

СЕРВИСНО-ОРИЕНТИРОВАННОЕ МОДЕЛИРОВАНИЕ ФУНКЦИОНИРОВАНИЯ ЦЕНТРОВ ОБРАБОТКИ ДАННЫХ

В.И. Грекул,

кандидат технических наук, профессор кафедры корпоративных информационных систем Государственного университета — Высшей школы экономики, старший научный сотрудник, e-mail: grekoul@hse.ru.

И.В. Пырлина,

преподаватель кафедры корпоративных информационных систем Государственного университета — Высшей школы экономики, e-mail: irina_30@mail.ru.
Адрес: 105187, Москва, ул. Кирпичная, 33/5.

В статье рассматриваются возможности и особенности применения сервисно-ориентированного подхода для описания услуг аутсорсинга, предоставляемых центрами обработки данных (ЦОД). Использование моделей сервисов позволит улучшить взаимодействие с заказчиками за счет формализованного и понятного представления услуг ЦОД, обеспечит методическую основу для оперативного и гибкого формирования необходимых заказчиком бизнес-процессов.

Ключевые слова: бизнес-процесс, сервис-ориентированная архитектура, архитектура предприятия, центры обработки данных, бизнес-логика, гранулированности, композитные приложения, BPM.

В соответствии с современными представлениями [1] центр обработки данных (ЦОД) является основным поставщиком услуг ИТ-аутсорсинга и аутсорсинга бизнес-процессов. Можно рассматривать ЦОД как некоторый целостный программно-аппаратный комплекс, предоставляющий бизнесу необходимые услуги с требуемым качеством — с соответствующим уровнем надежности, доступности и производительности. В тоже время, различия в требованиях потребителей услуг аутсорсинга вынуждают ЦОД проявлять большую гибкость в формировании портфеля предлагаемых услуг и оперативно реагировать на изменения требований заказчиков, что вызывает необходимость

выделения набора элементарных (базовых) услуг и формирования на их основе комплексных (агрегированных) услуг поддержки разнообразных бизнес-процессов заказчиков. Как следствие, возникает необходимость в методическом и технологическом обеспечении разработки описания отдельных услуг ЦОД (сервисов) и их агрегации. В статье рассматривается возможность применения для этих целей концепций сервисно-ориентированной архитектуры (SOA).

Понятие сервисно-ориентированной архитектуры все чаще встречается сегодня в литературе. Основная сфера применения сервисно-ориентированных подходов связана с обеспечением

эффективности бизнеса за счет адекватной реакции на изменения, способности оперативно и гибко выстраивать новые бизнес-процессы, эффективно управлять ландшафтом корпоративных систем или использованием существующих систем для построения новых бизнес-процессов. Этот перечень задач во многом совпадает с целевым предназначением центров обработки данных, вследствие чего представляется интересным и целесообразным рассмотреть возможности применения сервисно-ориентированного подхода для формализации описания функционирования ЦОД.

Сервисно-ориентированная архитектура описывает применение в приложениях повторно используемых бизнес-сервисов. Бизнес-сервисы предоставляются информационными системами, а методология SOA позволяет строить новые приложения (или бизнес-процессы) быстрее и эффективнее. Обычно концепция сервисно-ориентированной архитектуры нацелена на поддержку идеи создания архитектуры приложений с хорошо структурированной топологией интерфейсов, предоставляющих возможность доступа к функциональности приложений из других программ при помощи открытых стандартов. Это позволяет решить проблему интеграции различных систем, используемых в рамках организации, дает возможность многократного их использования для различных задач. В отличие от предыдущих концепций корпоративных архитектур, SOA подчеркивает такой аспект, как тесная связь информационных технологий с динамикой бизнеса. Мы полагаем, что применение концепции SOA окажется полезным для организации взаимодействия ЦОД с заказчиками: будут обеспечены формализованные описания услуг ЦОД, что позволит оперативно выстраивать необходимые заказчику бизнес-процессы.

