

# Подход к построению сервис-ориентированной электронной библиотеки в среде Semantic Web на основе репланирования

© А.В. Новицкий

Институт программных систем НАН Украины, г. Киев  
ale-nov@yandex.ru

## Аннотация

В работе рассмотрены вопросы построения сервис-ориентированной электронной библиотеки. Описан формализованный подход описания сервисов электронной библиотеки на основе дескриптивной логики. Особое внимание уделено решению проблемы планирования веб-сервисов в среде с нечеткими целями на основе знаний о предметной области.

## 1 Введение

Усовершенствование технологии дали новый толчок к развитию электронных библиотек. Основным требованием к будущим электронным библиотекам, есть постоянное расширение функциональности.

Использование сервис-ориентированной архитектуры является важным элементом достижения интероперабельности. В такой среде сложное программное обеспечение, такое как электронные библиотеки, может быть спроектировано на основе имеющихся в распоряжении сервисов, которые доступны не только локально, но и через Интернет. Причем доступ к сервисам будет происходить динамически.

Для того чтобы обеспечить свободный доступ служб к библиотеке она должна быть реализована на открытой архитектуре. Т.е. взаимодействие сервисов должно быть описано с помощью стандартизированных правил и сред. Что даст возможность свободно вводить новый сервисный элемент, не перестраивая систему заново.

Под *сервисом* мы понимаем некоторую услугу цель, которой удовлетворить запросы пользователя. Сервисы реализуются посредством веб-сервисов. *Веб-сервис* это программное приложение. Основное отличие между сервисом и веб-сервисом состоит в том, что сервис одновременно можно реализовать разными программными веб-сервисами. В свою

очередь веб-сервисы могут между собой объединяться, создавая новый веб-сервис, этот процесс называют композиция веб-сервисов.

Существует много способов композиции веб-сервисов [2]. Для того чтобы произвести композицию сервисов или обратиться к нему в рамках технологии Semantic Web сервис необходимо описать.

Консорциум W3C рекомендует использовать язык Консорциум W3C стандартизует язык OWL-S, который предназначен для описания онтологии сервисов, реализуемых через веб-сервисы. OWL-S включает три подонтологии, которые называются профилем, моделью процесса и основой. Онтология профиля сервиса описывает типы ввода и вывода, определяет предыдущие условия, которые сокращенно записывается как *IOPE* (Inputs, Outputs, Preconditions and Effects). Модель процесса описывает, как сервис работает. Онтология основание сервиса определяет информацию, необходимую для сервисного обслуживания [9].

## 2 Веб-сервисы и планирование их композиции

### 2.1 Обнаружение и композиция веб-сервисов

Взаимодействия сервисов – это динамический процесс, для проведения анализа которого необходима формализация. В основу языка OWL-S положен математический аппарат дескриптивной логики. Формализация сервисов отличается от обычного описания онтологий. Это связано с тем, что при описании сервисов мы имеем дело с динамической средой. Дескриптивная логика (DL) предназначена для того, чтобы описывать логику действий и поведение сервисов для достижения цели в зависимости от входных и выходных условий.

Дескриптивная логика играет значительную роль в Semantic Web, особенно с того времени, когда язык OWL был рекомендован W3C, которым мы можем воспользоваться для семантического описания страниц.

Под запросом  $r$  мы будем понимать, некоторые предоставленные в заранее оговоренном формате параметры, которые мы передаем веб-сервису. Обозначим веб-сервис через  $S_i^{in}$  где  $i$  - уникальный

номер сервиса,  $in(out)$  – набор входных (выходных) параметров сервиса.

Когда с помощью одного веб-сервиса невозможно достичь исполнения запроса  $r$  необходимо создать композицию нескольких веб-сервисов  $\{S_1, S_2, S_3, \dots, S_n\}$  при этом должно выполняться соотношение  $(r_{in} \cup S_1^{out} \cup \dots \cup S_n^{out}) \supseteq r_{out}$ . Это так называемая проблема композиции веб-сервисов (Web Service Composition - WSC).

Аннотации, описывающие семантику сервиса, могут упростить задачу композиции веб-сервисов.

Язык OWL-S развивает OWL для семантического описания веб-сервисов и включает в себя разные аспекты веб-сервисов, в том числе и функциональные. Чтобы описать функциональные возможности, сервисы рассматриваются в качестве процессов, которые имеют начальные условия и действия. Однако для представления динамического поведения таких процессов у статического языка OWL нет достаточных возможностей.

