгих распределенных действий с семантикой «действие либо выполняется полностью, либо не выполняется совсем». Атомарная транзакция требует таких свойств, как атомарность, непротиворечивость, изоляция и длительность (ACID — atomicity, consistency, isolation and durability), и таким образом поддерживает блокировку измененных ресурсов до сохранения или отмены этих изменений. Результатом службы будет либо успешное изменение всех ресурсов, координированных службой, либо восстановление состояния, которое существовало перед взаимодействием службы;

- **транзакции деловой активности.** Транзакции деловой активности происходят при создании хореографий, требующих непротиворечивого соглашения о результатах скоординированных продолжительных распределенных взаимодействий служб, где блокировки не могут сохраняться на измененных ресурсах в масштабе времени хореографии. Так как у служб имеются обычные аспекты, не сохраняющие свои состояния, то изменения, происходящие в ресурсах, управляемых компонентом службы, после взаимодействия службы не могут восстанавливаться обратно к прежним значениям. Таким образом, в случае сбоя в координации необходимо предпринимать обратные меры посредством надлежащих взаимодействий служб с целью восстановления приемлемого состояния ресурсов, управляемых компонентами службы. Эти действия называются компенсацией, однако они могут приводить к зафиксированным изменениям. Например, некоторые службы могут не разрешать удаление недавно созданного ресурса, в этом случае действием компенсации будет установка состояния в «неактивно». По своему характеру транзакции деловой активности могут быть рекурсивными, что включает многократные области видимости деловой активности.

В.3 Доступность

Доступность решения SOA — это доля времени, в течение которой решение выполняет свои бизнес-функции соответствующим образом. Для ее измерения важно идентифицировать и отслеживать ключевые показатели эффективности решения в целом. Измерение доступности отдельной службы не будет отражать доступность и выполнение деловых функций.

Доступность служб связана с долей времени, в течение которой служба может быть успешно вызвана. Доступность служб может быть измерена от границы поставщика, сети и потребителя, тем самым обеспечивается проверка функционирования службы, системы и сети. Доступность службы варьируется в зависимости от того, где проходит граница службы. Зачастую доступность службы можно измерить инфраструктурой и инструментами на

стороне службы, с использованием обычных метрик, таких как процент времени, когда служба выполнялась, и число недоставленных сообщений.

Доступность службы часто является ключевой метрикой в соглашении об уровне обслуживания, политиках и контрактах.

B.4 Надежность службы

Надежность в контексте SOA является качеством службы, направленным на взаимодействие между потребителем службы и поставщиком службы, и используется для указания на различные уровни гарантий посредникам и сети, включенных в обмен сообщениями, во время вызовов службы. Надежность — сквозное свойство, означающее, что один и тот же уровень качества должен сохраняться на всем протяжении пути от потребителя службы к поставщику службы.

При разработке службы ключевым принципом является понятие идемпотентности. Если служба является идемпотентной, это означает, что одинаковые копии обращения к службе (например, путем повторений) приведут к одинаковому результату. Другими словами, достигается идентичный результат независимо от того, происходит запрос однократно либо повторяется несколько раз. Любые операции, не изменяющие состояние (чтение/получение), по определению являются идемпотентными. Чтобы произвести действие, изменяющее состояние, требуется тщательная разработка идемпотентной системы для обеспечения одинакового результата при одинаковых обращениях (запросах). При невозможности разработки идемпотентной службы к надежности задаются другие требования, например «доставка без дублирования сообщений с возможностью потери» и «гарантированная доставка».

Системы могут предоставлять четыре качества надежности:

- **без потери с дублированием:** запрос отправлен один раз, но есть вероятность доставки копий. К этой семантике подходят идемпотентные операции;

- **с потерями без дублирования:** запрос отправлен однократно, при этом не разрешается отправлять любые копии, поскольку эффект двойного запроса приводит к неправильному поведению, например повторному списанию денег с банковского счета. Если не удается отправить запрос, нет риска для корректного поведения службы;

- **без потерь без дублирования:** запрос отправлен однократно, и его не разрешается размножать или дублировать. Потеря запроса может представлять риск для корректного поведения службы. Другими словами, должны быть предприняты все разумные попытки, чтобы отправить запрос исключительно однократно;

- **асинхронно и упорядоченно:** потребитель службы может инициировать отправку нескольких различных запросов к одному и тому же поставщику на протяжении некоторого интервала времени. Так как промежуточное программное обеспечение и сети могут нарушать порядок доставки таких запросов, важно, чтобы порядок запросов сохранялся, если это требуется, между потребителем службы и поставщиком службы. Следует обратить внимание на то, что асинхронность и упорядочение могут быть объединены с тремя другими свойствами.

