

**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В. ЛОМОНОСОВА**

**НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ ИНСТИТУТ
ЯДЕРНОЙ ФИЗИКИ имени Д.В. СКОБЕЛЬЦЫНА**

А.П.Крюков , М.Н. Жижин, А.А. Пойда, Д.Ю. Мишин, Д.П. Медведев,
А.П.Демичев, Д.С. Коковин

**РАЗРАБОТКА RESTFUL-ВЕБ-СЕРВИСОВ УДАЛЕННОГО ДОСТУПА К
РЕСУРСАМ ХРАНЕНИЯ ДАННЫХ ДЛЯ РАСПРЕДЕЛЕННЫХ СИСТЕМ
В ОБЛАСТИ НАУК О ЗЕМЛЕ**

Препринт НИИЯФ МГУ № 2012-4/882

Москва, 2012 г.

УДК 001.89:004

**А.П.Крюков , М.Н. Жижин, А.А. Пойда, Д.Ю. Мишин, Д.П. Медведев, А.П.Демичев,
Д.С. Коковин**

e-mail: kryukov@theory.sinp.msu.ru

**РАЗРАБОТКА RESTFUL-ВЕБ-СЕРВИСОВ УДАЛЕННОГО ДОСТУПА К РЕСУРСАМ
ХРАНЕНИЯ ДАННЫХ ДЛЯ РАСПРЕДЕЛЕННЫХ СИСТЕМ В ОБЛАСТИ НАУК О
ЗЕМЛЕ**

Препринт НИИЯФ МГУ № 2012-4/882

Аннотация

В статье представлены RESTful-веб-сервисы удаленного доступа к ресурсам хранения данных для распределенных систем в области наук о Земле, обладающие следующими особенностями: поддержка работы с метаданными и большими объемами данных, включая поиск и автоматическую генерацию запросов к данным на основе метаданных, поддержка общей модели данных, совместимой с UNIDATA Common Data Model, позволяющей интегрировать разнородные ресурсы в области наук о Земле, поддержка обобщенного языка запросов, позволяющего формировать универсальные запросы к нескольким разнородным источникам данных, горизонтальная масштабируемость, потоковый асинхронный запрос данных, позволяющий избежать ошибок таймаута и переполнения памяти на удаленном сервере.

Направлено в журнал Программные продукты и системы.

Работа проводилась при финансовой поддержке Министерства образования и науки Российской Федерации (Государственный контракт № 07.514.11.4022 от 20.09.2011).

**A.P.Kryukov, M.N.Zhizhin, F.A.A.Poyda, D.Yu.Mishin, D.P.Medvedev, A.P.Demichev,
D.S.Kokovin**

**DEVELOPMENT OF RESTFUL-WEB SERVICES FOR REMOTE ACCESS TO DATA
STORAGE RESOURCES FOR DISTRIBUTED SYSTEMS IN EARTH SCIENCES**

Preprint MSU SINP 2012-4/882

Abstract

RESTful-web services for remote access to data storage resources for distributed systems in the Earth sciences are presented. They possess the following features: operation support with meta data and large volumes of data including search and automatic generation of requests to data on the basis of meta data, support of the general data model compatible to UNIDATA Common Data Model and allowing to integrate heterogeneous resources in the field of Earth sciences, support of the generalized language of the requests allowing to create the universal requests to several heterogeneous data sources, horizontal scalability, stream asynchronous request of the data allowing to avoid timeout errors and memory overflow on a remote server.

© А.П.Крюков , М.Н. Жижин, А.А. Пойда, Д.Ю. Мишин, Д.П. Медведев, А.П.Демичев, Д.С. Коковин
© НИИЯФ МГУ, год, <http://www.sinp.msu.ru>

Введение

При работе с данными в области наук о Земле возникает ряд требований, вытекающий из специфики этой области наук, среди которых можно выделить:

- Большой объем используемых данных. Например, модели погоды генерируют гигабайты производных данных, причем с увеличением точности измерительных приборов, производительности процессоров и увеличения точности моделей, объем данных возрастает экспоненциально. В случае глобального анализа погоды результат тем точнее, чем больше параметров и территории охватывается, поэтому для работы таких моделей требуются большие объемы исходных данных. Это ставит вопрос о скорости выборки и записи данных. Например, скорость выборки может негативно сказаться на запросах из-за опции таймаута на ответ сервера, которая может привести к ошибке выполнения запроса. Еще одна особенность запроса ресурса данных - необходимость получения не только полного набора данных, но и его части (как непрерывной подобласти, так и прореженного по определенному закону участка).
- Сложность моделей данных. Окружающая среда – сложно-связный комплекс многих параметров. Это приводит к тому, что для описания окружающей среды требуются модели со сложными связями (например, модель данных, использовавшаяся при создании реанализа погодных данных NCEP/NCAR [1], включала как модельную сеточную структуру, так и наблюдения с движущихся объектов и со стационарных станций). Это ставит вопрос о мощном языке запросов, поддерживающем терминологию наук о Земле.
- Обособленность моделей данных для различных подобластей и требование междисциплинарного характера исследований. Как в ходе исторически сложившихся особенностей обособленного развития различных подобластей наук о Земле, так и по причине стремления ученых подобрать максимально выразительную систему связей для конкретной подобласти, сложилась ситуация сильного отличия схем и моделей данных используемых в различных направлениях наук о Земле. В то же время, как упоминалось выше, для повышения точности вычислительных моделей требуется учитывать как можно больше параметров, в том числе и междисциплинарных, что ведет к необходимости адаптировать данные различных моделей в одном вычислительном процессе. Это ставит вопрос о разработке обобщенной модели данных, способной объединить данные из различных подобластей.
- Разнообразии наборов данных. Науки о Земле - довольно обширная область с большим числом институтов и специалистов. В результате, за долгие годы работы, появилось много независимых наборов данных, с разными моделями, с разными параметрами и различным синтаксисом и семантикой. В данном случае под различием в синтаксисе и семантике понимается различие как в названиях одних и тех же переменных, так и различие единиц измерения и представления данных. Еще одна проблема большого числа источников данных: необходимость учитывать инвенторную информацию (т.е. информацию о временном и пространственном покрытии данных). Это важно как при правильном формулировании запросов (чтобы не выйти за допустимые границы параметров), так и при выборе между источниками данных. Это ставит вопрос о:
 - Создании ресурса метаданных, агрегирующего метаинформацию по множеству наборов данных, их описанию, списку параметров и их инвенторной информации, единицам измерения. Ресурс метаданных должен позволять поиск по описательной информации.
 - Обеспечении возможности самих сервисов предоставлять информацию о себе.

- Обеспечении возможности запрашивать сервисы с использованием общей терминологии с переводом общих терминов в термины конкретных моделей.

В настоящей статье представлены RESTful-веб-сервисы удаленного доступа к ресурсам хранения данных для распределенных систем в области наук о Земле, обладающие следующими особенностями: поддержка работы с метаданными и большими объемами данных, включая поиск и автоматическую генерацию запросов к данным на основе метаданных, поддержка общей модели данных, совместимой с UNIDATA Common Data Model [2], позволяющей интегрировать разнородные ресурсы в области наук о Земле, поддержка обобщенного языка запросов, позволяющего формировать универсальные запросы к нескольким разнородным источникам данных, горизонтальная масштабируемость, потоковый асинхронный запрос данных, позволяющий избежать ошибок таймаута и переполнения памяти на удаленном сервере.

Чтобы реализовать сервисы с заявленными функциональными возможностями, потребовалась работа в нескольких направлениях: разработка общей модели представления данных и обобщенного языка запросов, создание механизмов работы с метаданными и генерации автоматического запроса на основе метаданных, разработка и реализация схем доступа к данным, создание демонстрационного веб-портала для управления сервисами.

В статье собраны разделы, описывающие наработки и рекомендации по каждому из представленных направлений. Исследования выполнялись в рамках федеральной целевой программы «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007-2013 годы» по лоту шифр «2011-1.4-514-015» «Проектирование и разработка RESTful-веб-сервисов для создания специализированных проблемно-ориентированных распределенных инфраструктур с обеспечением упрощенного интегрирования ресурсов для научных исследований» по теме «Проектирование и разработка RESTful-веб-сервисов доступа к суперкомпьютерным ресурсам и ресурсам хранения данных для создания специализированных проблемно-ориентированных распределенных инфраструктур в области нанонаук и наук о Земле, обеспечивающих упрощенное интегрирование ресурсов для научных исследований» (шифр заявки «2011-1.4-514-015-081»).

1. Общая модель представления данных

При реализации сервисов был сделан упор на сверхбольшие наборы научных данных, которые представимы в виде совокупности многомерных массивов. Одной из наиболее мощных по функциональности моделей для представления многомерных массивов данных на сегодня является модель UNIDATA Common Data Model (CDM) [2].

Абстрактная модель CDM включает три уровня, построенных по иерархическому принципу (Рисунок 1) - каждый следующий уровень строится над предыдущим, реализуя более высокий уровень абстракции:

1. Синтаксический уровень (data access layer) описывает синтаксис данных и доступ к ним.
2. Уровень координатных систем определяет координаты элементов данных. Под координатами понимается любая абстрактная система координат, к которой может быть привязан тот или иной набор данных.
3. Уровень научных типов данных определяет специализированную группировку данных, например, регулярную сетку, произвольные точки, траекторию и т.п., поддерживая специализированные методы обработки этих типов.