В работе мы будем рассматривать сервисы как действия, которые имеют начальные и конечные условия. Эти условия выражаются с помощью дескриптивной логики. Дескриптивная логика оперирует базой знаний, которая включает два компонента Tbox и Abox. Tbox определяет терминологию, т.е. словарь прикладной области, а Abox содержит утверждения в терминологии данного словаря [3]. Для композиции веб-сервисов окружающее состояние мира частично описывается в Abox.

Единичный законченный веб-сервис который может удовлетворять запрос назовем атомным сервисом.

Мы остановимся на двух основных проблемах композиции сервисов: выполнимость (executability) и проектирование (projection)[8].

Обе задачи уместны для исследования сервисов. Очевидно, что нужно выбирать сервис, который гарантировано, будет выполняться при данной, возможно точно неизвестной ситуации. Кроме того, выполнение сервиса подразумевает достижение какой-то цели, поэтому следует использовать именно те сервисы, которые эту цель могут достичь – следовательно, решать проблему обнаружения сервиса.

## 2.2 Дескриптивная логика

Основными понятиями дескриптивной логики являются атомные понятия (atomic concepts) и атомные роли (atomic roles). Сложные описания могут быть индуктивно построены из них с помощью конструкторов понятий (concept constructors). В дальнейшем мы будем использовать буквы **A**, **B** для обозначения атомных понятий и **R** для обозначения атомных ролей и буквы **C**, **D** для описания произвольных понятий. Более детальную информацию о дескриптивной логике можно найти в книге [3]. Известно, что в полной мере язык OWL-DL описывается дескриптивной логикой, которая имеет диалект SHOIN(D).

## 2.3 Семантическое описание сервисов

Профиль сервиса (ServiceProfile) предназначен для объявления и обнаружения сервиса. Модель процесса (ServiceModel) дает детализированное описание операций сервиса. Основание (ServiceGrounding) описывает процесс взаимодействия с сервисом через сообщения.

Вообще говоря, ServiceProfile необходим для обнаружения сервиса, а ServiceModel и ServiceGrounding вместе дают информацию о том, как использовать найденный сервис.

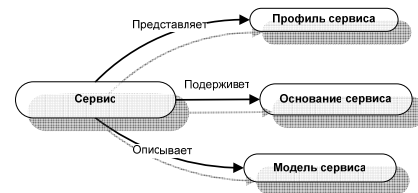


Рис. 1. Структура сервиса

Основание сервиса определяет детали доступа к сервису. Это главным образом протокол и формат сообщения, порядок поступления, транспортировка и адресация [9].

## 2.4 Формальное определение сервисов

Введем формальное описание веб-сервисов. Для упрощения будем рассматривать так называемые основные сервисы (ground services) т.е. сервисы в которых все параметры индивидуальные имена.

**Определение.** Пусть  $T$  – нециклический Tbox. Атомным сервисом для нециклического Tbox  $T$  будем называть выражение  $S = (pre, sname, inputs, outputs, post, pref)$ , где  $pre$  – ограниченное множество утверждений в Abox, которые образуют предыдущие статические (и динамические) условия,  $sname$  – идентификационное имя автономного сервиса,  $inputs$  – область определения класса входных параметров сервиса, которая формируется в виде  $A(a)$  или  $r(a, b)$ , где  $A$  – примитивное понятие в  $T$ ,  $r$  – роли и  $a, b$  атомные понятия.

$outputs$  – область определения класса исходных параметров сервиса, которая формируется в таком же виде, как и  $inputs$ ,

$post$  – ограниченное множество пост-условий в форме  $P/E$ , где  $P$  – утверждение в Abox, а  $E$  – примитивное понятие со ссылкой на Tbox  $T$  т.е.  $A(a)$ ,  $\neg A(a)$ ,  $r(a, b)$  или  $\neg r(a, b)$  [7].

Формальная семантика сервисов может быть определена на основе перехода к интерпретации. Сервис  $S$  может трансформировать интерпретацию  $I$  к  $I'$  то есть  $I \Rightarrow_{S, A}^T I'$  [8].

## 2.5 Отношение между веб-сервисами

При композиции веб-сервисов естественно стоит вопрос об установлении отношения между ними. Введем следующие виды отношений между сервисами.