Надежность контрастирует с доступностью и транзакционностью. В то время, как доступность сама по себе контролирует функционирование службы, выполнение операций, готовность к обработке запросов, надежность связана с тем, что сообщения запросов и ответов доставляются получателям. С другой стороны, надежность заботится о доставке сообщений между потребителем службы и поставщиком службы; она не рассматривает, обработан ли запрос поставщиком. Это транзакционность рассматривает, обработан запрос или нет.

B.5 Управление SOA

Те виды управления, которые применяются к бизнесу в целом, важны и для управления службами и решениями SOA и, возможно, требуют расширений для управления сервис-ориентированностью и междоменностью многих решений SOA. Управление службами фокусируется на управлении службами, в то время как управление SOA имеет более широкую область применения и управляет решением SOA целиком или набором решений SOA. Общие типы управления, используемые для решений SOA, таковы:

- **мониторинг и управление ИТ-системами:** обеспечивает мониторинг и управление инфраструктурой ИТ и системами, включая возможность отслеживать и собирать данные метрик и состояния ИТ-систем и инфраструктуры, что также подразумевает управление виртуализированными системами;

- **мониторинг и управление приложениями и решениями SOA:** обеспечивает мониторинг и управление программным обеспечением служб и приложением, что включает возможность собирать данные метрик, отслеживать и управлять состоянием решения и приложения;

- **мониторинг и управление деловой активностью:** обеспечивает мониторинг и управление деловой активностью и бизнес-процессами, что дает возможность анализировать информацию о событиях в режиме реального времени/почти реального времени, а также информацию о сохраненных событиях, рассматривать и проводить оценку деловой активности в форме информации о событии, а также определять ответы и предупреждения/уведомления о проблемах;

- **управление безопасностью:** управляет и отслеживает безопасность и защищенные решения, что дает возможность управлять ролями и идентификационными данными, правами доступа и наделением правами; защи-

щать неструктурированные и структурированные данные от несанкционированного доступа и потери данных; рассматривать, как разрабатывается и сохраняется программное обеспечение системы и службы на протяжении всего жизненного цикла программного обеспечения; поддерживать состояние безопасности через превентивные изменения, реагирующие на идентифицированные уязвимости и новые угрозы, позволяя ИТ-организации управлять рисками, связанными с ИТ, соответствовать и обеспечивать основание для автоматизации управления безопасностью;

- **управление событиями:** обеспечивает возможность управления событиями; включает в себя обработку событий, ведение журнала и аудит;

- **управление конфигурацией и управление изменениями:** данная категория дает возможность изменять конфигурацию и описание решения и службы;

- **мониторинг и применение политики:** обеспечивает механизм для мониторинга и применения политик и деловых правил к службам и решению SOA; включает обнаружение и доступ к политикам, оценку и применение политик в контрольных точках. Применение политики подразумевает ее применение после того, как получены, доставлены и записаны данные метрик. Применение также подразумевает отправку метрики или статуса соответствия (политике), а также уведомлений и ведение журнала нарушений (политики);

- **управление жизненным циклом:** обеспечивает механизм для развертывания, запуска/включения, остановки/отключения и удаления служб и решения SOA.

Кроме того, руководство SOA и службами используют управление для фактического выполнения и применения политики и процессов управления. Эти политики управления вместе с другими политиками в системе (безопасности, надежности, доступности и т. д.) являются теми правилами, которые формируют системы управления.

