

Конструирование оптимального набора сервисов для решения комплексных задач

Александр Дегтярёв
Санкт-Петербургский
Государственный
Университет
Санкт-Петербург, Россия
e-mail: deg@csa.ru

Вадим Гуськов
Санкт-Петербургский
Государственный
Электротехнический Университет
“ЛЭТИ”
Санкт-Петербург, Россия
e-mail: vadim.p.guskov@gmail.com

Александр Ерошкин
Санкт-Петербургский
Государственный
Электротехнический Университет
“ЛЭТИ”
Санкт-Петербург, Россия
e-mail: eroshkin.a@pochta.ru

АННОТАЦИЯ

Формирование требуемой системы из готовых многократно используемых компонент позволяет снизить время разработки и даёт возможность адаптировать систему к изменяющимся условиям на этапе исполнения. Проблемой подхода является сложность выбора комплексированных блоков в случае множественности и нетривиальности связей между ними. Решением проблемы может стать автоматизация процесса связывания. В работе рассматриваются требования к подобной системе. На основании существующих разработок предложен пример её реализации.

Ключевые слова

Adaptive Service Grid, динамическая композиция сервисов, коэффициенты уверенности, сервис-ориентированная архитектура.

1. ВВЕДЕНИЕ

Существует подход к разработке программного обеспечения, в рамках которого результирующая система не строится с нуля, а собирается из уже готовых компонентов. В качестве подобных составных блоков могут выступать различные сущности: программные модули, множество которых решает какую-либо сложную задачу; обособленные компьютерные системы, используемые поочерёдно для преобразования данных; информационные источники, комплексирование которых приводит к принятию того или иного решения и т.д. Главными требованиями, предъявляемыми к используемым компонентам, являются их слабая связность и наличие стандартизованных средств их использования. Эти свойства позволяют комплексировать компоненты в независимости от их внутренней реализации, а также взаимно заменять блоки, решающие однотипные задачи. Такой подход даёт возможность не только однократно сконструировать систему для решения конкретной задачи, но и динамически адаптировать её к изменяющимся условиям, переставив на этапе исполнения.

Проблема такого подхода состоит в трудности принятия решения о выборе конкретных компонент. Для каждой из подзадач могут быть доступны разные её реализации. Даже выбор одной компоненты из множества альтернатив может быть сложным, если зависит от большого числа предъявляемых к результирующей системе требований. Если проектируемая система

комплексная (необходимо решать большое число вложенных задач), сложность проектирования увеличивается многократно, так как выбор нужно осуществлять, учитывая особенности взаимодействия компонент между собой. В случае множественности и нетривиальности связей между доступными блоками задача их композиции оказывается плохо формализуемой. Решение такой задачи зачастую не под силу рядовому пользователю либо из-за недостатка квалификации, либо времени, отводимого на решение.

Автоматизация процесса формирования результирующих систем из множества доступных компонент может решить описанные проблемы. С одной стороны, пользователь освобождается от ответственности за выбор оптимального набора блоков, что снижает требуемый от него уровень знаний. С другой стороны, конструирование системы проходит быстрее, если все необходимые знания заранее выявлены и представлены в удобной для обработки форме. Как результат, автоматическое связывание сервисов позволяет пользователю сосредоточиться на решении своих задач, не углубляясь в особенности конфигурации вычислительной среды.

Прежде чем приступить к разработке решения по автоматизации, удобно составить общее описание тех областей, для которых подход применим. В данной работе в качестве основы была выбрана сервис-ориентированная архитектура (SOA). В рамках этой архитектуры решение задачи строится в виде последовательности взаимодействий сервисов - слабосвязанных программных модулей со стандартизованными интерфейсами. Такая формулировка совпадает с описанными выше требованиями к исследуемым предметным областям. Таким образом, SOA можно рассматривать в качестве универсальной структурной модели.

2. СЕРВИС-ОРИЕНТИРОВАННАЯ АРХИТЕКТУРА

Работы по автоматизации процессов обнаружения доступных реализаций и их композиции для решения поставленных задач ведутся уже достаточно долгое время. Существующие средства для представления информации об услугах будем рассматривать в рамках наиболее популярной реализации сервис-ориентированной архитектуры – веб-сервисов [1]. Под веб-сервисами понимаются компоненты, предоставляющие доступ к своим услугам через сеть

Интернет по стандартизованным протоколам. В архитектуре системы, основанной на веб-сервисах, выделяется специальный блок для хранения информации о них и предоставления к такой информации доступа – брокер (см. рисунок 1). В этом блоке хранятся описания сервисов, опубликованные их поставщиками. По запросу предоставляется доступ к этим описаниям. На этапе проектирования потребитель сервисов выбирает поставщиков из реестра брокера, исходя из своих потребностей. Далее взаимодействие между поставщиками и потребителем ведётся напрямую, минуя дополнительное звено.

