

# ФУНДАМЕНТ ДЛЯ SOA

*Сервис-ориентированная архитектура превратилась из предмета научных построений отраслевых проповедников в реальную практическую методологию создания информационных систем. IBM предлагает инициативу, получившую название IBM SOA Foundation, которая упрощает переход компаний к SOA с помощью набора интегрированного программного обеспечения, методических руководств и проверенных практик внедрения*

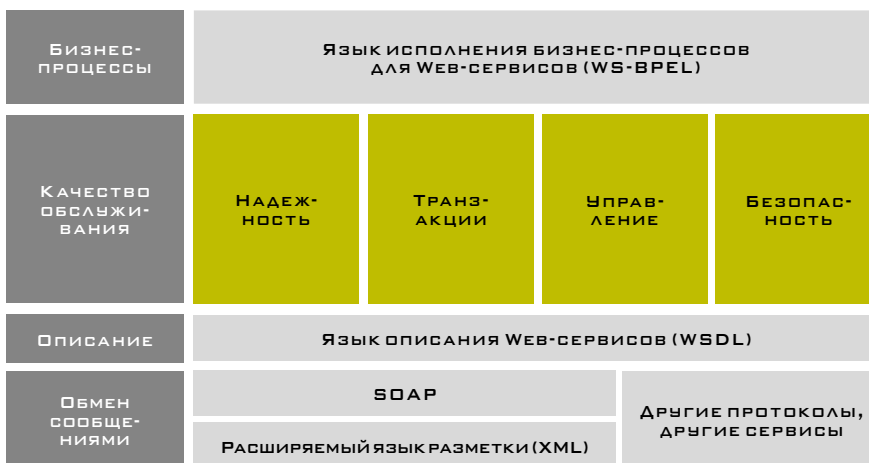
По мере стремительного роста количества систем ИТ и повышения их вклада в бизнес, компании вынуждены внедрять все более сложные программные архитектуры. Индустрия программного обеспечения прошла через множество архитектурных решений, стремясь реализовать полностью распределенную обработку информации. Традиционные подходы уже достигли предела своих возможностей, но от подразделений ИТ продолжают требовать все более быстрого реагирования на новые запросы бизнеса, снижения стоимости ИТ-решений и прозрачной интеграции с новыми партнерами и заказчиками. Сервис-ориентированная архитектура (Service-Oriented Architecture, SOA) помогает подразделениям ИТ достойно встретить возникающие задачи, являясь следующим эволюционным шагом в истории ИТ.

Всего за несколько лет сервис-ориентированная архитектура превратилась в реальную практическую методологию создания информационных систем. SOA — это архитектура, а не готовое решение. Следовательно, она не поставляется как коробочный продукт, а создается под определенное предприятие, и в последующем, в процессе своего жизненного цикла совершенствуется и адаптируется к изменениям условий бизнеса. Компания IBM предлагает инициативу, получившую название IBM SOA Foundation, которая облегчает внедрение SOA на предприятии и таким образом повышает ценность приложений и бизнес-процессов, отвечающих за ваш бизнес.

С помощью средств, входящих в SOA Foundation, современные предприятия могут оперативно создавать решения на основе SOA. В некотором смысле SOA Foundation напоминает наборы для детского творчества с исходными материалами, инструментами, методическими руководствами, примерами конструкций или опытов, которые позволяют начинающему построить действующую модель. С инструментально-продуктовой точки зрения SOA Foundation представляет собой набор интегрированного про-

## РЕАЛИЗАЦИЯ SOA НА ОСНОВЕ WEB-СЕРВИСОВ

*В ее нынешнем виде IBM SOA Foundation построена на базе существующего стека открытых стандартов. Такое решение отражает тот факт, что информационные системы гетерогенны, и ни одна компания не в состоянии удовлетворить все потребности собственными прикладными программными интерфейсами (API). Часть используемых стандартов уже признана de-юре, остальные — de-факто*



граммного обеспечения. Он основан на открытых стандартах и состоит из подмножества продуктов, входящих в семейство ПО промежуточного слоя WebSphere. Однако принципиально важно то, что, помимо программных систем, SOA Foundation комплектуется примерами лучших реализованных решений, руководствами и справочными материалами. С помощью руководств SOA Foundation будущий пользователь SOA может выбрать те компоненты, которые ему нужны на начальном этапе, а потом на протяжении жизненного цикла созданной им архитектуры развивать ее, выбирая для этой цели средства из уже имеющегося портфеля готовых программных решений IBM.