B.6 Службы управления

Службы управления представляют собой набор инструментов управления, используемых для мониторинга потоков службы, состояния нижележащей системы, использования ресурсов, идентификации отключений электричества и узких мест, достижения целей службы, применения административных принципов и восстановления после отказов. Например, поскольку в качестве модели можно взять бизнес-проект и использовать его для компоновки служб решения, реализующих этот проект, будет видна корреляция между бизнесом и ИТ-системой. Эту корреляцию, если перенести ее в среду развертывания, могут использовать службы управления для установления приоритетов разрешения проблем, которые появляются в информационной системе, или выделения вычислительных ресурсов различным частям системы, взяв за основу цели уровня обслуживания, которые были установлены в бизнес-проекте.

Службы управления могут использоваться в моделях службы как часть решения SOA по включению и управлению функциональными службами и решениями SOA.

Службы управления отличаются от функциональных интерфейсов, и интерфейсы управления могут быть реализованы непосредственно службами для обеспечения управляемости.

B.7 Управление метаданными службы

В окружениях SOA важно обеспечить возможность управления дополнительными метаданными службы и физическими документами, такими как описания служб и политик, связанных со службами. Реестры и репозитории включают эти возможности. Такая функциональность необходима как дополнение к обнаружению служб.

Совокупность информации о службах необходима для обеспечения управления и усиления слабого связывания. Указанная совокупность информации также включает возможности использования реестров, репозиториев, функций интеграции и междоменной федерации.

Информация, которая должна храниться в реестрах, может включать физические документы, описывающие службу, наличествующие логические отклонения в коммуникации и связанные со службой сущности для поддержания отношений с сущностями, на которые потенциально оказано воздействие.

Метаданные для описаний службы, которые должны быть доступны, могут включать: свойства службы (другая информация о службе), отношения службы (информация об отношениях со службой) и классификацию служб (для поиска соответствий).

Существуют также другие описания, которые могут потребоваться наряду с пониманием доменов служб, интеграции (или шины предприятия) и федерации служб.

B.8 Безопасность SOA

Безопасность SOA рассматривает защиту от угроз в контексте уязвимостей сервис-ориентированной архитектуры; это — защита взаимодействий между потребителями службы и поставщиками службы, а также защита всех элементов, которые участвуют в архитектуре.

Подход, используемый для безопасности SOA, должен следовать лучшим образцам передового опыта в этой отрасли, таким как рекомендация X.805, разработанная ITU-T, и аспекты, описанные в серии стандартов WS-Security (см. [28]).

Ниже приведены угрозы, которые должна рассматривать безопасность SOA:

- **разрушение (атака на доступность):** разрушение информации и/или ресурсов, и/или компонентов, к которым получен доступ через службы или которые связаны со службами или жизненным циклом служб;

- **повреждение (атака на целостность):** несанкционированное вмешательство в актив, доступ к которому получен через службы, или актив связан со службами или жизненным циклом служб;

- **удаление (атака на доступность):** кража, удаление или потеря информации и/или других ресурсов, влияющих на службы;

- **раскрытие (атака на конфиденциальность):** несанкционированный доступ к активу или службе;

- **прерывание (атака на доступность):** прерывание службы. Служба становится недоступной или непригодной для использования.

Для защиты от перечисленных угроз необходимы восемь аспектов безопасности:

a) **управление доступом:** ограничение и управление доступом к службам и элементам, с использованием маркеров, компонентов и элементов: пароль, списки управления доступом, брандмауэр;

b) **аутентификация:** предоставление идентификационных данных для использования служб или любых компонентов SOA. Например, общий секрет, инфраструктура открытых ключей или цифровые подписи. Нужно отметить, что, поскольку SOA определяет взаимодействие между потребителями служб и поставщиками служб, подтверждение идентификационных данных должно выполняться между исходным потребителем и окончательным поставщиком службы. В хореографии служб используются различные агенты, и, возможно, аутентификация потребуется для всех. При аспекте, которым должен управлять SOA, когда службы предоставляются двумя или несколькими организациями, существует риск, что из-за подключения и отключения агентов среда станет трудноуправляемой. Поэтому обычно в SOA используются интегрированные идентификационные данные, основанные на распространении маркеров, которые обеспечивают непротиворечивую аутентификацию начальной запрашивающей стороны посредством доверенного протокола между компонентами и реализуют доверенную модель аутентификации. В качестве примера может служить WS-Federation (см. [29]);