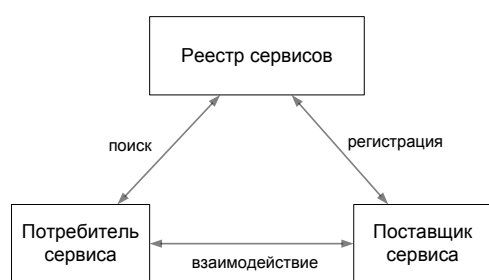


Рисунок 1 – Схема использования веб-сервисов

В области автоматизации композиции сервисов особенно интересен проект ASG (Adaptive Service Grid) – система предоставления услуг по семантическому запросу [2]. Распознавая семантику пользовательских требований, такая система сначала ищет единственный сервис, который может этим требованиям удовлетворить. Если таковой не найден, планируется последовательность вызовов сервисов. Строится схема с пустыми слотами определённых типов, в каждый из которых должна быть помещена некоторая реализация. Далее анализируются описания доступных альтернатив, выбираются лучшие среди имеющихся в распоряжении. Результатом работы становится описание на языке VPEL, которое уже можно загрузить в соответствующий исполнительный механизм для решения задач потребителя.

К сожалению, в результате своих исследований разработчики проекта ASG пришли к неутешительным выводам. Машины логического вывода, способной удовлетворить требованиям их проекта, на рынке не существует. А разработка её слишком трудоёмка. К тому же для работы такой машине логического вывода будет требоваться база знаний с детальным и точным описанием каждого из сервисов. Для реально существующих областей использования сервисов составление баз знаний подобных объёмов требует весьма больших ресурсов [3]. На сегодняшний день работы по проекту ASG свёрнуты, последняя публикация датируется 2008 годом.

Несмотря на возникшие в рамках проекта ASG трудности, общая архитектура системы обработки пользовательского запроса на услуги, представленная в работе, заслуживает внимания (см. рисунок 2). Построение схемы исполнения сервисов, поставляющих для пользователя требуемые ему услуги, разбивается на два основных шага. Сначала модуль планирования, получая запрос на набор услуг и список налагаемых на решение ограничений, строит схему композиции сервисов. В такой схеме отсутствует упоминание конкретных реализаций. В ней содержится только последовательность сочетания типов предоставляемых услуг. Затем блок связывания заполняет схему

композиции наиболее эффективными из доступных альтернатив. Каждому из типов услуги ставится в соответствие одна её реализация. Получается схема исполнения, с помощью которой можно развернуть конечные точки для получения пользователем запрошенных услуг. Если на этапе связывания обнаруживается, что такой набор реализаций, который бы удовлетворял всем пользовательским требованиям, составлен быть не может, то запускается механизм перепланирования – предпринимается попытка построить альтернативную схему композиции.

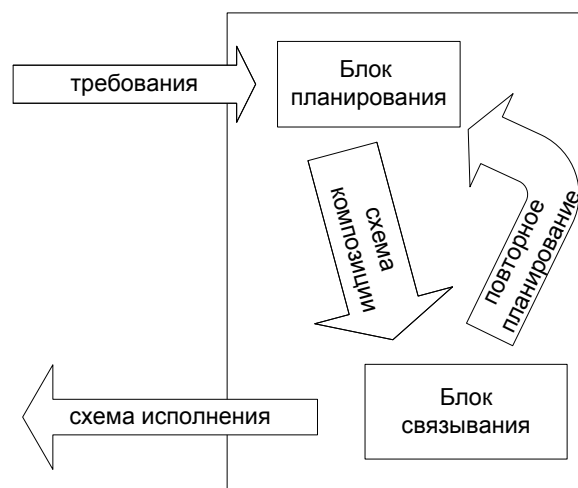


Рисунок 2 – Схема обработки запроса на услуги

Функциональность описанного блока планирования на сегодняшний день уже имеет реализации. Такой блок входил в состав прототипа системы ASG. Примеры подходов к планированию можно найти и в других работах. Решения требует проблема реализации такого блока связывания, который, с одной стороны, имел бы возможность подбирать эффективные для решения поставленных задач сочетания реализаций сервисов, а с другой стороны, использовал бы для своей работы достаточно компактную базу знаний о комплексировании сервисов. Для создания описанного блока требуется формат знаний, не только достаточно выразительный, но и достаточно простой в обработке.