Необходимо учесть также, что под SOA понимаются не столько программные технологии, сколько система отношений между бизнесом и ИТ, и в этом состоит основное преимущество SOA над предшествующими архитектурными решениями. Главное достоинство SOA заключается в том, что сервисная архитектура может быть использована в качестве средства для

отображения бизнес-процессов на технические ресурсы и в последующем обеспечивать поддержку и совершенствование этих бизнес-процессов. Вот почему SOA Foundation представляет собой не только полноценное архитектурное решение, но и набор технологий и практик, позволяющих связать бизнес с ИТ. Еще раз подчеркнем, что SOA не сводится только к технологиям, это понятие гораздо шире, чем просто программно-аппаратная платформа.

## SOA С ПОЗИЦИЙ БИЗНЕСА

Основная причина резко возросшего интереса к SOA, наблюдаемого в последние годы, заключается в том, что вместе с SOA впервые открылась возможность преодолеть хронический разрыв между бизнесом и информационными технологиями. Не требует доказательств, что подлинная интеграция бизнеса с ИТ открывает огромные перспективы, но ее реализация требует преодоления серьезных препятствий. Начать следует хотя бы с того, что проектирование SOA предполагает наличие бизнес-проекта

(business design). С позиций классической теории управления бизнес-проект есть не что иное, как модель, в которой описывается поведение управляемой системы. В данном случае средствами бизнес-проекта описывается, как работает бизнес, то есть выполняемые процессы, организационные структуры, объединяющие все — от людей до финансов, тактические и стратегические цели и задачи, учет влияния рыночных факторов, правила и политики ведения бизнеса. Без достоверной модели невозможно создание системы управления, в том числе управления бизнесом.

Обычно предприятия имеют задокументированные бизнес-планы, но гораздо реже — бизнес-проекты. Еще реже бизнес-проекты, созданные на начальном этапе проектирования, сохраняют свою актуальность в процессе существования системы. Наличие актуального бизнес-проекта является обязательным условием для создания SOA, но он может оказаться полезным для бизнеса и в более широком смысле. Бизнес-проект позволяет найти возможности для совершенствования бизнес-процессов, мониторинга текущей активности, распределения обязанностей, выделения функций, нуждающихся в автоматизации, и много другого.

Подход к созданию информационных систем, основанный на SOA, обладает тем преимуществом, что, модульный по своей природе, он позволяет поставить определенные компоненты систем в соответствие автоматизируемым функциям. Далее, поскольку система становится производной от бизнес-проекта, открывается возможность ее модификации по мере эволюции бизнес-проекта. Адаптация к текущему состоянию бизнес-проекта составляет суть жизненного цикла SOA.

Для того чтобы проявился синергетический эффект от совместного видения проблем со стороны бизнеса и со стороны ИТ, необходимо выполнение нескольких условий:

■ Требуются формальные средства и язык для описания бизнес-проекта (язык моделирования).

■ Требуются средства для трансляции описания бизнес-проекта на языке моделирования в конструкции и другие артефакты информационной системы.

■ Для внедрения этих артефактов необходима инфраструктура, обладающая не меньшей гибкостью, чем сам бизнес. Существовавшие до SOA инфраструктурные подходы не обеспечивали нужного уровня гибкости.

■ Для сохранения соответствия между бизнес-функциями и ИТ-компонентами необходимы специально

выделенные места — контрольные точки, куда будут вноситься изменения.

■ Необходима собственная система управления ИТ для обеспечения поставленных выше требований.

Реализация этих условий обеспечивает очевидный успех SOA.

#### РОЛЬ СЕРВИСОВ В СЕРВИС-ОРИЕНТИРОВАННОЙ АРХИТЕКТУРЕ

Термин «сервис» в названии SOA переключается с Web-сервисами, но Всемирной Паутиной область действия сервисов не ограничивается. Приставка Web отражает всего лишь связь с совокупностью Web-технологий и стекком стандартов, но в корпоративном контексте можно говорить о сервисах как таковых. По сути, любой бизнес-процесс представляет собой композицию, состоящую из сервисов, оказываемых участниками процесса, а бизнес-проект позволяет спроецировать эти сервисы, политики и условия на сервисную архитектуру информационной системы.

Однако что же в таком случае можно назвать сервисами? Функции, реализуемые приложениями? И можно ли считать все функции приложений сервисами? Включает ли SOA в себя системные сервисы? Прежде всего, определим руководящие принципы,

### ОСНОВНАЯ ПРИЧИНА РЕЗКО ВОЗРОСШЕГО ИНТЕРЕСА К SOA, НАБЛЮДАЕМОГО В ПОСЛЕДНИЕ ГОДЫ, ЗАКЛЮЧАЕТСЯ В ТОМ, ЧТО ВМЕСТЕ С SOA ВПЕРВЫЕ ОТКРЫЛАСЬ ВОЗМОЖНОСТЬ ПРЕОДОЛЕТЬ ХРОНИЧЕСКИЙ РАЗРЫВ МЕЖДУ БИЗНЕСОМ И ИНФОРМАЦИОННЫМИ ТЕХНОЛОГИЯМИ

которые являются общими для рассмотрения сервис-ориентированных архитектур: это слабая связанность (loose coupling) и декомпозиция сложных задач на отдельные сервисы.