c) **неотказуемость:** предотвращение возможности отказаться от факта выполнения действия в компонентах или элементах сервис-ориентированной архитектуры. Примеры: системные журналы и цифровые подписи, защищающие сообщения посредством XML-Signature. В хореографии служб взаимодействуют множество агентов, и может потребоваться поддержка трассировки всех действий от всех агентов;

d) **конфиденциальность данных:** гарантия конфиденциальности обмена данными при взаимодействии служб или управлении компонентами архитектуры. Чтобы избежать нежелательных разглашений, вероятно, потребуется обеспечить сквозную конфиденциальность между начальным потребителем службы и окончательным поставщиком службы. Пример: защита сообщения с использованием шифрования XML-Encryption;

e) **коммуникационная безопасность:** гарантия того, что информация исходит только из ожидаемого источника к ожидаемому месту назначения. Примеры: использование HTTPS, VPN, MPLS, L2TP. В архитектуре SOA на границе компонентов могут потребоваться проверки уровня приложения или решения для контроля адресов источников и точек назначения входящих и исходящих сообщений;

f) **целостность данных:** гарантия того, что полученные данные тождественны отправленным после всех взаимодействий между компонентами службы, будь то данные для управления SOA или целей жизненного цикла либо для взаимодействий между потребителями службы и поставщиками. Примеры: цифровая подпись XML-Signature, антивирусное программное обеспечение. Обработка подписи XML должна происходить только на конце взаимодействия службы, чтобы избежать нарушения целостности на различных узлах, которые могут составлять архитектуру между потребителем и поставщиком службы;

g) **доступность:** гарантия того, что элементы, службы и компоненты доступны законным пользователям. Данный аспект должен быть скоординирован с SLA и другими метаданными контракта, которые определяют условия использования службы;

h) **персональные данные:** гарантия того, что идентификация, использование и управление службой не разглашаются.

B.9 Руководство безопасностью SOA

Руководство безопасностью SOA — это специализированный домен руководства, который связан с безопасностью SOA.

Функции безопасности SOA: управление безопасностью SOA управляет жизненным циклом компонентов, которые обеспечивают функции для восьми размерностей безопасности, для защиты от перечисленных выше пяти угроз.

Инфраструктура политики безопасности: руководство безопасностью SOA определяет и реализует компоненты, которые администрируют политику безопасности, ее распространение и преобразование надлежащих компонентов SOA и элементов, решение политики безопасности и компонентов осуществления и наконец компонентов, которые отслеживают и сообщают об использовании политики безопасности и соответствии.

Процессы безопасности SOA: руководство безопасностью SOA определяет ИТ и бизнес-процессы для риска и соответствия, доверительного управления, идентификационных данных и жизненного цикла управления доступом, защиты данных и управления раскрытием и наконец безопасность для всех компонентов в SOA, таких как реестры, ESB и т. д. В среде мультиобъекта эти процессы включают в себя определение и обмен сертификатами или открытыми ключами между доверенными агентами, прежде чем разрешить взаимодействие других агентов в объектах.

Следует отметить, что конфиденциальность и отсутствие отказа могут привести к увеличению размера сообщения и объема из-за требуемой дополнительной информации, такой как сертификаты и дополнительное время обработки управлением подписями.

Библиография

[1] ISO/IEC 2382, Information technology — Vocabulary

[2] ISO/IEC/IEEE 15288, Systems and software engineering — System life cycle processes

[3] ISO/IEC 12207, Systems and software engineering — Software life cycle processes

[4] ISO/IEC/IEEE 42010, Systems and software engineering — Architecture description

[5] ISO/IEC 10746-2, Information technology — Open distributed processing — Reference model: Foundations — Part 2

[6] ISO/IEC JTC 1/SC 38 N0043, Research Report on China's SOA Standards System

[7] ISO/IEC JTC 1/SC 38 N0022, Chinese National Body Contribution on Proposed NP for General Technical Requirement of Service Oriented Architecture

[8] OASIS. Reference Model for SOA, Version 1.0, OASIS Standard, October 2006: Available from World Wide Web: gost.e-ecolog.ru

[9] ISO/IEC 18384-3, Information technology — Reference Architecture for Service Oriented Architecture (SOA RA) — Part 3: Service Oriented Architecture Ontology