Сегодня ведущие компании — производители программных продуктов, такие как SAP, IBM, Oracle, представляют свои решения, поддерживающие SOA. В основу таких решений положена единая открытая интеграционная платформа, реализующая на базе web-сервисов (стандартов для интеграции приложений или компонентов приложений) доступ к разнородным системам, и позволяющая компоновать необходимые процессы. В последние годы специалисты по SOA пытаются расширить рамки этой архитектуры, включив в рассмотрение не только архитектуру информационных систем, но и остальные ресурсы организации, которые участвуют в исполнении бизнес-процессов компании.

Естественно, что такие решения интересны и создателям ЦОД: развитие SOA ведет к востребованности «технологий виртуализации», к росту популярности концепции Software as a Service (SaaS) [1].

Поскольку в рамках данной статьи рассматриваются возможности сервисно-ориентированного описания деятельности ЦОД, используемое понятие бизнес-сервиса оказывается значительно шире, чем распространенное понятие web-сервиса. Бизнес-сервис потребляет некоторую информацию (данные), обрабатывает ее и предоставляет потребителю результат своей работы. Бизнес-сервис может быть определен как на уровне процессов второго уровня детализации, так и на более низких уровнях (рис. 1). Бизнес-сервис включает в себя одну или несколько функций компании, строго описанный интерфейс обращения (или контракт, описывающий правила обращения с сервисом), а также ресурсы, необходимые для его исполнения (человеческие ресурсы, информационные ресурсы и т.д.). Бизнес-сервис следует выделять на та-

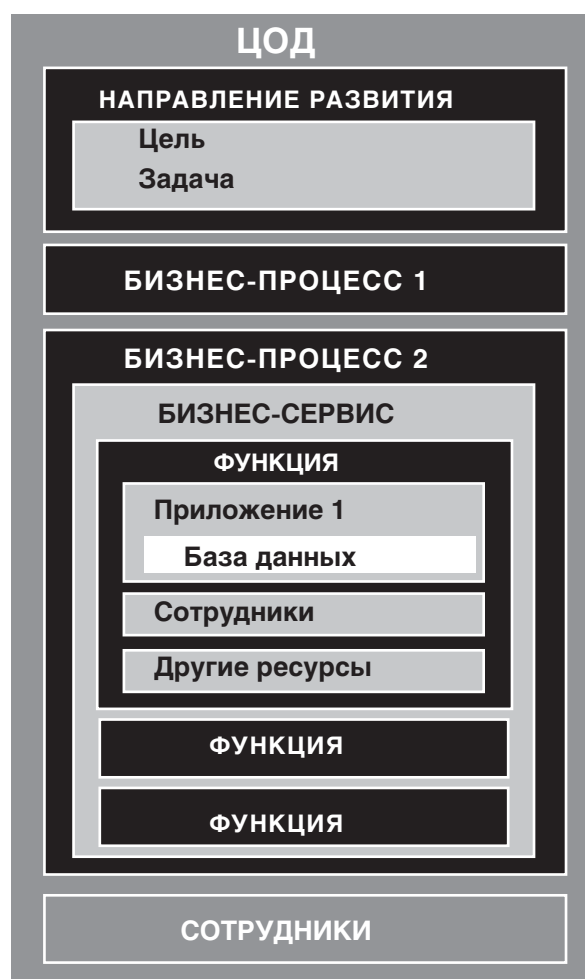


Рис. 1. Модель бизнес-сервиса в рамках ЦОД

ком уровне бизнес-процесса, который обеспечит управление этим элементом.

При моделировании бизнес-сервисы будут представляться отдельными блоками, имеющими унифицированный интерфейс для обращения к их функциям. Выделенные блоки будут отображать сервисы, из которых можно будет собирать тот или иной процесс клиента (в соответствии с концепцией SOA — потребителя сервисов).

Для разработки методики сервисно-ориентированного моделирования функционирования ЦОД необходимо, прежде всего, детально рассмотреть следующие вопросы:

1. Определить правила выделения бизнес-сервисов ЦОД.
2. Определить потребляемые сервисами ресурсы для анализа возможности их совместного исполнения в ЦОД.
3. Определить требования к гранулярности описываемых сервисов и выбрать нотацию для их описания.