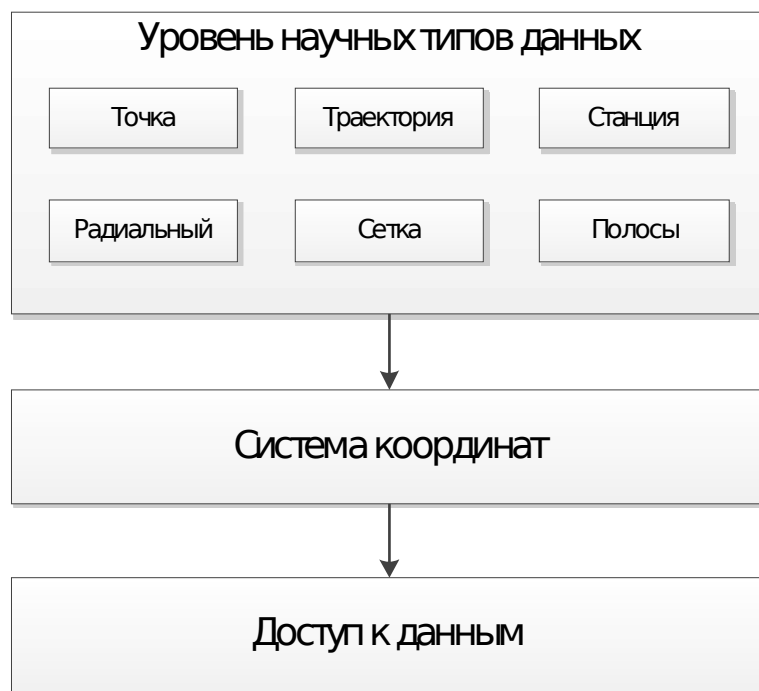


Рисунок 1. Уровни модели представления данных UNIDATA Common Data Model

Объектная модель CDM синтаксического уровня представлена на рисунке 2.

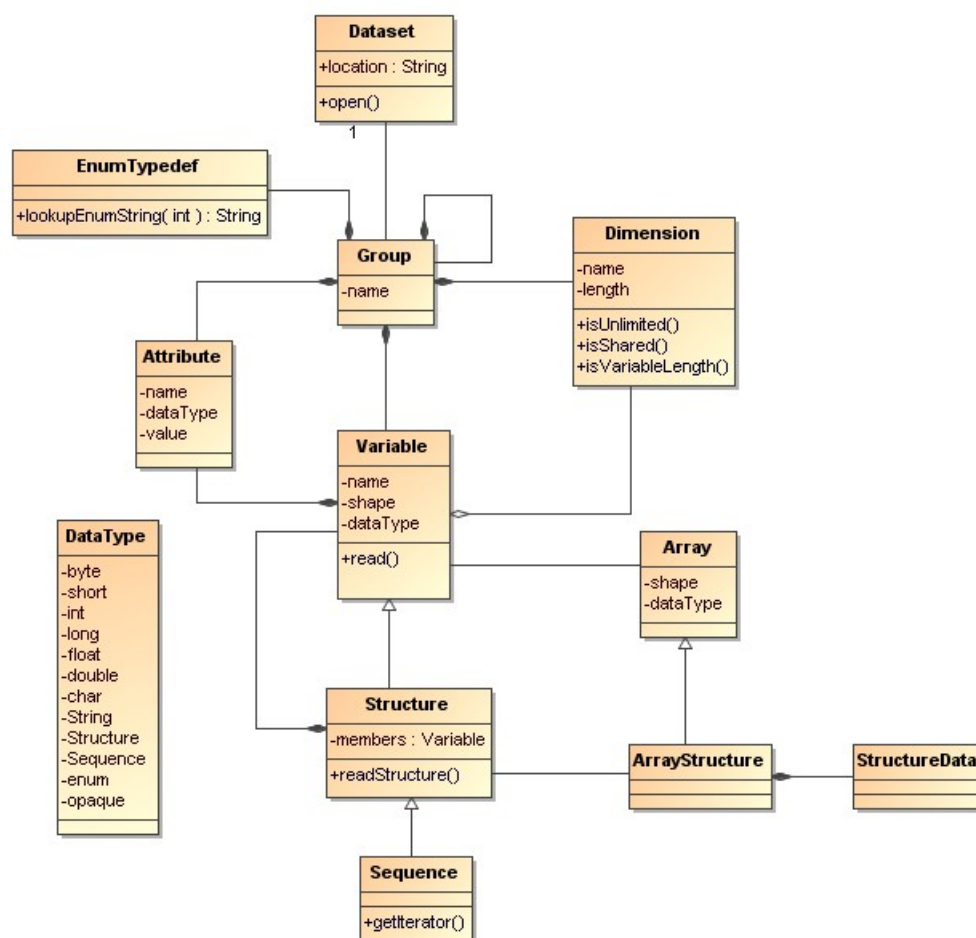


Рисунок 2. Объектная модель CDM синтаксического уровня

Dataset (набор данных) может быть файлом (в формате NetCDF, HDF5, GRIB или другом), потоковым веб-сервисом данных по протоколу OPeNDAP [3], коллекцией файлов или другой сущностью, доступ к которой может быть получен через прикладной программный NetCDF [4].

Group (группа) – совокупность атрибутов, измерений, перечислимых типов, переменных и вложенных групп.

Variable (переменная) – набор данных, имеющий тип данных и набор измерений, на которых он определен.

Dimension (измерение) – используется для определения измерений переменных. Несколько переменных могут разделять одно и то же измерение.

Attribute (атрибут) – элемент метаданных, имеющий название и значение, относится либо к группе, либо к переменной.

Structure (структура) – переменная, содержащая в себе другие переменные.

Sequence (последовательность) – одномерная структура неопределенной длины (длина не определена в метаданных, т.е. неизвестна до тех пор, пока не будут полностью считаны данные, содержащиеся в последовательности).

Array (массив) – массив, содержащий данные переменной.

ArrayStructure (структурный массив) – подкласс массива, содержащий данные структурной переменной.

Объектная модель уровня координатных систем для UNIDATA CDM представлена на рисунке 3.

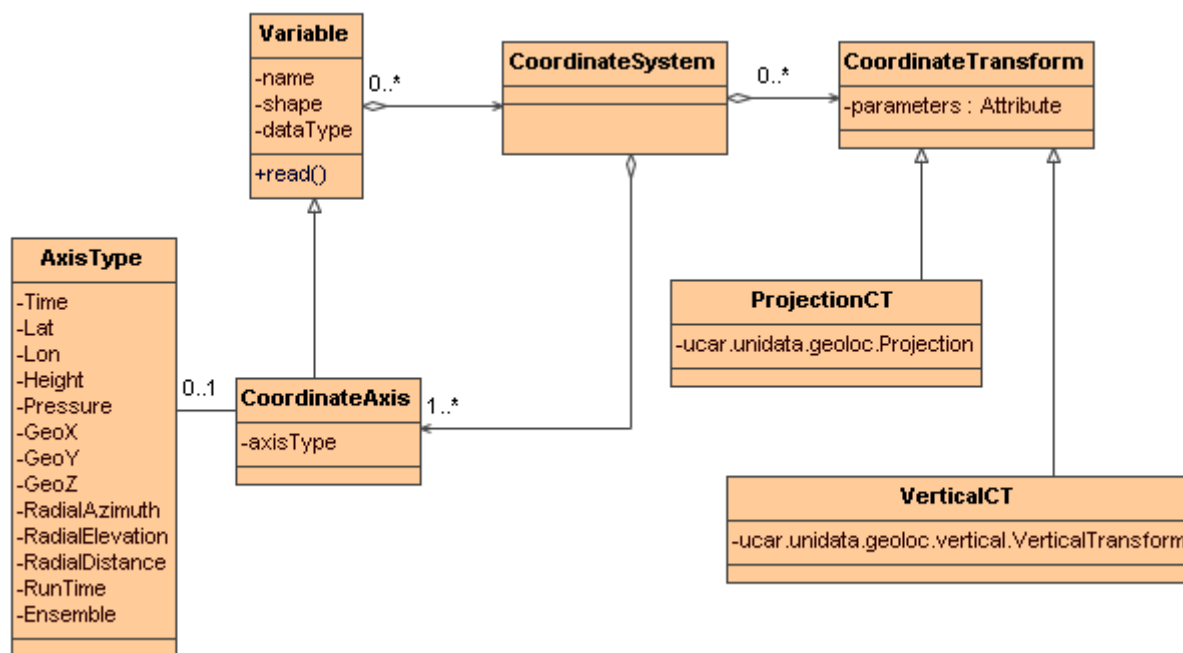


Рисунок 3. Объектная модель уровня координатных систем для UNIDATA CDM

Здесь *Variable* (переменная) может иметь ноль или более координатных систем, содержащих одну или более координатных осей.

Coordinate System (координатная система) имеет одну или более координатных осей и ноль или более трансформаций координат.

CoordinateAxis (координатная ось) – подтип переменной и опционально классифицирован в соответствии с типами в *AxisType*.

CoordinateTransform – абстрактное представление преобразования между координатными системами.

AxisType - специфические термины, используемые в той или иной координатной системе. Например, время, широта, долгота, высота, давление и т.п.

