

УДК 004.27

ВОПРОСЫ ПОСТРОЕНИЯ ИНТЕГРИРОВАННОЙ СИСТЕМЫ НА ОСНОВЕ ПРИНЦИПОВ СЕРВИС-ОРИЕНТИРОВАННОЙ АРХИТЕКТУРЫ

НГУЕН КУОК ХАНЬ

*Белорусский государственный университет информатики и радиоэлектроники
П. Бровка, 6, Минск, 220013, Беларусь*

Поступила в редакцию 4 марта 2011

Исследуются сервис-ориентированная архитектура (СОА), основные принципы СОА, преимущества использования СОА. Рассмотрены трудности построения интегрированной системы на основе принципов СОА и предложены решения для избежания этих трудностей.

Ключевые слова: сервис-ориентированная архитектура, веб-сервис, интеграция приложений.

Введение

Индустрия программного обеспечения прошла через множество архитектурных решений, стремясь реализовать полностью распределенную обработку информации. Традиционные подходы уже достигли предела своих возможностей, но от подразделений информационной технологии (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 помогает подразделениям ИТ достойно встретить возникающие задачи, являясь следующим эволюционным шагом в истории ИТ [1].

СОА интегрирует существующие «лоскутные» подсистемы корпорации в единую информационную систему. Кроме того, при расширении корпорации, при изменении ее структуры СОА позволяет без больших затрат ресурсов преобразовать существующую систему в новый информационный комплекс.

Методика сервис-ориентированной архитектуры

Сервис-ориентированная архитектура – это парадигма организации и использования распределенных информационных ресурсов, таких как приложение и данные, находящихся в сфере ответственности разных владельцев для достижения желаемых результатов потребителя, которым может быть как конечный пользователь, так и другое приложение.

Основными преимуществами использования СОА являются [2]:

- уменьшение стоимости и сокращение времени разработки;
- упрощение структуры информационных систем (ИС) и более тесная координация подразделений ИС за счет схожих задач и схожести структур бизнес-подразделений предприятия;
- уменьшение стоимости интеграции приложений;
- снижение рисков разработки: в процессе модернизации системы новые риски возникают из-за появления новых бизнес-функций или их изменения; СОА позволяет снизить эти риски путем изменения только некоторых сервисов;
- высокая адаптируемость к изменениям бизнес-процессов предприятия;
- возможность повторного использования служб в различных системах;
- возможность модернизировать и объединить всю инфраструктуру предприятия.

Все компоненты СОА-приложений могут быть условно разделены на три группы: службы, клиенты служб и брокеры служб. Первые два понятия очевидны; брокером называется приложение, которое предоставляет услуги по распространению информации о службах и по поиску необходимых служб.

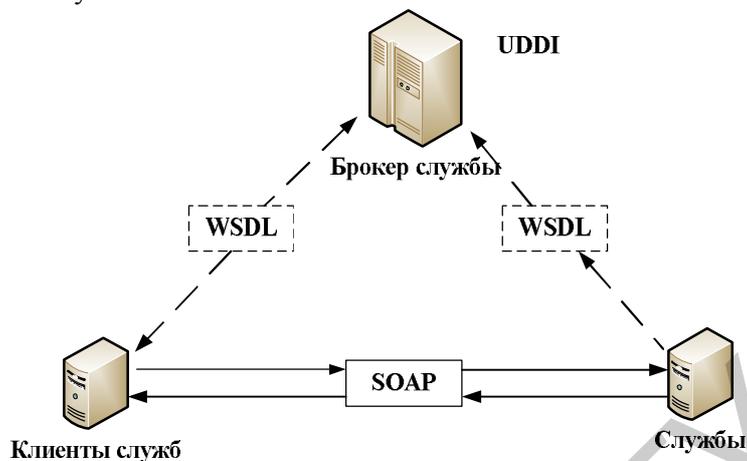


Рис. 1. Модель СОА в случае использования веб-служб

На рис. 1 обозначены: UDDI (Universal Description Discovery and Integration) – универсальное описание, поиск и взаимодействие; WSDL (Web Services Description Language) – язык описания веб-сервисов; SOAP (Simple Object Access Protocol) – простой протокол доступа к объектам.