Порядок выделения бизнес-сервисов ЦОД

ЦОД можно рассматривать как хранилище некоторых элементарных сервисов (достаточно автономных аппаратных и программных инструментов), которые по требованию заказчика должны объединяться в более сложные комплексные услуги или бизнес-процессы. В этом случае функционирование ЦОД естественным образом описывается в терминах распределенных систем, например, в виде моделей CORBA (Common Object Request Broker Architecture) или DCOM (Distributed Component Object Model). Однако и в CORBA, и в DCOM взаимодействующие объекты являются сильносвязанными, поэтому любые изменения компонентов должны быть жестко скоординированы между собой. Появившаяся позднее концепция SOA оказывается для нашей задачи более подходящей, поскольку позволяет строить слабосвязанные приложения, в которых одну часть можно менять, не затрагивая другую [3]. Можно выделить следующие основные характеристики SOA, которые и делают её особенно интересной для формализации описания услуг ЦОД:

- ◆ слабая связанность сервисов;
- ◆ независимость от аппаратной платформы и языков программирования;
- ◆ архитектура не привязана к конкретной техно-

логии (SOA может быть реализована на основе технологий RPC, REST и др.);

- ◆ функционирование в сетевой среде, когда клиенты и сервисы связаны между собой подсистемой коммуникации [3, 4].

Для того чтобы определить правила идентификации бизнес-сервисов в задаче описания ЦОД, воспользуемся опытом реализации концепции SOA в решениях известных поставщиков информационных платформ.

Несмотря на некоторые различия, взгляды ведущих ИТ-компаний на SOA во многом совпадают. С точки зрения компании IBM [5], SOA позволяет «представить повторяющиеся бизнес-процессы и функции в качестве сервисов, базирующихся на открытых стандартах и унифицированных интерфейсах, а корпоративная сервисная шина обеспечивает связующий инфраструктурный слой». По определению компании Oracle [6], SOA «облегчает развитие легко интегрируемых и повторно используемых модульных бизнес-сервисов — таким образом, создавая действительно гибкую и легко приспособляемую ИТ-инфраструктуру». Обе компании выделяют в качестве основной характеристики SOA эффективное использование повторяющихся бизнес-функций через сервисную шину. Компания SUN делает упор на упрощении и унификации доступа к базам данных через сервисы, возможность быстрой замены компонентов и доступ к любым данным. Но концепция все та же: выделяются бизнес-функции, которые в виде сервисов объединяются в бизнес-процессы.

В целом на уровне выделения сервисов компании IBM и Oracle демонстрируют одинаковый подход. Первичным считается определение бизнес-миссии и бизнес-целей компании. Затем описываются основные бизнес-процессы, они детализируются, выделяются повторяющиеся процессы. На следующем этапе формируются интеграционные бизнес-процессы или workflow-процессы, описываются на языке BPEL¹, выделяются сервисы. Сервисы уже программно реализуются в средах, которые предоставляют компании. Для каждого сервиса определяются его входы и выходы, форматы данных и бизнес-логика. Из сервисов составляются цепочки, которые и обозначают бизнес-процессы.

Основное отличие в реализации SOA в IBM и Oracle — использование различных методик при

¹ BPEL (англ. Business Process Execution Language) — язык на основе XML для формального описания бизнес-процессов и протоколов их взаимодействия между собой. BPEL расширяет модель взаимодействия веб-служб и включает в эту модель поддержку транзакций.

переходе от эмпирического описания бизнес-процессов к BPEL. Компания IBM использует водопадную модель, устанавливая зависимость реализации BPEL-описаний от бизнес-процессов, а Oracle строит двухстороннюю взаимосвязь, выполняя процессы синхронизации.