Научные типы данных позволяют разбить научные данные на категории. Уровень научных типов данных позволяет пользователю запрашивать данные в терминах координатных систем, например с использованием пространственно-временных ограничений. Ниже приведен список научных типов, поддерживаемых в настоящее время в модели данных UNIDATA CDM.

```
ANY // any type
NONE // no type
GRID
RADIAL
SWATH
IMAGE
ANY_POINT // any of the following
POINT
PROFILE
SECTION
STATION
STATION_PROFILE
TRAJECTORY
```

В случае создания RESTful сервисов было решено пойти на упрощение данной модели, отказавшись от ряда иерархических структур, организация поддержки которых в условиях больших массивов данных и сетевого доступа достаточно сложна. В результате была выбрана следующая синтаксическая модель данных (Рисунок 4), очень близкая по своей структуре к модели NetCDF-3 [5] и совместимая с моделью UNIDATA CDM. Элементы данной модели аналогичны соответствующим элементам синтаксической модели UNIDATA CDM.

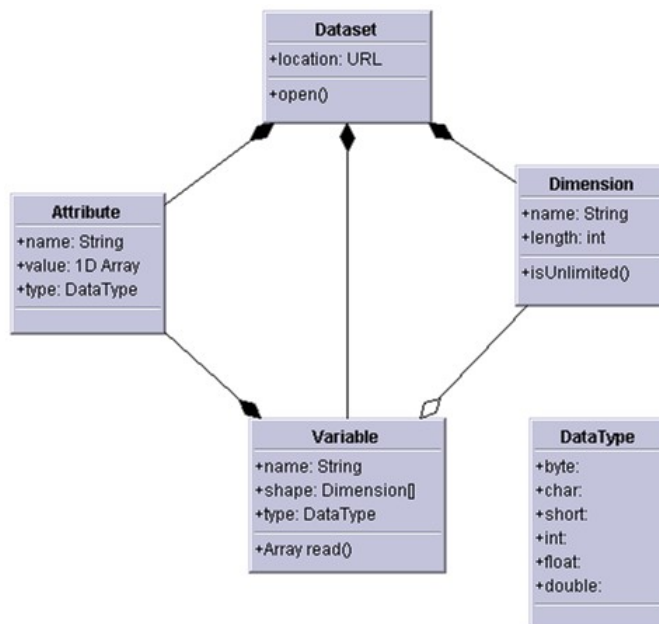


Рисунок 4. Синтаксическая модель, используемая в настоящем проекте

Семантический уровень, аналогичный уровню координатных систем модели UNIDATA CDM не является обязательным в текущей реализации, однако поддерживается уровень координатных систем Unidata CDM.

Семантический уровень, аналогичный уровню научных типов данных перенесен с уровня модели на уровень языка запросов.

2. Обобщенный язык запросов данных

Для предоставления конечному пользователю интерфейса запросов данных к источникам с различной моделью данных, в запросах вместо того, чтобы использовать различные наборы параметров для каждого источника, используется ограниченный набор параметров, релевантных для всех источников.

В рамках наук о Земле рассматриваются четыре модификации запросов:

- запрос данных с ориентацией «регулярная сетка»;
- запрос данных, ориентированных вдоль траектории;
- запрос нерегулярных данных, имеющих стационарную или глобальную координатную привязку
- запрос спутниковых изображений.

Данные модификации являются широко распространенными в науках о Земле и имеют аналоги в семантическом уровне модели данных UNIDATA CDM (см. предыдущий раздел).

В зависимости от модификации, запросы могут включать специфические параметры либо иметь различную семантику.

Поддерживаются следующие параметры запроса данных (Таблица 1).

Таблица 1. Описание параметров запросов данных

Параметр	Описание	Значения
command	команда сервису данных	get (по умолч.) – запрашивает сервис интерфейса источника данных describe – запрос метаданных источника в формате XML
dataset	запросить указанный источник данных по теме @theme	param[.vlevel[.#]]@theme[.source] geom_z@Geom geom_y.Geom_yr@Geom.yr SkinTemperature.Surface@Weather
format	формат представления данных результата	xml, json, ncml, csv, csv_noheader, jpeg, jpg, png, tif, tiff
datefrom	дата и время для выбора ближайшего первого отсчёта данных источника, сразу до либо совпадающие с ним	yyyy-mm-ddThh:mm:ssUTC
dateto	дата и время после последнего отсчёта данных источника, либо совпадающие с ним	yyyy-mm-ddThh:mm:ssUTC
location	пространственные ограничения запроса	Точка - Point: (56.0,38.0) область - ROI (Region of Interest): (нач.широта, нач.долгота, конечн.шир., конечн.долгота)
async	установить асинхронный режим работы	true – асинхронные запросы
asyncorder	идентификатор асинхронного запроса для получения статуса или результатов	

На рисунке 5 приведены примеры различных представлений выходных данных запроса. Например, выборка данных из базы данных реанализа погоды по измерениям температуры на изобарическом уровне 100 в географической точке 56,38 (широта, долгота), ограниченной интервалом дат:

http://93.180.14.101:18080/restful/servlet/GetData2?format=csv&datefrom=2011-03-01T00:00:00UTC&dateto=2011-03-03T00:00:00UTC&dataset=Temperature.IsobaricLevel.100@Weather&location=(56,38)

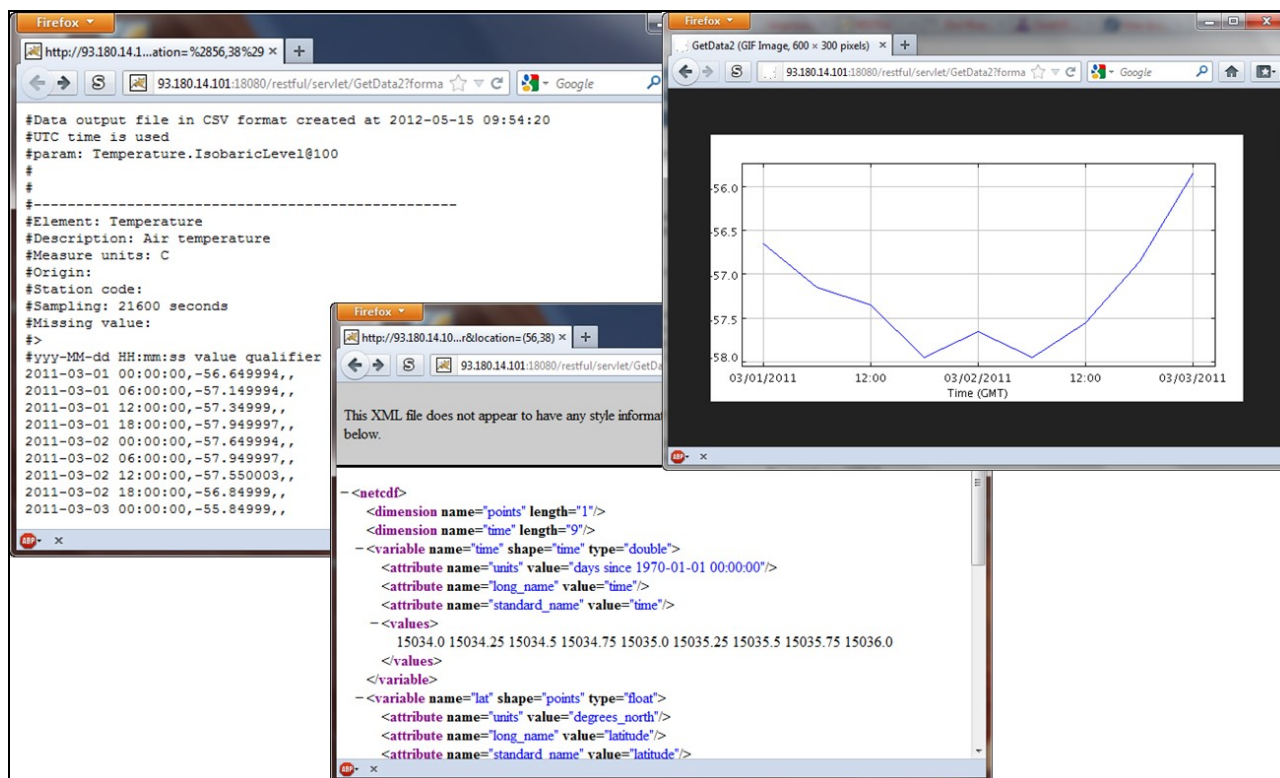


Рисунок 5. Представления форматов выходных результатов запросов данных

3. Модуль работы с метаданными

Основные компоненты

В ходе НИР был создан программный модуль для управления, поиска и каталогизации метаданных. Ниже описывается общая структура модуля (рисунок 6) и основные функции, позволяющие решить поставленные задачи.