3. КОЭФФИЦИЕНТЫ УВЕРЕННОСТИ

В данной работе предлагается подход, решающий проблему излишней сложности формализованных знаний в области композиции сервисов. В рамках разработанного подхода для осуществления выбора среди реализаций сервисов конкретной предметной области выделяется ряд критериев, оценки по которым могут повлиять на эффективность результирующего решения. К примеру, большая часть вычислительных сервисов может быть оценена по таким критериям, как скорость обработки данных, достоверность полученных результатов, надежность реализации, стоимость обслуживания и т.д. Помимо существующих для предметной области критериев в базу знаний заносится информация о возможных типах предоставляемых услуг, имеющихся поставщиках таких услуг, а также о конкретных реализациях разными поставщиками сервисов соответствующих типов.

Основой для осуществления выбора среди альтернатив являются заносимые в базу знаний экспертные оценки. Под такой оценкой понимается числовое значение,

формализующее степень уверенности эксперта предметной области в успешности функционирования заданного сочетания компонент по заданному критерию и в заданных условиях. Для каждой конкретной альтернативы может быть дано множество оценок по разным критериям и в разных контекстах. Задача выбора сводится к поиску таких реализаций, которые имеют лучшие оценки по интересующим пользователя критериям.

От машины логического вывода скрывается факт того, на каком основании сделана та или иная экспертная оценка. Ей доступны только числовые значения, характеризующие соответствующие ситуации. При таком подходе реализация экспертного движка, способного решать поставленные задачи, становится достижимой при текущем уровне развития области искусственного интеллекта.

4. ЗАКЛЮЧЕНИЕ

В рамках данной работы были проанализированы существующие средства автоматизации проектирования и развёртывания сложных многокомпонентных систем. Схема решения, сформулированная в рамках проекта Adaptive Service Grid (см. рисунок 2), была выбрана в качестве основы для полноценного решения в этой области, несмотря на сложности с реализацией блока связывания сервисов, возникшие у разработчиков этого проекта.

В процессе исследования был предложен подход к представлению и последующей обработке экспертных знаний по композиции сервисов, способный разрешить препятствующие разработке блока связывания проблемы. Для испытания этого подхода был спроектирован и реализован прототип блока связывания, способного, вместе с другими разработками этой области, в перспективе послужить фундаментом для создания решения промышленного уровня.

Для демонстрации универсальности созданной формы представления знаний и её применимости к реальным областям экспертизы, в работе проведено заполнение баз знаний по композиции модулей OpenFOAM и выбору мобильного веб-браузера.

С целью оценки эффективности прототипа и подхода в целом проведён ряд испытаний. Сделан вывод, что прототип выдаёт достоверные рекомендации с точки зрения экспертов соответствующих предметных областей. Времени выработки рекомендаций на текущем этапе может быть недостаточно для системы, в которой изменения схемы связывания происходят в режиме реального времени в процессе исполнения. Тем не менее, производительность прототипа уже позволяет использовать подход для модульных систем на этапе проектирования.

Основным направлением дальнейших исследований является интеграция разработанного блока связывания с существующими смежными блоками. Изначально основной целью работы являлось не столько создание механизма выбора среди доступных альтернатив, сколько заполнение существующего пробела в области формирования готового решения на основе доступных сервисов. Хотя блок связывания может использоваться отдельно для помощи принимающему решение лицу в проектировании модульных систем, основной потенциал

решения раскрывается при взаимодействии с блоком планирования и вспомогательными блоками.

БЛАГОДАРНОСТИ

Работа частично поддержана Российским Фондом Фундаментальных Исследований, проект № 13-07-00747-а и Санкт-Петербургским Государственным Университетом, проект № 9.38.674.2013.

ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА

[1] D. Sprott, L. Wilkes, "Understanding Service-Oriented Architecture", - <http://msdn.microsoft.com/en-us/library/aa480021.aspx>.

[2] D. Kuroпка, M. Weske, "Implementing a semantic service provision platform. Concepts and Experiences" – <http://bpt.hpi.uni-potsdam.de/pub/Public/MathiasWeske/submission.pdf>.

[3] D. Kuroпка, A. Bog, M. Weske, "Semantic Enterprise Services Platform: Motivation, Potential, Functionality and Application Scenarios". *Proceedings of the tenth IEEE international EDOC Enterprise Computing Conference, Hong Kong*, pp. 253 – 261, 2005.