Каждая служба состоит из интерфейса и реализации этого интерфейса. Интерфейс предоставляет данные о службе, данные для идентификации службы и описание входных/выходных параметров методов службы. Реализация интерфейса представляет собой программный модуль, для клиента службы это "черный ящик".

Клиентом службы может быть приложение абонента сети, серверное приложение, веб-приложение или другая служба.

Брокером в случае использования веб-служб может быть UDDI-сервер, содержащий регистр служб, или собственное решение.

Сервисы регистрируются брокером путем своего описания. Это описание называется формальным контрактом, в нем перечисляются функции сервиса и его параметры.

Принципы СОА

Не существует официального набора принципов СОА. Однако есть набор общепризнанных принципов СОА [3, 4], перечисленных ниже.

1. Сервисы слабо связаны друг с другом.

Слабо связанные сервисы взаимодействуют, опираясь на описание на языке WSDL, которое не зависит от выбранной платформы и языка программирования. Интерфейс скрывает внутреннюю логику приложения, делая сервис независимым от используемой технологии.

2. Сервисы могут использоваться повторно.

Сервисы следует проектировать так, чтобы иметь возможность их повторного использования. При этом экономятся ресурсы (в том числе и время) на построение нового сервиса, а также уменьшается сложность системы.

3. Сервисы обеспечивают формальный контракт для взаимодействия с другими сервисами.

Один сервис обнаруживает и распознает другой сервис через его формальный контракт. Формальный контракт сервиса следует описать так, чтобы другие сервисы могли понять и использовать его в соответствии с описанием.

4. Сервисы скрывают все свои части, кроме той части, которая взаимодействует с внешней средой (формальный контракт).

Недоступность частей сервиса необходима для обеспечения его безопасности и независимости. Сервисы взаимодействуют друг с другом через формальные контракты и работают по принципу «черного ящика».

5. Сервисы могут создаваться из других сервисов. Разрешено создавать новый абстрактный слой, который может ссылаться на любые части сервиса.

Сервисы проектируются с возможностью повторного использования: новые сервисы можно создать путем интеграции существующих сервисов. Сервисы могут создаваться для возможности использования других сервисов на более высоком уровне.

6. Сервисы являются автономными, они не зависят от других сервисов напрямую.

Сервисы следует проектировать так, чтобы они не зависели от платформы и могли сами управлять обработкой своих процессов.

7. Сервисы не имеют механизма запоминания состояния.

Идеальный сервис всегда готов к обслуживанию процесса, при обработке сообщения его работа не прерывается и поэтому не требуется запоминания текущего состояния обработки. Поэтому сервисам не следует запоминать текущее состояние, так как это противоречит принципам слабой связанности сервисов.

8. Сервисы являются общедоступными и видимыми для других сервисов.

Сервис через формальный контракт разрешает доступ всем другим сервисам, этот формальный контракт позволяет другим сервисам определить его функции.

Трудности построения интегрированной системы на основе основных принципов СОА

Построение информационно-управляющей системы (ИУС), интегрированной на основе СОА – это задача, которая должна жестко соответствовать основным принципам СОА. Некоторые часто встречающиеся препятствия на этом пути перечислены ниже.

1. Определение набора сервисов и выполняемых ими функций

Выбор необходимых сервисов и их функций является самым важным шагом в процессе проектирования интегрированной системы на основе принципов СОА. Сервисы проектируются так, что полностью соответствуют взаимосвязям и процессам системы и слабо связаны друг с другом. Для успешного построения сервисов на этапе моделирования системы необходимо тщательно работать с бизнес-аналитиками. Без консультации бизнес-аналитиков все СОА-начинания обречены на неудачу. На этом этапе построены бизнес-процессы системы, на основе которых определены необходимые сервисы для системы СОА. С помощью аналитиков и BPEL (Business Process Execution Language) – язык на основе XML для формального описания бизнес-процессов и протоколов их взаимодействия между собой мы можем облегчить построение сервисов.

