

Формирование композиций сервисов на основе статистических данных пользователей

Р. К. Фёдоров, И. В. Бычков, Г. М. Ружников

*Институт динамики систем
и теории управления имени В. М. Матросова СО РАН
Иркутск, Россия*

Аннотация

Рассмотрено решение задачи автоматического формирования композиций сервисов. Предложенный метод формирует композиции сервисов на основе обработки статистических данных об отдельных применениях сервисов (заданий) пользователями. В основе метода лежит определение связей вызовов сервисов друг с другом по данным. Далее выделяются параметры сервисов, значения которых жестко заданы композицией сервисов, и параметры, значения которых может изменить пользователь. Формируются композиции сервисов в виде направленного графа заданий DAG. Разработаны методы редукции множества получаемых композиций сервисов, позволяющие выделить завершённые и ранжировать их по степени используемости. В частности, определяются эквивалентные композиции сервисов на основе изоморфизма графов DAG, отбрасываются тривиальные и оставляются только композиции, приводящие к публикуемому результату.

Ключевые слова

SOA, OGC, WPS, обработка пространственных данных, семантическая сеть, композиция сервисов

Благодарности

Работа выполнена при поддержке ЦКП ИИВС ИРНОК, базового проекта АААА-А17-117032210079-1, гранта Минобрнауки России № 075-15-2020-787, проектов РФФИ 18-07-00758, 17-57-44006, блока интеграционной программы ИНЦ СО РАН 4.1.2

Для цитирования

Фёдоров Р. К., Бычков И. В., Ружников Г. М. Формирование композиций сервисов на основе статистических данных пользователей // Вестник НГУ. Серия: Информационные технологии. 2021. Т. 19, № 2. С. 115–130. DOI 10.25205/1818-7900-2021-19-2-115-130

Building Service Compositions Based on data on Use of Services by Users

R. K. Fedorov, I. V. Bychkov, G. M. Ruginov

*Matrosov Institute for System Dynamics and Control Theory SB RAS
Irkutsk, Russian Federation*

Abstract

The automatic service composition is discussed in the article. The method is proposed for building the service composition based on the processing of statistical data on individual applying services (tasks) by users. The method is based on linking tasks to each other, determining data dependencies, parameters of services whose values are rigidly set by the composition of services, and parameters whose values can be changed by the user are highlighted. Service compositions are built in the form of a directed graph of DAG. The methods have been developed for reducing the set of obtained service compositions, which allow us to highlight useful ones and rank them by degree of use. In particular, equivalent service compositions based on isomorphism of DAG graphs are determined, trivial ones are discarded, and only compositions that lead to the published result are left behind.

Keywords

SOA, OGC, WPS, spatial data processing, semantic network, composition of services

Acknowledgements

This work was supported by the Center for Collective Use of the IISS IRNOK, no. AAAA-A17-117032210079-1, no. 075-15-2020-787 of the Ministry of Science and Higher Education of the Russian Federation, no. 0348-2018-0003 of the SB RAS, RFBR Projects no. 18-07-00758, 17-57-44006, ISC SB RAS integration program 4.1.2

For citation

Fedorov R. K., Bychkov I. V., Ruznikov G. M. Building Service Compositions Based on data on Use of Services by Users. *Vestnik NSU. Series: Information Technologies*, 2021, vol. 19, no. 2, p. 115–130. (in Russ.) DOI 10.25205/1818-7900-2021-19-2-115-130

Введение

В современном мире активно растет количество Web-сервисов [1], выполняющих предоставление, обработку и публикацию данных, в частности сервисы для обработки и предоставления данных дистанционного зондирования земли (ДЗЗ). Широко используются облачные хранилища данных, такие как интернет-сервисы: Dropbox, OneDrive, Google Drive, iCloud, Яндекс.Диск, Облако Mail.Ru и др. Применяются различные стандарты и спецификации, унифицирующие вызов сервисов, передачу и получение данных, например стандарт обработки пространственных данных OGC WPS [2], набор спецификаций REST [3]. Формируются каталоги сервисов, содержащих метаданные, для поиска необходимых сервисов. Сервисы активно применяются в решении сложных информационно-вычислительных задач, упрощая использование программного обеспечения за счет исключения процессов его установки и настройки.

Обычно для пользователей, не являющихся специалистами в области информационных технологий, применение сервисов реализовано с помощью специальной формы, где можно указать значения параметров и запустить сервис. Часто решение задачи может требовать многократного вызова последовательности (цепочки) сервисов, при этом пользователю приходится придерживаться определенного порядка вызова сервисов, вводить в форме значения параметров для каждого сервиса и т. д. В этом случае для автоматизации работы пользователя требуется создание композиции Web-сервисов [4]. Обычно композиции сервисов задаются с помощью языков для формального описания бизнес-процессов, например BPEL (Business Process Execution Language), BPMN (Business Process Modeling Notation), BPML (Business Process Management Language) и XPDЛ (XML Process Definition Language). Также для задания композиций сервисов активно используются популярные языки программирования, например Python, JavaScript. Существует ряд систем workflow («поток задач»), позволяющих формировать композиции сервисов в различных предметных областях: Pegasus [5], Kepler [6], Swift [7], KNIME [8], Taverna [9], Galaxy [10], Trident [11] and Triana [12], Everest (Mathcloud) [13], CLAVIRE [14]. Для обработки пространственных данных активно используется Geo-Processing Workflows [15]. Общепринятой формализацией композиции сервисов является направленный ациклический граф (Directed acyclic graph, DAG) [16; 17], в котором вершинами являются вызовы сервисов, а дугами – зависимости между сервисами по данным. Создание композиции сервисов – это сложный процесс, требующий навыков программирования от пользователя, поиска подходящих сервисов среди их большого количества, знания языков задания композиций сервисов, больших временных издержек на разработку и отладку. Существует несколько подходов к автоматизации создания композиций сервисов [18].

1. Интерактивное построение композиций сервисов на основе знаний. В этом подходе пользователь вручную строит композицию сервисов, а система вывода и знания (метаописания, онтологии) используется для поиска подходящих сервисов и значений параметров [19–21].

2. Автоматическое построение композиций сервисов. В этом подходе производится (полу) автоматическое обнаружение и составление необходимых сервисов на основе семантического анализа и логического вывода [22–25].