В широком смысле сервисы можно определить как повторяющиеся задачи, выполняемые в бизнес-процессе. Например, введение нового клиента компании или получение для него остатка по счету. Понятно, что практически любой бизнес-процесс можно представить как сервис для процесса следующего уровня, то есть сервисы образуют иерархическую структуру. Теоретически возможна рекурсивная декомпозиция бизнес-процесса до некоторого «атомарного» уровня, до уровня простейших сервисов. При проецировании бизнес-сервисов на сервисы информационной системы приходится следовать

иерархии бизнес-сервисов, то есть ограничивать глубину декомпозиции до адекватного уровня гранулярности.

Исходя из такого определения сервисов, можно сказать, что сервисная ориентация — это способ интеграции бизнес-процесса в форме набора сервисов, сервис-ориентированная архитектура — это архитектурный стиль, основанный на принципах сервисной ориентации, а корпоративная информационная система, построенная на принципах SOA, представляет собой композитное приложение, составленное из набора интегрированных сервисов.

#### ПРЕИМУЩЕСТВА СЛАБОЙ СВЯЗАННОСТИ

Переход на SOA открывает возможность для реализации распределенных решений, полностью соответствующих требованиям инфраструктуры бизнеса. Распределенность информационных систем отражает реальные свойства бизнеса. Практически все предприятия состоят из отдельных подразделений или, как их теперь принято называть, бизнес-направлений (Line-Of-Business, LOB). Они представляют собой островки активности, деятельность которых нужно собирать воедино, чтобы обеспечить целостность бизнеса. LOB в системном

смысле являются слабосвязанными, именно поэтому попытки построить распределенные информационные системы на принципах жесткой связанности, как это делалось до появления SOA, в подавляющем большинстве случаев оканчивались неудачей.

Жесткосвязанные системы предполагают технологическую однородность всех компонентов, тогда между ними сохраняется взаимодействие. Но реальная жизнь свидетельствует о том, что отдельные направления бизнеса внутри предприятия и тем более родственные предприятия используют различные программные и аппаратные средства. В результате работоспособность системы оказывается в зависимости от ряда факторов, с которыми сложно справиться в условиях жесткой связанности. Прежде всего, возникают временные ограничения (temporal

constraint), которые проявляются в различном времени реакции на локальные и удаленные вызовы и, как следствие, в рассинхронизации работы системы. Практика показывает, что время реакции на локальные вызовы процедур измеряется микросекундами, а на внешние вызовы (Remote Procedure Call) — миллисекундами, то есть различие составляет, как минимум, три порядка. Кроме того, надежность распределенных систем с сетевой инфраструктурой существенно ниже, чем систем, работающих локально на одном сервере, и жесткая связанность не обеспечивает сохранения работоспособности в менее стабильной среде. И наконец, в условиях ограничений административного порядка отдельные LOB могут развивать свои внутренние функции независимо от других, что усложняет сохранение системной целостности.

Переход на SOA позволит избежать повторения тех ошибок, которые были допущены ранее не только при использовании таких технологий, как SNA, DCE, DCOM и CORBA, но и технологий на базе обмена сообщениями типа MOM. Последние, будучи в значительной мере слабо связанными, позволяли преодолеть практически все перечисленные выше ограничения, но требовали использования одного и того же типа ПО промежуточного слоя для реализации процедур передачи сообщений.

Идея слабой связанности информационных систем соответствует реальным бизнес-проектам еще и потому, что во многих компонентах проектов подразумевается участие сотрудников — исполнителей. Люди являются слабосвязанными по своей природе, они медлительнее компьютеров, не столь реактивны и т. д. Только системы со слабой связанностью позволят органичным образом включить человека в контур управления.

#### АРХИТЕКТУРЕ — АРХИТЕКТОРОВ

С появлением SOA усиливается роль архитектора предприятия — Enterprise Architect (EA). В качестве EA может выступать специально выделенный специалист, технический директор (Chief Technology Officer, CTO) или даже исполнительный директор (Chief Information Officer, CIO). Основная функция EA — в установлении соответствия между бизнес-проектом и информационной инфраструктурой. Метафорически можно сказать, что EA выступает в качестве внутреннего моста организации, он не только обеспечивает трафик, но организует участников движения. Для этого он

должен знать принципы создания бизнес-проектов и принципы транслирования этих проектов на информационные системы. В процессе трансляции проекта EA может работать совместно с архитектором по техническим решениям (Technical Solution Architect).

