УДК 681.3

В. Дерецкий

ПОДХОД К ОПРЕДЕЛЕНИЮ ПОЛНОТЫ КОМПОЗИЦИИ СЕМАНТИЧЕСКИХ ВЕБ-СЕРВИСОВ

Основная задача сервис-ориентированной парадигмы – это облегчение автоматизированной композиции сервисов мотивирована сценариями, развиваемыми в электронной коммерции и e-science. Проблема автоматизированной композиции начинается со спецификации цели "goal" сервиса и набора доступных для поиска сервисов. Композиция зависит от обнаружения соответствующих сервисов на семантическом уровне, в основном, через входные и выходные условия активностей, и сборки их в "приложение" целевого сервиса. Мы исследуем проблему полноты композиции, путем усиления запросов поиска сервисов, начиная от условий целевого сервиса. Более сильное условие может найти более релевантный сервис, без которого сборка целевого сервиса может быть невозможной. Проблема усиления условий изучается в различных логических языках, в данном случае, используются условия с порядковыми ограничениями.

Введение

Веб-сервисы описываются функциональными и нефункциональными свойствами и могут быть размещены и доступны через стандартизированные интерфейсы [1]. Композиция веб-сервисов обеспечивает механизм связывания многоразовых сервисов с целью формирования нового сервиса, который порождает новые значения. Автоматизированная композиция сервисов важна для приложений многих областей, в частности, таких как электронная коммерция, e-science и других.

Фундаментальной проблемой композиции сервисов является поиск сервисов. В результате поиска могут быть найдены сервисы, релевантные условиям поиска, реализующие целевой сервис. Целевой сервис представлен действиями или активностями, которые связаны в цепочку потока данных. Каждая активность в целевом сервисе содержит семантические описания. Проблема спецификации целевого сервиса состоит в назначении каждой активности в целевом сервисе существующих сервисов из реестра.

Композиция, основанная на модели целевого сервиса, представляет собой естественный подход, развиваемый от интерактивной композиции сервисов на основе бизнес модели приложения [2–4] и моделей потока данных научных приложений [5, 6]. Предлагаемый подход развивает задачи композиции как в части усиления поиска сервисов на основе семантической спецификации входных и выходных условий, так и исследований, связанных с принципиальной возможностью реализации целевого сервиса с учетом условий и окружения, в отличии от подходов к композиции, при которых сервисы заранее известны и отсутствует семантика в их спецификации.

Модель целевого сервиса представляется в виде ациклического графа с потоком данных, соответствующем входным и выходным условиям. Семантика активностей выражается в их условиях. Подход напоминает работы [7–10], в которых OWL-S расширялся семантическими описаниями. Для каждой активности формируется запрос к сервису в соответствии с входными и выходными условиями активности. Осуществляется семантическое согласование запросов с условиями существующих сервисов в реестре.

Поиск сервисов на основе семантики исследовался в различных работах

[10–13]. В данной работе предлагается подход, при котором поиск сервисов осуществляется с использованием семантики в модели целевого сервиса, при котором результаты запроса поиска одного сервиса используются в запросах поиска следующих сервисов в модели. Формирование поискового запроса на основе семантических условий целевого сервиса влияет на условия в соответствии с которыми сервис может быть найден. Учитывая входные условия в целевом сервисе можно значительно уменьшить число сервисов-кандидатов, реализующих заданную активность целевого сервиса, тем самым повысить производительность системы.

В данной работе исследуется подход, основанный на усилении условий целевого сервиса. Условия формально выводятся из условий предшествующих активностей для усиления условий текущей активности. Условия предшествующей активности называются пост-условиями активности. Заданные входные условия и существующие условия активности определяют задачу усиления или уплотнения пост-условий активности.

Алгоритм уплотнения, рассмотренный в данной работе, может использовать различные логические языки. В частности нами исследовалась возможность использования языка COLOG и исчисление ситуаций [14].

1. Композиция сервисов на основе семантических спецификаций

Автоматизированная композиция сервисов – основная задача парадигмы сервисов, которая обеспечивает и облегчает такие возможности сервисов как публикацию, поиск, повторное использование. Проблема композиции в целом имеет три составляющих: спецификация цели, сбор доступных сервисов, и «склеивание» найденных сервисов в один сервис. Кратко опишем модель композиции сервисов и метод усиления условий активностей, на основе которого реализуется композиция сервисов.

Проблема композиции сервисов исследуется во многих работах, в которых исследуются теоретические модели. В частности Латинская модель [15, 16], модели, основанные на переговорах [17, 18], модели потока данных [19–21], модели, основанные на внутренних состояниях сервисов [22, 23], модели планирования искусственного интеллекта [24, 25], и недавно, модели Коломбо [26]. Мы исследуем модель композиции сервисов, основанную на спецификациях целевого сервиса. Подход использует спецификации цели, используемый в модели WSMO [27, 28]. Целевой сервис состоит из активностей, которые связаны потоком управления и данных. Каждая активность представляет сервис, который должен быть найден, чтобы реализовать эту активность. Активность описывается своими входными и выходными дугами и соответствующими входными и выходными условиями.