Основные компоненты:

- Система авторизации пользователей
- Хранилище метаданных
- Профили объектов
- Хранилища объектов
- Веб-интерфейс
- Сервисы
- Источники данных

- Служебные плагины и сопровождающее программное обеспечение

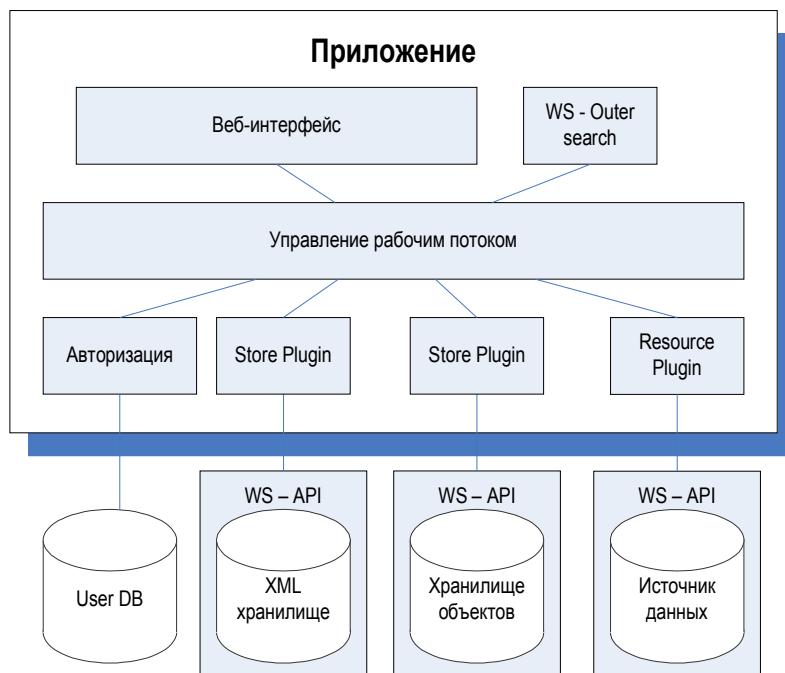


Рисунок 6. Структура программного модуля управления, поиска и каталогизации метаданных

Для обеспечения выполнения задачи управления метаданными, реализовано взаимодействие нескольких частей приложения. В рамках этого взаимодействия, разработана схема связей между метаданными, объектами данных, служебной информацией и деятельностью пользователей (рисунок 7).

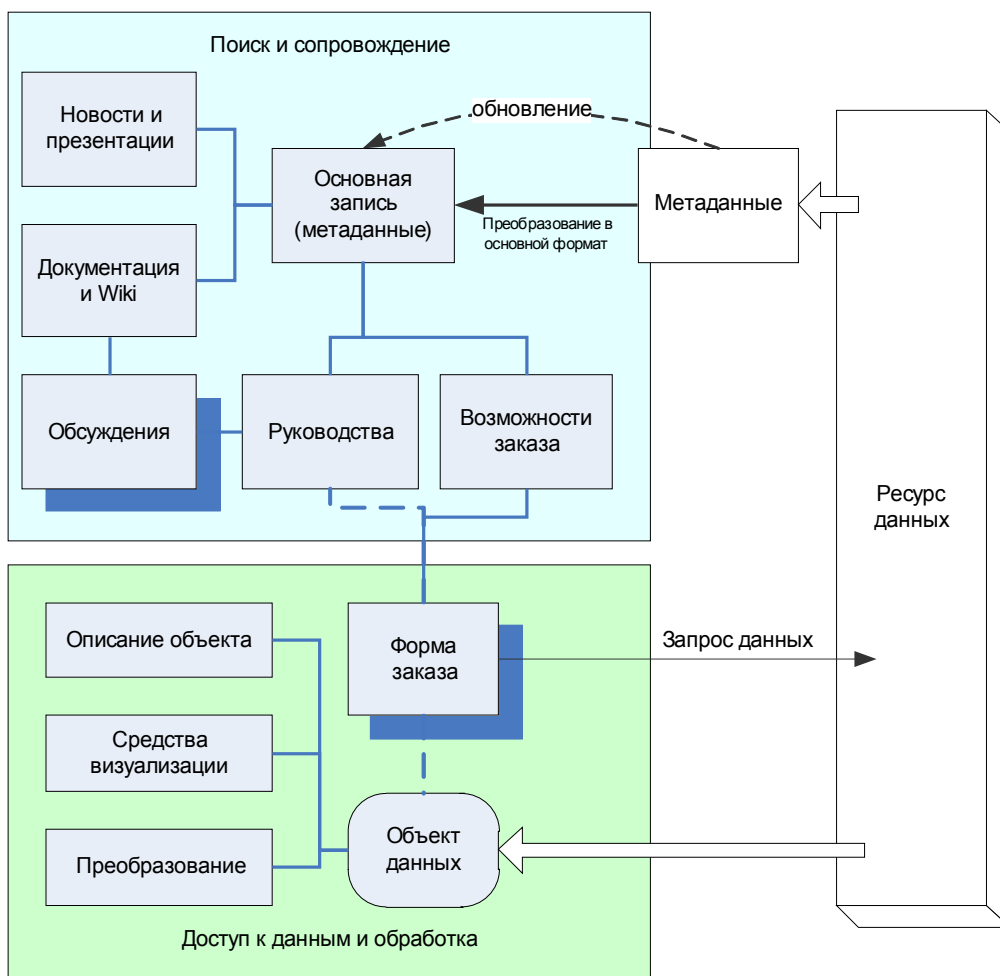


Рисунок 7. Принципиальная схема работы приложения с метаданными и данными

Одной из функций приложения является хранение записей в XML формате. Записи хранятся в разделах, каждый из которых настраивается индивидуально для записей того или иного типа.

XML хранилище

Основным хранилищем, в котором содержатся сами записи и служебная информация, является XML хранилище. К этому же хранилищу обращаются основные запросы при поиске записей.

Каждая запись в хранилище имеет идентификатор, который используется во всех случаях работы с записью (редактировании, связке с другой записью, в результате поиска и т.д.). Запись хранится в XML формате, специально определенном для записей данного типа (схема записи). Расположение записи внутри хранилища определяется настройкой раздела и привязывается к категории (дерево раздела, структура каталога раздела) к которой запись была отнесена при добавлении в систему.

Каждая запись, при попадании в хранилище, снаряжается служебной информацией, содержащей все данные о записи, накапливающиеся в процессе работы приложения. К служебной информации относится информация:

- о пользователе добавившем запись;
- об уровне доступа к записи;
- о редактировании записи и редакторах (со ссылками на устаревшие версии);
- о количестве просмотров, рейтинге (и другая статистическая информация);
- о связанных записях и объектах;
- и прочая информация, накапливаемая с течением времени.

В XML хранилище так же содержатся служебные файлы приложения, содержащие настройку всего приложения и каждого раздела в отдельности. От этих конфигурационных файлов зависят как схемы добавляемых записей, так и функциональность приложения (доступность поисковых сервисов, подключение хранилищ объектов и т.д.).

Хранилище объектов

Дополнительным хранилищем является хранилище объектов, которое подключается к разделу данных (рисунок 8), если вместе с XML записью необходимости загружать файлы в других форматах (документы, изображения, многомерные массивы, и т.д.).

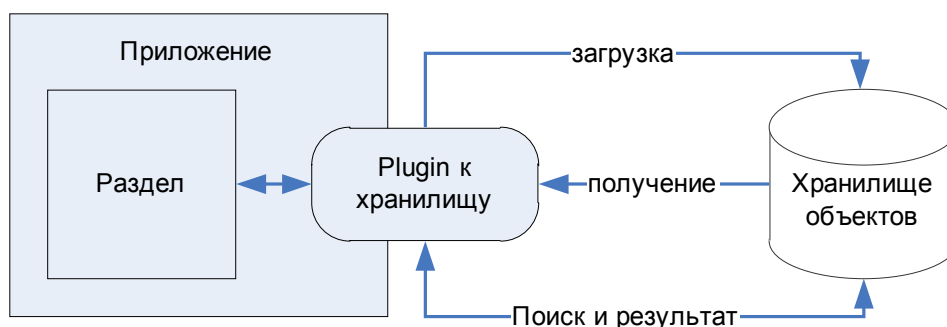


Рисунок 8. Схема взаимодействия с объектным хранилищем

Хранилище объектов расширяет функциональные возможности приложения за счет собственных функций работы с объектами. Если с разделом связано хранилище, то пользователи имеют возможность загрузить в него файлы, давая при этом описания их в качестве основных XML записей. Хранилище может иметь собственные функции (поиск, индексацию содержимого, предобработку объектов, переформатирование и т.д.) которые так же становятся доступны пользователям при обращении к разделу.

Сопровождение данных и служебная информация

Кроме поиска пользователи имеют и другие функции применимые к каждой из хранящихся записей. После добавления записи в хранилище, она может быть отредактирована (в рамках прав доступа пользователей), перемещена внутри раздела, пользователи могут добавлять реплики в обсуждение записи, или проставлять рейтинг. Многие функции связаны между собой, так при появлении новой реплики в обсуждении, автор записи получит уведомление об этом, через систему внутренних сообщений. Пользователи могут писать сообщения и напрямую друг другу. Пользователь имеет

возможность сохранить ссылку на запись в своих «закладках», что позволяет ему быстро вернуться к интересующей записи.

Некоторая часть информации из записей переносится в область служебной информации, где соотносится с общими для всех записей приложения данными, такими как: название, описание, географическая привязка, привязка по времени и т.д. Наличие таких обобщающих полей позволяет производить поиск записей по привязкам, не зависимо от их типа и внутренней схемы записи (рисунок 9).

Возможен поиск записей по служебной информации, и сопровождающей информации (по репликам обсуждений или пользователям, участвующим в обсуждении).

Статистическая информация, такая как количество просмотров, рейтинг, количество закладок на запись, используется для сортировки записей внутри каталога или в результатах поиска.

Каждая запись может быть связана с другой записью через систему «привязок». Можно создать обоюдную (двустороннюю) связку документов, или одностороннюю. Также можно указать тип связки, в каких отношениях находятся связываемые документы. Это позволяет создавать логические цепочки и группы отдельных самостоятельных записей, которые имеют логическую или историческую связь между собой.

