

ПРИМЕНЕНИЕ КОМПОНЕНТНЫХ СЕРВИСНЫХ РЕШЕНИЙ ПРИ РАЗРАБОТКЕ ИНФОРМАЦИОННЫХ СИСТЕМ

Манвелидзе А.Б.¹, Ковшов Е.Е.¹

¹ ФГБОУ ВПО Московский государственный технологический университет «СТАНКИН», Москва, Россия (127055, г. Москва, Вадковский пер., д. 3А), e-mail: e.kovshov@stankin.ru

В статье изложен подход к разработке информационных систем строящихся на базе Web-сервисов. Данный подход основан на использовании сервисно-ориентированной архитектуры в качестве основы для среды взаимодействия Web-сервисов. Рассмотрены основные понятия, обозначенные в сервисно-ориентированной модели архитектуры Web-сервисов, разработанной консорциумом W3C «The Service Oriented Model, Web Services Architecture». Модель взаимодействия Web-сервисов в информационной системе представлена на основании цветной сети Петри. Для реализации среды взаимодействия Web-сервисов рассмотрены наиболее интересные и перспективные технологические решения совместимые с Web-технологиями. Одним из преимуществ рассмотренных решений является наличие множества самых разнообразных инструментальных средств для их реализации. Преимуществами представленного подхода является простота расширения и развития информационной системы, а также - возможность беспрепятственной интеграции с аналогичными информационными системами.

Ключевые слова: сервисно-ориентированная архитектура, Web-сервис, компонентные сервисные решения, бизнес-процесс, сети Петри

APPLICATION SERVICE COMPONENT SOLUTIONS FOR DEVELOPING INFORMATION SYSTEMS

Manvelidze A.B.¹, Kovshov E.E.¹

¹ Moscow State University of Technology "STANKIN", Moscow, Russia (127055, Moscow, Vadkovsky ln., 3A), e-mail: e.kovshov@stankin.ru

The article describes an approach to the development of information systems built on the basis of Web services. This approach is based on use of service-oriented architecture as a basis for environment interaction of Web services. Basic concepts defined in a service-oriented architecture, Web services, developed by the W3C in "The Service Oriented Model, Web Services Architecture". The model of interaction of Web services in the information system presents on the basis of colored Petri nets. For the implementation of the environment Web services interactions are considered the most interesting and promising technological solutions compatible with Web technologies. One of the advantages of the considered solutions is the presence of great variety of tools for its implementation. The advantages of the presented approach is the ease of expansion and development of the information system, as well as the opportunity to seamlessly integrate with similar information systems.

Keywords: service-oriented architecture, Web service, component service decisions, business-process, Petri nets

В настоящее время уже разработано и разрабатывается множество информационных систем, функционирующих на основе Web-технологий [1, 8-10]. Существенной проблемой при разработке крупных информационных систем является разнообразие методов и подходов, позволяющих осуществлять решения многочисленных технических и технологических задач [2,3]. Несмотря на то, что эти методы позволяют справляться с различными задачами, их разнообразие приводит к возникновению проблемы несовместимости программных решений.

В свою очередь, проблема несовместимости программных решений имеет большое значение не только в рамках технических аспектов разработки информационных систем предприятия, но и при осуществлении взаимодействия с информационными системами

партнеров по бизнесу. К тому же возрастающая сложность самих информационных систем увеличивает разрыв между технологическими возможностями информационной системы и задачами, для решения которых она предназначена.

Одним из наиболее популярных решений для их разработки является использование Web-сервисов. При этом в качестве перспективного направления выступает применение сервисно-ориентированной архитектуры (СОА) в качестве среды взаимодействия Web-сервисов [9].

СОА представляет собой программное обеспечение в виде Web-сервисов, имеющих слабо связанный характер взаимодействия, но при этом обладающих высокой совместимостью, гибкостью и возможностью многократного использования. Основная задача архитектуры такого рода заключается в организации взаимодействия между сервисами и программными агентами, предоставляющими и запрашивающими сервисы [8, 9].

В качестве действий выступают любые изменения объекта информационной системы при осуществлении исходящих и/или входящих сообщений от программных агентов.

Программные агенты – это программы, которые участвуют в действиях, реализуют и запрашивают сервисы. По сути, программные агенты представляют собой работу программ, которые управляют сервисами, то есть выполняют их и осуществляют доступ к ним.