Дуги представляют контейнеры данных, вытекающие от одной активности к следующей. Если удовлетворяется начальное условие (истина), то активность может быть выполненной, и порождает образец выходной дуги, которая является входом для следующей активности. После выполнения активности, фиксируется условие выхода для пары входной и выходной дуг. Целевой сервис такого вида соответствует сервисным приложениям e-business, например, amazon.com, opentable.com, научным технологическим процессам Workflow [20, 21] и другим приложениям. Ключевая характеристика таких приложений ориентированная на потоки данных логика вычислений, которая управляет композицией сервисов. Модель композиции – пример парадигмы, положенной в основу WSMO [28, 29], в которой целевой сервис семантически описан пред- и пост- условиями, предположениями и результатами. Входные и выходные условия в нашей модели сосредоточиваются на ограничениях входов и выходов активностей.

Композиция сервисов, ориентированная на целевой сервис также расширяет естественный подход, используемый в диалоговой композиции [15, 26]. Использование целевого сервиса в форме модели

потока данных для композиции исследуется в [20, 23]. В работе [23], композиция сервисов использует нефункциональные свойства, например, условия входа и выхода. Семантическое расширение спецификации Веб-сервисов исследовалась в моделях OWL-S [30], WSDL-S [31], SWSO [27] и WSMO [28]. Атомарные процессы в OWL-S и SWSO используют входные условия и результаты.

1.1. Пред- и пост-условия активностей. Модель целевого сервиса представим в виде направленного графа (Q, M), где Q – представляет множество активностей, M – множество дуг, которые отображают потоки данных, производимые одной активностью и используемые другими как входные. В общем случае, для одной активности может быть более одной входной дуги и более одной выходной дуги. Для того чтобы рассмотреть алгоритм уплотнения условий, мы рассмотрим более простую модель, в которой целевой сервис представлен в виде "линейной" схемы, в которых активность имеет один вход и один выход. Вход каждой активности связан с начальными условиями, а выход, – с выходными условиями. Отметим, что дуги активности представляют кортеж переменных. Входные и выходные условия представляют собой логические формулы на некотором логическом языке над переменными активностей. Мы рассматриваем логический язык L, который содержит арифметические неравенства для выражения порядковых ограничений, как в области целых, так и реальных чисел.

Определение 1. Входным условием дуги является конъюнктивная формула в L над переменными дуги. r- ым покрытием выходных условий над входной дугой $[x_1, ..., x_n]$ и выходной дугой $[y_1, ..., y_m]$ является r-кортеж формул в L следующего вида:

$$\varphi_i(x_1,...,x_n) \to \psi_1(z_1,...z_l)$$
$$\dots$$
$$\varphi_r(x_1,...,x_n) \to \psi_r(z_1,...z_l),$$

где $k \ge 0, r \ge 1, \varphi_i, \psi_i$ – конъюнктивные формулы в L такие, что φ_i – парные дизъюнкции и $\vee_i \varphi_i$ – повторения, такие, что $\{z_1, ..., z_l\} \subseteq \{x_1, ..., x_n\} \cup \{y_1, ..., y_m\}$, и мощность $\{z_1, ..., z_l\} - \{y_1, ..., y_m\}$ является k.

Выходные условия являются *ортогональными*, если существует 0-покрытие и входные/выходные дуги не разделяют общих переменных.

На рис. 1 в соответствии с определением, дуга $e_1 = [x_1,\dots x_n]$ с входными условиями $\varphi_1(x_1,...,x_n)$. Дуга $e_2 = [y_1,\dots,y_m]$ с входными условиями φ_2.

Выходные условия $P(e_1, e_2)$:

if $\varphi_i(x_1,...,x_n)$ then $\psi_1(z_1,...z_l)$;
$\vdots$
if $\varphi_r(x_1,...,x_n)$ then $\psi_r(z_1,...z_l)$;

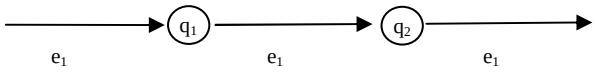

Рис. 1. Дуги, входные и выходные условия активностей линейного целевого сервиса

Очевидно, что входные условия должны удовлетворять результатам выполнения активности, которая выполнилась ранее. На рис. 1 показаны входные и выходные условия для активности q_1. Отметим, что выходная дуга q_1 является также входной дугой для последующей активности q_2. Выходное условие $P(e_1, e_2)$ имеет структуру в виде операторов выбора (если, то): Если $\varphi_i (1 \le i \le r)$ истинна, тогда ψ_i получается после выполнения активности q_1. Каждая пара φ_i и φ_j, $i \ne j$ не пересекается, таким образом, $\neg\exists x_1 ... \exists x_n \varphi_i(x_1,...x_n) \wedge \varphi_j(x_1,...x_n)$ всегда истинна. После реализации активности, хотя бы один случай в выходном условии должен быть истинным.