Запись может быть привязана к именованной группе, для чего она помечается с помощью «тега». Записи, помеченные одним тэгом, относятся к одной категории, но не связаны жестко между собой, в отличие от системы связок.

За счет этой функциональности приложение позволяет не только хранить записи и документы, но и организовывать работу с ними, следить за их использованием, а так же, облегчить эти функции.

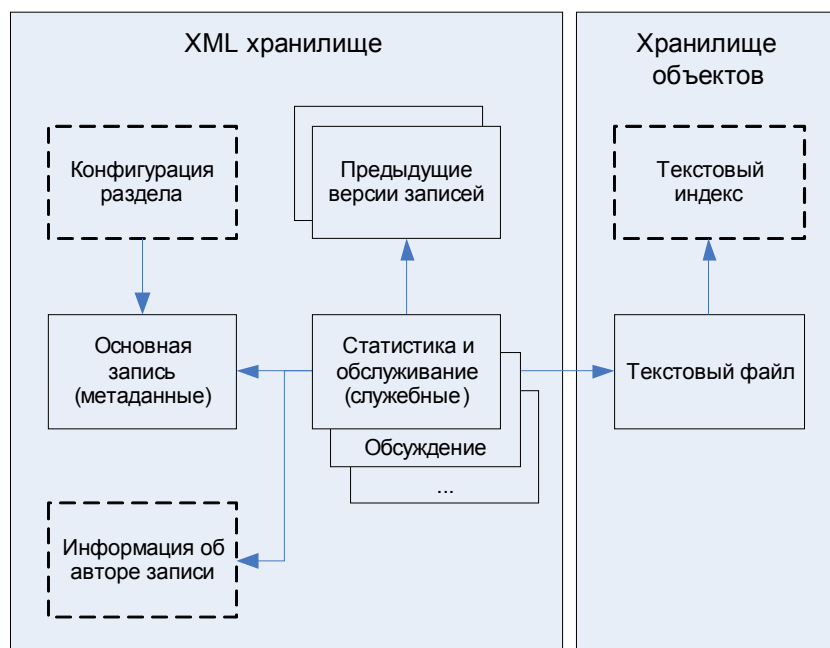


Рисунок 9. Связи объектов метаданных и служебной информации

Поисковые функции

Приложение предоставляет несколько вариантов поиска доступных пользователям. Эти инструменты предназначены для поиска записей в базе. Различаются следующие виды поиска: атрибутивный, контекстный, внешний, поиск в хранилищах.

Контекстный:

Поиск в разделе осуществляется с учетом особенностей типа записей и нацелен на содержимое записи. При этом пространство поиска может быть ограничено одним или несколькими элементами записи.

Атрибутивный:

Сквозной полнотекстовый поиск сразу по всем разделам на основании их свойств (например, статистике использования и просмотров). Возможны другие условия и области поиска, такие как поиск в репликах обсуждений или среди объектов, связанных с конкретным пользователем.

Внешний:

Кроме поиска доступного для пользователей через веб-портал, администраторы могут сделать разделы доступными для поиска клиентскими приложениями, используя механизм веб-сервисов. Для этого, необходимо определить разделы, доступные внешним приложениям-клиентам через веб-сервис, и в описании типа указать доступные для поиска и отображения элементы метазаписей, что делается отдельно при настройке каждого раздела приложения.

Веб-сервис внешнего поиска доступен клиентам по протоколу REST. Веб-сервис по запросу возвращает самоописание - описание типов хранящихся записей со списком разделов, доступных для поиска. На основании самоописания можно составить контекстный поисковый запрос, результатом которого будет список удовлетворяющих ему объектов. Важным свойством веб-сервиса внешнего поиска является возможность динамической настройки формата ответа: в поисковом запросе можно указать, какие из доступных элементов метазаписей необходимо вернуть в ответе.

Поиск в хранилище объектов:

Подключаемые к разделу хранилища могут обладать собственными функциями, в том числе функциями поиска. Обращение к хранилищу запроса на полнотекстовый поиск внутри

хранящихся в нем документов, в качестве результата выдаёт список идентификаторов документов. На основании полученного списка, приложение выбирает соответствующие XML записи и возвращает описательную информацию пользователю, пользователь убедившись в необходимости ему найденных документов загружает их из хранилища.

Принципиальная схема работы с сервисами данных

Через портал пользователи могут получить доступ к данным нескольких видов (рисунок 10).

Самый низкий уровень – уровень файлов. В этом случае пользователи могут самостоятельно загружать объекты и давать им описания, тем самым создаётся общий каталог записей, с которым и происходит дальнейшая работа. Пользователи могут в этом случаи загружать объекты данных к себе, просматривать их, визуализировать, при соблюдении договоренности о типе объектов, и наличии таких возможностей.

Следующий уровень – уровень перехода к сторонним приложениям, имеющим собственные формы заказа и поиска данных. Например, THREDDS OPeNDAP [3] сервер позволяет заказать выборку данных из него, но загрузить в него данные проблематично. В этом случае, приложение позволяет объявить такой сервер хранилищем, и пользователи могут загружать свои данные на сервер, а получать их уже с существующей формы заказа.

Третий вид сервисов позволяет получать как метаданные, так и данные через API. Такой сервис можно подключить в виртуальную обсерваторию через плагин, данные полученные из него могут сохраняться в пользовательском пространстве и хранилищах виртуальной обсерватории, что позволяет проводить дальнейшую интеграцию.

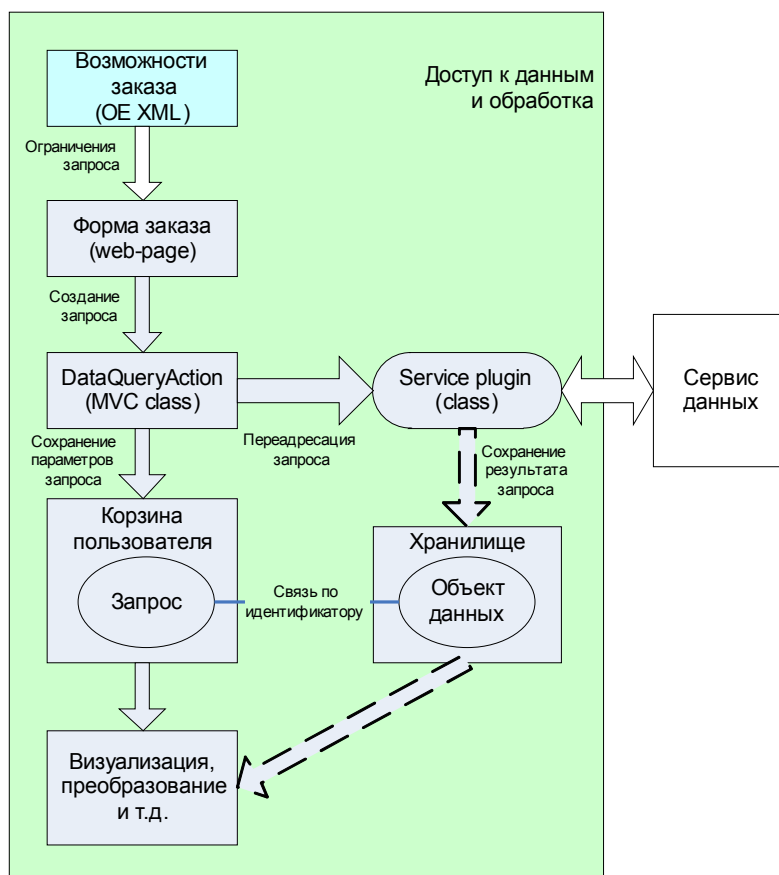


Рисунок 10. Общая схема обращения к сервисам данных на основе их метаданных

Для подключения таких сервисов в программном модуле существует специально разработанный формат метаданных, в который преобразуются метаданные предоставляемые сервисом, или запись про него в этом формате создаёт администратор. Это позволяет использовать запись для построения формы запроса и передавать запрос сервису, а полученные данные сохранять в корзине пользователя.

В корзине пользователя сохраняется не только ссылка на объект, подготовленный на удаленном ресурсе в ответ на запрос, но и сам объект данных, а так же сам запрос. Полученный объект данных пользователь может сохранить как объект обсерватории доступный другим пользователям (опубликовать), и даже преобразовать в самостоятельный источник данных.

4. Модуль автоматической генерации запросов к данным на основе метаданных

В связи с тем, что сервисы данных имеют разные формы обращений, и язык запросов к сервисам не ограничивается никакими требованиями, параметры запроса, определяющие одни и те же понятия, могут различаться по форме вызова. Например, указание на географическое положение может определяться как «latlon» или «location» или любым другим способом. Такая ситуация требует наличия информации по каждому сервису, о том, как параметры одного соответствуют параметрам другого, и создает трудности для пользователей и связанные с этим ошибки при запросах.

Модуль обработки метаинформации подразумевает унификацию запросов на уровне каталога, и создает пользовательский интерфейс, в котором поддерживается общая структура

заказа к разным сервисам. Пользователи имеют возможность сформировать заказ через интерфейс, после чего заказ автоматически переводится в язык запроса конкретного сервиса.

Это достигается через создание на стороне пользовательского интерфейса каталога метаданных, в котором зарегистрированы доступные сервисы, и регистрационная запись (метаданные в формате «OE») обеспечена обработчиком (плагином) переводящим заказ пользователя в запрос к сервису.