[10] ISO/IEC 17998, Information technology — SOA Governance Framework

[11] ISO/IEC 15474-1, Information technology — CDIF framework — Part 1: Overview

[12] ISO/IEC 16500-8, Information technology — Generic digital audio-visual systems — Part 8: Management architecture and protocols

[13] ISO/IEC 38500:2008, Information technology — Governance of IT for the organization[1)]

[14] OMG. Business Process Management Notation (BPMN), see gost.e-ecolog.ru

[15] ISO/TR 9007, Information processing systems — Concepts and terminology for the conceptual schema and the information base

[16] The Open Group Architecture Framework (TOGAF), section 8.1.1 Version 9 Enterprise Edition, February 2009; see gost.e-ecolog.ru

[17] OASIS. Reference Architecture for SOA Foundation, Version 1.0, OASIS Committee Specification, 4 December 2012: see gost.e-ecolog.ru

[18] W3C Web Services Description Language (WSDL) 1.1, W3C Note 15 March 2001, see gost.e-ecolog.ru

[19] OASIS. Web Services for Remote Portlets Specification v2.0 OASIS Standard, 1 April 2008 (WSRP), see gost.e-ecolog.ru

[20] OMG. Model Driven Architecture (MDA) Guide, Version 1.0.1, Object Management Group (OMG), June 2003: see gost.e-ecolog.ru .pdf

[21] OMG. Unified Modeling Language (OMG UML), Superstructure, Version 2.2, OMG Doc. No.: formal/2009-02-02, Object Management Group (OMG), February 2009: see gost.e-ecolog.ru

[22] OMG. SOA Modeling Language (OMG SoaML) Specification for the UML Profile and Metamodel for Services (UPMS), Revised Submission, OMG Doc. No.: ad/2008-11-01, Object Management Group (OMG), November 2008: see gost.e-ecolog.ru

[23] W3C Web Ontology Language (OWL), World Wide Web Consortium (W3C), April 2009: see gost.e-ecolog.ru

[24] Garrett J.J. A New Approach to Web Applications, Feb 18, 2005 see gost.e-ecolog.ru

[25] OASIS Web Services Coordination (WS-Coordination) Version 1.2, OASIS Standard, Feb 2, 2009, gost.e-ecolog.ru

[26] Web Services Atomic Transaction (WS-Atomic Transaction) Versions 1.2 OASIS Standard, Feb 2, 2009, gost.e-ecolog.ru

[27] Web Services Business Activity (WS-Business Activity) Version 1.2 OASIS Standard, Feb 2, 2009, gost.e-ecolog.ru

[28] Web Services Security (WS-Security) Version 1.1 OASIS Standard, Feb 1, 2006, gost.e-ecolog.ru /

[1)] Заменен на ISO/IEC 38500:2015.

[29] Web Services Federation (WS-Federation) Version 1.2 OASIS Standard, May 22, 2009,
gost.e-ecolog.ru
[30] ISO/IEC 20926, Software and systems engineering — Software measurement — IFPUG functional size measurement method 2009

[31] ISO/IEC 19505-2, Information technology — Object Management Group Unified Modeling Language (OMG UML) — Part 2: Superstructure

УДК 004:006.354 ОКС 35.100.05

Ключевые слова: информационные технологии, эталонная архитектура, сервис-ориентированная архитектура (SOA), структура SOA, терминология, решения SOA, управление SOA, безопасность SOA

БЗ 8—2017/26

Редактор *Л.И. Нахимова*
Технический редактор *В.Н. Прусакова*
Корректор *Е.Д. Дульнева*
Компьютерная верстка *А.Н. Золотаревой*

Сдано в набор 07.09.2017. Подписано в печать 03.10.2017. Формат 60 × 84 1/8. Гарнитура Ариал.
Усл. печ. л. 5,58. Уч.-изд. л. 5,05. Тираж 22 экз. Зак. 1675.
Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

Издано и отпечатано во ФГУП «СТАНДАРТИНФОРМ», 123001 Москва, Гранатный пер., 4.

gost.e-ecolog.ru gost.e-ecolog.ru

gost.e-ecolog.ru