3. Автоматическая композиция Web-сервисов с учетом планирования выполнения. В этом подходе [26; 27] входные данные представляются сервисами, которые семантически аннотированы с использованием онтологий, например Web Ontology Language for Services (OWL-S) [28], сервисы обработки аннотированы с помощью Web Service Modeling Ontology (WSMO) [29]. Выбор среди сгенерированных цепочек сервисов может проводиться на основе оценки качества обслуживания (QoS) с использованием алгоритмов, таких как генетические алгоритмы или алгоритмы теории игр [30; 31].

Следует отметить, что автоматическое построение композиций сервисов на основе семантического анализа, логического вывода и планирования является сложной задачей по следующим причинам:

- большая часть сервисов имеет недостаточно метаданных, либо она отсутствует или не формализована;
- требуется проработанная онтология предметной области, ее построение не тривиальный процесс, в котором нужно рассмотреть все возможные связи между сервисами;
- требуется согласованность метаданных сервисов и онтологии предметной области.

Обычно при решении конкретной задачи пользователь составляет некоторую последовательность выполнения сервисов на основе их неформализованных описаний и примеров решений подобных задач. Одна и та же последовательность выполнения сервисов может повторяться многократно. Автоматизация выполнения последовательности сервисов позволит упростить и ускорить работу пользователя. Выполняя последовательность сервисов, пользователь задает неявную информацию о связях между сервисами, используемых значениях параметров сервисов, которая может быть полезна для автоматического составления композиций сервисов. Поэтому актуальна задача автоматизации формирования композиций сервисов на основе статистических данных о применении сервисов пользователями. Авторами предлагается метод, который связывает одиночные вызовы сервисов друг с другом и определяет зависимости по данным. В статье будем рассматривать анализ только статистических данных и намеренно опускаем возможность применения метаданных и онтологий, оставляя их для дальнейшего исследования.

Сбор статистических данных о применении сервисов

В рамках геопортала ИДСТУ СО РАН [32] создана и развивается распределенная информационно-вычислительная среда, которая в настоящее время применяется преимущественно для задач обработки пространственных данных. Разработаны информационно-вычислительные сервисы, реализованные в соответствии со стандартом WPS (OGC, Open Geospatial Consortium) и сервисы хранения данных, каталог WPS-сервисов и системы выполнения сервисов и их композиций. Реализовано взаимодействие между WPS-сервисами и сервисами хранения данных, что позволяет упростить использование сервисов пользователем. Для каждого сервиса известно его расположение, список входных и выходных параметров, типы данных. При выполнении сервисов и их композиций автоматически формируются данные о произведенных вызовах сервисов, которые включают название сервиса и его адрес, значения входных и выходных параметров, время выполнения сервисов, успешность выполнения, ошибки выполнения и т. д. Данные сохраняются в таблице, где значения параметров хранятся в формате JSON (см. далее). Статистические данные о применении сервисов используются для формирования рекомендаций сервисов для пользователей [33; 34].

Таблица заданий
Task table

Идентификатор задания	Идентификатор сервиса	Статус	Входные параметры	Выходные параметры	Начало	Конец
3986	309	Success	{"table": "ticki"}	{"file": "http://84"}	18:20	18:22
3989	307	Failed	{"disp": "20"}		18:21	
3985	308	Success	{"file": "http://84..."}	{"res": "http://84"}	18:21	18:22
3987	307	Success	{"file": "http://84..."}	{"dist": "28",	19:18	19:19

Введем следующие обозначения, необходимые для описания метода:

$s = \langle name, I, O \rangle$ – сервис, где *name* – это имя сервиса, *I* – множество входных параметров, *O* – множество выходных параметров сервиса. Далее будем обозначать *s.I* и *s.O* параметры, принадлежащие определенному сервису. Пользовательский вызов сервиса имеет одинаковую структуру: исходные параметры, известные пользователю, и параметры, которые должны быть получены;

$t = \langle s, V_I, V_O \rangle$ – задание, т. е. вызов сервиса *s* с множеством значений *V_I* входных параметров *s.I* и результаты *V_O* выходных параметров *s.O*;

Log – это множество заданий (успешных вызовов сервисов), совершенных пользователями. Не все вызовы сервисов успешные. Причиной неудачного вызова сервиса могут быть некорректные значения параметров, поэтому данные фильтруются, и оставляются только успешные вызовы.

На основе данных *Log* необходимо сформировать множество композиций сервисов в виде DAG.

Определение передачи данных между двумя заданиями

Первым этапом построения композиции сервисов является фиксация факта передачи данных между заданиями, которая производится на основе анализа данных, передаваемых в качестве значений параметров. Параметры сервисов в общем можно разделить на строковые, числовые и идентификаторы ресурсов URI. Предполагается, что если значение выходного параметра совпадает со значением входного параметра, то между заданиями может быть факт передачи данных. При этом необходимо учитывать, что равенство значений входных и выходных параметров может быть случайным. Однако вероятность ввода пользователем значения входного параметра совпадающего со значением выходного параметра другого задания обычно не высока, но и не нулевая. Определение факта передачи данных между заданиями в общем случае сводится к задаче распознавания, которую планируется исследовать в будущем. На текущий момент рассматриваются только идентификаторы ресурсов URI, не требующие применения методов распознавания. Обычно URI – это ссылки на файлы, сервисы предоставления данных и т. д., которые однозначно идентифицируют данные. Например, в современных распределенных базах данных в качестве первичного ключа вместе GUID часто используют URI. Большинство сервисов для каждого результата обработки генерируют уникальный URI. Это необходимо, чтобы гарантировать передачу результата, соответствующего запросу, в условиях часто одновременной обработки данных для множества клиентов сервиса. Следовательно, по совпадению URI значений параметров можно однозначно определить, что данные передаются от одного задания другому, т. е. если сервис однажды произвел данные, которые были использованы другим сервисом, то эти сервисы точно могут быть связаны по данным (достаточное условие связанности). Со строковыми и числовыми параметрами такой критерий не сработает, их можно будет использовать только как «необходи-

мое условие» связывания сервисов. Строковые и числовые параметры планируется рассмотреть в будущем.

Приведем алгоритм проверки передачи данных между двумя заданиями.