Общими усилиями эти специалисты должны понять принципы, процессы и инструменты, необходимые для создания бизнес-проекта. Они обязаны выстроить программную модель, поддержать ее технологически, создать топологию информационной системы и определить способы ее интеграции средствами ПО промежуточного слоя. Кроме того, они должны оценить существующие возможности информационной системы и потребности в будущем. Им предстоит также решить, как средствами ПО промежуточного слоя обеспечить необходимые эксплуатационные качества системы, включая надежность и готовность в условиях изменяемой внешней среды. При этом основная задача архитектора предприятия заключается в том, что принято называть оркестровкой, то есть в интеграции разнородных компонентов в единую целостную систему. Он должен выстроить общую архитектуру предприятия (Enterprise Architecture) в виде набора стандартов, принципов и политик, руководствуясь которыми можно оценить эффективность применения конкретных информационных технологий.

Работа архитекторов, создающих архитектуру предприятия, в отличие от тех, кто проектирует здания, редко начинается с чистого листа. Это скорее перестройка, чем новое строительство, приходится учитывать наличие унаследованных приложений. Поэтому в SOA Foundation предусматривается возможность реинжиниринга продуктов, построенных на платформах Microsoft, Oracle, BEA, SAP и других платформах ПО промежуточного слоя. Итогом деятельности архитекторов должно быть отображение логической архитектуры на физическую с учетом как функциональных, так и нефункциональных требований, таких как качество обслуживания и безопасность.

#### SOA FOUNDATION И СТАНДАРТЫ

В ее нынешнем виде SOA Foundation построена на базисе существующего стека открытых стандартов. Такое решение отражает тот факт, что информационные системы гетерогенны, и ни одна компания не в состоянии удовлетворить все потребности собственными прикладными программными интерфейсами (API). Часть используе-

мых стандартов уже признана де-юре, остальные — де-факто. Основными составляющими нынешнего стека являются стандарты XML и Web-сервисов, в том числе WSDL, SOAP, WS-Security и другие, а также стандарт описания бизнес-процессов BPEL. Кроме того, используются рекомендации по интерпретации стандартов, предлагаемые организацией WS-I.org.

Из перечисления стандартов, приведенного выше, следует, что в SOA Foundation за Web-сервисами остается основная роль. Однако не надо полностью отождествлять сервис-ориентированные архитектуры лишь с этим типом технологии, существуют и иные подходы к их реализации. Web-сервисы представляют собой всего лишь один из возможных стандартов, способных обеспечить техническую сторону взаимодействия между функциональными сервисами, размещенными на разных платформах. В общем контексте SOA Foundation Web-сервисам отведена служебная роль носителя, они занимают примерно ту же позицию, что и, скажем, технологии физического уровня в семиуровневой сетевой модели OSI. При этом Web-сервисы в SOA сделаны прозрачными для верхних уровней архитектуры.

Для тех, кто занимается проектированием или аналитикой бизнеса, нюансы протоколов не представляют ни малейшего интереса. В конечном итоге ориентация на сервисы, присущая SOA, не зависит от конкретной технологии, которая ее поддерживает. Поэтому архитектура SOA Foundation остается открытой по отношению к языкам, инфраструктурным технологиям и протоколам. В ней есть принципиальная возможность для реализации сервисной архитектуры без использования стандартов, предложенных для Web-сервисов.

#### ЧЕТЫРЕ ЭТАПА ЖИЗНЕННОГО ЦИКЛА SOA

Построение SOA представляет собой итерационный циклический процесс, который включает в себя следующие этапы:

■ **СОЗДАНИЕ МОДЕЛИ.** В данном контексте под созданием модели понимается преобразование общего представления о целях и задачах бизнеса в бизнес-проект или в формальную модель. Если модель обладает достаточной степенью формализма, то она может быть использована для оценки работы бизнеса. Для этого в модель должны быть включены датчики, так называемые ключевые индикаторы производительности

ти (key performance indicators).

|| **СБОРКА СИСТЕМЫ.** В сборке системы участвуют архитектор предприятия, бизнес-аналитик, архитектор, ответственный за программное обеспечение. Совместно они переводят модель в компоненты системы, конечная цель сборки — создание программной модели.

|| **ВНЕДРЕНИЕ.** На этапе внедрения создаются информационная среда и приложения, работающие в этой среде.

|| **УПРАВЛЕНИЕ.** Управлением замыкается петля обратной связи, здесь оцениваются результаты и полученные сведения используются для совершенствования бизнес-проекта и моделей на следующей итерации создания SOA.