Наиболее важная задача компаний, создающих продукты для SOA-внедрений – это эффективно связать BPM и SOA для того, чтобы можно было осуществить переход от бизнес-логики к программной реализации. (Для наших целей – это задача формирования бизнес-процессов заказчика из сервисов ЦОД) Существующие модели для реализации данной связи представлены на рис. 2 [8].

Сначала с помощью инструментов BPM компания описывает свои бизнес-процессы. Затем эти описания переводятся в формат BPEL. Но далее возникает трудность: если мы построим сервисы, то каким образом их менять при изменении бизнес-процесса? По методологии Oracle это происходит через взаимную синхронизацию моделей, а в способе, представленном IBM, это происходит сверху вниз. Разработанные сервисы хранятся в репозитории, куда также есть дополнительный доступ для их доработки. Наиболее подходящей для решения нашей задачи является модель с синхронизацией.

Сервисы ЦОД должны представлять собой автономные (слабосвязанные) компоненты, выполняющие определенный набор функций и взаимодей-

ствующие через стандартизованные интерфейсы. Сервисы реализуют модули *бизнес-логики* достаточно высокого уровня, и для них можно выделить следующие обобщенные характеристики [2, 9, 10]:

- ◆ ориентация на бизнес (сервисы ориентируются на нужды бизнеса, в нашем случае – выполняют определенную бизнес-функцию, необходимую заказчику услуг ЦОД);
- ◆ автономность (сервисы самодостаточны и описываются в терминах интерфейсов, операций, семантики, потребляемых ресурсов, динамических характеристик, политик и свойств сервиса);
- ◆ повторное использование (сервисы могут многократно использоваться для построения новых бизнес-процессов пользователей);
- ◆ инкапсуляция деталей реализации;
- ◆ размещение и видимость (в течение своего жизненного цикла сервисы размещаются и становятся видимы благодаря сервисным метаданным, реестрам и хранилищам);
- ◆ агрегация (на слабосвязанных сервисах строятся объединяющие бизнес-процессы и сложные приложения для одного или нескольких пользователей услуг ЦОД).

Таким образом, применительно к разработке сервисной модели ЦОД можно определить следующие её составляющие:

- разработка моделей типовых бизнес-процессов заказчика;
- выделение сервисов ЦОД на основе декомпози-

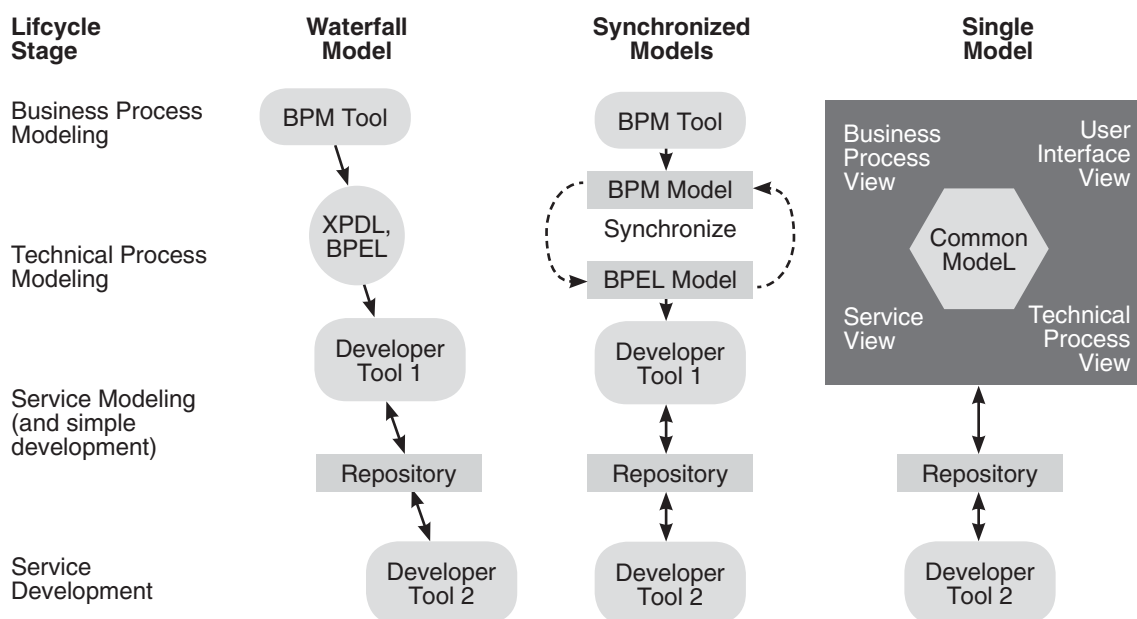


Рис. 2 Модели разработки SOA-приложений [8].