Input: t_i, t_j – два задания

Output: flag – наличие связи между заданиями.

```

1  function islinked( $t_i, t_j$ )
2  flag <- False
3  for each  $v_m$  in  $t_i.VO$ :
4    for each  $v_n$  in  $t_j.VI$ :
5      If (typeof  $v_m$  = URI and typeof  $v_n$  = URI and  $v_m = v_n$ ) then
6        flag <- true
7         $V_n.linkedwith <- V_m$ 
8      end if
9    end for
10 end for
11 return flag

```

Алгоритм проводит сравнение значений выходных параметров одного задания со значениями входных параметров другого задания, которые являются ссылками на файлы (URI). При совпадении значений URI параметров фиксируется наличие связи между двумя заданиями и определяется, по каким параметрам они связаны. Далее эти параметры будем называть связанными, а остальные параметры свободными.

Создание направленного ациклического графа заданий

Применение алгоритма проверки передачи данных для каждой пары заданий приводит к созданию направленного ациклического графа DAG_{gen} (рис. 1), где вершины задания t_i , а направленные дуги обозначают передачу данных между заданиями. Дополнительно получаем значения параметров заданий, введенных пользователем.

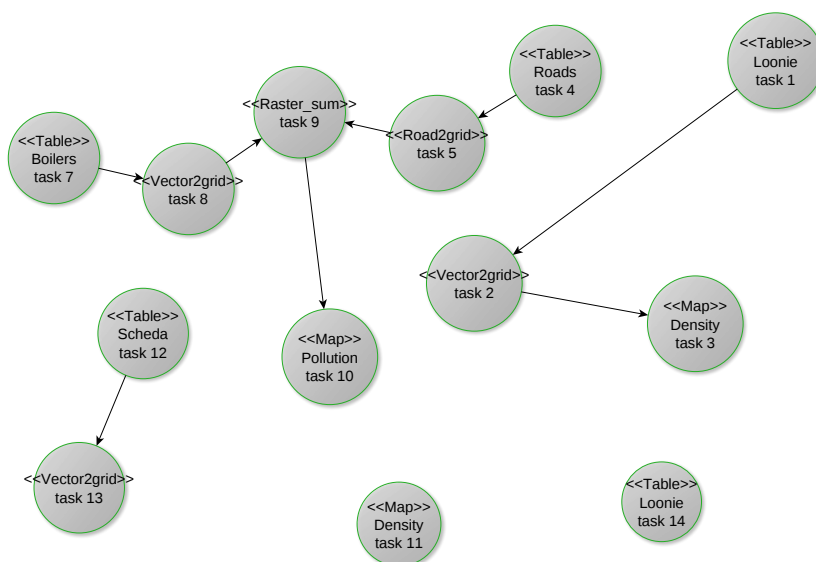


Рис. 1. Граф заданий DAG_{gen}
Fig. 1. The graph of tasks DAG_{gen}

Если при выполнении данного алгоритма сохранять связи между сервисами вместо заданий, то получим семантическую сеть сервисов. На рис. 2 представлена часть полученной сети. Сервисы представлены окружностями с подписанными названиями сервисов, семантические связи – направленными дугами. Необходимо отметить, что при таком подходе семантические связи между сервисами подтверждаются рабочими примерами.

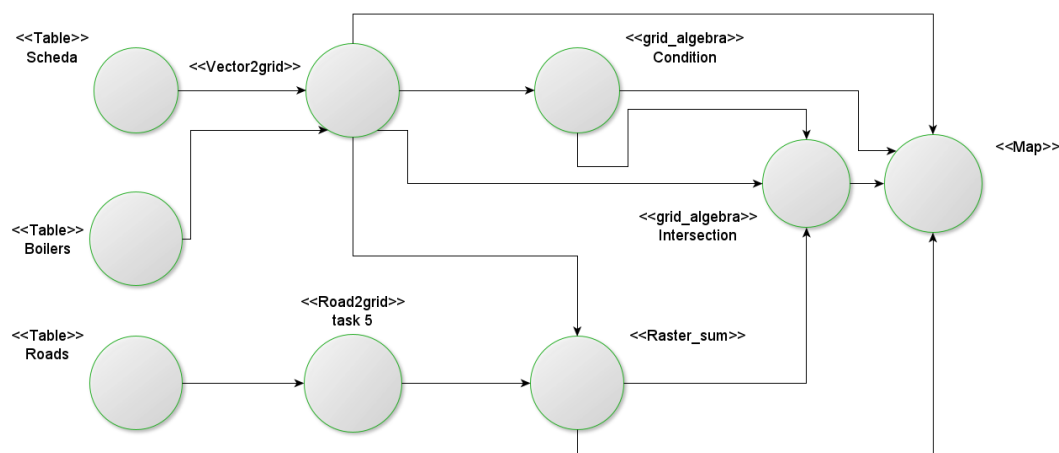


Рис. 2. Полученная семантическая сеть

Fig. 2. The resulting semantic network

Декомпозиция графа заданий

Граф заданий DAG_{gen} содержит результаты решения пользователем разных задач. Следовательно, его необходимо декомпозировать и выделить задания, соответствующие решаемым задачам.

Пользователь может выполнить некоторую последовательность сервисов, задающую последовательность заданий. Если задания связаны по данным (на текущий момент учитываются связи по URI), то они образуют связанный подграф D_i (компонент связности графа) в DAG_{gen} . Почти всегда выполнение композиции сервисов приводит к созданию такого связанного подграфа D_i . Если задания, сформированные композицией сервисов, не связаны, тогда такую композицию сервисов можно разделить. Для получения подграфа, соответствующего решению задачи, не достаточно найти компоненты связности в DAG_{gen} . Так как в подграфе связности могут быть задания получения данных. Например, задания могут выполняться сервисами получения данных ДЗЗ или сервисами предоставления векторных и реляционных данных. Эти данные могут использоваться для решения большого количества различных задач. Например, данные ДЗЗ Landsat 8, которые можно получить с помощью сервисов, используются для задач оценки подтоплений разливов рек, динамики изменения ледников, классификации породного состава лесов и т. д. Каждому решению задачи будет соответствовать свой подграф в DAG_{gen} , и они будут все связаны между собой через задание получения данных ДЗЗ, т. е. объединяться в компоненты связности. Поэтому при выделении компонентов связности необходимо разделять подграфы, связанные только через задания получения данных. Учет и выделение заданий, выполняемых сервисами предоставления данных, при выполнении анализа может быть полезно для определения параметров композиций сервисов, т. е. сервисов, которые могут быть заменены на другие при ее повторном выполнении.

