

The analysis of service oriented architecture methodologies (SOA)

Yuriy Gik

Roseurobank

gikjuri@yandex.ru

Abstract

Service oriented architecture is one of the most interesting architecture paradigms for recent time in the IT industry. In spite of 10 years long history and plenty of implemented for that time projects, theory of SOA has no unity. Several vendors are promoting own methodologies, common understanding of base terms is absent and so on. There is obvious task of systematization of existing methodologies. This paper deals with analysis and systematization of actual SOA methodologies.

Keywords: Enterprise architecture, SOA

Анализ методологий сервисно-ориентированной архитектуры (COA)

Юрий Гик

Росевробанк

gikjuri@yandex.ru

Аннотация

Сервисно-ориентированная архитектура является одной из наиболее интересных парадигм, рожденных в последнее время в IT отрасли. Несмотря на более чем 10-летнюю историю и обилие внедренных проектов, в теории COA нет единства. Многие вендоры развивают собственные методологии, отсутствует общее понимание базовых терминов. Назрела задача систематизации существующих методологий. Работа посвящена анализу и систематизации существующих методологий COA.

Ключевые слова: архитектура предприятия, COA

I. ВВЕДЕНИЕ

Несмотря на многолетнюю историю существования COA и обилие внедренных в соответствии с этой архитектурой масштабных проектов, отрасль до сих пор не имеет единой общепризнанной методологии COA. Фактически каждый достаточно большой вендор имеет собственную методологию COA или, что еще хуже, набор собственных методологий. COA по своей природе независима от конкретных вендоров и ориентирована на решения, где используются технологии разных вендоров, и подобный разноречивый в методологиях ведет в конечном

итоге к снижению качества внедряемых в архитектуре COA проектов.

Наличие разнообразных методологий выгодно прежде всего самым крупным вендорам, которые, пользуясь отсутствием единой теоретической базы, подгоняют собственные методологии COA под свои продуктовые линейки. Если теория имеет размытое содержание, которое, в общем-то, невозможно проверить, например, по международным стандартам, то вендор не упускает возможности подстроить эту теорию под свои маркетинговые потребности.

Основными владельцами и разработчиками методологий COA являются IBM (в блоке с Open Group) и компания Arcitura, бывшая SOA School. Другие крупные вендоры (Oracle, Microsoft, Red Hat, Progress Software) имеют собственные не столь проработанные методологии, использующие так или иначе идеи IBM и/или Arcitura. В последнее время налицо центробежные тенденции, т.е. вендоры склонны более прорабатывать собственные методологии COA, чем договариваться о единой стандартной методологии.

Целью этой работы является анализ имеющихся методологий COA с учетом их специфических деталей и попытка систематизации методологий COA. Источниками информации являются прежде всего открытые ресурсы в интернете на сайтах вендоров, многие вендоры имеют отдельные большие разделы, посвященные COA. Методология Arcitura изложена по книгам Томаса Эрля. Методология Oracle изложена по соответствующим книгам, рекомендованным компанией Oracle.

II. КРАТКАЯ ИСТОРИЯ COA

COA не является революционной идеей. Концепция COA является эволюционным развитием множества теорий и концепций, имеющих отношение к распределенным компьютерным вычислениям и интеграции: от объектно-ориентированной парадигмы до концепций CORBA и DCOM.

Термин «COA» впервые появился в статье вице-президентов «Гартнер» Роя Шульте и Ефима Натиса «Сервис-ориентированные архитектуры. Часть 1» в 1996 году. Статья содержала определение COA как стиля многозвенных вычислений и описание преимуществ сервис-ориентированных конфигураций.

Практическое развитие COA получил в начале 2000-х, с появлением и популярностью веб-сервисов. Веб-сервисы хорошо подходили для реализации теоретических идей COA и поддерживались всеми основными вендорами на уровне международных стандартов. Де-факто, веб-сервисы являются основной реализацией COA – хотя сейчас они испытывают конкуренцию со стороны REST-сервисов.

Через несколько лет появились первые стандарты и методологии. В 2004 году Али Арсаньяни в IBM написал статью, положившую начало SOMA – сервис-ориентированному моделированию и анализу. В 2006 году группа OASIS опубликовала «Справочную модель сервис-ориентированной архитектуры», содержащую основные определения COA.

В это же время получают развитие концепции интеграционной шины (ESB), управления бизнес-процессами (BPM), второго поколения веб-сервисов (стандарты WS-*) – эти концепции начинают играть важную роль в COA и находят свое отражение в методологиях вендоров. В 2007 году Томас Эрль издает книгу «Шаблоны проектирования COA», взятую на вооружение основными вендорами. В это время COA становится мейнстримом, в крупных компаниях создаются COA-подразделения, внедряется множество проектов. Вендоры создают специализированные пакеты для

внедрения COA (SOA Suites). Возникает система сертификации COA-специалистов.