Сервисы представляют собой ресурс, представленный в виде программного обеспечения, а также - совокупности компонентов, объектов и взаимосвязанных задач, функциональное назначение которых совпадает, как с точки зрения поставщика, так и потребителя Web-сервисов. Компоненты, организованные в виде сервисов являются автономными, имеют определенный интерфейс и зачастую поддерживают несколько различных технологий интеграции. В качестве сервиса в СОА может выступать, как целое программное приложение, так и отдельные его функциональные модули. Сервисы могут реализовать как бизнес-логику, так и функции более низкого уровня, в том числе некоторые системные функции [9, 10].

СОА позволяет представить программное обеспечение или его компоненты в виде сервисов со стандартными интерфейсами, которые могут быть многократно использованы в различных информационных задачах. Слабая связанность сервисов в концепции СОА дает возможность потребителю самостоятельно выбрать необходимые программные компоненты, и, соответственно, не нести финансовые издержки на дополнительные невостребованные функции.

Благодаря унифицированным интерфейсам сервисов в СОА, возможно осуществлять свободный доступ к ним через различные приложения, по различным каналам связи. Главное

достоинство применения СОА – это возможность интеграции между Web-сервисами и традиционными информационными технологиями потребителей [9].

Модель взаимодействия Web-сервисов в информационной системе реализуется посредством цветной сети Петри:

$$C = (P, T, I, O, \mu), \quad (1)$$

где $P = \{p_1, p_2, \dots, p_n\}$ – конечное множество позиций, $n \geq 0$;

$T = \{t_1, t_2, \dots, t_m\}$ – конечное множество переходов, $m \geq 0$;

Множества позиций и переходов не пересекаются $P \cap T = \emptyset$;

I – входная функция отображения переходов в комплекты позиций $I : T \rightarrow P^\infty$ или отображения позиций в комплекты переходов $I : P \rightarrow T^\infty$;

O – выходная функция отображения переходов в комплекты позиций $O : T \rightarrow P^\infty$ или отображения позиций в комплекты переходов $O : P \rightarrow T^\infty$;

μ – функция, отображающая множество позиций P во множество переменных задачи N , $\mu : P \rightarrow N$.

При таком рассмотрении множество позиций P соответствует переменным задачи, а множество переходов соответствует функциям, осуществляемым над этими переменными.

Сервисы взаимодействуют друг с другом в рамках бизнес-процессов. При этом бизнес-процесс может быть определен как набор взаимосвязанных задач, относящихся к деятельности, имеющей функциональные границы. Бизнес-процесс управляет потоком событий, вызывает и координирует сервисы и создает контекст для их взаимодействия [4]. Бизнес-процесс представляет собой абстракцию и, будучи отделенным от реализации сервисов, бизнес-процесс заботится о ходе выполнения деятельности производимой сервисами, в рамках конкретной информационной задачи, характерной для данного бизнес-процесса.

В большинстве случаев представленная модель взаимодействия Web-сервисов целенаправленно описывает бизнес-процесс, и лишь в отдельных случаях модель может описывать некоторую совокупность различных бизнес-процессов.

В общем случае бизнес-процесс включен в модуль хореографии и оркестровки сервисов, который необходим для объединения Web-сервисов при создании высокоуровневых бизнес-процессов, в которые вовлечены, например, несколько промышленных предприятий [1, 9].

Хореография определяет последовательность и условия, при которых несколько взаимодействующих независимых программных агентов обмениваются сообщениями в целях выполнения задач, предназначенных для достижения целевого состояния, и

представляет собой модель последовательности операций, состояний и условий, которые управляют сервисами, принимающими участие во взаимодействии. Результатом такого взаимодействия, установленного на основании хореографии сервисов, является выполнение определенных функций [4, 5].

Оркестровка предназначена для определения бизнес-процесса, в рамках которого взаимодействуют различные сервисы. Она заключается в определении механизма сервисного взаимодействия, основывающегося на сформулированной для бизнес-процесса логике, и представляет собой последовательность действий совершаемых сервисами [9].

Стандарты оркестровки и хореографии должны удовлетворять ряду технических требований, обеспечивающих разработку бизнес-процессов с привлечением Web-сервисов. Эти требования касаются как языка, служащего для описания последовательности действий в бизнес-процессе, так и инфраструктуры для его выполнения.

Для обеспечения механизма обнаружения сервисов в большинстве случаев используется модуль, представляющий собой реестр сервисов [8]. В случае применения реестра сервисов спецификации UDDI при организации взаимодействия корпораций и пользователей в среде сети Internet [1, 9], итерационный процесс корректировки запросов практически исключается, поскольку уже в процессе формулирования запроса, запрашивающий получает исчерпывающую информацию об источниках информации (например, адреса, контакты, сфера интересов, отношение к предметной области, перечень конкретных разработок и их спецификации и т.д.).