В SOA Foundation предусмотрено создание логической, программной и физической моделей SOA.

**ЛОГИЧЕСКАЯ МОДЕЛЬ SOA**

Логическую модель SOA можно представить как совокупность групп близких по функциональности сервисов, объединенных корпоративной сервисной шиной (Enterprise Service Bus, ESB):

|| **СЕРВИСЫ ВЗАИМОДЕЙСТВИЯ (INTERACTION SERVICES)** обеспечивают связь между приложениями и источниками внешней информации. Источниками могут быть как люди, так и устройства, в том числе датчики RFID, промышленные роботы, устройства систем промышленной автоматизации и другие. Взаимодействие может осуществляться в ролевом контексте (role-sensitive context), где важны не только собственно передаваемые параметры, но и то, кто их передает, что позволяет осуществлять целый ряд вспомогательных операций, в том числе аутентификацию, выбор источников по приоритетам и т. д.

|| **СЕРВИСЫ ПРОЦЕССОВ (PROCESS SERVICES)** реализуют логику управления, в том числе организуют потоки бизнес-процессов (business process flows) и контролируют машины состояний (business state machines). В терминологии, принятой в IBM, машина состояний — это управляемая событиями бизнес-транзакция, в которой внешние события (воздействия) переводят систему из одного дискретного состояния в другое. Эти же сервисы осуществляют оркестровку процессов.

|| **ПРИКЛАДНЫЕ СЕРВИСЫ (BUSINESS APPLICATION SERVICES)** реализуют ядро бизнес-логики, они могут быть декомпозированы на более мелкие сервисы или, наоборот, собраны в более круп-

ные сервисы следующего уровня. Прикладные сервисы связываются в бизнес-процессы с помощью сервисов процессов, но могут вызываться и сервисами взаимодействия.

|| **ИНФОРМАЦИОННЫЕ СЕРВИСЫ (INFORMATION SERVICES)** оперируют данными бизнес-процессов и реализуют два типа функций. Во-первых, это непосредственное обеспечение бизнес-процессов данными, которые могут поступать по запросам из реляционных баз

является не законченным продуктом, а архитектурным шаблоном, с помощью которого может быть построена интеграционная платформа.

Помимо основных сервисов в состав логической модели SOA Foundation входят следующие группы вспомогательных поддерживающих сервисов:

|| **СЕРВИСЫ РАЗВИТИЯ И ОПТИМИЗАЦИИ БИЗНЕСА (BUSINESS INNOVATION AND OPTIMIZATION SERVICES)** включают в себя инстру-



**ЛОГИЧЕСКАЯ МОДЕЛЬ SOA**

*Логическую модель SOA можно представить как совокупность групп близких по функциональности сервисов, объединенных корпоративной сервисной шиной (Enterprise Service Bus, ESB)*

данных, различных каталогов баз данных, файлов систем, репозиториев XML и других источников. Второй тип функций относится к управлению жизненным циклом информации и осуществляет миграцию данных, наполнение хранилищ данных, поддержку бизнес-аналитики и управление контентом.

|| **СЕРВИСЫ ДОСТУПА (ACCESS SERVICES)** предназначены для включения в сервисную архитектуру унаследованных приложений. Это достигается путем заключения таких приложений в специальную «оболочку» и представления в последующем в форме сервисов.

|| **СЕРВИСЫ ДЛЯ РАБОТЫ С ПАРТНЕРАМИ (PARTNER SERVICES)** сочетают в себе качества сервисов взаимодействия и сервисов доступа, но ориентированы на работу с организациями-партнерами.

Взаимодействие между сервисами осуществляется посредством корпоративной шины ESB. Для создания ESB могут быть использованы разные технологии, в том числе WebSphere Enterprise Service Bus. Важно уточнить, что WebSphere ESB

предназначены для кодирования бизнес-проектов. К ним относятся также средства для моделирования бизнес-процессов, измерения метрик процессов и анализа производительности.

|| **СЕРВИСЫ РАЗРАБОТКИ (DEVELOPMENT SERVICES)** состоят из архитектурных инструментов, инструментов для разработки и сборки приложений, из методик, средств отладки, а также специализированных механизмов публикации, необходимых для создания приложений, предназначенных для работы в среде SOA.

|| **СЕРВИСЫ МЕНЕДЖМЕНТА ИТ (IT SERVICE MANAGEMENT)** служат для мониторинга работы других сервисов, анализа сбоев и узких мест, восстановления после сбоев и реализации административной политики.

|| **ИНФРАСТРУКТУРНЫЕ СЕРВИСЫ (INFRASTRUCTURE SERVICES)** образуют ядро ИТ-среды, в которой работают приложения SOA. С помощью этих сервисов осуществляется виртуализация компьютерной платформы.