Сейчас тема COA утратила ореол модности и популярности и перешла в режим спокойного функционирования и эволюционного развития. Объединение концепции COA и идей событийного программирования (SEP) получило развитие в COA 2.0. Идеи COA используются в популярных в наше время облачных технологиях.

III. ПОДХОД ARCITURA

Компания Arcitura (бывшая SOA School) работает в области образования и теории COA, управления бизнес-процессами (BPM) и облачных технологий со второй половины 1990-х годов. Ее лидер Томас Эрль является автором книг по COA и смежным направлениям, а также является признанным в отрасли авторитетом, консультирующим многие крупнейшие корпорации: от Пентагона до российского Сбербанка. Методология COA компании Arcitura является одной из наиболее проработанных и популярных в отрасли.

A. Основы COA

Arcitura (бывшая SOA School) определяет сервис как единицу прикладной логики с сервисной ориентацией¹. Сервисная ориентация – это парадигма проектирования, состоящая из набора принципов дизайна². Таким образом, сервис определяется как прикладная сущность, удовлетворяющая конкретному набору принципов дизайна:

- стандартизированный контракт;
- слабая связь;
- абстракция;
- повторное использование;
- автономия;
- отсутствие сохранения контекста;
- понятность (discoverability)
- способность к композиции³

Сервис-ориентированная архитектура определяется как архитектурная модель, состоящая из сервисов⁴.

Теория COA у Arcitura, помимо общих принципов, состоит из набора важных для COA концептов, развитой системы шаблонов проектирования и организационных аспектов методологии построения корпоративной COA.

¹ Thomas Erl. SOA Design Patterns. Upper Saddle River, NJ: Prentice Hall, 2009, p.37

² Thomas Erl. SOA Principles of Service Design. Upper Saddle River, NJ: Prentice Hall, 2008, p.39

³ Thomas Erl. SOA Principles of Service Design. Upper Saddle River, NJ: Prentice Hall, 2008, p.71-74

⁴ Thomas Erl. SOA Design Patterns. Upper Saddle River, NJ: Prentice Hall, 2009, p 37

В. Концепты SOA

Сервисная композиция – это согласованная совокупность сервисов⁵. Сервисы в композиции могут занимать роли контроллера, подконтроллера или члена композиции. В теории говорится мало о возможной прикладной реализации сервисных композиций. Но в прикладных книгах Эрля (“SOA with .NET”, “SOA with Java”), служащих дополнением к основной методологии, достаточно большое внимание уделено вопросам интеграционных шин (ESB) и бизнес-процессов (BPM). Поэтому реализацией сервисных композиций являются прежде всего процессы BPM и ESB.

Хранилище сервисов – это коллекция независимых, стандартизированных и управляемых сервисов, ограниченных функциональностью предприятия или его части⁶. Хранилища могут быть разных уровней: уровня предприятия или его домена (части предприятия).

Arcitura выделяет следующую классификацию сервисов (сервисные модели):

- сервисы сущностей
- сервисы задач
- утилитарные сервисы⁷

В теории подчеркивается, что SOA не зависит от конкретной прикладной платформы и технологии. Однако большая часть материала ссылается на веб-сервисы, и эта технология воспринимается как основная, на базе которой можно строить SOA масштаба предприятия. У Томаса Эрля издана также отдельная книга, посвященная построению SOA с помощью REST-сервисов.

Интеграционные шины (ESB) упоминаются в теории в двух ролях: как предтеча и источник идей для SOA, а также как шаблон проектирования. В шаблоне ESB представлены основные компоненты шины и их роли. Шаблон представляет абстрактную интеграционную шину с минимальной функциональностью. В практических руководствах, описывающих построение SOA на Java и .NET, акцентируется возможность построения SOA без шин и приводятся примеры сервисных композиций на базе интеграционных шин.

Аналогичную ESB роль имеет в теории SOA и теория управления бизнес-процессами (BPM), с теми же двумя ролями – предтечи-источника идей и практической реализации сервисных композиций.

SOA как архитектура обладает следующими преимуществами⁸ по сравнению с предшествующими типами архитектур:

- тесные внутренние взаимодействия сервисов

⁵ Thomas Erl. SOA Principles of Service Design. Upper Saddle River, NJ: Prentice Hall, 2008, p.39

⁶ Thomas Erl. SOA Design Patterns. Upper Saddle River, NJ: Prentice Hall, 2009, p 40

⁷ Thomas Erl. SOA Design Patterns. Upper Saddle River, NJ: Prentice Hall, 2009, p 43

⁸ Thomas Erl. SOA Principles of Service Design. Upper Saddle River, NJ: Prentice Hall, 2008, p.55

- объединение (федерация) сервисов в рамках организации

- диверсификация вендоров

- лучшее согласование работ подразделений бизнеса и ИТ

- увеличение отдачи от инвестиций (ROI)

- повышение гибкости организации

- сокращение нагрузки на ИТ-подразделения

С. Шаблоны проектирования

Arcitura обладает развитым набором шаблонов проектирования SOA по следующим направлениям:

