

Метод композиции сервисов обработки пространственных данных

Роман Константинович Фёдоров ¹

¹ *Институт динамики систем и теории управления им. В.М. Матросова СО РАН, ул. Лермонтова, д. 134, 664033, Иркутск, Россия*

Аннотация

Разработаны оригинальные эвристические алгоритмы и метод создания композиций сервисов, который в отличие от существующих методов проводит комплексный анализ метаданных, онтологий, экспертных знаний и статистики применения сервисов, что позволяет находить композиции сервисов на основе комбинации данных. Использование метода может значительно упростить работу пользователя и автоматизирует часто повторяющиеся его действия. В результате применения метода создания композиций опыт применения пользователем сервисов автоматически распространяется среди всех пользователей.

Ключевые слова

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

1. Введение

Для современного этапа решения крупномасштабных фундаментальных и прикладных задач актуально повышение эффективности и надежности процессов обработки и передачи данных в вычислительных машинах, комплексах и компьютерных сетях путём внедрения сервис-ориентированной архитектуры (COA, англ. service-oriented architecture, SOA), на основе которой реализуют интеллектуальные технологии и машинное обучение, системы обработки больших объемов пространственных данных (ПД). Активно растет количество сервисов, реализующих предоставление, обработку и публикацию данных. Например, это сервисы предоставления данных дистанционного зондирования земли, обработки пространственных данных, расшифровки генома и т. д. Созданные сервисы значительно упрощают решение многих задач. В области обработки ПД определены и активно используются стандарты Open Geospatial Consortium (OGC). При этом возникает ряд сложных научно-технических задач нахождения сервисов, построения их связей между собой, проверки корректности совместного выполнения этих сервисов и др.

Объединение этих сервисов, т. е. создание композиции сервисов, позволяет решать большое количество задач. COA значительно упрощает и ускоряет интеграцию программного обеспечения, созданного разработчиками из разных предметных областей, за счет упрощения и стандартизации интерфейсов. Композиции сервисов обеспечивают повышение уровня автоматизации решения задач, начиная от ввода данных и заканчивая публикацией результатов.

В то же время создание композиций сервисов является нетривиальной задачей. Большое количество сервисов, с одной стороны, увеличивает возможности исследователей, а с другой стороны, значительно усложняет поиск нужных сервисов для решения конкретной задачи. Составление найденных сервисов в композиции в некоторых случаях может оказаться комбинаторно сложной задачей, при этом возможна генерация достаточно большого числа

6th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2024), July 01–05, 2024, Irkutsk, Russia

EMAIL: fedorov@icc.ru (A. 1)

ORCID: 0000-0002-2944-7522 (A. 1)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2024 Workshop Proceedings

DOI: 10.47350/ICCS-DE.2024.22

альтернативных композиций сервисов. Все эти альтернативы необходимо будет оценить, выбрать наиболее релевантные для решаемой задачи, а также их проверить на возможность взаимодействия. Часто композиция из двух потенциально возможных сервисов не реализуема из-за того, что структуры данных и форматы входных и выходных данных отличаются.

Композиции сервисов позволяют объединить результаты работы исследователей, но создание таких композиций в силу сложного процесса требует высокой квалификации. Поэтому актуальной научной проблемой является разработка нового метода автоматизации построения и применения композиций сервисов.

2. Обзор методов композиции сервисов

Одна важнейших особенностей создания композиции сервисов – это спецификация того, как результаты конкретного сервиса могут быть приняты входами других сервисов, т. е. сопоставление данных. Композиция сервисов обычно выполняется как обратный вывод, т. е. от целевого состояния к исходному состоянию. Существует несколько подходов к автоматизации создания композиций сервисов [1].

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 [2, 3, 4].

2. Автоматическое построение композиций сервисов. В этом подходе производится (полу) автоматическое обнаружение и составление необходимых сервисов на основе семантического анализа и логического вывода [5, 6, 7].

3. Автоматическая композиция Web-сервисов с учетом планирования выполнения. В этом подходе [8, 9] входные данные представляются сервисами, которые семантически аннотированы с использованием онтологий, например Web Ontology Language for Services (OWL-S) [10], сервисы обработки аннотированы с помощью Web Service Modeling Ontology (WSMO) [11]. Выбор среди сгенерированных цепочек сервисов может проводиться на основе оценки качества обслуживания (QoS) с использованием алгоритмов, таких как генетические алгоритмы или алгоритмы теории игр [12, 13].