2. Логические области сервисов

Сервисы следует разместить в соответственной области так, чтобы не усложнять архитектуру системы и чтобы при конструировании новых сервисов можно было без труда повторно использовать имеющиеся в наличии сервисы. Здесь предлагается два подхода к распределению сервисов в соответствующей области: по их применению в бизнес-процессе или по их назначению в системе. При первом подходе сервисы распределены по течению бизнес-процесса (в глубину) и конструируемые сервисы описывают этот бизнес-процесс. При втором подходе сервисы распределены по их назначению (в ширину) и тогда образуемые сервисы описывают конкретную функцию процесса. На основе предлагаемых здесь подходов по управлению сервисами их интегрирование и использование становятся легче.

3. Структуры сообщений для обмена информацией между сервисами

Существует ряд важных стандартов, позволяющих описать сервисы и определить структуру сообщений между сервисами; это протоколы XML, SOAP, WSDL и др. На основе этих стандартов можно создать собственную структуру сообщений.

4. Физическое распределение сервисов

Сервисы следует распределить в системе так, чтобы беспрепятственно обеспечить множественный доступ к сервисам.

5. Интегрирование сервиса для построения нового сервиса

Сервисы размещаются по некоторому критерию в соответствующих областях. Если при этом ставится задача построения нового сервиса, то обычное решение – интеграция существующих сервисов. При этом необходимо выбрать подходящие сервисы, которые следует интегрировать в соответствии со стандартом.

6. Управление сервисами

Управление сервисом должно обеспечивать:

- быстрый и надежный поиск сервисов;
- доступность ресурсов для функционирования сервисов;
- легкую регистрацию нового сервиса;
- удобство контроля, коррекции, диспетчеризации сервисов.

Для избежания трудностей, которые были описаны в пунктах 4, 5, 6 необходимо использовать распределенные серверы, в которых содержатся сервисы. Эти серверы должны иметь возможности управления сервисами и обеспечить их ресурсами для беспрепятственной работы системы.

7. Маршрутизация сообщений между сервисами

Неэффективная маршрутизация сообщений между сервисами системы увеличивает интервал ожидания ответа на запросы и поэтому снижает эффективность системы. Каждый объект в системе должен иметь уникальный адрес, по которому потребитель отправляет свой запрос. Для обеспечения надежности, стабильности и безопасности передачи сообщений между сервисами необходимо зарезервировать для системы SOA сервисную шину. Сервисная шина присваивает сервисам адреса и обеспечивает маршрутизацию сообщений.

8. Безопасность SOA

Построение механизма для обеспечения безопасности SOA является большой и трудной задачей. Это обобщенное решение, которое зависит от всей инфраструктуры подразделений, то есть от программного и аппаратного обеспечения. Подход к безопасности SOA должен включать в себя три важных компонента:

- безопасность на уровне сообщений (защита сообщений в процессе их передачи); она основана на стандартах WS-Security, WS-Policy, WS-Trust, WS-Privacy, защищающих сообщения в процессе его передачи;
- создание системы безопасности, которую можно было бы интегрировать с уже имеющимися на предприятии системами безопасности и управления;
- детализированную авторизацию, которая может не только обеспечить защиту доступа к сервисам, но и контролировать доступ к бизнес-функциям и данным, используемым в сервисе.

PROBLEMS OF BUILDING AN INTEGRATED SYSTEM BASED ON FUNDAMENTAL PRINCIPLES OF SERVICE-ORIENTED ARCHITECTURE

NGUYEN QUOC KHANH

Abstract

Service-oriented architecture (SOA), emphasizes main advantages of SOA exploiting and the basic principles of SOA is discussed. Problems of integrated system designing based on SOA principles and suggested solutions to resolve them are discussed.

Литература

1. *Артухов Л.* // Инновации в технологиях и бизнесе. 2008. №1. С. 2.
2. *OASIS* // Эталонная модель сервис-ориентированной архитектуры. 2006, №1. С. 1–3.
3. *Er T.* // The principles of service-orientation [Электронный ресурс] Режим доступа: <http://searchsoa.techtarget.com/tip/The-principles-of-service-orientation-part-1-of-6-Introduction-to-service-orientation>.
4. *Erl T.* Service-Oriented Architecture: Concepts, Technology, and Design. Boston, 2005.