Создание отдельной записи для регистрации сервиса решает несколько задач, облегчая работу пользователя:

1. Заказы от пользователя к любому из сервисов создаются в рамках одного языка, вне зависимости от языка запроса самого сервиса.
2. Имеется возможность создания нескольких регистрационных записей для одного сервиса, при этом каждая запись будет содержать лишь те параметры которые нужны одной группе пользователей. Это бывает необходимо, если сервис предоставляет данные по длинному списку параметров, счет которых идет на сотни. Многим пользователям достаточно ограниченного списка в десятках параметров для полноценной работы в своей области.
3. Возможно создание виртуальных ресурсов, заказ к которым объединяет в себе обращение сразу к нескольким сервисам данных. Работа по разделению запроса и объединению результата возлагается на связанный с регистрационной записью «плагин».

Общая схема работы модуля представлена на рисунке 11.

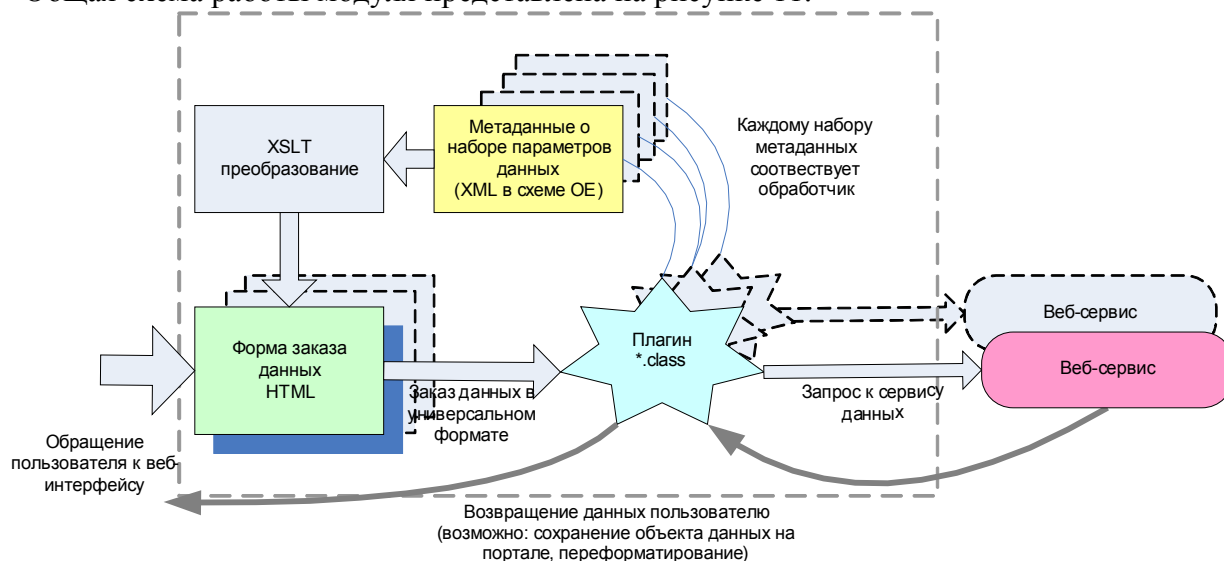


Рисунок 11. Общая схема работы модуля автоматической генерации запросов на выборку данных

При обращении пользователя посредством веб-интерфейса к одному из источников данных, модуль анализирует метаданные о наборе параметров данного источника и посредством соответствующего XSLT преобразования генерирует форму заказа данных. После заполнения пользователем всех необходимых полей, форма заказа перенаправляется на соответствующий данному формату плагин, который на основании полученных из формы параметров формирует правильный запрос к источнику данных, отправляет полученный запрос на удаленный сервис данных, получает и преобразует результат в требуемый формат.

Запросить метаданные о наборе параметров источника данных можно через RESTful сервисы модуля запроса метаданных об источнике данных.

Чтобы добавить новый источник данных требуется:

1. Добавить метаданные о наборе параметров данного источника в формате OE.
2. Если ни один из существующих плагинов не обеспечивает правильное

формирование запросов, то требуется реализовать новый плагин для подключаемого источника данных.

Все плагины должны удовлетворять единым требованиям. Плагин должен иметь стандартный метод для обращения со стороны приложения, принимать идентификатор объекта, строку запроса, и таблицу параметров запроса:

```
public org.jdom.Element process(Map params, String paramsString, String
basketObjectId) {
// ... Код запроса данных
// ... Код сохранения результата в промежуточное хранилище.
// Данный метод следует использовать по умолчанию:

FileStoreSave.storeFile(inp, basketObjectId, "XML file", null,
basketObjectId, "fileAction");

// ... Сообщение о статусе запроса
Element returnElm = new Element("result");
returnElm.addContent( (new Element("status")).setText("0"));
returnElm.addContent( (new Element("error")).setText(""));
return returnElm;
}
```

В случае ошибки при обработке данных, плагин должен вернуть в результирующем документе статус «1» и текст ошибки в соответствующем элементе документа

3. Прописать требуемый плагин и источник данных в конфигурационные файлы модуля.

Плагин должен быть прописан в конфигурационном файле “vo_store.conf” следующим способом:

```
<имя плагина>.pluginClass=<путь к классу плагина>
```

на плагин должна быть правильная ссылка из ОЕ. Плагин должен быть указан в дополнительных параметрах к запросу, иметь имя «voPlugin» и значение – имя задействованное в конфигурационном файле.

```
<QueryFormAdditions>
  <Addition>
    <AdditionName>voPlugin</AdditionName>
    <AdditionValue>имя плагина </AdditionValue>
  </Addition>
</QueryFormAdditions>
```

Элементы модуля написаны на языке Java с использованием Struts технологии для реализации веб-интерфейса, XML формата для хранения метаданных и XSLT преобразования для генерации HTML страниц. Для работы веб-интерфейса модуля необходим сервер приложений Tomcat.

5. Модуль доступа к данным

Функциональное назначение модуля – обработка RESTful запросов данных пользователя, выборка данных из источников, сериализация результата в требуемом формате, возврат результата пользователю с возможностью работы в асинхронном режиме.

Модуль позволяет осуществлять параллельные запросы данных с использованием нескольких ограничений атрибутов к большим базам данных и другим ресурсам хранения из различных научных областей.

Источниками данных могут быть, например:

- многомерные массивы;
- гранулы бинарных данных (например, спутниковых снимков);
- данные для построения графиков или слоёв электронных карт;
- интерфейсы к специализированным базам данным.

Для большинства наборов данных на координатной сетке используется формат данных, совместимый с UNIDATA CDM, после приведения к которому, над наборами данных можно гибко выполнять общие и специальные операции запросов и преобразований. Преобразование осуществляется в сервисе интерфейса, предоставляющего конкретный тип ресурса данных.

Данные из модели CDM могут быть сериализованы в различные форматы, например: XML, JSON [6-7], бинарный NetCDF файл, NetML.

Для предоставления конечному пользователю интерфейса запросов данных к источникам с различной моделью данных, в запросах вместо того, чтобы использовать различные наборы параметров для каждого источника, используется ограниченный набор параметров, релевантных для всех источников (см. раздел 2). В этом случае параметры запроса могут быть инкапсулированы в контейнер данных запроса, который передается менеджеру запросов в качестве входного параметра. Менеджер запросов с использованием интерфейса источника данных позволяет производить над данными операции выгрузки неформатированных данных, получения форматированных данных, преобразования в нужное выходное представление (в т.ч. графического формата изображения).

Асинхронные RESTful запросы данных

Выбор большого объема данных из базы данных или с ленты может потребовать слишком много памяти или может занять настолько много времени, что web-сервис получит ошибку таймаута или переполнения памяти. Чтобы избежать этой проблемы, сервисы выборки данных оснащены функцией, проверяющей размер запрошенных данных. Если этот запрос слишком велик, то клиенту выдается сообщение об ошибке (throttling технология). Однако такой подход ограничивает применение RESTful сервисов.

Возможное решение - асинхронный RESTful сервис. В этом случае можно разбить большой запрос на более мелкие части каждая из которых помещается в память сервера, а клиент запрашивает данные путем вызова сервера несколько раз подряд, получая каждый раз одну порцию данных. Протокол асинхронного запроса становится «stateful» и требует нескольких типов сообщений, включая запрос идентификатора заказа, запрос статуса и данных. В первом вызове, сервис данных вернет клиенту идентификатор заказа. Используя этот идентификатор, клиент может периодически запрашивать сервис, каждый раз получая статус выполнения заказа либо порцию данных.

Архитектура асинхронного RESTful сервиса данных имеет три компонента: контейнер REST сервиса, контейнер менеджера запроса и бэкэнда хранилища данных. Целью RESTful сервиса является поддержание асинхронные сессии с клиентом, перевод получаемых HTTP-запросов от клиента в запросы к менеджеру, и доставка ответа менеджера обратно клиенту. Менеджер запроса обеспечивает специальные действия для запроса данных у бекэнда хранилища данных и их обработки. Он также имеет очередь для хранения состояния запроса данных и пула данных. Дискковый пул данных используется для работы с "медленным" ленточной библиотекой. "В памяти" пул данных будет использоваться для хранения небольших порций данных, полученных от бэкэнда хранилища данных. Бэкэнд хранилища данных может быть как базой данных так ленточной библиотекой.