Пусть  $T$  – нециклический Tbox,  $A$  – Abox,  $S_i$  и  $S_j$  – сервисы, которые являются подсервисами сложного сервиса  $S$ . Причем  $S_i$  предоставляет отличный от  $S_j$  тип сервиса. Отношение  $R$  между подсервисами  $S_i$  и  $S_j$  могут быть определены следующим образом:

Отношение независимости:  $S_i + S_j = S_j + S_i$  – это означает, что каждый из сервисов независимый один от другого и порядок выполнения этих сервисов не будет влиять на конечный сервис. Отношение идентичности:  $S_i = S_j$  – это значит, что сервисы предоставляют один и тот же сервис тем не менее имеют разные атрибуты. Отношение условной идентичности:  $S_i \cong S_j$  – это значит, что сервис  $S_i$  предоставляет те же самые функции что и  $S_j$  в некоторых ситуациях. Отношение замены:  $S_i \rho S_j$  – это значит, что сервис  $S_i$  можно заменить на сервис  $S_j$  в любом случае. Отношение условной замены:  $S_i \% S_j$  – это значит, что сервис  $S_i$  можно заменить на сервис  $S_j$  в некоторых ситуациях. Отношение перекрытия:  $S_i \otimes S_j$  – это значит, что существует общая часть в этих сервисных обслуживаниях. При композиции отношений перекрытия общая часть должна быть исключена. Отношение последовательности:  $S_i \rightarrow S_j$  – это значит, что сервис  $S_i$  должен полностью закончить свою работу перед стартом сервиса  $S_j$ .

## 2.6 Базовые сервисы

Для построения электронной библиотеки на основе сервис-ориентированного подхода необходимо выделить сервисы, на основе которых будет выполняться композиция. Этот набор сервисов обусловлен анализом работы конечного пользователя (на примере научного сотрудника) а также на основе анализа требования к современным электронным библиотекам и принято во внимание ПО Eprints. При выделении базовых сервисов мы руководствовались тем, что каждый базовый сервис отвечает определенному уровню абстракции структуры библиотеки. Сервисы возможна разделить на две группы, внутренние сервисы которые не предназначены для конечных пользователей (имеют программные интерфейсы) и сервисы которые имеют веб-интерфейсы конечного использования. Приведем список последних:

*Сервис преобразования форматов.* Предназначенный для преобразования документов одного

формата в другие. Этот сервис является необходимым, поскольку он обеспечит работу библиотеки с разными форматами данных.

*Сервис построения графа цитирования.* Выполняет функции построения графа цитирования, где узлами графа будут объекты или документы, а связи между узлами являются собственно ссылками. При построении семантического графа узлами могут выступать, как ресурс или часть ресурса, а также элементы базы знаний, а связи между узлами выступают в качестве отношений между объектами.

*Сервис конвертации метаданных.* Отвечает за конвертацию одних схем метаданных в другие.

*Сервис управления ресурсами.* Обеспечит управления авторским правом, политиками и дубликатами объектов электронной библиотеки.

*Сервис базового поиска.* Обеспечивает поиск по базовым наборам метаданных.

*Сервис расширенного поиска.* Обеспечивает поиск по более широкому набору полей с возможностью логической комбинации.

*Сервис навигации и классификации по определенному типу метаданных*

*Сервис управление доступом.* Обеспечивает авторизацию пользователей, а также восстановления и установления паролей доступа.

*Сервис сохранения результатов поиска.* Предназначен для пользователей, которые прошли авторизацию, причем возможно выборочное сохранения результатов.

В последнее время в связи с развитием среды Semantic Web мы вводим *сервис аннотирования информации*. Сервис аннотирования представит еще один способ активного взаимодействия пользователя с электронной библиотекой. Объемы информационных ресурсов в электронной библиотеке очень значительны и использование аннотаций значительно упростит поиск информации.

Формальная модель, которая предложена в [12], выделяет два подхода к пониманию аннотаций: аннотации как метаданные и аннотации или как контент.

В первом случае мы имеем дело с разнообразными схемами метаданных (Dublin Core, MARC и др.) которые используются для описания информационных ресурсов. Эти аннотации в первую очередь направлены на пользователя.

Во втором случае аннотации как предоставления контента предназначены для автоматизированной машинной обработки. Эти аннотации предоставляют семантику документа. Семантическая аннотация – аннотация написана формальным языком с хорошо определенной семантикой и основана на онтологиях.

## 2.7 Обобщенная формальная модель аннотации

Цифровые объекты  $DO$  могут быть разными по содержанию: текст, аудио, картинки, также разнообразные форматы представление этих данных. Модель, которая будет представлять цифровой объект должна отображать фактическое содержание данно-

го объекта. Среди множества разработанных моделей, мы используем модель, которая предложена в [12, 13] с некоторыми изменениями которые касаются логики выделение потоков, исключили управления идентификации создателей аннотации и функции *hsm* что на наш взгляд делает модель эффективней.

