

УДК 681.327

Принципы построения семантической сервис-ориентированной архитектуры

Кашалкин Д.Ю., Курчидис В.А.
Ярославский государственный университет
150 000, Ярославль, Советская, 14

получена 1 марта 2007

Аннотация

Предлагаются принципы построения семантической сервис-ориентированной архитектуры как средства достижения адаптивности и гибкости инфраструктуры распределенных информационных систем. Рассматриваются различные варианты построения данной архитектуры, ее основные элементы, их взаимодействие, методы работы с ними. Предлагаемая архитектура основана на определении семантического сервис-ориентированного архитектурного стиля и предоставляет семантические средства взаимодействия распределенных компонентов, обеспечивая новый уровень решения проблемы интеграции информационных систем.

1. Введение

Создание распределенных информационных систем, способных быстро адаптироваться к изменяющимся условиям динамичной среды, является важной проблемой для непрерывно развивающихся сфер деятельности. В настоящее время информационные службы компьютерных сетей, в частности WWW, представляют собой синтаксические сети, в которых представлена только структура контента, предназначенная для восприятия людьми, а не машинами. Между тем адаптивные системы могут быть получены только на базе эффективного решения проблемы интеграции приложений в распределенных системах. В связи с этим следует указать два параллельных подхода к решению проблемы интеграции.

Первый подход заключается в разделении бизнес-логики приложения и логики представления. Бизнес-логика может быть полностью отдельной и предлагаться в виде службы с определенным API (Application Program Interface) - так называемая идея сервис-ориентированной архитектуры (service-oriented architecture, SOA) [1,2], которая представляет собой развитие компонентной, объектно-ориентированной и распределенной систем.

Второй подход к решению проблемы интеграции приложений - использование архитектурных решений Semantic Web, которые предоставляют средства для эксплицитного задания контента ресурсов во Всемирной сети [3,4]. Главными элементами этой технологии являются языки разметки и онтологии. В настоящее время в основе Semantic Web лежат языки разметки ресурсов RDF, OWL [4,5]. Они обладают строгой семантикой, что позволяет делать логические выводы, основываясь на утверждениях языка относительно того или иного ресурса.

Для построения гибких распределенных информационных систем в работе предлагается создание гибридной архитектуры, сочетающей в себе достоинства обоих вышеуказанных направлений. Сервисная ориентация может значительно улучшить семантическую архитектуру, предоставляя средства для автоматического взаимодействия программ. В свою очередь, семантическое наполнение позволит сервисам решать следующие задачи:

- представлять задачу, которую выполняет сервис, и рассуждать о ней, обеспечивая автоматический поиск службы, основанный на эксплицитном развертывании службы и описании ее функциональности;
- понимать смысл обмениваемой информации;
- представлять предусловия, необходимые для использования служб, и эффекты, характеризующие собой вызов сервиса;
- описывать процесс композиции сервисов для выполнения сложных служб.

В качестве основы построения архитектуры выбрана теория архитектурных стилей Роба Филдинга [6], одного из авторов таких стандартов, как HTTP 1.1, URI, а также создателя так называемого архитектурного стиля REST - модели, представляющей собой образец построения ПО для Всемирной сети. На базе данной модели в работе определяется семантический сервис-ориентированный стиль и порождаемая им семантическая сервис-ориентированная архитектура (CCOA, SSOA - semantic service-oriented

architecture), основные принципы ее построения, структурные элементы, необходимые для полноценного функционирования системы, особенности их взаимодействия и методы работы с ними.

2. Определение конечных свойств ССОА на основе теории архитектурных стилей

Согласно теории Роба Филдинга, любая архитектура представляет собой конфигурацию (т.е. структуру архитектурных взаимоотношений) элементов: данных, компонентов, коннекторов, на взаимодействия которых наложены определенные ограничения с целью достижения желаемого конечного набора архитектурных свойств.