Рассмотрим пример, в котором активность имеет входную дугу $[x_1, x_2]$ и выходную дугу $[y_1]$. Входное условие – $(0 \le x_2 \wedge x_2 \le 9)$, выходное условие – $(3 \le x_1 - x_2 \to 2x_1 < y_1, x_1 - x_2 < 3 \to y_1 < x_1)$. Отметим, что условия $3 \le x_1 - x_2$ и

$x_1 - x_2 < 3$ взаимоисключающие, и их дизъюнкция всегда истинна.

Определение. Пусть [x_1, ..., x_n], [y_1, ..., y_m] – входные и выходные дуги для активности q, $\varphi(x_1,...,x_n)$ и $\psi(x_1,...,x_n,y_1,...,y_m)$ – входные и выходные условия соответственно. Пост-условием для q является формула $\xi(y_1,...,y_m)$ в дизъюнктивной нормальной форме в L над выходными дугами, такими, что имеет место следующее:

$$\forall x_1 ... \forall x_n \forall y_1 ... \forall y_m (\varphi \wedge \psi \rightarrow \xi).$$

Пост-условие $\xi 1$ является уплотнением другого пост-условия $\xi 2$, если

$$\forall x_1 ... \forall x_n \forall y_1 ... \forall y_m (\varphi \wedge \psi \rightarrow \xi_1 \rightarrow \xi_2).$$

Самым плотным пост-условием активности q является самое уплотненное пост-условие из пост-условий активности q.

Например, пусть [x_1] и [y_1] – входные и выходные дуги. Входным условием является $3 < x_1$ и выходным – $x_1 < y_1$. Пост-условие может быть представлено как $0 < y_1$, но самое сильное уплотнение условия – $3 < y_1$.

Пусть задан целевой сервис, процедура уплотнения генерирует уплотнение пост-условий для усиления входных условий последующей активности. Процедура выполняется последовательно от первой активности к следующей и на каждом шаге вычисляются самые уплотненные пост-условия.

Отметим, что пост-условия представляются в дизъюнктивной нормальной форме вместо того, чтобы быть представленным в конъюнктивной форме. Это следует из факта, что не всегда возможно осуществить уплотнение пост-условий.

Рассмотрим следующий пример. Пусть [x_1] и [y_1] – переменные входной и выходной дуг активности. Входное условие является истинным и выходные условия – $0 \le x_1 \rightarrow x_1 < y_1, x_1 < 0 \rightarrow y_1 < x_1$. Самое плотное пост-условие – $(0 < y_1 \vee y_1 < 0)$, которое представлено в дизъюнктивной нормальной форме.

На рис. 1, входным условием является φ_1 в конъюнктивной форме, и выходное условие – $P(e_1, e_2)$. Самое плотное условие для q_1 – формула в DNF, а именно $\exists t_1 ... \exists t_s (\varphi_1 \wedge P(e_1, e_2))$, где t_s представляет все переменные в e_1, но не в e_2. Уплотнением начальных условий для q_2 является конъюнкцией из самого плотного пост-условия с φ_2.

2. Подход к поиску семантических Web-сервисов с использованием пред- и пост-условий

Рассмотрим свойство "полноты" композиции целевого сервиса. Под полнотой, мы подразумеваем нахождение выполнимой реализации целевого сервиса, если она существует. Понятие полноты "схемы" композиции измеряет возможность поиска выполнимых реализаций для алгоритма композиции схемы. В данном случае учитываются только семантические характеристики целевой схемы и сервисов в реестре.

Алгоритмы композиции направлены на реализацию целевого сервиса на основе реализации активностей целевого сервиса. Информация о сервисах, связанных с активностями, существует в описаниях сервисов. Мы показываем, что стратегия уплотнения "вширину" полностью использует описание сервисов, в том смысле, что алгоритм композиции схемы с первой стратегией уплотнения является полной схемой.

Определение. Композиция сервисов является полной, если для целевого сервиса и сервисов из реестра существует выполнимая реализация. На рис. 2 представлен пример, который показывает целевой сервис с тремя активностями. S_{11} и S_{12} – два сервиса в реестре для активностей q_1, S_{21}, S_{22}, и S_{23} – три сервиса для активности q_2, S_{31} и S_{32} – два сервиса для активности q_3.