ции типовых бизнес-процессов заказчика;
 ● формирование репозитория сервисов ЦОД.

Потребляемые сервисами ресурсы

SOA разбивает деятельность компании на сервисы – своеобразные черные ящики с интерфейсами входа и выхода. Реализация бизнес-процесса в SOA – это вызов сервисов в определенном порядке. Однако их исполнение должно обеспечиваться доступностью соответствующих ресурсов.

Для выделения типов используемых ресурсов рассмотрим связь между компонентами SOA и ЦОД.

Для начала рассмотрим техническую архитектуру SOA [11].

Техническую архитектуру сервисно-ориентированного подхода можно разделить на слои, начиная от конечного потребителя к поставщику сервиса:

Front-end приложения

В данном контексте имеются в виду приложения, с которыми непосредственно работает пользователь при работе с информационной системой. В равной степени могут быть представлены как настольными приложениями, так и web-приложениями, разработанными на основе использования сервисной архитектуры. Front-end приложения так же известны как композитные приложения (composite applications).

Сервисы

Основные строительные блоки сервисной архитектуры. Каждый сервис инкапсулирует в себе некий бизнес-сервис, то есть набор операций, объединенных одной предметной областью. Сервис сам по себе закрыт от потребителя, использующего нужный функционал сервисов непосредственно или через композитные приложения. Реализация сервиса закрыта, в том числе и от разработчика композитного приложения, известно лишь описание интерфейса сервиса для взаимодействия с ним.

Платформа

Комплекс программно-аппаратных средств, необходимых для работы сервисной архитектуры. Обеспечивает управление инфраструктурой, обмен информацией, преобразование данных, качество сервиса и пр. Платформа полностью закрыта для конечного пользователя, контролируется поставщиком.

Теперь рассмотрим основные ресурсы ЦОД. Выстроим составляющие дата-центров в пирамиду

(рис. 3), упорядочив их снизу вверх по степени доступности потребителю услуг ЦОД.

В частности, физический объект, в котором размещается дата-центр, является сложным комплексом и обладает высокой стоимостью. Он скорее будет арендован у некоего поставщика, нежели спроектирован и построен потребителем. Система жизнеобеспечения (противопожарная система, системы контроля температуры и влажности, обеспечения безопасности) так же скорее будет объектом аренды, а не будет разворачиваться потребителем.

На верхних уровнях пирамиды возможны различные варианты стыковки ответственности потребителя и поставщика. Например, SaaS приложения, иначе известные как «Программное обеспечение в аренду», стыкуются на самом верхнем элементе, когда потребитель пользуется лишь услугой. Более низкие слои скрыты от него, и за них ответственен поставщик. В услуге collocation (размещение сервера), потребитель закрепляет за собой элементы верхнего уровня, включая аппаратную платформу (которую он размещает у поставщика), остальные более низкие уровни предоставляются поставщиком.

Теперь нанесем на получившуюся схему выделенные ранее слои технической архитектуры сервисного подхода (рис.3). Услуга, необходимая потребителю, может быть конкретизирована как функционал front-end приложений, реализующих доступ к информационной системе. В разрезе сервисного подхода роль приложений исполняют сервисы, которые контролируются поставщиком или потребителем, как элемент, предоставляющий некоторый функционал.

Сервисная платформа обеспечивается инфраструктурным программным обеспечением (ПО), которое содержит как классические элементы, характерные для любого ПО (СУБД, ОС), так и средства поддержки сервисно-ориентированных систем, а также более низкими слоями ресурсов ЦОД.