От выбора и упорядочивания элементов системы, в частности от ограничений, накладываемых на элементы архитектуры, зависят конечные архитектурные свойства системы как функциональные, так и нефункциональные (относительная простота дальнейшей эволюции, возможность повторного использования компонентов, эффективность, динамическая расширяемость). Этот набор ограничений, лежащий в основе архитектуры и определяющий ее свойства, в работе [6] назван архитектурным стилем. Четкое определение конечных архитектурных свойств позволяет уже на стадии проектирования подобрать необходимые ограничения, накладываемые на компоненты системы и на их взаимоотношения, и, таким образом, получить представление об общем виде архитектуры в целом. Поэтому для построения определяемой семантической сервис-ориентированной архитектуры в данной работе решаются следующие задачи:

- определение конечных свойств архитектуры;
- определение составляющих элементов системы;
- определение ограничений, накладываемых на элементы системы для получения желаемых свойств.

В работе предполагается, что конечные свойства семантической сервис-ориентированной архитектуры и составляющие ее элементы следуют из происхождения самой сервис-ориентированной архитектуры (COA) и Semantic Web.

COA представляет собой развитие распределенного объектного стиля. Примером реализации COA является архитектура web-сервисов [7]. Цель этого архитектурного стиля - построение такого типа системы, при котором все элементы слабо связаны. Это свойство возникло как решение одной из главных проблем распределенного объектного стиля. Для этого применяется два основных ограничения.

1. Все участвующие компоненты имеют простые интерфейсы, в которых закодирована основная семантика. Интерфейсы должны быть универсально доступны для всех провайдеров сервисов и потребителей.
2. Через интерфейсы происходит обмен сообщениями, которые обладают следующими характеристиками:
 - дескриптивность - описательный, а не инструктивный характер сообщения;
 - ограниченность структуры - запрос на выполнение сервиса не может быть выполнен только из-за того, что сообщение имеет структуру, использует словарь или закодировано в формате, который не понятен провайдеру сервиса. Таким образом, ограничение словаря и структуры сообщения необходимо для эффективной коммуникации. Чем более ограничено сообщение, тем легче его понять.

На рис. 1 определены архитектурные свойства сервис-ориентированного стиля. Один из существенных недостатков COA с точки зрения архитектуры состоит в том, что обмен сообщениями не несет практически никакой семантической нагрузки. Вся семантика достигается через ограничение структуры сообщения и используемого словаря. Именно этот недостаток приводит к снижению такой характеристики архитектуры, как расширяемость. На примере web-сервисов этот недостаток проявляется в следующем: XML гарантирует синтаксическую межоперабельность, что позволяет сервисам разбирать сообщения, передаваемые друг другу, но этого недостаточно для семантического понимания контента сообщения. Отсутствие эксплицитной семантики инфраструктуры web-сервисов при их реализации требует, чтобы интерпретация обмениваемой информации была жестко привязана к коду компонентов. Любые изменения в модели требуют от программистов модификации кода. Подобное условие является неприемлемым в условиях расширяемости и спонтанной масштабируемости в информационных сетях [6].

В качестве основного метода отделения семантики данных от их структуры предлагается использование онтологий - основных элементов представления ресурсов в Semantic Web [4]. Для устранения вышеописанного недостатка COA предлагается введение дополнительного ограничения на элементы архитектуры COA - отделение смысловой нагрузки данных от их структуры и формальное представление

Рис. 1. Качественные свойства сервис-ориентированной архитектуры

семантического контента с целью машинной обработки. Введение нового ограничения на элементы сервис-ориентированного стиля способствует уменьшению степени связи между компонентами архитектуры, повышению ее гибкости и расширяемости и определяет новый семантический сервис-ориентированный архитектурный стиль (рис. 2).

Рис. 2. Качественные свойства семантической сервис-ориентированной архитектуры

3. Модель ССОА

Использование онтологий как дополнительного архитектурного элемента позволяет решить основное ограничение, накладываемое в семантическом сервис-ориентированном стиле: отделение семантики от структуры контента. Архитектурная модель ССОА основывается на использовании следующих основных элементов:

- компоненты: агенты пользователя, сервисы, посредники;
- коннекторы: протоколы передачи сообщений, менеджер сервисов (например, реестр), онтологии (сервиса, пользователя, доменные);
- данные: сообщения.

Модель семантической сервис-ориентированной архитектуры может быть представлена в виде структуры на рис. 3.

Особенности модели ССОА определяют способы организации взаимодействия структурных элементов между собой. Один из предлагаемых способов представлен в виде модели на рис. 4.