Логическая модель отображается в программную и фи-

зическую модели SOA.

**ПРОГРАММНАЯ МОДЕЛЬ SOA**

Программная модель SOA существует в виде набора ролей, задач, правил кодирования, языков программирования, которым нужно следовать и которые нужно использовать в процессе создания ПО. Эта модель создавалась исходя из необходимости удовлетворения двух требований:

|| Модель должна вбирать в себя языки программирования, приложения и компонентные модели, которые уже существуют и используются, такие как, например, CICS, IMS, MQ, J2EE, .NET, BPEL, XML, DB2, продукты Oracle, SAP и др.

|| Модель должна маскировать различия между этими языками и средами, чтобы в максимальной степени обеспечить повторное использование компонентов и создать условия для простой сборки компонентов в соответствии с бизнес-проектом.

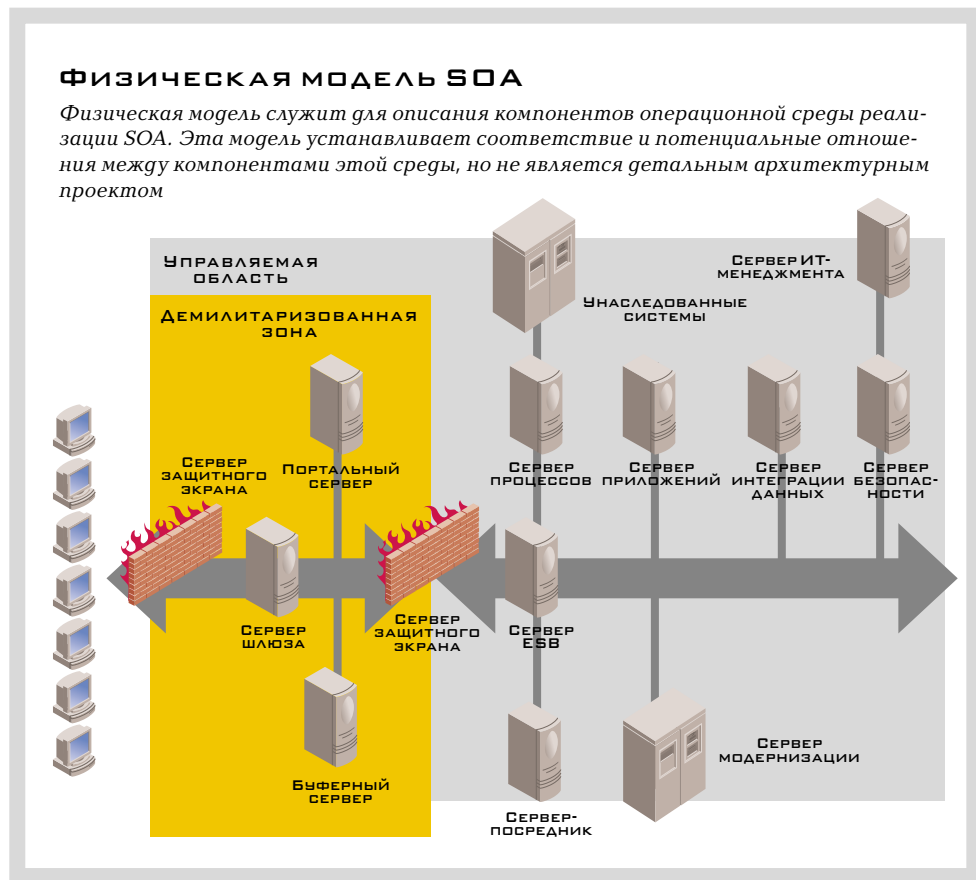
В основе программной модели лежит концепция последовательного раскрытия (progressive disclosure), которая предполагает возможность использования программных компонентов как своего рода черных ящиков, снабженных метаданными. Ящик открывается только в крайнем случае, если в процессе программирования на верхнем уровне не хватает возможностей, предоставляемых нижним уровнем. Такой подход выбран в связи с желанием сосредоточиться на программировании бизнес-логики сервиса, не уделяя при этом излишнего внимания технологическим аспектам. В процессе создания программной модели реализуются следующие основные цели: программирование логики пользовательского взаимодействия (презентационной логики), управления бизнес-логикой, логики информации, композиционной логики и логики взаимодействия сервисов. Эти цели напрямую соотносятся с логической моделью SOA.