При заказе услуги ЦОД потребитель каждый раз делает выбор, в какой именно точке он встретится с поставщиком, за какую часть ИТ-услуги он будет нести ответственность сам, а какую передаст поставщику. Это определяет перечень ресурсов, которые должен обеспечить поставщик услуг ЦОД для обеспечения функционирования сервиса. В самом общем случае перечень ресурсов включает в себя:

- ◆ Человеческие ресурсы (кто и какое время тратит на реализацию сервиса).
- ◆ Программные ресурсы (используемые приложения).

◆ Аппаратные ресурсы (используемые элементы и характеристики потребления, например, машинное время для исполнения сервиса).



Рис. 3. Слои сервисной архитектуры (техническое представление), наложенные на составляющие дата-центра

Рассмотренная схема позволяет выделить различные типы сервисов, характерные для представления деятельности ЦОД, и сопоставить им различные виды потребляемых ресурсов.

Классификация сервисов применительно к ЦОД

1. Прикладные сервисы

Данный тип сервисов – основной поставщик для верхних уровней бизнес-процессов, определяет корпоративные приложения и механизмы их интеграции для реализации бизнес-процессов. Прикладные сервисы включают в себя:

1.1 Бизнес-сервисы – обеспечивают доступ к прикладным компонентам и унаследованным системам и реализуют автоматизацию бизнес-процессов. Используют ресурсы от уровня «объект» до уровня «услуги».

Ориентированы на представление только бизнес-логики. Делятся на две группы:

1.1.1 Сервисы уровня задачи – это сервисы, чья функциональность основана на определенном бизнес-процессе. Сервисы уровня задачи инкапсулируют логику конкретного бизнес-процесса, поэтому потенциал их повторного использования ниже, чем у других типов сервисов. Сервисы уровня задачи выполняют специфич-

ные, узкоспециализированные задачи, которые необходимы для выполнения данного бизнес-процесса.

1.1.2 Сервисы сущности – это сервисы, функциональность которых порождается одной или несколькими однотипными бизнес-единицами. Примерами бизнес-единиц являются заказ, клиент, счет-фактура. Так как выполняемые сервисами сущности функции основаны на бизнес-единицах, сервис сущности может участвовать во многих бизнес-процессах.

1.2 Базовые сервисы – сервисы нижнего уровня, представленные в сервисном каталоге и доступные для включения в бизнес-сервисы. Используют ресурсы от уровня «объект» до уровня «приложения». Базовые сервисы делятся на две группы:

1.2.1 Базовые прикладные сервисы – стандартизируют различные аспекты доступа к прикладным компонентам и унаследованным системам (например, идентификатор клиента, право на выполнение сервиса и т.д.), причем сами базовые прикладные сервисы могут быть реализованы как компоненты корпоративных приложений. Обычно базовый сервис представляет собой практически не меняющуюся во времени функцию, описываемую в бизнес-терминах. Примером базового прикладного сервиса является проверка действительности кредитной карты. Этот сервис, предоставляемый сегодня специализированными поставщиками, используется практически всеми приложениями, принимающими заказы с оплатой по кредитной карте.

1.2.2 Сервисы интеграции включают в себя механизмы поддержки модели выполнения бизнес-процессов, средства интеграции корпоративных приложений и интеграции корпоративной информации. Средства интеграции информации реализуют унифицированное, логическое представление данных из разных корпоративных приложений и поддерживают агрегирование данных, анализ, корреляцию, синхронизацию модификаций, трансформацию разных моделей и доступ к данным в реальном времени.

2. Инфраструктурные сервисы

Уровень инфраструктурных сервисов представляет собой платформу для прикладных сервисов. Здесь реализуются базовые функции ИТ-среды, которые используются для построения и выполнения приложений, но независимы от них. Используют ресурсы от уровня «объект» до уровня «инфраструктурное ПО».