Для поиска связанных подграфов D_i применяется алгоритм поиска в ширину. При этом сложность поиска линейно зависит от суммы числа вершин и числа рёбер графа DAG_{gen} .

Алгоритм поиска связанного подграфа композиции сервисов

В статье для описания алгоритмов применяется псевдокод, использующий ключевые слова императивных языков программирования. Алгоритм поиска связанного подграфа реализует алгоритм поиска в ширину.

Input: nodes – список заданий, у каждого задания есть список предшествующих prevs и последующих заданий nexts.

Output: subgraphs – список связанных подграфов.

```
1  Stack <- [];
2  while(nodes is not empty):// перебираем все задания
3      new subgraph
4      add subgraph into subgraphs
5      first <- nodes.pop()
6      add first into subgraph
7      add first into stack
8      while(stack is not empty){//перебор всех связанных заданий
9          cur <- stack.pop()
10         for each n in cur.nexts:
11             if n exists in nodes then
12                 add n into subgraph.nodes
13                 add n into stack
14                 remove n in nodes
15             end if
16         end for
17         for each n in cur.prevs:
18             if n exists in nodes then
19                 add n into subgraph.nodes
20                 add n into stack
21                 remove n in nodes
22             end if
23         end for
24     end while
25 end while
```

В результате выполнения алгоритма получаем множество связанных подграфов D_i (рис. 3). На рисунке задания получения данных выделены пунктирной линией. Задания публикации данных имеют более темную заливку.

Методы редукции множества композиций сервисов

Алгоритм поиска связанных подграфов D_i может получить их достаточно много. Часть подграфов может соответствовать экспериментам, которые привели к недостаточно хорошим результатам. Также среди них может быть достаточно много дублирований, которые возникают из-за многократного решения задачи на разных данных. Поэтому возникает задача редукции множества D_i . Композиции сервисов, состоящие из одной вершины (одного сервиса), являются тривиальными, и их можно отбросить. Одним из объективных критериев необходимости связанного подграфа заданий может быть получение пользователем конечного результата. Предполагается, что результат будет конечным, если пользователь его публикует. Например, на основе полученных данных создаются карты, графики, таблицы и т. д. В общем среди всех связанных подграфов выделяются D_i , состоящие как минимум из двух вершин и имеющие хотя бы одно задание публикации данных.

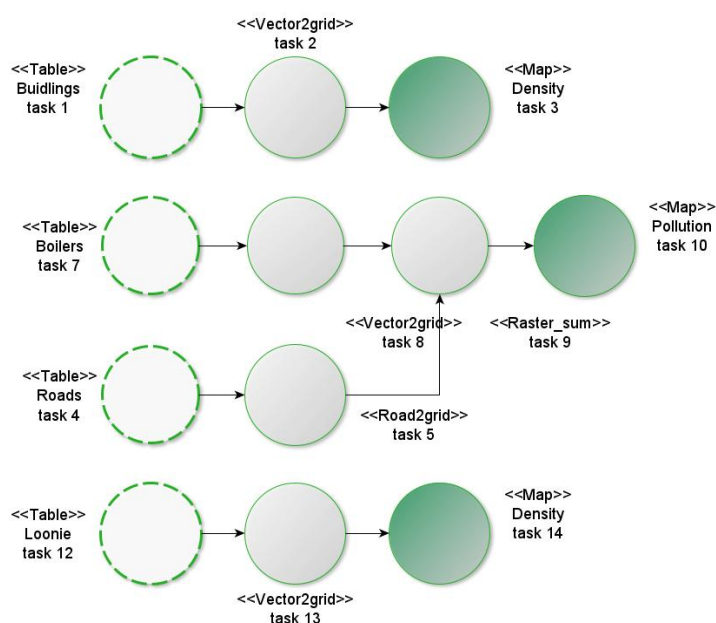


Рис. 3. Связанные подграфы D_i
 Fig. 3. Linked subgraphs of D_i

Для декомпозиции графа заданий требуется предварительная классификация сервисов, которая производится вручную при регистрации сервиса в каталоге. Выделим три класса сервисов:

- предоставления данных, как правило, с них начинается выполнение композиции сервисов, в композиции сервисов они могут быть заменены;
- публикация данных, все сервисы данного класса должны требовать от пользователя метки семантики;
- вычислительные сервисы, выполняющие обработку данных.

Поиск эквивалентных подграфов

Часто последовательность применения сервисов многократно повторяются на разных наборах данных. При этом могут меняться входные данные, часть которых предоставляется сервисами. Это приводит к созданию нескольких подграфов D_i , в которых производятся вызовы одних и тех же вычислительных сервисов, но отличаются значениями параметров и сервисами предоставления данных. Заменяем в связанных подграфах D_i задания t на соответствующие сервисы s . Ребра оставим без изменений, и в новом графе SD_i они вместо заданий будут связывать сервисы. Повторение одной и той же последовательность вызовов сервисов приведет к созданию изоморфных подграфов SD_i относительно сервисов и связей по данным без учета сервисов предоставления данных. Проверка на изоморфность двух графов осуществляется ниже следующим алгоритмом.

Алгоритм выделения изоморфных графов

Графы SD_i являются направленными, что значительно упрощает их сравнение. Предварительно для каждого графа добавляется узел, с которого начинается выполнение композиции. Начальное задание не имеет реальный вызов сервиса. Все задания, не зависящие по данным

от других, добавляются к начальному заданию. Сравнение двух графов начинается с начальных заданий и выполняется рекурсивно. Два задания являются изоморфными, если названия сервисов совпадают, и являются изоморфными все зависящие от них задания.

Input: $n1, n2$ – графы SD_i .