Таким образом, целевой сервис имеет выполнимые реализации. Например, схема $L(q_1 \rightarrow S_{11}, q_2 \rightarrow S_{21}, q_3 \rightarrow S_{31})$, является выполнимой реализацией с предусловиями, S_{31} гарантируются постусловиями S_{21} и S_{11}. Другие выполнимые

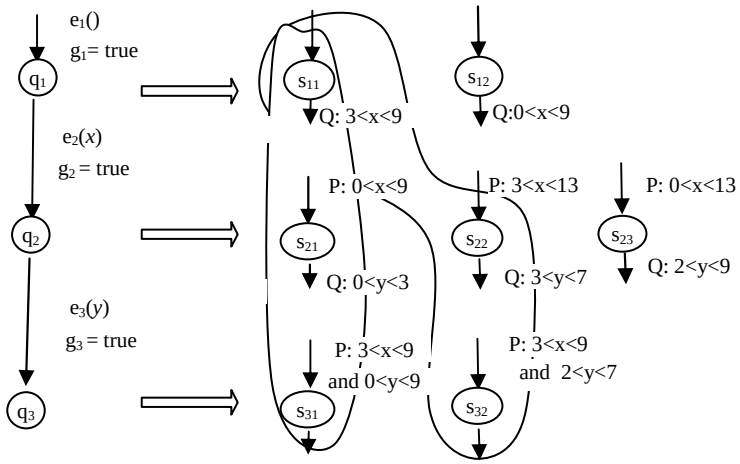

Рис. 2. Иллюстрация полноты композиции

реализации включают $L_1(q_1 \rightarrow S_{11}, q_2 \rightarrow S_{22}, q_3 \rightarrow S_{31})$, $L_2(q_1 \rightarrow S_{11}, q_2 \rightarrow S_{22}, q_3 \rightarrow S_{32})$ и т.п. Алгоритм полной схемы композиции сформирует одну из выполнимых реализаций.

Для того чтобы найти выполнимую реализацию, алгоритму композиции нужно усилить условия с целью нахождения сервисов-кандидатов для каждой активности. Эти условия включают начальные условия предыдущих активностей, и пост-условия отобранных сервисов. Все агрегированные условия получаются из описаний сервисов.

Рассмотрим два алгоритма (две стратегии) на основе поиска «вглубину» и поиска «вширину», которые обеспечивают полноту схемы. Также рассмотрим эффективность алгоритма в терминах избыточных вычислений.

Алгоритм композиции использует первую стратегию уплотнения условий, которая должна обеспечить полноту схемы. На рис. 3 показано поисковое пространство алгоритма композиции сервисов с первой стратегией уплотнения, рассмотренной на примере (рис.2). Каждый путь в области поиска является выполнимой реализацией целевого сервиса.

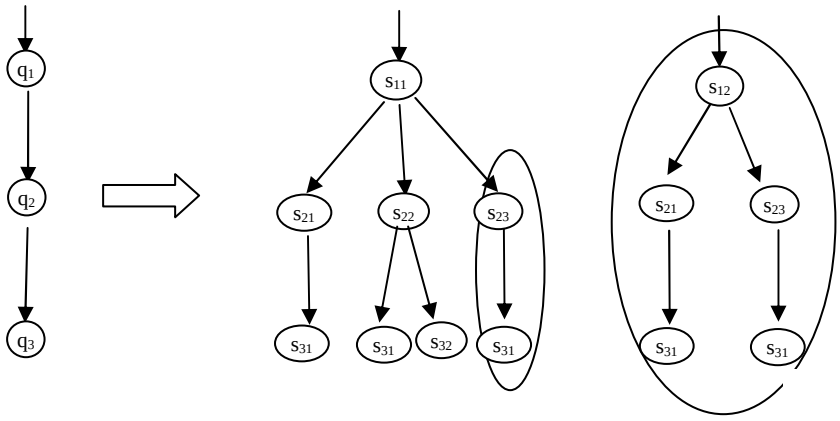

Рис. 3. Пространство поиска для стратегии уплотнения пост-условий

Если для активности q_1 выбирается сервис S_{11}, то пост- условие S_{11} $(3 < x < 9)$ используется для уплотнения входного условия активности q_2, в результате чего могут быть найдены сервисы S_{21}, S_{22} и S_{23}. Если для активности q_2 выбран сервис S_{21}, уплотняется входное условие q_2 $(3 < x < 9)$ вместе с пост-условиями S_{21} $(0 < y < 3)$, которые толкают активность q_3. Уплотнение входных условий q_3 $(3 < x < 9 \wedge 0 < < y < 3)$ определяет S_{31} сервисом-кандидатом. При таких условиях S_{32} не является сервисом-кандидатом, потому что для сервисов S_{11} и S_{21}, агрегированные условия не могут гарантировать предусловия S_{32} которые должны быть истинными.