Выбор методики моделирования

Одна из основных функций сервиса заключается в обеспечении связи между бизнесом и используемыми для его поддержки технологиями. На уровне сервисов должна обеспечиваться формализация требований бизнеса и перевод их на язык технологий. Таким образом, методика моделирования сервисов должна обеспечивать связь между функциональными моделями, характерными для бизнеса, и объектно-ориентированными, характерным для сферы информационных технологий.

Опыт проектов построения SOA показывает, что существующие методологии разработки программного обеспечения и информационных систем, и нотации, такие как OOAD (Object-Oriented Analysis and Design, OOAD), Enterprise Architecture (EA) и Business Process Modelling (BPM) охватывают лишь часть из того, что необходимо для поддержки парадигмы SOA [12].

В нашем случае рассматриваются бизнес-процессы ЦОД и их составляющие – бизнес-сервисы. В такой ситуации естественным является стремление выбрать единую методологию для описания этих объектов.

Описание функционирования бизнес-системы наиболее естественно и понятно осуществляется с использованием методологии структурного анализа. Организация рассматривается как набор функций, осуществляющих преобразование входных потоков (материальных, информационных, финансовых и пр.) в выходные [13,14]. Наиболее распространенными стандартами структурного моделирования являются:

IDEF0 – стандарт описания функциональной архитектуры системы;

IDEF3 – стандарт описания причинно-следственных связей между составляющими процесса;

DFD – стандарт описания потоков данных.

Эти стандарты обеспечивают приемлемое качество отображения бизнес-деятельности, но ни в одном из них не предусматриваются возможности выделения некоторых типовых компонентов (в нашем понимании – сервисов). Такие средства присутствуют в объектно-ориентированных методиках анализа. Однако главная проблема их использования при описании сервисов состоит в том, что степень детализации объектных моделей находится на уровне класса [15]. Класс является слишком низким уровнем абстракции для моделирования сервисов, а принятые в объектных моделях типы связей порождают сильные зависимости между

взаимодействующими объектами, что противоречит идеологии SOA. Невзирая на это, объектно-ориентированный подход в наибольшей мере подходит для поддержки сервисно-ориентированной парадигмы, которая способствует гибкости и возможности быстрого перестроения систем посредством слабого связывания её компонентов.

Таким образом, в качестве инструмента моделирования следует выбрать программный продукт ARIS ToolSet, обеспечивающий возможности совмещения структурного и объектного подхода. Это средство моделирования предлагает множество нотаций для моделирования бизнес-процессов, среди которых можно выделить следующие, наиболее приемлемые для решения нашей задачи [16]:

eEPC – расширенная нотация описания цепочки процесса, управляемого событиями. Позволяет строить описание бизнес-процесса как последовательности чередующихся событий и функций, и может рассматриваться как расширение возможностей IDEF3 и BPMN (Business Process Modeling Notation); возможности совмещения функционального и объектного подхода;

UML – объектно-ориентированный язык проектирования.

Оценка гранулярности сервисов ЦОД

Гранулярность сервисов является одним из важных аспектов SOA. В концепции SOA этот термин означает уровень бизнес-функций и технических решений, поддерживающих эти функции, а также количество процессов, охватываемых или выполняемых сервисом [17].

«Крупнозернистые» сервисы содержат большое количество функций/операций и позволяют выполнять сразу большое количество действий в рамках одного вызова сервиса. Соответственно, сервисы более низкого уровня или «мелкозернистые» сервисы содержат ограниченное количество функций или исполняемых операций. Мелкозернистые задачи (или мелкогранулярные задачи [18]) обычно фокусируются на обмене параметрами или небольшими объемами данных для выполнения специфичной задачи или операции низкого уровня. Для исполнения конкретной операции используется соответствующий сервис. Недостаток использования мелкозернистых сервисов заключается в том, что приложение, в котором вызов каждой мельчайшей операции в процессе оформлен в виде отдельного сервиса, является сложным и дорогостоящим.