Output: true или false, являются графы изоморфными или нет.

```

1  function isonodes(n1,n2)
2  if n1.nexts.length <> n2.nexts.length then
3    return false
4  end if
5  for each nx1 in n1.nexts:
6    found <- false
7    for each nx2 in n2.nexts:
8      if nx1.service = nx2.service and isonodes(nx1,nx2) then
9        found <- true
10       break
11     end if
12   end for
13   if not found then
14     return false
15   end if
16 end for
17 return true

```

Изоморфные подграфы SD_i не всегда соответствуют одной и той же композиции сервисов. Например, один и тот же набор сервисов может применяться для получения разных результатов при изменении входных параметров. При дешифрировании данных ДЗЗ могут использоваться параметры, задающие различные пороговые значения. Их изменения могут привести к получению принципиально других результатов. С другой стороны, изменение параметров не всегда означает применение другой композиции вычислительных сервисов. Например, композиция сервисов может быть применена для другой территории, т. е. изменен параметр, задающий территорию, но решает ту же самую задачу. Возникает проблема определения, соответствуют ли два изоморфных графа решению одной задачи. Одним из способов решения этой проблемы является определение соответствия семантики результатов изоморфных графов. Автоматическое определение семантики результатов является нетривиальной задачей, поэтому предлагается воспользоваться метками пользователей. Пользователи часто сами дают такие метки, например названия карт, таблиц и графиков. В рамках среды можно обязать пользователей при публикации данных вручную указывать эти метки.

На рис. 4 приводится результат определения двух изоморфных подграфов SD_i , которые имеют одинаковую метку «Density» на сервисе публикации карт «Map». Изоморфные подграфы SD_i с одинаковой семантикой результатов объединяются (рис. 5). Количество изоморфных подграфов SD_i является оценкой частоты использования этой композиции сервисов и может применяться для их ранжирования и выделения наиболее полезных композиций сервисов, например, как работе [33].

Выполнение композиций сервисов

Подграфы SD_i являются искомыми композициями сервисов. Для того чтобы выполнить композицию сервисов, представленную подграфом SD_i , необходимо пользователю задать все свободные входные параметры, и переопределить сервисы предоставления данных. Все эти данные могут быть введены пользователем на генерируемой форме выполнения композиции

сервисов. В качестве значений по умолчанию предлагаются наиболее часто используемые значения среди всех изоморфных подграфов SD_i .

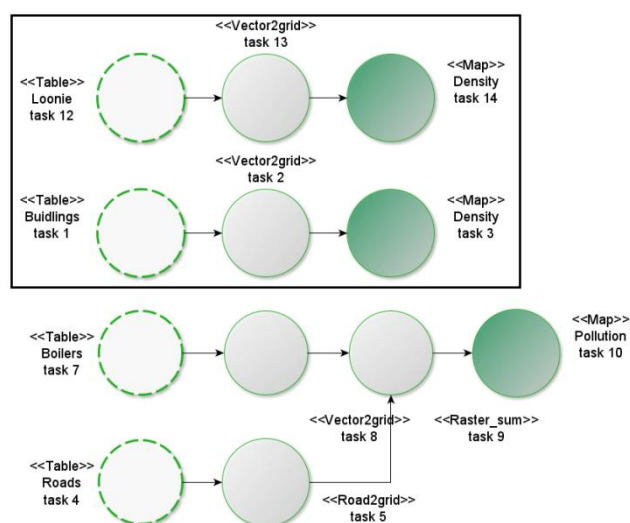


Рис. 4. Изоморфные подграфы выделены прямоугольником

Fig. 4. Isomorphic subgraphs are highlighted with a rectangle

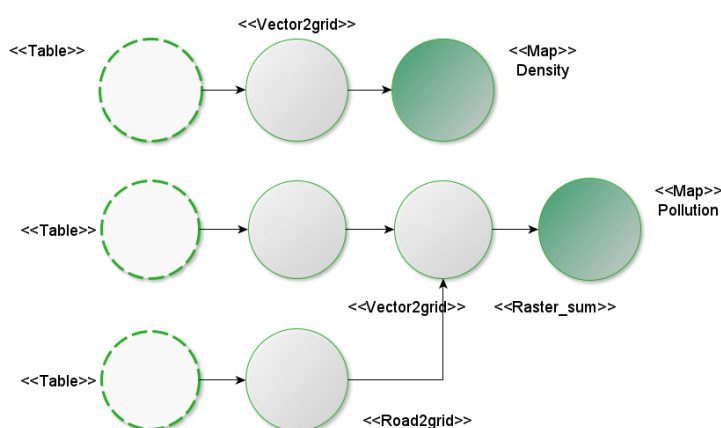


Рис. 5. Композиции сервисов

Fig. 5. Service compositions

Апробация

Для многих пространственных данных часто строят карты плотности (тепловые карты), которые дают представление о пространственном распределении объектов. Например, при изучении того или иного вида растения является важным исследование ареала его распространения. Специалистами собирается информация о находках различных видов растений в виде таблицы. Для создания карты распространения вида на основе этой таблицы в рамках геопортала используется три сервиса:

1) Table – сохраняет таблицу геопортала в векторный формат SHAPE, пользователю необходимо указать таблицу, файл результата;

2) Vector2grid – подсчитывает количество растений в ячейках регулярной сетки и сохраняет в формате GeoTIFF, пользователю необходимо указать размер ячейки, файл в формате SHAPE, файл результата;

3) Map – создает карту ареала распространения вида, пользователю необходимо указать файл в формате GeoTIFF, название карты, классы и стили отображения данных на карте.

Для сравнения эффективности автоматизации выполнения сервисов произведем подсчет условных действий пользователя. Будем учитывать количество действий поиска сервисов и ввод параметров. Первый сервис требует три действия, второй и третий – четыре. Итого имеем 11 действий пользователя для создания карты ареала распространения вида.

Выполнив последовательно все три сервиса, пользователь автоматически получает композицию сервисов (рис. 6).

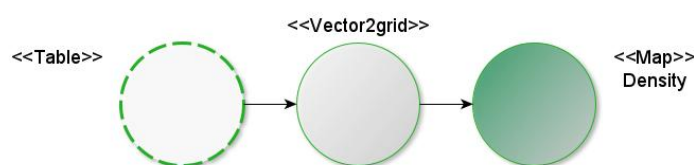


Рис. 6. Композиция сервисов для построения ареала распространения
Fig. 6. The service composition for calculating distribution area

При выполнении данной композиции пользователю необходимо выбрать композицию сервисов и указать таблицу, т. е. всего два действия пользователя. Конечно, любые свободные параметры сервисов могут быть изменены пользователем. В этом случае количество действий будет равно 5. Построение карты ареала распространения может выполняться для разных видов, по разным годам. Использование автоматически созданной композиции сервисов может значительно упростить работу пользователя.