В алгоритме композиции сервиса со стратегией уплотнения сервис выбирается для каждой активности. Для конкретного алгоритма композиции поиск выполнимой реализации состоит в выполнении поиска «вглубину» или поиска «вширину». Например, на рис. 3 поиск «вглубину» начинается от первой активности q_1 и первого сервиса-кандидата S_{11}, затем первому кандидату сервиса S_{21} активности q_2 после того, как для q_1 назначен S_{11}, поиск переходит к следующим сервисам, вчастности S_{31} для активности q_3, и выполнимая реализация найдена. В случае если сервисы для активности не могут быть найдены, осуществляется возврат к предыдущей активности и осуществляется попытка найти следующий сервис-кандидат на предыдущем уровне (S_{22} или S_{23}).

Утверждение 1. Алгоритм поиска «вглубину» порождает полную схему.

Грубое доказательство этого утверждения состоит в том, что, поиск «вглубину» проверяет все пути в пространстве поиска. На примере исследуется область поиска слева направо, пока не будет найдена выполнимая реализация. Поиск «вглубину» осуществляет полный перебор, для того чтобы гарантировать полноту схемы. В наихудшем случае, исследуется вся область поиска.

Скорость поискового процесса для алгоритма поиска «вглубину» может быть увеличена путем сокращения области поиска. Например (рис. 3), все подветви S_{12} также являются подветвями S_{11}. Ветвь, начинающаяся с вершины S_{12} на рис. 3 может быть удалена из области поиска. Сокращение области поиска не изменяет свойство полноты схемы поиска «вглубину» потому что, если существует выполнимая реализация в ветви (S_{12}), то должна быть выполнимая реализация в одном из путей, начинающихся от S_{11}. Если мы рассмотрим уплотнение начальных условий q_2 после того как выбран S_{11} или S_{12}, то очевидно, что уплотнение условия q_2 после S_{11} и S_{12} является $C_1 = 3 < x < 9$, и $C_2 = 0 < x < 9$. C_1 сильнее, чем C_2 (т. е. $C_1 > > C_2$), таким образом, все сервисы, найденные для C_2, нашлись бы в C_1. Таким образом, ветвь (S_{12}) становится избыточной и может быть удалена. Ветвь (S_{11}) на рис. 3 также может быть удалена по той же причине, условие после S_{11} и S_{22} сильнее, чем после S_{11} и S_{23}. Если путь (S_{11}, S_{23}, S_{31}) является выполнимой реализацией, то путь (S_{11}, S_{22}, S_{31}) – также выполнимая реализация.

Алгоритм композиции с сокращением области поиска может быть описан следующим образом. Основная идея подхода горизонтального поиска «вширину» состоит в том, что для всех сервисов-кандидатов активности, если существует отношение между условиями по двум путям, заканчивающихся на одной активности, путь с более слабым условием может быть удален. Поисковый процесс начинается от первой активности целевого сервиса. Для каждой активности q_i за исключением последней, вначале находятся все сервисы-кандидаты по всем возможным комбинациям сервисов. Затем проверяются любые два пути, которые заканчиваются на активности q_i. Если есть отношение между двумя агрегированными условиями в обоих путях, тогда путь с более слабым условием удаляется. Этот алгоритм называется поиском «вширину» с сокращением.

Рассмотрим целевой сервис и область поиска, показанную на рис. 3. Поиск «вширину» с сокращением начинается от активности q_1, и находит сервисы S_{11}, S_{12}, как сервисы-кандидаты.

Поскольку агрегированное условие пути после (S_{11}) ($3 < x < 9$) является сильнее, чем условие пути после (S_{12}) ($0 < x < 9$), то путь (S_{12}) удаляется, т. е. ветвь (S_{12}) обрезается в пространстве поиска. Поиск «вширину» с сокращением переходит к следующей активности q_2. Для активности q2 по пути (S_{11}), найдены сервисы S_{21}, S_{22} и S_{23}. Агрегированное условие пути (S_{11}, S_{22}) составляет $3 < x < 9 \wedge 3 < y < 7$, а агрегированное условие пути (S_{11}, S_{23}) составляет $5 < x < 10 \wedge 3 < y < 10$. Первое условие является более сильным, следовательно путь (S_{11}, S_{23}) удаляется. Ветвь удаляется из поискового пространства.

Утверждение 2. Алгоритм поиска «вширину» с сокращением обеспечивает полную схему композиции.

Основная идея доказательства состоит в том, что для каждой выполнимой реализации, найденной посредством поиска «вглубину» или вертикального поиска можно найти выполнимую реализацию поиска «вширину» с сокращением. Поскольку поиск «вглубину» – полная схема, то поиск «вширину» с сокращением также является полной схемой.

Алгоритм поиска «вширину» с сокращением ограничивает область поиска. Фактически устраняются избыточные пути в области поиска. Пути являются избыточными, если алгоритм может все еще находить выполнимую реализацию, если она существует после того как была уже найдена. Для того, что бы гарантировать полноту схемы линейного целевого сервиса и произвольного реестра сервисов, не существует алгоритма, который может обеспечить меньшее пространство поиска, чем алгоритм поиска «вширину» с сокращением.