В первом вызове асинхронного сервиса данных всегда возвращается идентификатор заказа. Со стороны сервиса открывается «stateful» сессия для клиента и запрос перенаправляется менеджеру запросов, который помещает запрос в очередь. Клиент может специально указать, что возвращаемые данные следует разбить на мелкие части. Затем менеджер запросов поставит запросы в очередь: один запрос на каждую часть. Монитор очереди внутри менеджера запросов проверит - достаточно ли свободного места в пуле данных. Если пул не заполнен, а очередь не пуста, менеджер выберет необходимый объем данных из хранилища, сохранит в пуле и удалит запрос из очереди.

Каждый вызов клиента в сессии должен ссылаться на один и тот же идентификатор заказа. Ответ RESTful сервиса будет зависеть от статуса сессии. Если запрошенные данные еще не готовы, сервис вернет статус "processing". Если данные недоступны, сервис вернет статус "error" и закроет сессию. Если сессия была закрыта или не существует, сервис вернет "order <ID> not found".

Когда выборка данных готов, следующий вызов клиента получит от сервера ссылку на набор данных в пуле. Когда все данные будут отосланы клиенту, сессия будет закрыта. После этого вызов сервиса с данным идентификатором вернет статус "EOF".

Пример асинхронного запроса к источнику данных приведен на схеме Рисунка 12. Это типичный пример использования, когда спутниковые изображения с разных орбит сохраняются как бинарные гранулы в библиотеке. Перекачка изображений из библиотеки на сервер занимает некоторое время, поэтому можно использовать асинхронные сервисы для предотвращения таймаутов или ошибок отсутствия данных.

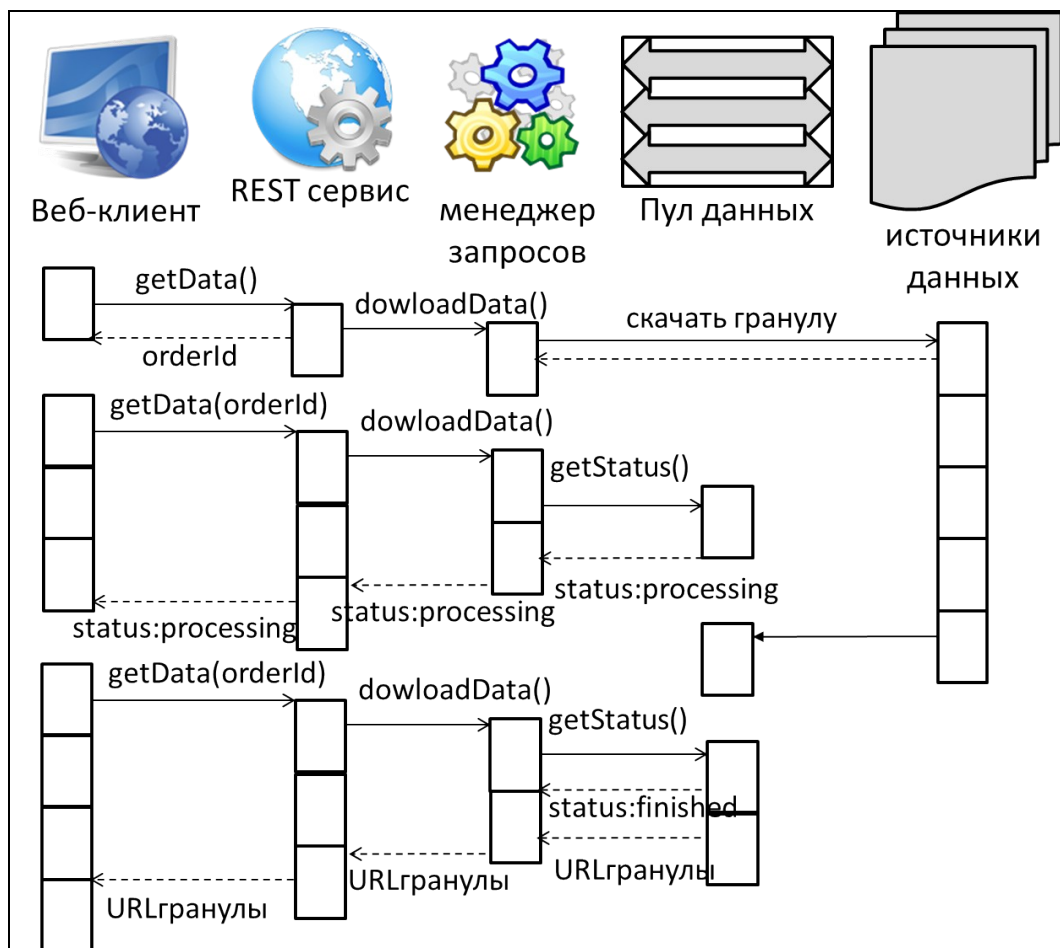


Рисунок 12. Схема асинхронного запроса

6. Демонстрационный веб-портал

Веб-портал (демонстрационное приложение) обеспечивает следующие функции:

- Хранит записи о доступных сервисах данных. Каждому сервису (или выделенной его части) соответствует XML-файл с метаданными в формате OE, что позволяет создавать на их основе пользовательские формы заказа данных. Записи о сервисах объединяются в каталог, доступный пользователям.

- Создает формы заказа данных с унифицированным интерфейсом. Заказ данных обеспечивается пользовательским интерфейсом со стандартными элементами (например, возможностью выбора географической точки или станции-поставщика данных на карте).
- В портал включаются специализированные плагины, которые обеспечивают стыковку универсально способа заказа с специфичным (для сервиса данных) языком запроса.

Портал может предоставлять функции сохранения получаемых от сервисов данных на своей стороне, средства просмотра данных и другие функции характерные и допустимые для веб-приложения.

7. Примеры

7.1. Пример источников данных

Погодный реанализ NCEP/NCAR

Погодный реанализ NCEP/NCAR [1] представляет собой архив глобальных данных о погоде и климате на период с 1948 года по настоящее время. Данные реанализа формируются на основе реальных наблюдений и глобальных математических моделей погоды. Использование математических моделей позволяет восстановить картину погоды за прошедшие годы, имея в наличие исторические результаты наблюдений с наземных погодных станций.

Данные реанализа NCEP/NCAR доступны на координатной сетке двух видов: регулярная сетка с шагом 2.5 x 2.5 градуса (73 точки по широте, 144 точки по долготe) и Гауссова сетка T62 (94 точки по широте, 192 точки по долготe). Шаг по времени составляет 4 часа.

Большинство параметров погоды определены на одном слое или поверхности (например: сигма-уровень или тропопауза), но некоторые параметры даны на нескольких изобарических уровнях. Количество изобарических уровней различно для разных параметр и составляет от 8 до 17.

На рисунке 13 показан пример обработки и визуализации данных погодного реанализа NCEP/NCAR. Показана среднеклиматическая температура у поверхности земли за январь, июль, а также за период с июля по январь.

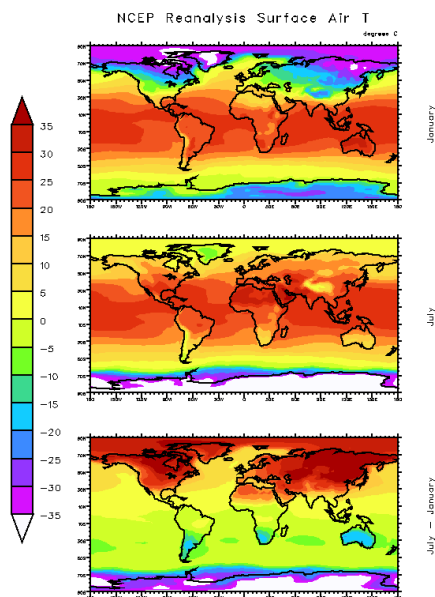


Рисунок 13. Среднеклиматическая температура по данным погодного реанализа NCEP/NCAR

Данные погодного реанализа NCEP/NCAR находятся в свободном доступе и выложены в интернет в виде NetCDF файлов. Архив реанализа насчитывает 74 погодных параметра, разбитых примерно на 5000 NetCDF файлов объемом по 30 – 500 мегабайт каждый.

Общий объем загружаемых данных погодного реанализа NCEP/NCAR составляет около 500 гигабайт.

Среднегодовые мозаики ночных огней по данным со спутников DMSP

Среднегодовые мозаики ночных огней представляют собой продукт обработки данных OLS-сенсора спутников DMSP (спутники F10 – F18). Мозаики доступны в виде изображений в формате TIFF с разрешением 30 угловых секунд на точку [8]. Пространственное покрытие составляет от -180 до 180 градусов по долготе и от -65 до 75 градусов по широте. При обработке исходные данные проходят фильтрацию по ряду критериев, включая отсутствие облаков, северных сияний и т.п. В результате получается детальная глобальная карта ночных огней по наблюдениям с одного спутника за один год. Мозаики доступны на период с 1992 по 2010 год. Время работы некоторых спутников частично пересекается, поэтому в некоторых случаях данные на один год доступны сразу с двух спутников, например, на 1994 год доступны данные со спутников F10 и F12.

Пример готовой мозаики за 2003 год по данным со спутника F15 показан на рисунке 14.