Заключение

В статье рассмотрено решение актуальной задачи автоматического формирования композиций сервисов. Предложенный метод формирует композиции сервисов на основе обработки статистических данных об отдельных применениях сервисов (заданий) пользователями. В основе метода лежит определение связей вызовов сервисов друг с другом по данным. Далее выделяются параметры сервисов, значения которых жестко заданы композицией сервисов, и параметры, значения которых может изменить пользователь. Формируются композиции сервисов в виде направленного графа заданий DAG. Разработаны методы редукции множества получаемых композиций сервисов, позволяющие выделить полезные и ранжировать их по степени использования. В частности, определяются эквивалентные композиции сервисов на основе изоморфизма графов DAG, отбрасываются тривиальные и оставляются только композиции, приводящие к публикуемому результату.

В геопортале формируется список композиций сервисов, где у каждой композиции в качестве заголовка указывается название получаемых результатов. Список композиций сервисов ранжируется по мере их используемости, т. е. по количеству эквивалентных композиций сервисов. При выборе композиции пользователем формируется форма ввода параметров сервисов. Пользователю предлагается воспользоваться введенными значениями параметров сервисов или определить новые значения. После ввода параметров пользователь может выполнить композицию сервисов.

Разработанный метод имеет практическую значимость, которая заключается в том, что значительно упрощается разработка новых композиций для автоматизации повторяющихся действий пользователя. Для формирования композиции пользователю достаточно последовательно выполнить необходимые сервисы. Кроме того запоминаются значения параметров, что упрощает применение композиций сервисов. Кроме композиции сервисов строится их семантическая сеть, в которой семантические связи между сервисами подтверждаются работоспособным примером выполнения.

В будущем планируется развитие метода для реализации совместного использования метаданных, онтологий и статистических данных при построении композиций сервисов, распознавания семантики сервисов, разработки ранжирования на основе многопользовательских статистических данных.

Список литературы

1. **Grimm S., Abecker A., Volker J., Studer R.** Ontologies and the Semantic Web. *Handbook of semantic web technologies: foundations and technologies*, 2011, vol. 1, p. 507–579.
2. **Schut P.** OpenGIS ® Web Processing Service. *Open Geospatial Consortium*, 2007, no. 6, p. 1–3.
3. **Pautasso C.** RESTful Web service composition with BPEL for REST. *Data knowledge*, 2009, vol. 68, no. 9, p. 851–866.
4. **Hoffmann J., Weber I.** Web Service Composition. In: *Encyclopedia of Social Network Analysis and Mining*. Springer-Verlag, 2014.
5. **Deelman E., Vahi K., Juve G.** Pegasus, a workflow management system for science automation. *Future Generation Computer Systems*, 2015, vol. 46, p. 17–35.
6. **Ludäscher B., Altintas C., Berkley C., Higgins D., Jaeger E., Matthew J., Edward A. L., Tao J., Zhao Y.** Scientific Workflow Management and the Kepler System. *Special Issue: Workflow in Grid Systems. Concurrency and Computation: Practice & Experience*, 2006, vol. 18 (10), p. 1039–1065.
7. **Wilde M., Hategan M., Wozniak J. M.** Swift: A language for distributed parallel scripting. *Parallel Computing*, 2011, vol. 37 (9), p. 633–652.
8. **Berthold M. R., Cebron N., Dill F.** The konstanz information miner. *SIGKDD Explorations*, 2009, no. 11, p. 26–31.
9. **Wolstencroft K., Haines R., Fellows D.** The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud. *Nucleic Acids Research*, 2013, vol. 41 (W1), p. 557–561.
10. **Blankenberg D., Kuster G. V., Coraor N.** Galaxy: A Web-Based Genome Analysis Tool for Experimentalists. Wiley, 2010.
11. **Simmhan Y., Barga R., Ingen C.** Building the trident scientific workflow workbench for data management in the cloud. In: *Advanced Engineering Computing and Applications in Sciences (ADVCOMP)*, 2009. DOI 10.1109/ADVCOMP.2009.14
12. **Churches D., Gombas G., Harrison A.** Programming scientific and distributed workflow with Triana services: Research articles. *Concurrency and Computation: Practice and Experience*, 2006, vol. 18 (10), p. 1021–1037.
13. **Smirnov S., Sukhoroslov O., Volkov S.** Integration and Combined Use of Distributed Computing Resources with Everest. *Procedia Computer Science*, 2016, vol. 101, p. 359–368.
14. **Бухановский А. В., Васильев В. Н., Виноградов В. Н., Смирнов Д. Ю., Сухоруков С. А., Яппаров Т. Г.** CLAVIRE: Perspective Technology for Second Generation Cloud Computing // *Современные тенденции развития распределенных вычислений*. 2011, Т. 54, № 10. С. 7–14.