- Хранилища сервисов

- Фундаментальные шаблоны

- Шаблоны реализации сервисов

- Шаблоны сервисов безопасности

- Шаблоны проектирования сервисных контрактов

- Шаблоны связи с унаследованными системами

- Шаблоны управления SOA

- Шаблоны композиции сервисов

- Составные шаблоны

Шаблоны SOA являются наиболее популярной частью теории SOA компании Arcitura, и ни в одной другой методологии не встречается столь проработанной коллекции шаблонов. Шаблоны охватывают весь цикл жизни сервисов: от первичного проектирования (фундаментальные шаблоны и, в частности, принцип функциональной декомпозиции) через проектирование (например, каноническая схема) и разработку (например, обратный вызов сервиса) к сопровождению и организационным вопросам практического построения SOA в организации.

Д. Организационные аспекты

Управление SOA в теории Arcitura делится на две части:

- Управление SOA-проектами

- Стратегическое управление SOA

В первой части описывается цикл жизни сервисов (аналогично моделям зрелости методологий-конкурентов): от анализа и дизайна через разработку и тестирование к внедрению и эксплуатации. Детально излагаются вопросы версий сервисов и особенностей проектных команд и ролей в SOA-проектах.

Во второй части излагаются вопросы политик, накладываемых на сервисы, актуализации сервисов, а также рекомендуются технологии и инструменты для управления SOA.

Деятельность компании Arcitura не ограничивается только вопросами теории SOA. Компания ведет

обширную образовательную программу. Доступна богатая сеть сертификаций по COA – от профессионала COA до архитектора и аналитика COA. Arcitura инициировала создание COA симпозиума – площадки для выработки решений, касающихся COA, где участвуют все основные вендоры, предлагающие решения по COA.

IV. Подход OASIS

OASIS – это некоммерческий консорциум, созданный для продвижения открытых стандартов. Консорциум основан группой компаний, в число которых входят IBM и Microsoft. Спонсорами OASIS являются Oracle, Red Hat и другие крупные вендоры. Методология COA изложена, в основном, в документе «Эталонная архитектура COA»⁹, имеющем статус стандарта. Методология OASIS оказала большое влияние на современную методологию COA компании IBM.

A. Основы COA

Методология OASIS описывает архитектуру, принципиально независимую от конкретных технологий и продуктов, т.е. обладает высокой степенью абстракции.

COA понимается как экосистема, т.е. место, где люди, машины и сервисы взаимодействуют друг с другом. Отметим, что такое понимание принципиально отличается от понятия COA у компании Arcitura, толкующей только о сервисах. В экосистеме отсутствует иерархия, т.е. каждый член экосистемы (человек, машина или сервис) является равным участником.

Ключевыми принципами экосистемы COA являются:

- COA – это среда обмена благами между независимо действующими участниками;
- участники (а также заинтересованные стороны) имеют одинаковые права претендовать на собственность ресурсов, доступных с помощью COA;
- поведение и производительность участников определяется правилами, зафиксированными в политиках и контрактах.

Целями COA являются:

- эффективность;
- конфиденциальность
- масштабируемость

B. Принципы COA:

- нейтральность по отношению к технологии;
- экономия (избегание сложностей в дизайне)
- разделение интересов
- применимость

⁹ OASIS Reference Architecture for Service Oriented Architecture, <http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/csd03/soa-ra-v1.0-csd03.pdf>

Отметим кардинальное отличие этих принципов от соответствующих принципов у Arcitura. Это объясняется более широким определением COA как экосистемы.

C. Модели COA

В методологии описывается большое количество моделей по следующим срезам:

- сервисы с точки зрения бизнеса
 - реализация COA
 - управление COA
- Список основных описываемых моделей:
- Модель заинтересованных сторон и участников
 - Модель ресурсов
 - Модель прав собственности
 - Модель необходимостей и возможностей
 - Модель социальной структуры
 - Действия в социальном контексте
 - Роли в социальных структурах
 - Управление социальными структурами
 - Модель предложения
 - Модель описания сервисов
 - Модель видимости сервиса
 - Модель взаимодействия с сервисом
 - Модель политик и контрактов
 - Модель управления
 - Модель безопасности
 - Модель сервисов как управляемых сущностей

Если сравнивать с методологией Arcitura, мы увидим, что большая часть этих моделей не имеет аналогов у Arcitura вследствие различных объектов анализа. Некоторые модели OASIS имеют похожие механизмы у Arcitura в виде принципов (модель исполнения контекстов у OASIS и принцип понятности у Arcitura) или шаблонов.

Уже на примере сравнения методологий Arcitura и OASIS мы видим, насколько сильно они отличаются: на уровне принципов, сущностей, а также самих предметов анализа.