Реестр сервисов хранит информацию о каждом сервисе, входящем в информационную систему, а также об интерфейсах, которые эти сервисы предоставляют. Он позволяет определить наличие или отсутствие того или иного сервиса, необходимого для реализации бизнес-процесса.

Для реализации среды взаимодействия Web-сервисов можно воспользоваться следующими популярными технологическими решениями совместимыми с Web-технологиями [1]:

- для организации обмена сообщениями используется протокол SOAP (Simple Object Access Protocol);
- для описания интерфейса Web-сервисов используется язык WSDL (Web Services Description Language);
- для организации системы распознавания сервисов используется спецификация UDDI (Universal Description, Discovery and Integration);
- для описания бизнес-процессов используется язык BPEL (Business Process Execution Language).

Протокол SOAP устанавливает стандарт взаимодействия «клиент-сервер» и регламентирует, как должен осуществляться вызов, передаваться параметры и возвращаемые значения в распределенных информационных системах.

WSDL определяет службу, как набор конечных точек или портов. При этом специфичные параметры реализации – формат данных, настройки транспортного протокола – отделены от определения абстрактных сообщений, операций и типов портов, что позволяет многократно использовать эти определения. Конкретный тип порта с указанным протоколом передачи и форматом сообщений образует связывание, ассоциация связывания с конкретным сетевым адресом образует порт, а множество портов образует службу.

Описания языка WSDL активно используют различные пространства имен. Кроме собственных имен, язык WSDL часто использует имена типов и элементов языка протокола SOAP. Пространство имен языка WSDL часто описывается как пространство имен по умолчанию [1, 4].

Спецификация языка BPEL поддерживает как структурные действия для управления потоком работ бизнес-процесса в целом, так и базовые действия, которые включают взаимодействия с внешними Web-сервисами [9]. Структурные действия управляют потоком работ бизнес-процесса в целом, определяя последовательность вызова Web-сервисов. Эти действия также поддерживают выполнение циклов и динамическое ветвление. По существу, они составляют основную логику программирования в BPEL. Два других важных элемента BPEL – партнерские связи и переменные.

Переменная идентифицирует определенные данные в потоке сообщений. Когда процесс BPEL получает сообщение, он присваивает значение соответствующей переменной, чтобы последующие запросы могли обратиться к этим данным. Переменные используются для управления долговременным хранением данных в ходе обработки запросов Web-сервисов.

Партнерская связь может обозначать любой сервис, с которым взаимодействует данный процесс. Каждой партнерской связи приписана определенная роль в реализуемом бизнес-процессе.

Все перечисленные выше технологические решения основаны на языке разметки XML (Extensible Markup Language), являющимся на сегодняшний день наиболее популярным и широко используемым в среде Web-технологий [2,5]. XML может обеспечить исчерпывающее представление набора данных. Популярность этого языка можно объяснить его высокой гибкостью и расширяемостью. На практике синтаксис XML позволяет модернизировать и разрабатывать специализированные XML-схемы для обработки различных данных [6].

Одним из преимуществ использования этих решений при разработке информационных систем на базе СОА является наличие множества инструментальных средств для их реализации. Среди них, наряду с коммерческими, существует множество средств разработки, предоставляемых по универсальной общественной лицензии GNU (General Public License), которая допускает свободную эксплуатацию, модифицирование, копирование исходных кодов программных продуктов, попадающих под ее область действия, в том числе и в коммерческих целях [5].

В случае применения перечисленных выше технологических решений, информационную систему можно представить в виде, изображенном на рис. 1.



Рис. 1. Модель информационной системы на основе Web-сервисов

Как видно из рис. 1, ключевым элементом модели является корпоративная сервисная шина ESB (Enterprise Service Bus), которая осуществляет взаимодействие всех программных приложений через единую точку, которая при необходимости обеспечивает транзакции, преобразование данных, сохранность обращений и т.д. Корпоративная сервисная шина ESB [7-9] применяется в качестве гибкой связывающей инфраструктуры для интеграции бизнес-процессов и реализует транспортный уровень, обеспечивающий:

- связывание запросов к службам по одному протоколу и адресу с другим протоколом;
- преобразование форматов данных;
- поддержку различных моделей безопасности и транзакционных моделей;
- поддержку работы коммуникационных протоколов между несколькими платформами;
- поддержку различных моделей обмена сообщениями (синхронная, асинхронная, на основе событий, публикация, подписка).