Утверждение 3. Пусть задан линейный целевой сервис и регистр сервисов, алгоритм поиска «вширину» с сокращением пространства поиска не порождает избыточных путей в области поиска.

Основная идея доказательства состоит в том, что мы удаляем путь из области поиска «вширину» с сокращением. Мы можем всегда конструировать ситуацию, в которой этим путем является только выполнимая реализация. Например, если (S_{11}, S_{21}) удаляется из области поиска, мы конструируем регистр сервисов, в котором существует только один сервис S_{31}, который соответствует активности q_3. И пред-условие S_{31} является агрегированным условием *C* в пути (S_{11}, S_{21}), если выбраны S_{11} и S_{21}, то может быть найден S_{31}. Условия *C* не можно отнести к категории агрегированных условий на пути (S_{11}, S_{22}). Иначе путь (S_{11}, S_{21}) удаляется из области поиска «вширину» с сокращением. Поэтому, только пути (S_{11}, S_{21}, S_{31}) являются выполнимой реализацией для целевого сервиса. Если удаляется путь (S_{11}, S_{21}), то схема становится неполной.

3. Алгоритм композиции на основе формирования полноты образца

Полнота схемы, измеряет полноту алгоритмов формирования схемы композиции. В данном случае семантические характеристики, используемые для алгоритмов композиции, заданы в спецификациях целевого сервиса и в реестре сервисов при описании каждого сервиса. Но существуют ситуации, в которых семантические характеристики могут быть определены только после выполнения сервиса.

Для таких ситуаций, мы определяем понятие "полнота образца". Отметим, что полнота образца допускает выполнимые реализации в случаях, когда: 1) для уплотнения условий должны использоваться параметры, получаемые в результате выполнения сервиса; 2) в процессе композиции возможен запуск на выполнение отобранных сервисов.

Рассмотрим способ, при котором алгоритм формирования образца вызывает на выполнение сервисы-кандидаты для активностей, которые инициируют другие активности, в соответствии с целевым сервисом.

Алгоритм композиции образца со стратегией уплотнения «вглубину» является полным. Такое определение полноты, возможно, слишком сильное, в том смысле, что с практической точки

зрения не целесообразно осуществлять запуск все возможных сервисов.

Определение. Алгоритм композиции обеспечивает полноту образца, если для заданного реестра сервисов, соответствующему целевому сервису, и реализации целевого сервиса, алгоритм возвращает выполнимую реализацию каждый раз, когда целевой сервис имеет выполнимую реализацию для запускаемых сервисов из реестра.

Алгоритм композиции полного образца в худшем случае должен осуществить полный перебор, который вызывает почти все сервисы-кандидаты.

Рассмотрим целевой сервис и реестр сервисов (рис. 4). Для любого алгоритма композиции, который не вызывает сервис-кандидат S_{13} для q_1 не обеспечивается полнота образца.

На рис. 4 целевой сервис содержит три активности q_1, q_2 и q_3, q_1 имеет три сервиса-кандидата, а q_2 и q_3 имеют по два. Для каждого сервиса значение x или y в круглых скобках связаны со значениями, получаемыми после выполнения сервиса. Без вызова сервиса для q_1 не может быть найдена выполнимая реализация, потому что ни для одного из входных предусловий S_{21} и S_{22} не может следовать постусловия S_{11}, S_{12} и S_{13}. Однако если сервис S_{13} выполнен, то определяется постусловие $S_{11,}$ которое становится $x = 3$, и которое делает S_{22} сервисом-кандидатом для q_2. Фактически, только (S_{13}, S_{22}, S_{32}) является выполнимой реализацией. Если алгоритм запустит только сервисы S_{11} и S_{12}, он не найдет выполнимую реализацию.

Утверждение. Алгоритм композиции полного образца в наихудшем случае должен запускать все сервисы-кандидаты для активностей кроме конечных активностей.

Алгоритм композиции полного образца использует стратегию вертикального поиска («вглубину») или горизонтального («вширину»), вызывая на выполнение все сервисы из реестра. Очевидно, что он не эффективный. Можно улучшить его через использование похожего алгоритма со стратегией уплотнения «вширину», выше представленной.