V. ПОДХОД THE OPEN GROUP

The Open Group – это глобальный консорциум, конкурент OASIS, занимающийся вопросами продвижения IT-стандартов. Платиновыми членами консорциума являются IBM и Oracle. Членами консорциума являются фактически все значимые IT-компании. Методология COA компании The Open Group изложена в книге «SOA Source Book», опубликованной в свободном доступе на своем сайте. Теория COA компании The Open Group является основой современной методологии компании IBM.

Методология SOA компании The Open Group является прямым конкурентом методологии Arcitura и претендует на роль ведущей методологии SOA.

Методология SOA компании The Open Group состоит из основных и дополнительных понятий SOA (именуемых строительными блоками SOA), инфраструктуры SOA, модели зрелости сервисов и SOA-онтологии.

А. Основы SOA

Сервис – это логическое представление повторяемого бизнес-процесса¹⁰. Сервис самодостаточен. Сервис может состоять из других сервисов. Сервис является черным ящиком для своих потребителей.

Сравнивая с соответствующим определением Arcitura, мы видим, что определение сервиса The Open Group является упрощением по отношению к Arcitura. Последняя добавляет в определение сервиса принципы стандартизованного контракта, слабой связи, отсутствия сохранения контекста и понятности. Другие принципы Arcitura (автономия, повторное использование, абстракция, способность к композиции) присутствуют в определении The Open Group. Справедливости ради отметим, что принципы SOA у Arcitura проработаны гораздо глубже, чем у The Open Group.

Сервис-ориентированная архитектура у The Open Group определяется как архитектурный стиль, который поддерживает сервисную ориентацию. Сервисная ориентация – это стиль мышления в терминах сервисов, сервисной разработки и сервисных результатов¹¹.

Архитектурный стиль SOA у The Open Group обладает следующими свойствами:

- Стиль строится на дизайне сервисов, отражающих реальную бизнес-активность
- Сервисы подчиняются бизнес-правилам. Для сервисных композиций применяется оркестровка сервисов
- В инфраструктуре SOA рекомендуется использовать открытые стандарты для повышения уровня взаимодействия и прозрачности
- Реализации сервисов зависят от технического окружения. Они должны определяться контекстом и описываться внутри этих контекстов.
- Стиль требует строгого управления описанием и реализацией сервисов
- Стиль требует «лакмусовых тестов», определяющих качество сервисов

Arcitura определяет сервисную ориентацию как парадигму, состоящую из 8 принципов сервисного дизайна. Arcitura не содержит на уровне базовых принципов рекомендаций применять конкретные технологии (и открытые стандарты рассматриваются наряду с закрытыми технологиями вендоров). Также

¹⁰ The Open Group. SOA Source Book. What is SOA? <http://www.opengroup.org/soa/source-book/soa/soa.htm>

¹¹ The Open Group. SOA Source Book. What is SOA? <http://www.opengroup.org/soa/source-book/soa/soa.htm>

Arcitura настаивает на построении сервисов, независимых от конкретных технологий и технических окружений. Эти изменения в подходах Arcitura и The Open Group принципиальны.

Сервис в методологии The Open Group имеет провайдера, потребителей и производит результат, необходимый потребителям.

В. Основные понятия SOA

Бизнес-процесс – деятельность бизнеса предприятия в соответствии со своей миссией, осуществляемая определенным повторяемым способом. Отметим, что в методологии Arcitura нет понятия бизнес-процесса.

Эктор (Actor) – сущность (человек или программа), производящая какие-либо действия. Нет аналогов у Arcitura

Событие – то, что может произойти. Необязательно связано с бизнес-процессами. Нет аналогов у Arcitura. Отсутствие аналогов этих терминов, присущих управлению бизнес-процессов, у Arcitura служит хорошей иллюстрацией разных картин мира у Arcitura и The Open Group. Arcitura пытается выделить чистую SOA, вне теорий BPM и ESB. The Open Group включает элементы теории BPM в свою методологию SOA

Сервисное описание – это описание сервиса словами и диаграммами. Аналогом у Arcitura служит принцип понятности сервиса.

Сервисный контракт – это соглашение между провайдером сервиса и его клиентами. Принцип сервисного контракта у Arcitura является одним из базовых и проработан намного лучше.

Сервисная политика – набор действий, удовлетворяющих требованиям провайдеров и потребителей. У Arcitura политики сервисов определяются как части контрактов.

Сервисная композиция – коллекция совмещенных сервисов, действующих как один сервис. У Arcitura принцип сервисной композиции является базовым и, соответственно, более глубоко проработанным.

Программа – набор компьютерных инструкций для решения определенной задачи. У Arcitura нет аналогов.

Информационный элемент – сущность, обладающая знанием о чем-либо. У Arcitura нет аналогов.

Элемент данных – представление информационного элемента. У Arcitura нет аналогов.

Хранилище данных – технологичный эктор, хранящий элементы данных.

Обмен сообщениями – программы могут обмениваться сообщениями с помощью сервисов обмена сообщениями. У Arcitura есть соответствующий шаблон проектирования.