15. **Chen N. C., Di L. P., Yu G. N., Gong J. Y.** Geo-processing workflow driven wildfire hot pixel detection under sensor web environment. *Computers & geosciences*, 2010, vol. 36, no. 3, p. 362–372.
16. **Kwok Y.-K., Ahmad I.** Static scheduling algorithms for allocating directed task graphs to multiprocessors. *ACM Computing Surveys*, 1999, vol. 31, no. 4, p. 406–471.
17. **Xie G., Li R., Xiao X., Chen Y.** A High-Performance DAG Task Scheduling Algorithm for Heterogeneous Networked Embedded Systems. In: Proc. of IEEE 28th International Conference Advanced Information Networking and Applications, 2014, p. 1011–1016.
18. **Zhi-Wei H., Cheng-Zhi Q., A-Xing Z., Peng L., Yi-Jie W., Yun-Qiang Z.** From Manual to Intelligent: A Review of Input Data Preparation Methods for Geographic Modeling. *ISPRS International journal of geo-information*, 2019, vol. 8, no. 9, article 376. DOI 10.3390/ijgi8090376
19. **Di L., Zhao P., Yang W., Yue P.** Ontology-Driven Automatic Geospatial-Processing Modeling Based on Web-Service Chaining. In: Proceedings of the Sixth Annual NASA Earth Science Technology Conference. College Park, MD, USA, 2006, p. 27–29.
20. **Zhao P., Di L., Yu G., Yue P., Wei Y., Yang W.** Semantic Web-based geospatial knowledge transformation. *Computers & Geosciences*, 2009, no. 35, p. 798–808.
21. **Scheider S., Ballatore A.** Semantic typing of linked geoprocessing workflows. *International Journal of Digital Earth*, 2017, vol. 11, p. 113–138.
22. **Jiang J., Zhu A.X., Qin C.Z., Zhu T., Liu J., Du F., Liu J., Zhang G., An Y.** CyberSoLIM: A cyber platform for digital soil mapping. *Geoderma*, 2016, no. 263, p. 234–243.
23. **Lutz M., Lucchi R., Friis-Christensen A., Ostländer N.** A Rule-Based Description Framework for the Composition of Geographic Information Services. In: Proceedings of the International Conference on GeoSpatial Semantics. Mexico City, Mexico, 2007, p. 114–127.
24. **Lutz M.** Ontology-based descriptions for semantic discovery and composition of geoprocessing services. *GeoInformatica*, 2007, vol. 11, p. 1–36.
25. **Lutz M., Lucchi R., Friis-Christensen A., Ostländer N.** A Rule-Based Description Framework for the Composition of Geographic Information Services. In: Proceedings of the International Conference on GeoSpatial Semantics. Mexico City, Mexico, 2007, p. 114–127.
26. **Yue P., Di L., Yang W., Yu G., Zhao P., Gong J.** Semantic Web Services-based process planning for earth science applications. *International Journal of Geographical Information Science*, 2009, vol. 23, p. 1139–1163.
27. **Farnaghi M., Mansourian A.** Automatic composition of WSMO based geospatial semantic web services using artificial intelligence planning. *Journal of Spatial Science*, 2013, vol. 58, p. 235–250.
28. **Martin D., Burstein M., Hobbs J., Lassila O., McDermott D., McIlraith S., Narayanan S., Paolucci M., Parsia B., Payne T.** OWL-S: Semantic markup for web services. W3C Member Submission, 2004.
29. **Roman D., Keller U., Lausen H., Bruijn J.D., Stollberg M., Polleres A., Feier C., Bussler C., Fensel D.** Web Service Modeling Ontology. *Applied ontology*, 2005, no. 1, p. 77–106.
30. **Li H., Zhu Q., Yang X., Xu L.** Geo-information processing service composition for concurrent tasks: A QoS-aware game theory approach. *Computers & Geosciences*, 2012, vol. 47, p. 46–59.
31. **Yue P., Tan Z., Zhang M.** GeoQoS: Delivering Quality of Services on the Geoprocessing Web. In: Proceedings of the OSGeo's European Conference on Free and Open Source Software for Geospatial (FOSS4G-Europe 2014). Bremen, Germany, 2014.
32. **Фёдоров Р. К., Бычков И. В., Шумилов А. С., Ружников Г. М.** Система планирования и выполнения композиций веб-сервисов в гетерогенной динамической среде // Вычислительные технологии. 2016. Т. 21, № 6. С. 18–35.
33. **Городничев М. А., Комиссаров А. В., Можина А. В., Прочкин П. В., Рудыч П. Д., Юрченко А. В.** Модели и проектные решения системы хранения и обработки исследо-

вательских данных Ecclesia // Вестник НГУ. Серия: Информационные технологии. 2018. Т. 16, № 3. С. 87–104.

34. **Linoff G. S., Berry M. J. A.** Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management. 3rd ed. John Wiley & Sons, 2004. 643 p.

References

1. **Grimm S., Abecker A., Volker J., Studer R.** Ontologies and the Semantic Web. *Handbook of semantic web technologies: foundations and technologies*, 2011, vol. 1, p. 507–579.
2. **Schut P.** OpenGIS ® Web Processing Service. *Open Geospatial Consortium*, 2007, no. 6, p. 1–3.
3. **Pautasso C.** RESTful Web service composition with BPEL for REST. *Data knowledge*, 2009, vol. 68, no. 9, p. 851–866.
4. **Hoffmann J., Weber I.** Web Service Composition. In: *Encyclopedia of Social Network Analysis and Mining*. Springer-Verlag, 2014.
5. **Deelman E., Vahi K., Juve G.** Pegasus, a workflow management system for science automation. *Future Generation Computer Systems*, 2015, vol. 46, p. 17–35.
6. **Ludäscher B., Altintas C., Berkley C., Higgins D., Jaeger E., Matthew J., Edward A. L., Tao J., Zhao Y.** Scientific Workflow Management and the Kepler System. *Special Issue: Workflow in Grid Systems. Concurrency and Computation: Practice & Experience*, 2006, vol. 18 (10), p. 1039–1065.
7. **Wilde M., Hategan M., Wozniak J. M.** Swift: A language for distributed parallel scripting. *Parallel Computing*, 2011, vol. 37 (9), p. 633–652.
8. **Berthold M. R., Cebren N., Dill F.** The konstanz information miner. *SIGKDD Explorations*, 2009, no. 11, p. 26–31.
9. **Wolstencroft K., Haines R., Fellows D.** The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud. *Nucleic Acids Research*, 2013, vol. 41 (W1), p. 557–561.
10. **Blankenberg D., Kuster G. V., Coraor N.** Galaxy: A Web-Based Genome Analysis Tool for Experimentalists. Wiley, 2010.
11. **Simmhan Y., Barga R., Ingen C.** Building the trident scientific workflow workbench for data management in the cloud. In: *Advanced Engineering Computing and Applications in Sciences (ADVCOMP)*, 2009. DOI 10.1109/ADVCOMP.2009.14
12. **Churches D., Gombas G., Harrison A.** Programming scientific and distributed workflow with Triana services: Research articles. *Concurrency and Computation: Practice and Experience*, 2006, vol. 18 (10), p. 1021–1037.
13. **Smirnov S., Sukhoroslov O., Volkov S.** Integration and Combined Use of Distributed Computing Resources with Everest. *Procedia Computer Science*, 2016, vol. 101, p. 359–368.
14. **Boukhanovsky A. V., Vasilev V. N., Vinogradov V. N., Smirnov D. Y., Sukhorukov S. A., Yapparov T. G.** CLAVIRE: Perspective Technology for Second Generation Cloud Computing. *Priboroostroenie*, 2011, vol. 54, no. 10, p. 7–14.
15. **Chen N. C., Di L. P., Yu G. N., Gong J. Y.** Geo-processing workflow driven wildfire hot pixel detection under sensor web environment. *Computers & geosciences*, 2010, vol. 36, no. 3, p. 362–372.
16. **Kwok Y.-K., Ahmad I.** Static scheduling algorithms for allocating directed task graphs to multiprocessors. *ACM Computing Surveys*, 1999, vol. 31, no. 4, p. 406–471.
17. **Xie G., Li R., Xiao X., Chen Y.** A High-Performance DAG Task Scheduling Algorithm for Heterogeneous Networked Embedded Systems. In: *Proc. of IEEE 28th International Conference Advanced Information Networking and Applications*, 2014, p. 1011–1016.
18. **Zhi-Wei H., Cheng-Zhi Q., A-Xing Z., Peng L., Yi-Jie W., Yun-Qiang Z.** From Manual to Intelligent: A Review of Input Data Preparation Methods for Geographic Modeling. *ISPRS In-*