Кроме того, корпоративная сервисная шина ESB обеспечивает поддержку посреднических функций, упрощающих взаимодействие служб, и поддерживает различные способы подключения/отключения существующих адаптеров или бизнес-соединений [9, 10].

Реализация информационных систем на базе Web-сервисов с использованием COA предоставляет следующие преимущества:

- возможность создания и многократного использования программных компонент;
- возможность формирования из имеющихся программных компонент новые бизнес-процессы;
- интеграцию существующих программных решений в единую системную архитектуру;
- независимость организации программных решений от языков программирования и программно-аппаратных платформ;
- наличие унифицированных интерфейсов обмена информацией между сервисами и доступа к ним.

Проприетарные инструментальные средства для поддержки полного жизненного цикла информационных систем, реализованных с применением COA, представлены крупными поставщиками программного обеспечения: IBM, Oracle, Microsoft, Sun Microsystems, Tibco, Software AG, SAP AG [8, 9]. Эти высокоэффективные инструменты позволяют в полной мере выполнить основные этапы разработки и внедрения COA такие, как: планирование (анализ потребностей в элементах COA), согласование (выработка подхода к управлению COA), разработка (создание программного обеспечения и ввод COA в эксплуатацию), функционирование (мониторинг внедренных программных решений) [1, 9]. Таким образом, информационные системы, построенные с применением концепции сервисно-ориентированной архитектуры, предоставляют конечному пользователю мощный, многофункциональный и удобный для применения программный комплекс, позволяющий с минимальными затратами обеспечить его эксплуатацию и планомерное, поступательное развитие на основе компонентных сервисных решений.

Список литературы

1. Баранова И.В. Управление интегрированными средствами поддержки распределенных приложений на основе анализа информационных ресурсов автоматизированной производственной среды: Автореф. дис. канд. экон. наук. – Москва, 2013. – 22 с.
2. Гофман В.Э., Хомоненко А.Д. Работа с базами данных в Delphi – 2-е изд. – СПб: БХВ-Петербург, 2002. – 624 с.:ил.

3. Ковшов Е.Е., Батова М.М., Митропольский Н.Н. Комплексный подход в интеллектуальном анализе данных прикладной информационной системы // Вестник Университета // Развитие отраслевого и регионального управления, № 9 М.: ГУУ. 2011. – С. 86-89.
4. Ковшов Е.Е., Батова М.М., Смирнов О.С. Разработка информационных систем инновационного промышленного предприятия на основе унифицированного модульного подхода // Инновации, 2011. – №5. – С.102-106.
5. Ковшов Е.Е., Борисенко Е.В. Применение инструментальных средств обработки корпоративной информации на основе программно-аппаратных технологий // Вестник МГТУ «Станкин». – М.: МГТУ «Станкин», 2010. – №3, с. 123-129.
6. Ковшов Е.Е., Мартынов П.Н., Хуэ Н.Н., Киен Ф.Ч. Автоматизация оценки эффективности взаимодействия конечного пользователя с обучающей информационной системой // Открытое образование. – М., 2010. – № 1. – С. 37-43.
7. Ковшов Е.Е., Смирнов О.С. Методы оценки и повышения надёжности информационных сред промышленного предприятия // Динамика сложных систем – XXI век. – 2012. – №2. – С. 112-116.
8. Костров И.А., Ковшов Е.Е. Построение и реализация информационной системы для управления персоналом промышленного холдинга // Современные проблемы науки и образования. – 2012. – № 6; URL: www.science-education.ru/106-7615 (дата обращения: 17.12.2013)
9. Костров И.А. Модели и инструментальные средства прикладной информационной системы на основе сервисно-ориентированной архитектуры: Автореф. дис. канд. экон. наук. – Москва, 2013. – 28 с.
10. Смирнов О.С. Конвергенция неоднородных информационных сред на основе кроссплатформенных программных компонент: Автореф. дис. канд. техн. наук. – Москва, 2013. – 23 с.

Рецензенты:

Климанов В.П., д.т.н., профессор, профессор кафедры информационных систем ФГБОУ ВПО МГТУ «СТАНКИН», г. Москва.

Волков А.Э., д.т.н., профессор, декан факультета МТО ФГБОУ ВПО МГТУ «СТАНКИН», г. Москва.