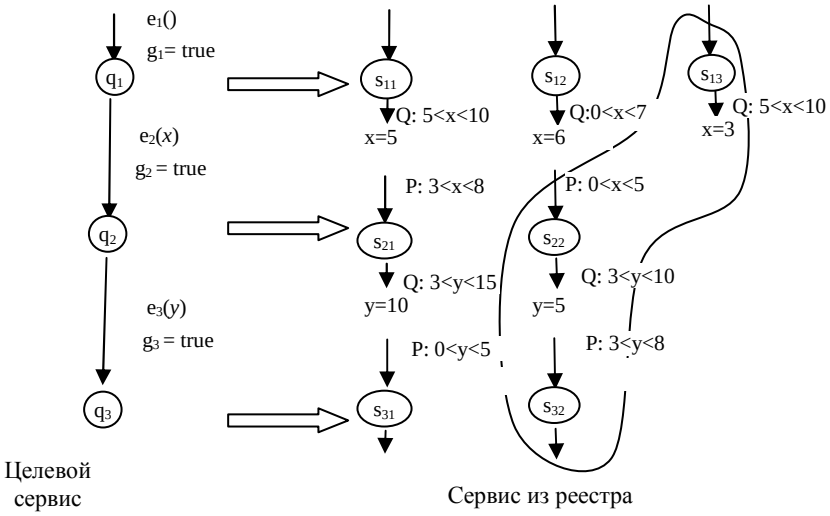

Рис. 4. Иллюстрация полноты образца

На рис. 5 показан пример, в котором целевой сервис имеет линейную структуру. Общий алгоритм композиции образца является полным, если развернута стратегия уплотнения – поиск «вширину». Сервис-кандидат для каждой активности q_i находится с уплотненным начальным условием. Путь в пространстве поиска будет неполным, если отсутствующий сервис не может быть найден следующей

активностью. Выполнимая реализация представляется полным путем в пространстве поиска, если каждой активности в целевом сервисе назначен сервис из реестра.

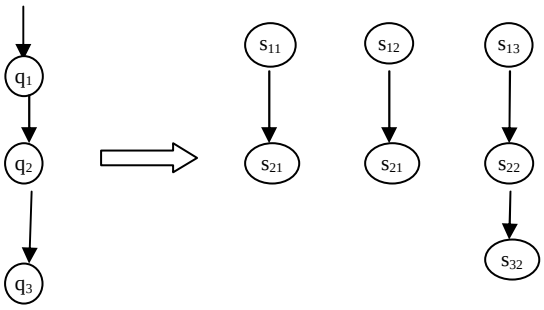

Рис. 5. Пространство поиска для композиции со стратегией горизонтального уплотнения

Вторая стратегия уплотнения запускает выбранные сервисы, связывая свойства в выходных дугах со значениями, полученными в результате выполнения сервиса и уплотнения выходных условий. Эти условия инициируют уплотнение начальных условий для последующей активности.

На рис. 5, если для активности q_1 выбрать S_{13}, то значение после выполнения S_{13} связывается с ограничением (x = = 3) и добавляется к пост-условиям q_1. Обновление пост-условий используется для уплотнения входных условий активности q_2 таким образом, что для q_2 может быть найден сервис S_{22}.

Алгоритмы композиции со второй стратегией уплотнения могут достигать полноты образца в процессе запуска сервисов и связывания значения ограничений, которые обеспечивают уплотнение начальных условий для последующих активностей. Однако, чтобы достичь полноты образца, требуется сделать полный перебор и таким образом вызывать на выполнение все связанные сервисы.

Алгоритмы композиции, которые обеспечивают полноту образца через вызов сервисов-кандидатов, не практичны. Проблема состоит в избыточности использования ресурсов и масштабируемости. Если композитный сервис основан на транзакциях и, если текущий путь не успешный, то алгоритму нужно вернуться на предыдущий уровень.

На практике, возможно, выполнить несколько сервисов выборочно. Понятие полноты образца может служить мерой качества Q_oS для алгоритмов поиска композиции.

Заключение

Автоматизированная композиция Web-сервисов, использующая поиск сервисов – важная практическая проблема. Через формулировку проблемы композиции, мы рассматриваем много связанных ее вопросов. Одно направление – развитие алгоритмов эффективности композиции, которые могут использовать семантические свойства предметных областей. На общем уровне, важно рассмотреть, как процесс поиска может использоваться совместно с запуском сервисов.

1. *Hull R., Su J.* Tools for composite web services: A short overview // SIGMOD Record. − 2005. − N 34(2). − P. 86–95.
2. *Berardi D., Calvanese D., De Giacomo G. et al.* Automatic composition of transition-based semantic web services with messaging // In Proc. 31st Int. Conf. on Very Large Data Bases (VLDB).− 2005.− P. 613–624.
3. *Berardi D., Calvanese D., De Giacomo G., et al.* Automatic composition of e-services that export their behavior // In Proc. 1st Int. Conf. on Service Oriented Computing (ICSOC).− 2003.− P. 43–58.
4. *Gerede C., Hull R., Ibarra O., Su J.* Automated composition of e-services: Lookaheads // In Proc. 2nd Int. Conf. onService-Oriented Computing (ICSOC). − 2004.
5. *Ludaescher B., Altintas I., Berkley C., et al.* Scientific workflow management and the kepler system // J. of Concurrency and Computation: Practice & Experience. Special Issue on Scientific Workflows.
6. *Андон П., Дерецкий В.* Проблемы построения сервис-ориентированных прикладных информационных систем в Semantic web среде на основе агентного подхода // Проблемы программирования. − 2006. − № 2-3, − С. 493–502.
7. *Rao J., Su X.* A survey of automated web-service composition methods // In Proc. of the 1st Int.Workshop on Semantic Web Services and Web Process Composition. − 2004.