Исследование сервисов – способ получения информации о сервисе. Алгоритм включает исследование хранилища сервисов. У Arcitura есть аналогичный базовый принцип понятности сервисов и описывается похожий алгоритм.

Обертка тела сервиса – принцип взаимодействия с сервисом через фасад. У Arcitura соответствует базовому принципу сервисного контракта, детально проработанному.

Виртуализация сервиса – функциональность сервиса определяется контрактом, реализация тела сервиса скрывается. У Arcitura есть соответствующий базовый принцип абстракции сервиса.

Обработка событий – поток событий обрабатывается специализированным событийным процессором. Нет аналогов у Arcitura.

C. Управление COA

Управление COA – это организация и контроль людей на предприятии, деятельность которых связана с разработкой, внедрением и применением COA.

Управление COA подчиняется следующим принципам:

Деятельность по управлению COA должна подчиняться корпоративным, IT и архитектурным стандартам предприятия

COA-сервисы и решения должны удовлетворять архитектуре предприятия

Существующие сервисы должны учитываться в первую очередь при построении COA-решений

Между провайдерами и потребителями сервисов должны существовать контракты

Процессы управления COA должны учитывать уровни и риски проектов

Разработка и внедрение COA регулируются следующими документами:

- Управление решениями
- Управление сервисами
- Управление жизненными циклами сервисов
- Управление жизненными циклами решений

Управление COA содержит строгий набор ролей и должностных обязанностей, например собственник домена, бизнес-аналитик, старший архитектор и т.п.

Отметим, что Arcitura имеет набор соответствующих ролей, но ориентированных более на специфику разработки и сопровождения сервисов.

Управление COA содержит собственный метод, содержащий 4 фазы: планирование, определение, внедрение и мониторинг. Этот метод является основным при внедрении COA на предприятии.

The Open Group рекомендует использовать методологию TOGAF для построения COA, где определены подробно необходимые шаги и расписаны соответствующие модели дизайна.

D. Модель зрелости (OSIMM)

Модель определяет уровень зрелости COA на предприятии по следующим атрибутам:

- Бизнес
 - Организация
 - Методы
 - Приложения
 - Архитектура
 - Информация
 - Инфраструктура
- Возможны следующие уровни зрелости:
- Силос
 - Интегрированный
 - Уровень компонент
 - Уровень сервисов
 - Уровень составных сервисов
 - Уровень виртуализируемых сервисов
 - Уровень динамически настраиваемых сервисов

Аналогичная модель зрелости Arcitura имеет существенные идеологические отличия.

E. COA-онтология

COA-онтология является технологическим стандартом, описывающем в графическом и текстовом виде все элементы COA и их взаимосвязи.

В онтологии даются основные определения COA и детально описаны все основные понятия COA, взаимоотношения с другими элементами, а также их свойства. Для описания используется UML-нотация, основные понятия рассматриваются как классы, взаимосвязи между ними рассматриваются как связи между классами. Для свойств построены xml-представления. Приводятся примеры использования классов.

Отметим идеологическое отличие у Arcitura. В ее методологии не применяется UML, поскольку COA рассматривается как надмножество над объектно-ориентированным программированием, и поэтому в методологии применяется собственная графическая нотация.

Итак, мы видим, что методологии COA компаний Arcitura и The Open Group описывают примерно одну область (в отличие от методологии OASIS). Однако в деталях эти методологии существенно отличаются. Иногда эти отличия касаются степени проработки материала, но многие отличия носят принципиальный идеологический характер – вопрос нотации описания сервисов, отношение к управлению бизнес-процессами и т.п.

VI. Подход IBM

Подход IBM к методологии COA чрезвычайно важен, поскольку компания стояла у истоков COA и многое определяет в отрасли. Поэтому мы рассмотрим эволюцию

методологий IBM. Здесь мы можем выделить три этапа: ранний (COMA), зрелый (Smart SOA) и текущий.

A. Ранний этап - COMA

Сервис-ориентированное моделирование и архитектура (COMA) – одна из первых методологий COA. COMA – это метод с ролями и последовательностями действий по производству артефактов, относящихся к идентификации, спецификации и реализации сервисных компонентов и процессов. Анализ и моделирование с помощью COMA являются технологически независимыми, но устанавливают контекст для применения определенных технологий на поздних стадиях жизненного цикла решения. Существует компонент COMA для унифицированного процесса разработки Rational (RUP)¹²

Идеи COMA изложил впервые Али Арсаньяни в статье «Сервис-ориентированное моделирование и архитектура» 2004 года. В статье обосновывается недостаточность объектно-ориентированного дизайна для проектирования сервисов. «Во-первых: текущие методы OOAD не обращаются к трем ключевым элементам SOA – сервисам, потокам и компонентам, реализующим сервисы. Вам необходимо также иметь возможность напрямую обращаться к методикам и процессам, необходимым для идентификации, определения и реализации сервисов, их потоков и композиции.»¹³

Сервис определяется как имеющий три стороны: поставщик, потребитель и посредник. Посредник определяется как специфический компонент для обслуживания реестра сервиса.