В последние годы многие исследования подчеркивали важность семантической информации о сервисе при композиции сервисов геообработки, утверждая, что при составлении сервисов необходимо обнаружение сервисов с соответствующими функциями [14, 15, 16]. Входы и выходы между соседними сервисами должны совпадать не только с точки зрения параметров, но также с точки зрения их синтаксиса и семантики. Онтология стала распространенным инструментом для описания семантической информации сервисов геообработки, а онтологические подходы в настоящее время являются наиболее распространенными методами автоматической компоновки сервисов геообработки.

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 метаинформации, 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. Достаточно сложно выполнить описание некоторых сервисов. В статье [6] приводится пример, в котором сервис, выполняющего деление, может возвращать плотность, если его входные данные представляют собой массу и объем, или скорость, если задано расстояние и период времени;
- требуется согласованность метаинформации сервисов и онтологии предметной области.

Поиск и создание новых композиций сервисов является сложной задачей, в которой необходимо проанализировать большое количество сервисов и предоставить наиболее релевантные их комбинации. Существующие методы и подходы на основе анализа метаданных и онтологий не позволяют значительно упростить поиск из-за частого отсутствия метаданных и трудоемкости разработки онтологий. Создание композиций сервисов ограничено сложностью

взаимодействия сервисов по данным из-за различий в программном интерфейсе доступа к данным, в структуре данных, используемых справочников и т. д.

2.1. Предлагаемый метод композиции сервисов

Для создания композиций сервисов требуется метод, который учитывает перечисленные требования и позволяет связывать различные сервисы между собой для достижения целей пользователей. При этом метод должен учитывать непроработанность метаданных сервисов, недостаточную онтологическую формализацию данных и процессов и давать возможность разрабатывать новые сервисы или средства обработки данных для создания композиций сервисов.

Работа метода основывается на новой вычислительной модели композиции сервисов, определяемой структурой

$CM = \langle Z, S, A, T, CR, F \rangle$ со следующими обозначениями.

Z – множество параметров. В стандарте WPS OGC обработки пространственных данных параметры связаны с определенными типами данных.

S – множество сервисов. В работе рассматриваются только сервисы без хранения состояния (stateless), т. е. результат выполнения сервисов не должен зависеть от предыдущих вызовов. Этого достаточно для решения широкого круга задач. Сервис $s \in S$ – это упорядоченная тройка $s = \langle name, Z^{Inp}, Z^{Out} \rangle$, где $name$ – это имя сервиса, $Z^{Inp} \in Z$ – множество входных параметров, $Z^{Out} \in Z$ – множество выходных параметров сервиса. Среди выходных параметров сервиса имеется параметр успешности выполнения сервиса. Далее будем обозначать $s.Z^{Inp}$ и $s.Z^{Out}$ параметры, принадлежащие определенному сервису. Сервисы разделяются на сервисы получения данных $S^d \subset S$, вычислительные сервисы $S^c \subset S$ и сервисы публикации данных $S^p \subset S$.

A – множество пользователей, которые взаимодействуют с СОИАС через применение сервисов S . Набор применяемых ими сервисов может варьироваться.

T – множество заданий выполнения сервисов из S . Каждое задание $t \in T$ определяется упорядоченной четверкой $t = \langle a, s, V_i, V_o \rangle$, т. е. вызов сервиса $s \in S$ пользователем $a \in A$ с множеством значений V_i входных параметров $s.I$ и результаты V_o выходных параметров $s.O$.

CR – множество вычислительных узлов, на которых выполняются сервисы. Каждый сервис может быть активирован на заранее определенных узлах. Автоматическое выделение новых узлов для сервисов в работе не рассматривается.

F – множество функций ранжирования сервисов.