Обычно инициатором взаимодействия является агент запрашивающей стороны. В этом случае агент пользователя должен получить адрес агента провайдера либо непосредственно от провайдера, либо через службу поиска. Поиск носит семантический характер - в процессе поиска используются онтологии пользователя и онтологии сервиса, а также посредник, осуществляющий преобразование запросов потребителя

Рис. 3. Семантическая сервис - ориентированная архитектура

сервиса из концептов и взаимоотношений пользовательской онтологии к концептам и взаимоотношениям онтологии провайдера сервиса.

Возможен вариант, когда агент провайдера является инициатором обмена. В этом случае агент провайдера получает адрес запрашивающего агента и осуществляется обмен сообщениями, в которых структура отделена от смысловой нагрузки благодаря использованию онтологий.

Рис. 4. Модель взаимодействия структурных элементов ССОА

Использование онтологий как отличительных структурных элементов ССОА определяет три различных способа построения архитектурной модели.

1. Подход с использованием единой онтологии - характеризует собой такое построение архитектуры, при котором используется одна глобальная онтология, фактически предоставляющая общий словарь для спецификации семантики ресурсов. Каждый информационный ресурс должен быть описан с использованием этой глобальной онтологии.
2. Множество онтологий - каждый информационный ресурс описывается своей собственной онтологией. В этом случае ресурс может развиваться независимо от других ресурсов или их онтологий,

поэтому при таком построении архитектуры сравнительно легко вносить изменения в ресурсы, добавлять новые ресурсы и т.д. (улучшаются качественные характеристики архитектуры: расширяемость, масштабируемость, способность к эволюционным изменениям). Однако отсутствие общего словаря делает почти невозможным сравнение онтологий различных ресурсов.

3. Гибридный подход - аналогично предыдущему подходу семантика каждого ресурса описывается своей собственной онтологией. Но для того, чтобы онтологии ресурсов были сравнимы друг с другом в семантическом плане, они строятся на основе глобальной онтологии, которая содержит базовые термины (примитивы) предметной области.

Другим основным структурным элементом архитектуры является сервис, который может быть представлен как некоторая функция, правило, осуществляющее преобразование набора входных параметров в выходные, т.е. $F : Inp \rightarrow Out$, где Inp - конечное множество входных параметров, а Out - конечное множество выходных параметров. Основная проблема, связанная с представлением сервиса в ССОА, заключается в эффективном описании изменений мира, порождаемых выполнением сервиса, и условий, предшествующих выполнению действия (так называемых предусловий - Preconditions). Таким образом, предусловия специфицируют условия (на основании входных данных), при выполнении которых действие может быть осуществлено. В этом случае в множестве Out следует выделить два подмножества:

- *output* - непосредственно выходные параметры;
- *effects* - эффекты, которые сервис может оказывать на предметную область, изменяя состояние окружающей его среды.

Тогда необходимое условие выполнения сервиса может быть записано в виде формулы:

$$S : Pre \rightarrow Output \cap Effects.$$

Необходимое и достаточное условие может иметь вид формулы:

$$S : Pre \cap Inp \rightarrow Output \cap Effects.$$

Смысл данной формулы заключается в том, что при выполнении всех предусловий и корректном вводе входных параметров будут получены выходные параметры и эффекты от выполнения службы. Для задания множеств In , Pre , $Output$, $Effects$ используется формализм дескриптивной логики ($TBox$ и $ABox$) [8]. Семантику перехода от множества входных данных (и предусловий) к множеству выходных данных (и эффектов) предлагается определять, как логическую программу (правило).

Целесообразно выделить следующие типы процессов, которые могут быть организованы в семантической сервис-ориентированной архитектуре: опубликование, обнаружение, селекция, композиция, вызов.

Опубликование семантического сервиса позволяет агенту пользователя обнаружить сервис на основе анализа конечного назначения сервиса и его свойств. Опубликование происходит через семантический регистр, в котором регистрируются описания сервисов с использованием онтологии.

Обнаружение сервиса представляет собой поиск семантического соответствия между описанием запроса и описанием опубликованного сервиса. Запросы могут включать имя сервиса, входные, выходные данные, предварительные условия, необходимые для выполнения сервиса, и прочие атрибуты, используемые для поиска по семантическому регистру. Поиск соответствия может проходить и на уровне сопоставления задач пользователя и целей, которым служит тот или иной сервис. В этом случае необходим процесс доказательства (через сравнение концептов соответствующих онтологий) того, что постусловия и эффекты от применения сервиса означают выполнение задачи.