В статье выделяются такие свойства сервиса как наличие слабой связи, способность к масштабированию и формированию композиций, возможность вызова через стандартные протоколы и наличие описания. В целом, эти свойства близки более поздним определениям Arcitura и The Open Group.

Для составных сервисов рекомендуется использовать хореографию сервисов и возможности интеграционной шины (ESB). Отметим важную роль интеграционной шины, возникшую еще в первых методологиях у IBM.

Архитектура COA делится на 7 уровней:

- Уровень операционной системы
- Уровень корпоративных компонентов
- Уровень сервисов
- Бизнес-процесс и составной уровень
- Уровень обеспечения доступа или презентации
- Уровень интеграции

¹² Bobby Woolf. Exploring IBM SMART SOA Technology & Practice, 605 Silverthorn Road Gulf Breeze, FL: Maximum Press, 2008, p.54

¹³ Ali Arsanjani. Сервис-ориентированное моделирование и архитектура, 2004, <http://www.ibm.com/developerworks/ru/library/ws-soa-design1>

Уровни описывают статичные архитектурные срезы COA, к каждому уровню рекомендуется разрабатывать свои пакеты документов. Отметим интеграцию на высшем уровне абстракции COA, т.е. COA обеспечивает прежде всего задачу интеграции, и здесь мы видим важное идеологическое расхождение как с видением Arcitura, так и с более поздними методологиями IBM.

В COMA приводится описание процесса моделирования COA, состоящее из шагов:

- Идентификация сервисов
- Спецификация сервисов
- Реализация сервисов, компонентов и потоков.

B. Smart SOA

Методология Smart SOA изложена в книге «Exploring IBM. Smart SOA Technology & Practice» (2008).

COA определяется как архитектурный стиль для создания архитектуры уровня предприятия на принципах сервисной ориентации с целью достичь более глубокого взаимодействия между бизнесом и ИТ-системами, поддерживающими бизнес. Сервисная ориентация – это принцип представления бизнеса как набора сервисов. Сервис определяется как повторяющаяся задача внутри бизнес-процесса. Сервисы являются самоописываемыми объектами, удовлетворяют требованиям качества сервиса, являются управляемыми структурами и могут объединяться в составные сервисы¹⁴.

Smart SOA строится на следующих концептуальных документах:

- Первые шаги COA
 - Модель зрелости сервисов интеграции (SIMM)
 - Сервис-ориентированное моделирование и архитектура (COMA)
 - Моделирование компонент бизнеса (CBM)
 - Метод управления COA
- «Первые шаги COA» определяют абстракции COA высокого уровня.
- Люди
 - Процессы
 - Информация
 - Повторное использование
 - Доступность

Эта часть методологии является развитием методологии OASIS Group.

Модель зрелости сервисов интеграции – это модель зрелости консорциума The Open Group, первоначально разработанная инженерами IBM

¹⁴ Bobby Woolf. Exploring IBM SMART SOA Technology & Practice, 605 Silverthorn Road Gulf Breeze, FL: Maximum Press, 2008, p.25

Моделирование компонент бизнеса – способ моделирования и анализа предприятия. Представляет собой карту областей бизнеса с определенными характеристиками, возможностями и процессами. Выделяются три уровня абстракции предприятия:

- Стратегический
- Управляющий
- Исполнимый

Метод управления COA включает стратегические вопросы управления COA: собственники сервисов, модели обмена сообщениями, вопросы безопасности сервисов, вопросы версионности, тестирования и мониторинга сервисов. Управление COA в принципе соответствует соответствующим идеям компаний Arcitura и The Open Group

Smart SOA предлагает для реализации COA набор IBM-продуктов из линеек Rational, WebSphere и Tivoli.

C. Современная ситуация

Современное понимание COA у IBM строится на развитии методологии The Open Group, в разработке которой IBM принимал важное участие. IBM берет эту методологию как основную и под эту методологию предлагает свои продукты по построению COA. IBM имеет подробные руководства по внедрению методологии The Open Group, а также карты соответствий принципов методологии возможностям своих продуктов.

IBM базирует современную методологию COA на следующих стандартах The Open Group:

- Модель зрелости открытых сервисов интеграции (OSIMM)
- Стандартная среда управления COA (SGMM)
- Стандарт COA онтологии
- Стандарт эталонной архитектуры COA

Отметим наличие у IBM шаблонов COA. Этот набор шаблонов не так богат как у Arcitura и частично перекрывается с ним по функциональности.

VII. ПОДХОД ORACLE

Методология COA компании Oracle не отличается строгостью и цельностью в сравнении с методологией IBM. В методологии используются идеи как Arcitura, так и IBM.