Отметим, что композиция сервисов является также сервисом с заданным именем, входными и выходными параметрами. Общеизвестным стандартом определения композиций является использование направленного ациклического графа (Directed Acyclic Graph, (DAG)) $s_i = DAG_i = \langle T_i, E_i \rangle$, $s_i \in S$, в котором вершинами являются задания $t \in T_i$, а ребра $e = \langle v_k, v_l \rangle$, $e \in E_i$ показывают передачу данных от выходного параметра одного задания к входному параметру другого задания, где $v_k \in V_k^i$, $v_l \in V_l^j$, $t_i = \langle a_i, s_i, V_i^i, V_o^i \rangle$, $t_j = \langle a_j, s_j, V_j^j, V_o^j \rangle$. Поскольку возможна передача данных одновременно по нескольким параметрам, то ребер между двумя вершинами (заданиями) также может быть несколько. В литературе чаще всего передача данных между заданиями обычно обозначается одним ребром. Для создания готовой к выполнению композиции необходимо определить все передаваемые данные между параметрами.

Определим типы данных, используемые для обработки пространственных данных. Частично примем обозначения, принятые в реляционной алгебре:

$D = \{D_1, \dots, D_l\}$ – множество скалярных типов, определенных в стандарте WPS;

$R = \{R_1, \dots, R_m\}$ – множество схем отношений;

$d = \{d_1, \dots, d_k\}$ – множество таблиц (отношений). В качестве параметров могут использоваться реляционные таблицы $d_i \in Z$.

Введем множество типов параметров $TZ = D \cup R$. Для каждого параметра определен тип $tz_j = type(z_j)$, $tz_j \in TZ$.

Для формирования композиций сервисов требуется информация о возможности передачи данных выходного параметра сервиса для использования во входном параметре другого сервиса. Однако соответствие типов параметров не является достаточным, например, тип «число» может иметь смысл – высота над уровнем моря, а в другом случае – температура. Поэтому соответствие типов является необходимым условием, но не достаточным. Получение информации о возможности передачи данных между параметрами может производиться на основе анализа метаданных, онтологий, экспертных знаний и статистики применения сервисов и т. д. Поэтому вводим понятие метки, которое позволяет абстрагироваться от способа получения информации о возможной передаче данных между параметрами. Метка используется для идентификации возможности передачи данных, т. е. если выходной параметр и входной параметр имеет одну и ту же метку, то между ними возможна передача данных. Каждому параметру может соответствовать ноль и более меток. Метками могут являться онтологические концепты, заданные с помощью URI, или некоторые сгенерированные уникальные значения.

$m \in M$ – множество меток. Для каждого параметра z_j определено подмножество меток M_j . Проверка возможности передачи данных между сервисами в рамках модели выполняется на основе соответствия типов параметров и меток. Для двух сервисов s^i и s^j возможна передача данных от параметра $z^i \in s^i.Z^{Out}$ к параметру $z^j \in s^j.Z^{Inp}$, если типы параметров совпадают $type(z_i) = type(z_j)$ и существует метка $m \in M$, которая имеется у обоих параметров ($m \in M_i$ и $m \in M_j$).

Выполнение этого условия не гарантирует корректное выполнение сервиса s^j . Оно позволяет значительно сузить количество возможных композиций сервисов.

2.1.1. Основные этапы метода

Процесс создания композиции сервисов представлен на рисунке 2.1 и состоит из трех основных этапов.