Выбранный уровень гранулярности позволя-

ет регулировать размер/формат сервиса и объем функциональности. Это в свою очередь напрямую влияет на возможность повторного использования сервисов и эффективность работы предоставляемых функций в виде сервиса (например, скорость отработки сервиса). В разработке приложений и интеграции систем наиболее удобны более крупные компоненты, которые охватывают определенный набор функциональности из сплоченных мелкозернистых объектов. Таким образом, granularity сервисов определяется в каждом конкретном случае в соответствии с целями проектирования SOA.

В рамках создания сервисно-ориентированной модели ЦОД преследуется цель создания модели, наиболее удобной для описания услуг ЦОД и расширения рынка их потребления. Использование такой модели позволит ЦОДу наиболее адекватно реагировать на изменения на рынке и получить до-

ступ к наиболее рентабельным видам деятельности. Поэтому в рамках проекта ЦОД уровень сервисов должен соответствовать уровню компонент, из которых собирается предлагаемая услуга. В редких случаях сервисы будут включать целиком услуги или, наоборот, выполняемые операции/процессы.

Выводы

Анализ методологических особенностей SOA приводит к заключению о возможности применения её концепций для описания функционирования ЦОД. На основе такого описания перечень услуг ЦОД может быть представлен в форме каталога сервисов, обеспечивающих удобное и понятное заказчику представление этих услуг, а также методические возможности гибкого и оперативного формирования из этих сервисов необходимых заказчику бизнес-процессов. ■

Литература

1. Центры обработки данных: актуальные проблемы и подходы. ВУТЕ/Россия. №3 (123), март 2009. Web:
2. <http://www.bytemag.ru/articles/detail.php?ID=14309> T. Erl, Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall PTR, August 04, 2005, 792 стр.
3. A. Trenaman Using Open Source Software for SOA IONA Technologies 2005
4. Н. Байбородин «SOA на пальцах» IT спец, №10, октябрь 2008
5. Б. Портье (Bertrand Portier), Обзор терминологии SOA: Часть 1. Сервис, архитектура, управление и бизнес-термины, 2008, Web: <http://www.ibm.com/developerworks/ru/library/ws-soa-term1/index.html>
6. Андрей Колесов, Путь к реализации SOA – взгляд Oracle. Web: <http://www.interface.ru/home.asp?artId=5668>
7. <http://www.opennet.ru/openforum/vsluhforumID3/5237.html>
8. V. Swanton, I. Finley, SOA and BPM for Enterprise Applications: A Dose of Reality. AMR Research Report, May 2007
9. А.В. Богданов, Е.Н. Станкова, В.В. Мареев «Сервисно-ориентированная архитектура: новые возможности в свете развития GRID-технологий»
10. Н. Байбородин «Веб-сервисы: проектирование и реализация» IT спец, №10, октябрь 2008
11. В. Боркус «Практическое построение SOA: борьба с мифами». 2006, <http://www.pcweek.ru/themes/detail.php?ID=73883>
12. О. Зиммерманн и др. Элементы сервисно-ориентированного анализа и проектирования. Междисциплинарный подход к моделированию в проектах построения SOA <http://www.ibm.com/developerworks/webservices/library/ws-soad1/>
13. Д.А. Марка., К. МакГоуэн SADT – методология структурного анализа и проектирования., М., Метатехнология, 1993
14. В.И. Грекул, Г.Н. Денищенко, Н.Л. Коровкина. Проектирование информационных систем. – М.: Интернет-Ун-т Информ. Технологий, 2005
15. Г. Буч, Д. Рамбо А. Джекобсон Язык UML. Руководство пользователя, 1999
16. В.В. Репин, Сравнительный анализ нотаций ARIS/IDEF и продуктов их поддерживающих (ARIS TOOLSET/БРWIN). Web: http://www.iteam.ru/publications/it/section_51/article_2518
17. Michael Bell, Service-oriented modeling. Service analysis, design, and architecture. Wiley & Sons, Inc., 2008, 387p.
18. В. Энгельс, Стандартизация определений web-сервисов. Web: http://www.oracle.com/global/ru/oramag/feb2008/feb-08_engels-web-x-57-62.pdf