Для того чтобы реализовать концепцию последовательного раскрытия, сервисы всех групп логической модели программируются как сервисные компоненты (service components), которые обмениваются сервисными данными (service data) и инициируются средствами сервисной шины (service bus):

|| **СЕРВИСНЫЙ КОМПОНЕНТ** — нейтральное по отношению к языкам и технологии представление сервиса, любая логика сервиса должна быть инкапсулирована в сервисный компонент. Сервисные компоненты — это фундаментальные строительные блоки SOA.

|| **СЕРВИСНЫЕ ДАННЫЕ** — нейтральные по отношению к языкам и технологии представления данных. Сервисные данные можно описать

Физическая модель служит для описания компонентов операционной среды реализации SOA. Эта модель устанавливает соответствие



как документы или сообщения, которыми обмениваются сервисы.

|| **СЕРВИСНАЯ ШИНА** — средства для обмена данными между сервисами и для обеспечения правил их взаимодействия. Шина остается абсолютно прозрачной для программной модели.

Для реализации программной модели должны быть выполнены следующие действия:

- || моделирование бизнес-проекта с включением основных индикаторов производительности;
- || преобразование модели в программную архитектуру;
- || кодирование потоков процессов и машин состояний;
- || поиск унаследованных приложений и ранее созданных сервисов;
- || превращение унаследованных приложений в сервисы и написание новых сервисов;
- || определение схем обмена данными и сообщениями между сервисами;
- || задание потоков управления, политик и бизнес-правил;
- || сборка сервисов;
- || тестирование и внедрение.

**ФИЗИЧЕСКАЯ МОДЕЛЬ SOA**

и потенциальные отношения между компонентами этой среды, но не является детальным архитектурным проектом. Физическая модель представлена в виде набора серверов, хотя в реальности аналогичного деления на физические серверы может и не быть. Физическая архитектура, как и логическая архитектура, концентрируется вокруг корпоративной шины ESB. В свою очередь центральную роль в реализации ESB играет специализированный сервер ESB server. Однако сервисная шина не сосредоточена только в этом сервере, она является виртуальным образованием и физически распределена по корпоративной сети. ESB server осуществляет управление шиной, но потоки данных при обмене между сервисами могут его обходить.

Фактически функции ESB может выполнять вся внутренняя корпоративная сеть с такими традиционными серверами, как:

|| **СЕРВЕР ЗАЩИТНОГО ЭКРАНА (FIREWALL SERVER)**, отделяющий «демилитаризованную» зону сети от внутренней части, защищенной от внешних воздействий.

|| **БУФЕРНЫЙ СЕРВЕР (PROXY**

SERVER), кэширующий страницы, распределяющий нагрузку и выполняющий ряд других вспомогательных функций.

■ **ПОРТАЛЬНЫЙ СЕРВЕР (PORTAL SERVER)**, обеспечивающий взаимодействие с внешними пользователями.

■ **СЕРВЕР ПРОЦЕССОВ (PROCESS SERVER)**, на котором выполняются бизнес-процессы и реализованы машины состояний.

■ **СЕРВЕР ПРИЛОЖЕНИЙ (APPLICATION SERVER)**, который поддерживает выполнение приложений и обеспечивает прямой доступ к сервисам данных.

■ **СЕРВЕР ИНТЕГРАЦИИ ДАННЫХ (INFORMATION INTEGRATION SERVER)**, содержащий данные, хранилища данных и сервисы бизнес-аналитики (business intelligence services).

■ **СЕРВЕР МОДЕРНИЗАЦИИ (ENTERPRISE MODERNIZATION)**, представляющий ту часть среды, которой еще предстоит преобразование в сервисы.

■ **СЕРВЕР БЕЗОПАСНОСТИ (SECURITY SERVER)**, который решает проблемы идентификации, авторизации и аудита и реализует другие действия по защите сервисов в рамках SOA Foundation.

■ **СЕРВЕР ИТ-МЕНЕДЖМЕНТА (MANAGEMENT SERVER)**, отвечающий за управление ИТ-инфраструктурой.

## SOA: НОВЫЕ ПЕРСПЕКТИВЫ И НОВЫЕ ПРОБЛЕМЫ

Создание нового класса информационных систем, которые, будучи слабосвязанными внутри, становятся тесно связанными с практикой бизнеса, порождает множество новых

оказывается пройденной. Внедрение управления бизнес-процессами на принципах SOA тоже в каком-то смысле представляет собой переход точки возврата. Здесь возникает проблема руководства SOA (SOA Governance), то есть определения ответственных лиц, которые используют систему и принимают решения об ее изменении, и описания необходимых для их работы структур и процессов.