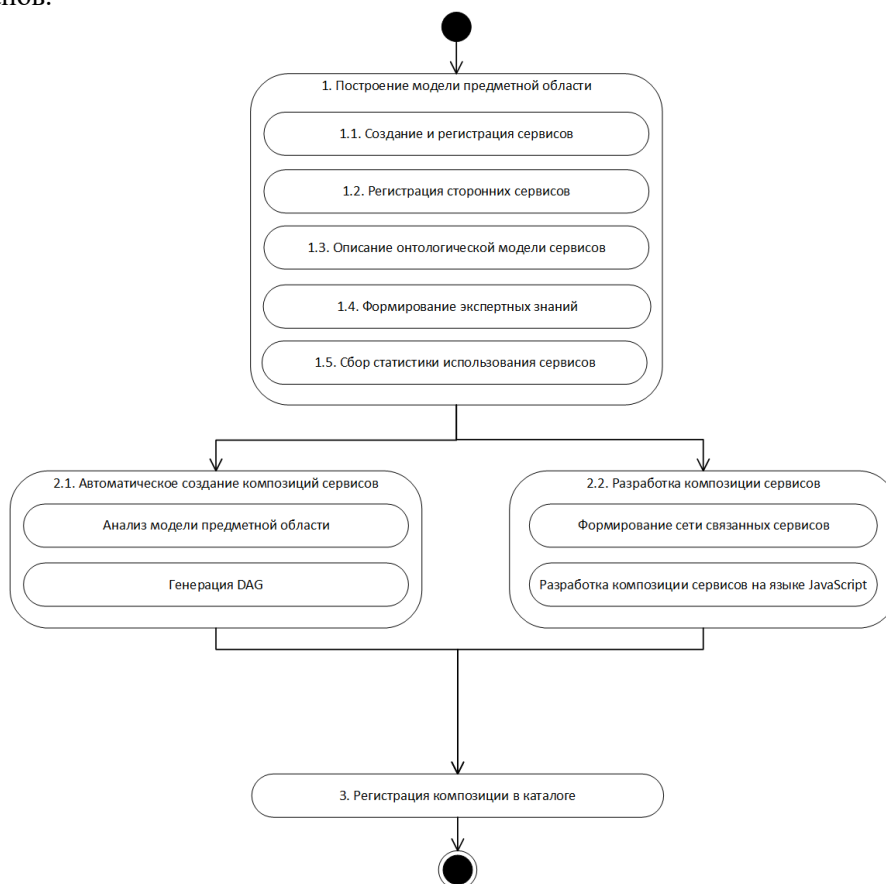


Рисунок 1: Схема метода формирования композиций сервисов

Этап 1. Построение модели предметной области

Состоит из 5 шагов, выполняемых независимо друг от друга.

1.1. Создание и регистрация сервисов – предполагается разработка вычислительных сервисов, сервисов данных и сервисов публикации данных. Вычислительные сервисы разрабатываются в соответствии со стандартом Web Processing Service (WPS, разработан Open Geospatial Consortium) на вычислительных ресурсах среды или на любых других серверах, к которым имеется сетевой доступ. Для создания сервисов данных используется компонент «Фабрика сервисов ввода и редактирования реляционных данных», который поддерживает все основные этапы работы с табличными данными. «Фабрика сервисов отображения графиков» и «Фабрика сервисов отображения пространственных данных» предназначены для создания сервисов публикации данных. Все создаваемые сервисы регистрируются в каталоге, обладают программным интерфейсом и готовы стать частью композиции сервисов.

1.2. Регистрация сторонних сервисов – осуществляется подключение WPS сервисов, созданных на сторонних вычислительных ресурсах. В каталоге сервисов указывается необходимая для запуска сервиса информация и метаданные.

1.3. Описание онтологической модели сервисов – предназначено для определения возможности передачи данных между параметрами сервисов на основе онтологий. В рамках описания онтологической модели сервисов предполагается определение концептов для параметров сервисов. Каждому параметру может соответствовать ноль и более меток. Метками могут являться онтологические концепты, заданные с помощью URI. Использование одного и того же концепта для выходного параметра сервиса и для входного параметра другого сервиса предполагает возможность передачи данных между ними и присвоение одной метки для них.

1.4. Формирование экспертных знаний – используется для определения возможности передачи данных пользователем. Разработка онтологий – достаточно сложный процесс. Часто с помощью онтологий не все сущности и отношения предметной области формализованы. Поэтому в качестве меток параметров сервисов могут использоваться некоторые сгенерированные уникальные значения, назначаемые пользователем для конкретных сервисов. Например, сервис построения карты плотности может применяться для всех точечных данных. Соответственно, выходным параметрам, которые предоставляют векторный точечный слой, можно назначить ту же самую метку, что и у входного параметра сервиса построения карты плотности.

1.5. Сбор статистики выполнения сервисов – производится сбор информации 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 виде таблицы 1, где значения параметров хранятся в формате JSON.

Таблица 1

Данные о вызовах сервисов

Id задания	Id 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

Далее данные этой таблицы будем обозначать как Log – множество заданий (успешных вызовов сервисов), совершенных пользователями. Причиной неудачного вызова сервиса могут быть некорректные значения параметров, поэтому данные фильтруются и оставляются только успешные вызовы.