Селекция сервиса выполняется в случае, если критерию запроса удовлетворяют более чем один сервис. Для селекции можно использовать так называемые нефункциональные атрибуты службы, как, например, цена или качество.

Композиция осуществляется в случае, если для выполнения пользовательского запроса недостаточно вызова одного сервиса - необходимо последовательное выполнение нескольких сервисов.

Вызов семантического сервиса осуществляется в несколько этапов:

- определение онтологий сервиса и онтологии домена (области);
- проверка входных данных через инициализированные онтологии;
- вызов сервиса (или инициализация некоторого потока в случае композиции сервисов) выполняется через сведение описания сервиса к физическим данным (машина, порт и т.д.) и обращения к полученным данным.

Расширение семантической сервис-ориентированной архитектуры новыми элементами приводит к тому, что для выполнения большинства вышеуказанных процессов требуется решение проблемы согласования онтологий (например, онтологии задач пользователя и онтологии сервиса) в процессах поиска, селекции, композиции, вызова сервиса - так называемая задача гетерогенности в Semantic Web. С целью решения этой задачи в рассматриваемую архитектуру вводится такой элемент, как посредник. Его задача сводится к процессу выравнивания онтологий (alignment) - нахождение соответствующих друг другу сущностей в онтологиях. Посредник может быть реализован на основе машины дескриптивного логического вывода, в которой для согласования онтологий используется табличный алгоритм дескриптивной логики. Помимо этого посредник используется для получения логических выводов в специально разработанных алгоритмах обнаружения и композиции сервисов.

Заключение

Предлагаемый архитектурный подход направлен на эффективное решение проблемы интеграции компонентов в распределенных информационных системах и основан на определении семантического сервис-ориентированного стиля. В нем сочетаются достоинства существующих технологий, получивших распространение при организации современных распределенных систем - COA и Semantic Web.

Использование онтологий как дополнительного архитектурного элемента позволяет решить основное ограничение, накладываемое в семантическом сервис-ориентированном стиле, - отделение семантики от структуры компонентов. Семантический уровень взаимодействия компонентов основывается на согласовании их онтологий, которое может быть выполнено с использованием процессоров вывода дескриптивной логики.

Построение распределенных систем на основе слабосвязанных компонентов семантической сервис-ориентированной архитектуры обеспечивает повышение гибкости и адаптивности информационных систем.

Список литературы

1. Stojanovic, Z. Service-oriented Software System Engineering / Z. Stojanovic. - Idea Group Publishing, 2005. - 435 с.
2. Черняк, Л. SOA - шаг за горизонт / Л. Черняк // Открытые системы. - 2003. - №9. - С. 31-35.
3. Berners-Lee, T. What the Semantic Web can represent / T. Berners-Lee. - <http://www.w3.org/DesignIssues/RDFnot.html>.
4. Кашалкин, Д.Ю. Принципы организации Semantic Web / Д.Ю. Кашалкин, А.Н. Лататуев // Модел. и анализ информ. систем. - 2005. - Т. 12, № 2. - С. 7-11.
5. Кашалкин, Д.Ю. Принципы создания метаописаний ресурсов во всемирной сети / Д.Ю. Кашалкин, А.Н. Лататуев // Модел. и анализ информ. систем. - 2005. - Т. 12, № 2. - С. 12-16.
6. Fielding, R.T. Architectural Styles and the Design of Network-based Software Architectures / R.T. Fielding. - <http://www.ebuilt.com/fielding/pubs/dissertation/top.htm>.
7. W3C Working Group. Web Services Architecture. - <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>.
8. Baader, F. The description logic handbook - Theory, implementation and applications / F. Baader. - Cambridge University Press, 2003. - 583 с.

Construction principles of semantic service-oriented architecture

Kashalkin D.Y., Kurchidis V.A.

Construction principles of service - oriented architecture as a means of achieving adaptability and flexibility of distributed information system's infrastructure are suggested. Different variants of constructing of this architecture, it's main elements, their interoperability, methods of working with them are considered. The suggested architecture is based on the definition of the semantic service-oriented architecture style and provides semantic means of distributed components interaction thus providing new level of decision of information systems integration problem.