Создание систем на базе SOA может дать импульс развитию таких перспективных технологий, как, например, технология самоуправляемого компьютеринга (autonomic computing), однако при этом возникает необходимость в появлении принципиально новых решений.

Поведение такого рода систем в корне отличается от поведения привычных монолитных конфигураций, что проявляется прежде всего при сбоях.

Несмотря на то что сервисы слабо связаны, распределены логически и могут использоваться совместно разными процессами, тем не менее выход из строя одного сервиса может повлиять на работу других. Однако обнаружить неисправность, в отличие от монолитной системы, можно будет только тогда, когда возникнет обращение к этому сервису.

Это означает, что первое обращение к сервису после его выхода из строя останется без обслуживания. Следовательно, потребуются системы с превентивным обслуживанием, которые смогут обнаруживать сбои в сервисах заблаговременно.

Поэтому, инфраструктура SOA должна быть построена так, чтобы

## НОВЫЙ АРСЕНАЛ SOA

В начале апреля IBM анонсировала программные продукты семейства IBM WebSphere и набор профессиональных услуг, предназначенных для преодоления барьеров на пути к успеху SOA. Решения IBM помогают спроектировать SOA, исключить лишние расходы и дать гарантию того, что выделенные средства пойдут на осуществление бизнес-стратегии, способной выдержать флуктуации рынка и изменения в компании. Среди представленных программных продуктов: порталный сервер WebSphere Portal 6.0 для интеграции технологий совместной работы IBM Workplace и построения составных приложений для индивидуальных отраслей, ролей и задач; WebSphere Business Monitor 6.0 для идентификации потенциальных проблем; сервер приложений WebSphere Application Server 6.1; инструментарий электронной коммерции WebSphere Commerce 6.0. Кроме того, корпорация представила отраслевые модели SOA-инициатив для банковской и страховой сферы.

## ПАРТНЕРЫ ПРОДВИГАЮТ SOA

Новые инициативы, призванные помочь заказчикам и партнерам в освоении сервис-ориентированной архитектуры (SOA), выдвинуты IBM на международной конференции Partnerworld. Для партнеров, подтвердивших поддержку платформы IBM SOA Foundation, в программе PartnerWorld for Industry Networks вводится направление SOA Specialty. В рамках нового направления будет предоставляться техническая поддержка и перспективные планы развития навыков партнерам, которые заинтересованы в возможностях растущего рынка SOA. Ведется работа над новым порталом IBM SOA Business Central, который представляет собой ресурс для поиска средств SOA.

## Альянс Cognos и IBM

Компании Cognos и IBM расширяют сотрудничество, направленное на содействие заказчикам в интеграции бизнес-процессов в масштабах предприятия и лучшем понимании показателей эффективности бизнеса. Новое соглашение предусматривает совместную разработку, продвижение и продажу решений, основанных на сервис-ориентированной архитектуре (SOA). В частности, IBM расширит свои консалтинговые ресурсы по решениям Cognos, а Cognos, в свою очередь, усовершенствует и оптимизирует свои продукты для интеллектуального бизнес-анализа для работы с аппаратными платформами, программным обеспечением и услугами IBM.

## СОЗДАНИЕ СИСТЕМ НА БАЗЕ SOA МОЖЕТ ДАТЬ ИМПУЛЬС РАЗВИТИЮ ТАКИХ ПЕРСПЕКТИВНЫХ ТЕХНОЛОГИЙ, КАК, НАПРИМЕР, ТЕХНОЛОГИЯ САМОУПРАВЛЯЕМОГО КОМПЬЮТИНГА (AUTONOMIC COMPUTING), ОДНАКО ПРИ ЭТОМ ВОЗНИКАЕТ НЕОБХОДИМОСТЬ В ПОЯВЛЕНИИ ПРИНЦИПИАЛЬНО НОВЫХ РЕШЕНИЙ

проблем. Прежде всего, возрастает ответственность тех лиц, которые создают, эксплуатируют и развивают информационные системы. До тех пор, пока любая система автоматизации остается вспомогательной, сохраняется возможность перехода на ручное управление, но с определенным моментом без нее функционирование бизнеса становится невозможным, «точка возврата»

она сама могла обнаруживать собственные неисправности и по возможности исправлять их, осуществляя доступную переконфигурацию.

Для решения этой задачи SOA Foundation предлагает инструментальные средства, позволяющие реализовать цикл мониторинг-анализ-планирование-выполнение (Monitor-Analyze-Plan-Execute, MAPE).

Наличие цикла MAPE при-