Этап 2. Создание композиций сервисов

После построения модели предметной области на этапе 2 применяются два альтернативных способа создания композиций:

- автоматическое формирование композиций сервисов. Данный способ позволяет создать композицию сервисов на основе анализа графа связности заданий;
- разработка композиций сервисов. Пользователь разрабатывает композиции сервисов с помощью языка JavaScript. Этот способ требуется при необходимости обработки промежуточных данных с помощью средств языка, или при заранее неизвестном количестве заданий в композиции $DA G_i$, которые формируются в зависимости от данных.

Этап 3. Регистрация композиции сервисов в каталоге

Последним этапом является сохранение полученных композиций сервисов в каталоге для последующего применения пользователями.

2.1.2. Автоматическое формирование композиций сервисов

Создание композиций сервисов должно проводиться на основе анализа возможных связей между сервисами. Применение меток обобщает использование информации о применении сервисов, экспертных знаний и онтологий. Каждая композиция задается графом $DA G$, с заданными параметрами заданий и связями по данным. Создаваемые композиции сохраняются в каталоге сервисов, где пользователь может найти нужную композицию в соответствии со своим запросом. Автоматическое создание композиций состоит из следующих шагов:

Шаг 1. Присвоение меток параметрам сервисов.

Шаг 2. Установление связей между сервисами и создание графа сервисов (граф $DA G_{gen}$).

Шаг 3. Поиск компонент связности графа $DA G_{gen}$.

Шаг 4. Декомпозиция графа сервисов.

2.1.3. Присвоение меток параметрам сервисов

Присвоение меток параметрам сервисов производится следующими способами:

1. Задаются вручную (экспертные знания).
2. Анализ статистики выполнения сервисов.
3. Анализ метаданных и онтологических понятий.

Рассмотрим подробнее все способы.

Задание вручную меток сервисам производится в основном для общеиспользуемых сервисов, например для всех сервисов редактирования реляционных данных ставится общая метка, которая связывает с сервисами экспорта в формат XLS, CSV. Для сервисов редактирования реляционных данных с пространственными атрибутами задается общая метка с сервисами экспорта в форматы shape, geojson и т.д.

Анализ статистики выполнения сервисов производится для того, чтобы определить факт передачи данных между параметрами сервисов. Определение связи между двумя вызовами сервисов может производиться на основе анализа значений параметров. Множество параметров можно разделить на входные и выходные. Предполагается, если значение выходного параметра совпадает со значением входного параметра, то между вызовами сервисов может быть передача данных. Значения параметров могут быть строковыми или числовыми, а также идентификаторами ресурсов URI, обычно это URL ссылки на файлы, сервисы предоставления данных и т. д. Каждый URI, используемый для получения данных WPS сервисом, однозначно идентифицирует передаваемые данные. Соответственно, по URI параметрам можно однозначно определить, что данные передаются от одного вызова сервиса другому. Несколько сложнее со строковыми и числовыми параметрами, совпадение значений которых может быть случайным. На текущий момент строковые и числовые параметры не рассматриваются. Если факт передачи данных между параметрами сервисов определен, то генерируется уникальная метка, которая добавляется указанным параметрам.

Анализ метаданных и онтологических понятий проводится пока только для сервисов, предоставляющих реляционные данные. Между данными часто имеется отношение общее – частное. В терминах объектно-ориентированного программирования это отношение называется наследованием (суперкласс – подкласс). Например, собираемые данные о конференциях, семинарах, лекциях обобщаются данными, описывающие классом мероприятие (название,

место, дата, организатор и т.д.). Применение какой-либо обработки к данным суперкласса за редким исключением, означает возможность ее применения к данным подклассов. Например, если обработка производит подсчет и строит график распределения мероприятий по годам, то ее можно применить к данным, описывающим конференции при условии соблюдения схемы данных суперкласса. Создан рекурсивный алгоритм, который все метки общего класса распространяет всем дочерним классам.

Отношение между данными суперкласс – подкласс определяется разными способами:

1. Получаются в результате создания сервисов данных на основе других.
2. Проводится анализ онтологических понятий метаданных сервисов.

Для описания понятий используется язык OWL (Web Ontology Language) [17], разработанный консорциумом W3C. Язык OWL позволяет описывать классы и отношения между ними. В основе языка – представление действительности в модели данных «объект – свойство». Язык OWL позволяет определить иерархию классов, включая суперклассы, под которыми объединяются подклассы, что позволяет применять правило ко всем классам. Определение иерархии классов задается с помощью элементов `subClassOf`. При анализе метаданных при наличии онтологических понятий производится запрос на наличие отношения `subClassOf` между двумя классами, если оно имеется, то применяется алгоритм распространения меток.