A. Основы COA

Сервис – это термин, который понимают и бизнес-подразделения и IT-подразделения. Сервис обладает следующими свойствами¹⁵:

- Инкапсуляция – сервис создает разграничение между своими провайдером и потребителем

¹⁵ Antony Reynolds, Matt Wright. Oracle SOA Suite 11g R1 Developer's Guide, 32 Lincoln Road, Olton, Birmingham, UK: Packt Publishing, 2010, p.12

- Интерфейс, определяемый в терминах входных и выходных параметров

- Контракт или соглашение об уровне сервиса (SLA) – атрибуты, описывающие качество сервиса, производительность, доступность и стоимость

Сервис – это объединение (инкапсуляция) данных и бизнес-логики. Сервис состоит из интерфейса, имеет реализацию и определенное поведение. Интерфейс определяет набор операций, обеспечивающих возможности сервиса. Провайдер сервиса обеспечивает контракты на основе интерфейса, потребители сервиса взаимодействуют с сервисом посредством контракта. Функциональности сервисов могут быть выделены из разных источников: баз данных, унаследованных приложений, программ на разных языках программирования¹⁶. Отметим сходство определений с соответствующими идеями Arcitura и расхождение с ними в деталях – у Arcitura не выделяются понятия интерфейса и операций, они являются атрибутами контракта.

COA не зависит от применяемой технологии. Однако COA ассоциируется с web-сервисами и связанными с ним стандартами OASIS GROUP¹⁷.

В построении архитектуры Oracle ссылается на документы The Open Group (например, SOA Reference Architecture¹⁸). Классификация сервисов на сервисы сущностей и сервисы задач берется из методологии Arcitura¹⁹

Для выполнения своих задач сервисы должны удовлетворять следующим условиям²⁰:

- Слабая связь
- Понятность
- Прозрачность обнаружения – т.е. возможность исполнения сервиса из любого места
- Автономия
- Управление состоянием – желательно не хранить состояния, но иногда этого невозможно избежать

¹⁶ Heidi Buelow, Manas Deb, Jayaram Kasi, Demed L'Her, Prasen Palvankar. Getting Started with Oracle SOA Suite 11g R – a hands-on tutorial, 32 Lincoln Road, Olton, Birmingham, UK: Packt Publishing, 2009, p. 10

¹⁷ Antony Reynolds, Matt Wright. Oracle SOA Suite 11g R1 Developer's Guide, 32 Lincoln Road, Olton, Birmingham, UK: Packt Publishing, 2010, p.14

¹⁸ Heidi Buelow, Manas Deb, Jayaram Kasi, Demed L'Her, Prasen Palvankar. Getting Started with Oracle SOA Suite 11g R – a hands-on tutorial, 32 Lincoln Road, Olton, Birmingham, UK: Packt Publishing, 2009, p.10

¹⁹ Heidi Buelow, Manas Deb, Jayaram Kasi, Demed L'Her, Prasen Palvankar. Getting Started with Oracle SOA Suite 11g R – a hands-on tutorial, 32 Lincoln Road, Olton, Birmingham, UK: Packt Publishing, 2009, p.11

²⁰ Heidi Buelow, Manas Deb, Jayaram Kasi, Demed L'Her, Prasen Palvankar. Getting Started with Oracle SOA Suite 11g R – a hands-on tutorial, 32 Lincoln Road, Olton, Birmingham, UK: Packt Publishing, 2009, pp.12-13

- Повторное использование
- Способность к композиции

Здесь мы видим (слегка обогащенные) принципы SOA компании Arcitura

Также в методологии упоминаются шаблоны проектирования SOA-решений от компании IBM:

- Бизнес-шаблоны
- Шаблоны интеграции
- Композитные шаблоны
- Шаблоны приложения
- Шаблоны периода исполнения

Методология SOA компании Oracle служит теоретической частью продукта Oracle Fusion Middleware.

VIII. ПОДХОД MICROSOFT

Microsoft имеет собственную методологию SOA, являющуюся теоретической основой для решений на основе продуктов своей линейки (WCF, WPF, .NET)

A. Основы SOA

Сервис – компонент, способный выполнить свою задачу. Возможности сервиса описываются WSDL.

Определение сервиса – сущность, с помощью которой удовлетворяются потребности клиентов сервиса в соответствии с контрактом.

Выполнение сервиса – осуществление возможностей экземпляра сервиса

Сервис-ориентированная архитектура – набор исполнимых компонентов с опубликованным описанием интерфейсов

Сервис должен удовлетворять следующим принципам дизайна²¹:

- Нейтральность по отношению к технологии
- Использование стандартных протоколов
- Понятность
- Повторное использование
- Абстракция реализации
- Опубликованные интерфейсы
- Наличие формального контракта
- Релевантность уровня гранулярности

Отметим сходство этих принципов с принципами дизайна SOA компании Arcitura.

К ключевым компонентам SOA относятся²²:

- Сервис

- Сообщение
- Динамичное обнаружение
- Масштабируемость сервисов
- Веб-сервис

Отметим наличие технологической привязки к веб-сервисам на уровне ключевых понятий методологии.