В каждый момент времени  $k$  мы имеем некоторое множество цифровых объектов  $DO(k)$  при аннотировании документа, мы получаем некоторое новое множество документов в момент времени  $DO(k+1)$ . Далее если переменная будет явно зависеть от  $k$  то она будет указываться, иначе она опускается.

Аннотация  $a \in A(k)$  это кортеж:

$$a = \left( \begin{array}{l} h_a \in H(k), \\ A_a \subseteq SN(k) \times LT \times ST(k) \times \\ \times SM(k-1) \times H(k-1) \end{array} \right)$$

где  $h_a$  - уникальный собственный идентификатор аннотации  $a$  то есть  $h(h_a) = a$ ;

$A_a$  множество  $n$ -арных отношений аннотации  $a$  определяется как произведение множеств  $SN$ ,  $LT$ ,  $ST$ ,  $SM$  и  $H$ ;  $SN$  - определяет множество типов аннотаций;  $ST$  - множество сегментов;  $SM$  - определяет множество потоков;  $H$  - множество уникальных идентификаторов

## 2.8 Описание сервиса аннотирования с помощью дескриптивной логики

Пусть  $S_0$  сервис аннотирование веб-ресурсов. В рамках дескриптивной логики он примет следующий вид:

$S_0$  :

$pre_1 \leq 1 \text{ hasCount.link}$

$sname_1 = \text{AnnotationProcess}$

$input_1 = \{sn, sm_i, st_{sm}, LT\}$

$outputs_1 = \{LT(a, do)\}$

Ациклический Tbox  $T_j$  для этих сервисов определяется следующим образом:

$link \equiv \text{URI} \sqcup \text{DOI} \sqcup \text{OpenURL} \sqcup \text{PURL}$

$sn \equiv \text{text} \sqcup \text{image} \sqcup \text{audio} \sqcup \text{video}$

$sm \equiv \exists \text{number.T}$

Используя дескриптивную логику возможно описать и другие сервисы.

## 3 Композиция веб-сервисов

Для крупномасштабных задач WSC, какими является электронная библиотека, алгоритмы полного перебора не являются эффективными, в данном случае более пригодными являются алгоритмы автоматического поиска подоптимальных решений

выборки сервисов на основе знаний о предметной области.

Это вызвано тем, что в процессе функционирования сервис-ориентированой электронной библиотеки среда в котором функционируют веб-сервисы постоянно изменяется. Изменение среды порождается двумя основными факторами:

во-первых, проблемы, которые порождают распределенные системы: временные задержки и ненадежность транспортного протокола; нехватка памяти общего использования между частями распределенной системы; проблемы отказа доступа и параллельных запросов, а также проблемы связаны с программной несовместимостью вследствие обновления части распределенной системы;

во-вторых, человеческий фактор, в процессе обслуживания пользователя, могут измениться требования пользователя, а, следовательно, это определенным образом будет влиять на результат. Рядовые пользователи целевой аудитории не имеют четкого представления об архитектуре библиотеки, следовательно, не могут заранее четко определиться с целями которые должны удовлетворять сервисы.

Мы предлагаем рассматривать композицию на основе целе-ориентированой парадигмы, то есть исходя из первоначальных условий и имеющегося множества сервисов, осуществить композицию. Причем, поскольку веб-сервисы размещены в семантической среде, то выбирать такие планы композиции, которые могут быть полезными для конечного пользователя. В отличие от классической постановки задачи, от спецификации цели к поиску сервисов, которые смогут эту цель достичь. Мы предлагаем выходить из того предположения, что существует множество сервисов, которые могут достигать некоторых, заранее неизвестных целей, которые может поставить перед собой пользователь.

В такой сложной системе цели, которые появляются перед композицией могут изменяться. Мы также выходим с того что знания об окружающем мир является не полными, а следовательно, и цели являются неполными. Для таких конфликтных целей Horst Rittel и Melvin Webber в [14], [15] ввели понятие викид (wicked) задачи. Поэтому классические методы планирования могут оказаться неэффективными. В среде, где цели не являются четко определенными и являются динамическими, процесс планирования принимает другое смысловое значение, а именно, план выполнения действий не является однозначным алгоритмом достижения цели. Функцию планов можно сформулировать следующим образом [3]: проверка ресурсов; начинают координационные процессы или помогают упростить координационные процессы в начале; устанавливают ответственность и идентификацию; трековый прогресс и, что более важно, генерация намерений; узнают и управляют рисками; поддерживают импровизацию, и формализуют представление человека (пользователя) о проблеме, которую план пытаются решить.