8. *Elgedawy I., Tari Z., Winikoff M.* Exact functional context matching for web services // In Proc. Int. Conf. on ServiceOriented Computing (ICSOC*).*− 2004.− P. 143–152.
9. *Andon Ph., Deretsky V.* Control Oriented Ontology and Process Description for Cooperation Agents in Information Retrieval // Sixth International Scientific Conference „Electronic Computers and Informatics ECI'2004". X Kosice – Herlany, Slovakia September 22-24.− 2004.− P. 14–18.
10. *Дерецкий В.* Подход к композиции Веб-сервисов на основе специификации функциональной семантики // Проблемы программирования. − 2009. − № 2. − С. 30–39.
11. *Bernstein A. and Klein M.* Discovering services: Towards high precision service retrieval // In Proc. of the CaiSE workshop on Web Services, e-Business, and the Semantic Web. – 2002.
12. *Cardoso J. and Sheth A.* Semantic e-workflow composition // Jnl. of Intelligent Info. Systems. – 2003. – N 21(3). – P. 191–225.
13. *Lu S.* Semantic Correctness of Transactions and Workflows // PhD thesis, SUNY at Stony Brook. –2002. – P. 8–18.
14. *McIlraith S. and Son T.C.* Adapting Colog for Composition of Semantic Web Services // In Proc. KRR. − 2002. − P. 482–493.
15. *Berardi D., Calvanese D., Giacomo G.D., Lenzerini M. and Mecella M.* Automatic composition of e-services that export their behavior. ICSOC. – 2003.
16. *Gerede C., Hul lR., Ibarra O., Su J.* Automated composition of e-services: Lookaheads // In Proc. 2nd Int. Conf. onService-Oriented Computing (ICSOC). − 2004.
17. *Fu X. Bultan T. and Su J.* Conversation protocols: A formalism for specification and verification of reactive electronic services // Theoretical Computer Science. − 2004. – N 328 (1-2).
18. *Fu X., T. Bultan, and Su J.* Realizability of conversation protocols with message contents. // International Journal of Web Services Research. – 2005.– N 2. – P. 68–93.
19. *W. M. P. van der Aalst.* On the automatic generation of Workflow processes based on product structures // Computer in Industry. – 1999. – Vol. 39. N.2. – P. 97 – 111.
20. *Tsalgatidou A., Athanasopoulos G. and et al.* Developing scientific WorkFlows from heterogeneous services // ACM Sigmod Record. – 2006. – N 2. – P. 22–28; 1999. – N 39(2). – P. 97–111.
21. *Ludaescher, Altintas I., Berkley C. and et al.* Scientific Workflow management and the kepler system // J. of Concurrency and Computation. Practice & Experience. Special Issue on Scientific Workflows. http://users. sdsc.edu/ ~ludaesch/Paper/kepler-swf.pdf
22. *Benatallah B., Dumas M., and et al.* Declarative composition and peer-to-peer provisioning of dynamic web services. ICDE. – 2002.
23. *Zeng L., Benatallah B. and et al.* Qos-aware middleware for web services composition // IEEE Transactions on Software Engineering. – 2004. – N 30(5). – P. 311–327.
24. *Narayanan S. and McIlraith S.* Simulation, verification and automated composition of web services. http://www2002.org/presentations/narayanan.pdf.
25. *Narayanan S. and McIlraith S.* Analysis and simulation of web services // Computer Networkds. – 2003. – N 42. – P. 675–693.
26. *Berardi D., Calvanese D., Giacomo G. D. and et al.* Automatic composition of transition-based semantic Web services with messaging. VLDB. – 2005.
27. *Semantic* Web Services Ontology (SWSO) 1.0. – 2005. September. http://www.w3.org/ Submission/SWSF-SWSO/.
28. Web Service Modeling Ontology. http://www.wsmo.org/.
29. *Rao J. and Su X.* A survey of automated Web service composition methods // In Proc. of the 1st Int.Workshop on SemanticWeb Services and Web Process Composition. – 2004.
30. OWL-S 1.1 Release.– 2004. November. http://www.daml.org/services/ OWL-S/1.1/
31. W3C. Web services semantics – WSDL-S 1.0. – 2005. Nov. http://www.w3.org/Submission/ WSDL-/

Получено 08.05.2009

Об авторе:

Дерецкий Валентин Александрович,
кандидат физико-математических наук,
ведущий научный сотрудник.

Место работы автора:

Институт программных систем
НАН Украины.
03187, Киев-187,
проспект Академика Глушкова, 40.
Тел.: 38 044 526 4342.
e-mail: dva@isofts.kiev.ua.