IX. ПОДХОД RED HAT

Red Hat имеет линейку продуктов JBoss, в т.ч. SOA Platform. Как теоретическую основу последнего продукта, Red Hat имеет собственную методологию SOA.

A. Основы SOA

Сервис-ориентированная архитектура – это не программа и не технология. Это парадигма дизайна программного обеспечения. Сервисы не обязательно должны являться веб-сервисами. Сущности, представленные протоколами FTP или JMS, также могут являться сервисами²³

SOA – это архитектурный стиль, имеющий целью достижение слабой связи между взаимодействующими программными агентами.

Сервис – это единица работы провайдера сервиса для осуществления результата, необходимого клиенту. Провайдер и клиент – это роли, которые берут на себя программные агенты²⁴. Отметим наличие концепта «программный агент», отсутствующего в других методологиях.

SOA имеет три роли:

- Провайдер сервиса
- Сущность, запрашивающая сервис
- Сервисный брокер, отвечающий за связь двух предыдущих ролей

Отметим ориентацию методологии на интеграционную шину, что является важным идеологическим отличием от других методологий SOA (у Arcitura интеграционная шина – это лишь один из архитектурных шаблонов).

Отметим также, что во многих публичных выступлениях, посвященных SOA, Red Hat ссылается на методологию Arcitura²⁵.

²² Service Oriented Architecture, <http://msdn.microsoft.com/en-au/architecture/aa948857>

²³ JBoss Enterprise SOA Platform 5 ESB Services Guide, http://docs.redhat.com/docs/en-US/JBoss_Enterprise_SOA_Platform/5/pdf/ESB_Services_Guide/JBoss_Enterprise_SOA_Platform-5-ESB_Services_Guide-en-US.pdf

²⁴ JBoss ESB. Basics of SOA, <http://www.jboss.org/jbossesb/resources/SOABasics.html>

²⁵ Презентация «Designing SOA with JBoss Tools», <http://www.slideshare.net/eschabell/designing-soa-with-jboss->

²¹ Understanding Service Oriented Architecture, <http://msdn.microsoft.com/en-us/library/aa480021>

Х. ПОДХОД PROGRESS SOFTWARE

Progress Software имеет линейку продуктов COA (Sonic, Actional). Теоретической основой этих продуктов является собственная методология COA. Наработки этой методологии используют и другие вендоры, например, Microsoft.

А. Основы COA

Сервис-ориентированная архитектура (COA) – это ориентированный на бизнес архитектурный подход, обеспечивающий интеграцию и повторное использование процессов и сервисов²⁶.

Принципы дизайна сервисов²⁷:

- Сервис инкапсулирует определенные данные и поведение. Вызываемый сервис называется провайдером сервиса, вызывающий сервис называется потребителем сервиса.

- Сервис должен использовать определенный интерфейс (контракт), который могут использовать потребители сервиса.

- Сервис должен внедряться таким способом, чтобы быть доступным для вызова со стороны другого программного обеспечения с помощью стандартных протоколов

- Сервис не имеет излишних зависимостей с программным обеспечением, что позволяет легко их комбинировать

Эти принципы хорошо соотносятся с принципами дизайна COA компании Arcitura.

Методология, помимо общих принципов COA, содержит модель зрелости COA, содержащую следующие уровни зрелости COA на предприятии:

- Начальные сервисы
- Архитектурные сервисы
- Совместные и бизнес-сервисы
- Изменяемые бизнес-сервисы
- Оптимизированные бизнес-сервисы.

Модель зрелости соотносится с аналогичными моделями компаний Arcitura и The Open Group

XI. COA И ОБЛАЧНЫЕ ТЕХНОЛОГИИ

Облачные технологии подчиняются стандартам COA. В частности эталонная архитектура COA (SOA RA) компании The Open Group применима к облачным технологиям²⁸. Основные понятия COA этой методологии

tools

²⁶ SOA Infrastructure Progress Software, <http://www.progress.com/en/soa-infrastructure.html>

²⁷ Principles of a service-oriented architecture, <http://www.progress.com/en/soa-infrastructure.html>

²⁸ The IBM advantage for SOA Reference Architecture Standard,

– сервисные компоненты, сервисы, бизнес-процессы и др. – относятся и к облачным технологиям. Для облачных технологий предъявляются специфические требования к архитектурным слоям, касающимся доступности, производительности, масштабируемости и безопасности.

В методологии Arcitura облачные технологии рассматриваются наряду с COA-архитектурой, даются важные определения, при управлении проектами выделяются роли, специфические для облачных технологий.

XII. ВЫВОДЫ

Методологии COA прошли достаточно долгий эволюционный путь. Сейчас мы имеем картину двух соперничающих и существенно отличающихся методологий: Arcitura, с одной стороны, и The Open Group/IBM, с другой. Остальные вендоры пытаются выстраивать собственные методологии COA прежде всего в угоду своим маркетинговым потребностям, используя идеи двух основных методологий.

<http://www.ibm.com/developerworks/webservices/library/ws-soa-ref-arch/>