Такая комбинация функциональных свойств приводит к тому, что планы необходимо дополнительно исследовать и проводить перепланирование. Т.е. изменять планы во время выполнения этих планов.

Рассмотрим самый простой случай в электронной библиотеке. Пользователю необходимо найти некоторую информацию, система генерирует возможные варианты поиска информации и пользователь решил, что лучше для этого воспользоваться базовым (простым) поиском, однако система не может однозначно определить цель (цель неточна) а следовательно и возможен план развития, поскольку в процессе выполнения плана пользователь может изменить цель (например, некоторая дополнительная обработка результатов поиска), а следовательно, нужно репланировать, то есть изменять первоначальный план. При этом пользователь появляется перед викид-проблемой, поскольку появляется дилемма выбора между важностью возможно найденной информации и времени затраченному на ее поиск.

Аннотирование информации значительно упрощает ее машинную обработку с целью систематизации и поиска.

Решение описанных проблем возможно только в среде Semantic Web. Что требует четкой формализации всех сервисов и построение модели процессов. В таком случае, возможно, получить гибкую, легко расширяемую систему с возможностью автоматически решать проблему WSC. Следующим этапом будет практическая реализация сервисов на основе их моделей.

## Литература

- [1] Yves Petinot, C. Lee Giles, Vivek Bhatnagar, Pradeep B. Teregowda, Hui Han, Isaac Council. A Service-Oriented Architecture for Digital Libraries
- [2] J. Rao, X. Su. A Survey of Automated Web Service Composition Methods. In Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition, SWSWPC 2004, San Diego, California, USA, July 6th, 2004.
- [3] The description logic handbook: theory, implementation, and applications Edited by Franz Baader Deborah L. McGuinness Daniele Nardi Peter F. Patel-Schneider
- [4] Lirong Qiu, Fen Lin, ChanglinWan, ZhongzhiShi. Semantic WebServices Composition Using AIplanning of Description Logics
- [5] Description Logics: Foundations for Class-based Knowledge Representation Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini Dipartimento di Informatica e Sistemistica Universit'a di Roma "La Sapienza" Via Salaria 113, I-00198 Roma, Italy
- [6] Description Logics: Foundations for Class-based Knowledge Representation Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini Dipartimento di Informatica e Sistemistica Universit'a di Roma "La Sapienza" Via Salaria 113, I-00198 Roma, Italy

- [7] Yingjie Li1, Xueli Yu1, Lili Geng1, Li Wang. Research on Reasoning of the Dynamic Semantic Web Services Composition. Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference on Volume , Issue , 18-22 Dec. 2006 Page(s):435 – 441
- [8] Franz Baader, Maja Milicic, Carsten Lutz, Ulrike Sattler, Frank Wolter. Integrating Description Logics and Action Formalisms for Reasoning about Web Services. LTCS-Report 05-02, Chair for Automata Theory, Institute for Theoretical Computer Science, Dresden University of Technology, Germany, 2005
- [9] OWL-S: Semantic Markup for Web Services <http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/>
- [10] Ragone, A., Di Noia, T., Di Sciascio, E., Donini, F.M., Colucci, S. Fully automated Web services orchestration in a resource retrieval scenario. Web Services, 2005. ICWS 2005. Proceedings. 2005 IEEE International Conference on Publication Date: 11-15 July 2005
- [11] WebServices Description Language (WSDL)1.1 <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>
- [12] GONCALVES, M.A., FOX, E.A., WATSON, L.T., AND KIPP, N.A. 2004a. Streams, structures, spaces, scenarios, societies (5S): A formal model for digital libraries. ACM Trans. Inform. Syst. 22, 2 (Apr.), 270–312.
- [13] M. Agosti, N. Ferro. A Formal Model of Annotations of Digital Content. ACM Transactions on Information Systems (TOIS), 2008, 26(1):3:1–3:57
- [14] Gary Klein, "Flexexecution as a Paradigm for Replanning, Part 1," IEEE Intelligent Systems, vol. 22, no. 5, pp. 79–83, Sept/Oct, 2007
- [15] Gary Klein "Flexexecution, Part 2: Understanding and Supporting Flexible Execution," IEEE Intelligent Systems, vol. 22, no. 6 Nov./Dec., pp. 108–112.

### **The approach to construction service-oriented to digital library in the environment of Semantic Web on a basis replanning**

A.V. Novytskyi

In work it is considered constructions service-oriented of electronic library. The formalized approach of the description of services of electronic library on the basis of descriptive logic is described. The special attention is given to the decision of a problem of planning of web services in the environment with the indistinct goals on the basis of knowledge of a subject domain.