2.1.4. Установление связей между сервисами и создание графа сервисов

В графе DAG_{gen} вершинами являются сервисы, зарегистрированные в каталоге. Дуги графа являются направленными и определяют связь (возможность передачи данных) между параметрами сервисов. Для поиска дуг производится поиск одинаковых меток у параметров сервисов. Если такие пары параметров найдены, то добавляется дуга в граф DAG_{gen} .

2.1.5. Поиск компонент связности

Граф сервисов DAG_{gen} содержит возможные применения для решения множества разных задач. Следовательно, его необходимо декомпозировать и выделить отдельные подграфы, соответствующие решаемым задачам, т.е. композиция сервисов образуют связанный подграф D_i (компонента связности графа) в DAG_{gen} . Для получения подграфа, соответствующего решению задачи, недостаточно найти компоненты связности в DAG_{gen} , так как в нем могут быть сервисы получения данных. Например, могут быть сервисы получения данных ДЗЗ или сервисами предоставления векторных и реляционных данных. Эти сервисы могут использоваться для решения большого количества различных задач. Например, данные ДЗЗ Landsat 8, которые можно получить с помощью сервисов, используются для задач оценки подтоплений из-за разлива 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 сервисы получения данных ДЗЗ, т. е. объединяться в компоненты связности. Поэтому при выделении подграфов, соответствующих только одной задаче, необходимо не рассматривать связи через сервисы получения данных.

Для поиска связанных подграфов D_i применяется алгоритм поиска в ширину. При этом сложность поиска является линейной от суммы числа вершин и числа рёбер графа DAG_{gen} . Это обосновывается направленностью графа и отсутствием циклов.

2.1.6. Декомпозиция графа сервисов

Алгоритм поиска компонент связности может получить достаточно много подграфов D_i . Часть подграфов может соответствовать экспериментам, которые привели к недостаточным хорошим результатам. Также среди них может быть достаточно много дублирований, которые возникают из-за многократного решения задачи на разных данных и с разными параметрами. Поэтому производится фильтрация множества D_i и выделение наиболее полезных композиций сервисов.

3. Апробация

Данные гербарных листов определены стандартами TDWG Darwin Core и Access to Biological Collections Data (ABCD). На основе этих стандартов создан сервис сбора и редактирования реляционных данных «BasicHerbarium». «BasicHerbarium» стал базовым классом для следующих сервисов: БД «Гербарий сосудистых растений Сибирского института физиологии и биохимии растений СО РАН» (IRK); БД «Гербарий мхов Сибирского института физиологии и биохимии растений СО РАН» (IRK) и т.д.

Пользователь для получения карты распространения видов последовательно использовал сервисы:

1. «BasicHerbarium».
2. table2shp – сохранил данные сервиса «BasicHerbarium» в формат SHAPE.
3. vector2grid52 – произвел обработку данных в формате SHAPE для получения растрового файла в формате GeoTIFF. В каждой ячейке (пикселе) записано количество точек.
4. map – создал карту распространения видов с указанием легенды (задание цветовой палитры отображений файла GeoTIFF в зависимости количества точек в ячейке).

На основе анализа статистики создана композиция сервисов densityMap (рисунок 2).