- ternational journal of geo-information*, 2019, vol. 8, no. 9, article 376. DOI 10.3390/ijgi8090376
19. **Di L., Zhao P., Yang W., Yue P.** Ontology-Driven Automatic Geospatial-Processing Modeling Based on Web-Service Chaining. In: Proceedings of the Sixth Annual NASA Earth Science Technology Conference. College Park, MD, USA, 2006, p. 27–29.
 20. **Zhao P., Di L., Yu G., Yue P., Wei Y., Yang W.** Semantic Web-based geospatial knowledge transformation. *Computers & Geosciences*, 2009, no. 35, p. 798–808.
 21. **Scheider S., Ballatore A.** Semantic typing of linked geoprocessing workflows. *International Journal of Digital Earth*, 2017, vol. 11, p. 113–138.
 22. **Jiang J., Zhu A.X., Qin C.Z., Zhu T., Liu J., Du F., Liu J., Zhang G., An Y.** CyberSoLIM: A cyber platform for digital soil mapping. *Geoderma*, 2016, no. 263, p. 234–243.
 23. **Lutz M., Lucchi R., Friis-Christensen A., Ostländer N.** A Rule-Based Description Framework for the Composition of Geographic Information Services. In: Proceedings of the International Conference on GeoSpatial Semantics. Mexico City, Mexico, 2007, p. 114–127.
 24. **Lutz M.** Ontology-based descriptions for semantic discovery and composition of geoprocessing services. *GeoInformatica*, 2007, vol. 11, p. 1–36.
 25. **Lutz M., Lucchi R., Friis-Christensen A., Ostländer N.** A Rule-Based Description Framework for the Composition of Geographic Information Services. In: Proceedings of the International Conference on GeoSpatial Semantics. Mexico City, Mexico, 2007, p. 114–127.
 26. **Yue P., Di L., Yang W., Yu G., Zhao P., Gong J.** Semantic Web Services-based process planning for earth science applications. *International Journal of Geographical Information Science*, 2009, vol. 23, p. 1139–1163.
 27. **Farnaghi M., Mansourian A.** Automatic composition of WSMO based geospatial semantic web services using artificial intelligence planning. *Journal of Spatial Science*, 2013, vol. 58, p. 235–250.
 28. **Martin D., Burstein M., Hobbs J., Lassila O., McDermott D., McIlraith S., Narayanan S., Paolucci M., Parsia B., Payne T.** OWL-S: Semantic markup for web services. W3C Member Submission, 2004.
 29. **Roman D., Keller U., Lausen H., Bruijn J.D., Stollberg M., Polleres A., Feier C., Bussler C., Fensel D.** Web Service Modeling Ontology. *Applied ontology*, 2005, no. 1, p. 77–106.
 30. **Li H., Zhu Q., Yang X., Xu L.** Geo-information processing service composition for concurrent tasks: A QoS-aware game theory approach. *Computers & Geosciences*, 2012, vol. 47, p. 46–59.
 31. **Yue P., Tan Z., Zhang M.** GeoQoS: Delivering Quality of Services on the Geoprocessing Web. In: Proceedings of the OSGeo's European Conference on Free and Open Source Software for Geospatial (FOSS4G-Europe 2014). Bremen, Germany, 2014.
 32. **Fedorov R. K., Bychkov I. V., Shumilov A. S., Ruzhnikov G. M.** System for planning and executing web service compositions in a heterogeneous dynamic environment. *Computational technologies*, 2016, vol. 21, no. 6, p. 18–35.
 33. **Gorodnichev M. A., Komissarov A. V., Mozhina A. V., Prochkin P. V., Rudych P. D., Yurchenko A. V.** Information Models and Project Solutions for the Ecclesia Research Data Storing and Processing System. *Vestnik NSU. Series: Information Technologies*, 2018, vol. 16, no. 3, p. 87–104. (in Russ.)
 34. **Linoff G. S., Berry M. J. A.** Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management. 3rd ed. John Wiley & Sons, 2004. 643 p.

Материал поступил в редколлегию

Received

23.04.2021

Сведения об авторах

Фёдоров Роман Константинович, кандидат технических наук, ведущий научный сотрудник, Институт динамики систем и теории управления имени В. М. Матросова СО РАН (Иркутск, Россия)
fedorov@icc.ru

Бычков Игорь Вячеславович, доктор технических наук, академик РАН, директор, Институт динамики систем и теории управления имени В. М. Матросова СО РАН (Иркутск, Россия)
bychkov@icc.ru

Ружников Геннадий Михайлович, доктор технических наук, зав. отделением, Институт динамики систем и теории управления имени В. М. Матросова СО РАН (Иркутск, Россия)
rugnikov@icc.ru

Information about the Authors

Roman K. Fedorov, Senior Researcher, V. M. Matrosov Institute of System Dynamics and Control Theory SB RAS (Irkutsk, Russian Federation)
fedorov@icc.ru

Igor V. Bychkov, Doctor of Technical Sciences, Academician of the Russian Academy of Sciences, Director, V. M. Matrosov Institute of System Dynamics and Control Theory SB RAS (Irkutsk, Russian Federation)
bychkov@icc.ru

Gennady M. Ruzhnikov, Doctor of Technical Sciences, Head of the Department, V. M. Matrosov Institute of System Dynamics and Control Theory SB RAS (Irkutsk, Russian Federation)
rugnikov@icc.ru