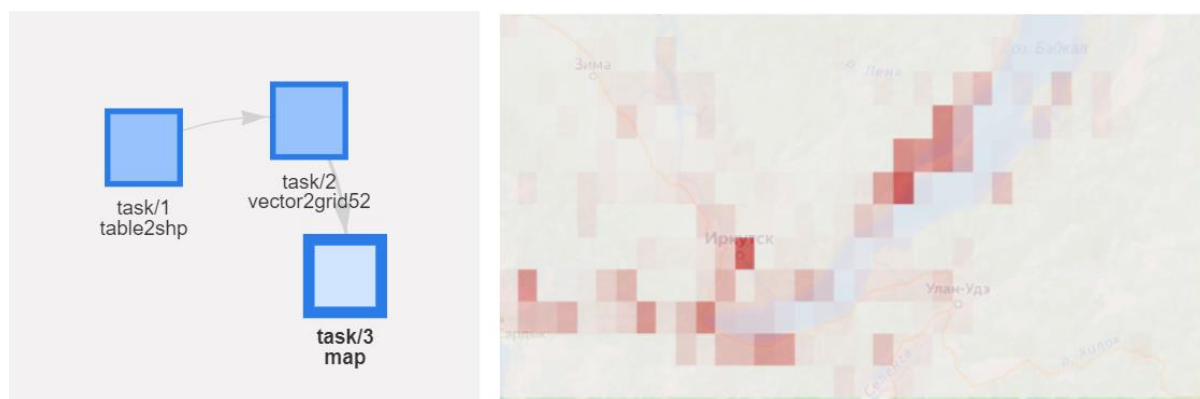


Рисунок 2: Сервис densityMap и результат его работы

Созданный сервис автоматически предлагается всем пользователям сервисов данных, созданных на основе «BasicHerbarium». При выборе этого сервиса автоматически создается новая композиция сервисов002E

4. Заключение

Разработаны оригинальные эвристические алгоритмы и метод создания композиций сервисов, базирующийся на применении предложенной модели, который в отличие от существующих методов проводит комплексный анализ метаданных, онтологий, экспертных знаний и статистики применения сервисов, что позволяет находить композиции сервисов на основе комбинации данных. Использование метода может значительно упростить работу пользователя и автоматизирует часто повторяющиеся его действия. В результате применения модели и метода создания композиций опыт применения пользователем сервисов автоматически распространяется среди всех пользователей.

Работа выполнена при поддержке ЦКП ИИВС ИРНОК, базового проекта 121030500071-2.

5. Список литературы

- [1] 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, № 9. article 376. DOI: 10.3390/ijgi8090376.
- [2] Di L., Zhao P., Yang W., Yue P. Ontology-Driven Automatic Geospatial-Processing Modeling Based on Web-Service Chaining // Proceedings of the Sixth Annual NASA Earth Science Technology Conference. College Park, MD, USA. 2006. P. 27-29.
 - [3] Zhao P., Di L., Yu G., Yue P., Wei Y., Yang W. Semantic Web-based geospatial knowledge transformation // Computers & Geosciences. 2009. Vol. 35, № 4. P. 798-808.
 - [4] Scheider S., Ballatore A. Semantic typing of linked geoprocessing workflows // International Journal of Digital Earth. 2017. Vol. 11. P. 113-138.
 - [5] 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. № 263. P. 234–243.
 - [6] Lutz M., Lucchi, R., Friis-Christensen A., Ostländer N. A Rule-Based Description Framework for the Composition of Geographic Information Services // Proceedings of the International Conference on GeoSpatial Semantics. Mexico City, Mexico, 29–30 November. 2007. P. 114-127.
 - [7] Lutz M. Ontology-based descriptions for semantic discovery and composition of geoprocessing services // GeoInformatica. 2007. Vol. 11. P. 1-36.
 - [8] 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.
 - [9] 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.
 - [10] 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.
 - [11] 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. № 1. P. 77-106.
 - [12] 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.
 - [13] Yue P., Tan Z., Zhang M. GeoQoS: Delivering Quality of Services on the Geoprocessing Web // Proceedings of the OSGeo's European Conference on Free and Open Source Software for Geospatial (FOSS4G-Europe 2014). Bremen, Germany. 15–17 July 2014.
 - [14] Cruz S.A., Monteiro A.M., Santos R. Automated geospatial web services composition based on geodata quality requirements // Computers & Geosciences. 2012. Vol. 47. P. 60-74.
 - [15] Huang W., Harrie L. Towards knowledge-based geovisualisation using Semantic Web technologies: A knowledge representation approach coupling ontologies and rules // International Journal of Digital Earth. 2020. Vol.13. P. 976-997.
 - [16] Sun Z., Yue P., Lu X., Zhai X., Hu L. A task ontology driven approach for live geoprocessing in a service-oriented environment // Transactions in GIS. 2012. Vol. 16. P. 867-884.
 - [17] Web Ontology Language (OWL) [Электронный ресурс]. – Режим доступа: <https://www.w3.org/OWL/> (дата обращения 25.02.2024).