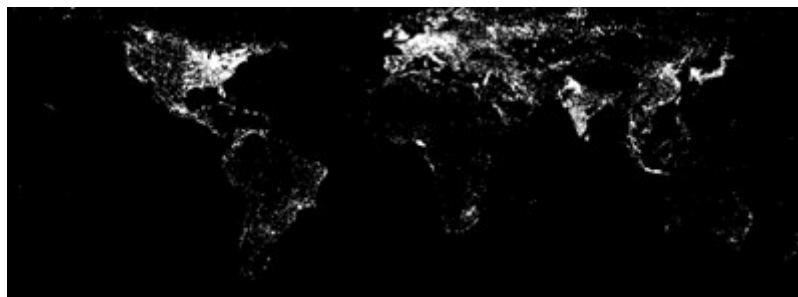


Рисунок 14. Среднегодовая мозаика ночных огней, спутник DMSP F15, 2003 год

Сами данные имеют достаточно простую организацию в виде двумерных массивов, зависящих от двух координатных переменных: широты и долготы.

7.2. Пример работы с веб-порталом

В данном разделе рассмотрен пример получения графика температуры на поверхности земли в одной точке за определенный временной интервал из источника прогноза погоды NWS. Последовательность действий включает несколько шагов:

1. Вход и авторизация на портал, а также выбор в каталогах требуемого источника «РЕСУРСЫ/NWS weather» (Рисунок 15).

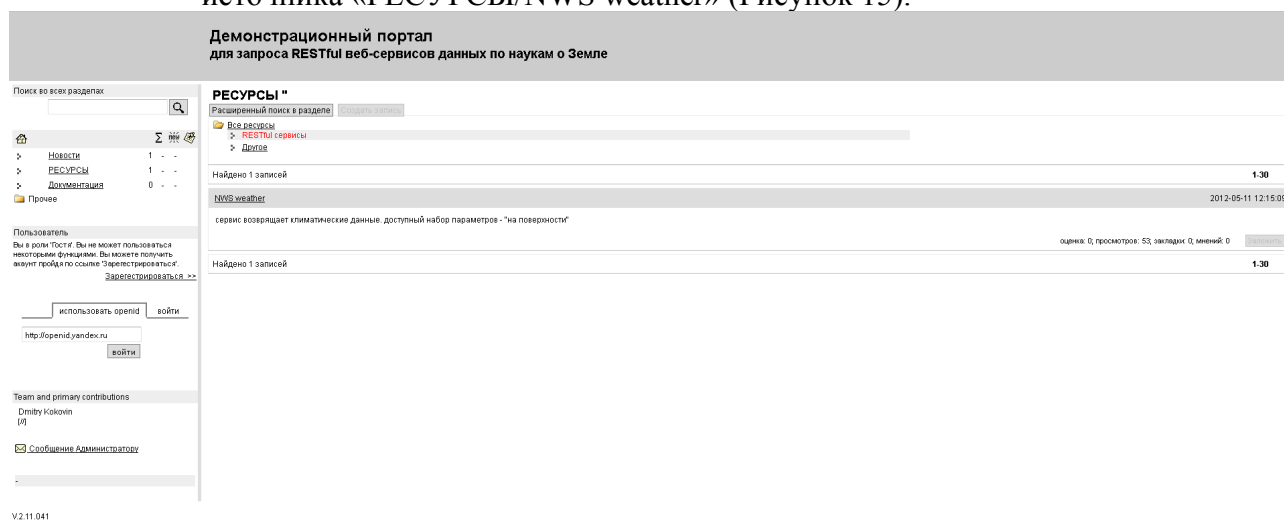


Рисунок 15. Выбор каталогов на веб-портале

2. В открывшейся форме, автоматически сгенерированной на основе метаданных источника, выбрать интервал дат, параметр, географическую точку и формат выдачи результата (в нашем случае «график») (Рисунок 16).

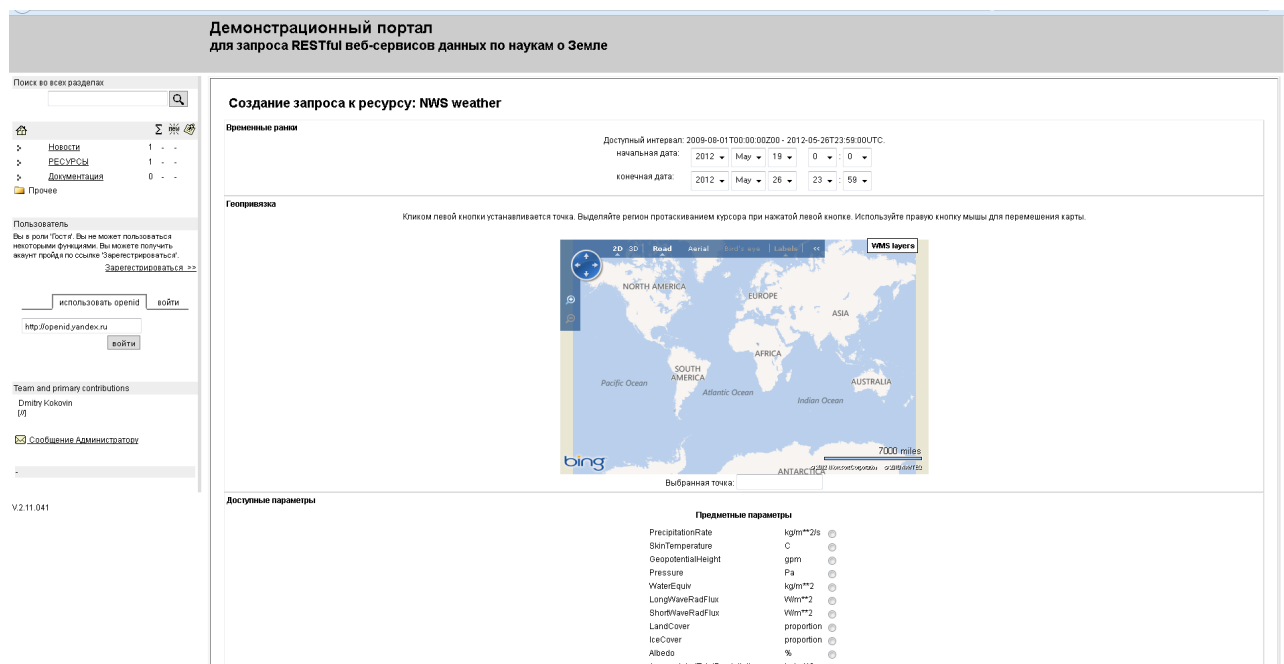


Рисунок 16. Заполнение формы параметров запроса

3. Нажать на кнопку «получить», в результате чего форма будет преобразована в запрос к источнику. Результат будет выведен на экран (Рисунок 17).

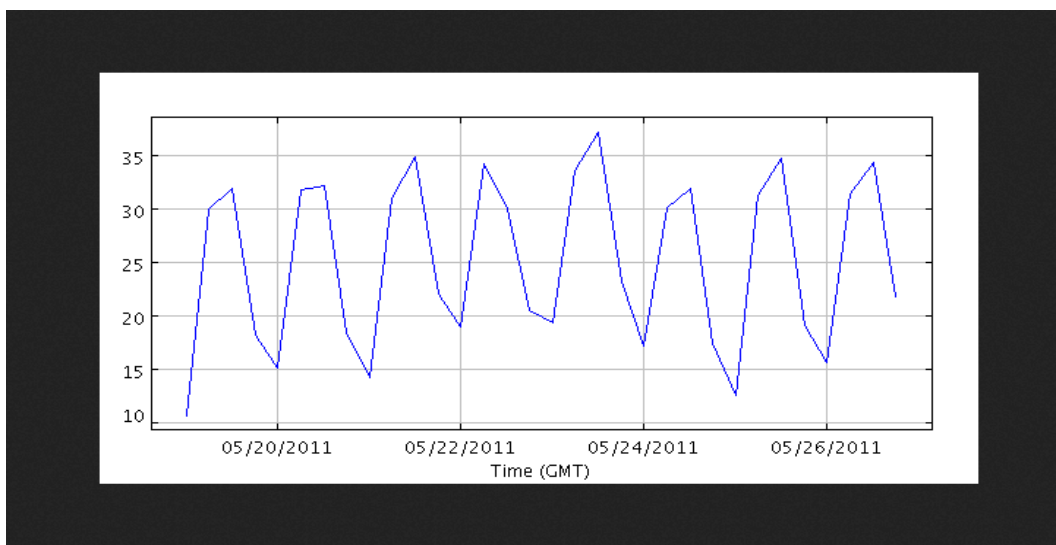


Рисунок 17. Получение результата

Выводы

В ходе работы над проектом были разработаны и реализованы RESTful-веб-сервисы удаленного доступа к ресурсам хранения данных для распределенных систем в области наук о Земле обладающие следующими особенностями: поддержка работы с метаданными и большими объемами данных, включая поиск и автоматическую генерацию запросов к данным на основе метаданных, поддержка общей модели данных, совместимой с UNIDATA Common Data Model, позволяющей интегрировать разнородные ресурсы в области наук о Земле, поддержка обобщенного языка запросов, позволяющего формировать универсальные запросы к нескольким разнородным источникам данных, горизонтальная масштабируемость, потоковый асинхронный запрос данных, позволяющий избежать ошибок таймаута и переполнения памяти на удаленном сервере.

Работа проводилась при финансовой поддержке Министерства образования и науки Российской Федерации (Государственный контракт № 07.514.11.4022 от 20.09.2011).

Литература

- [1]. Kalnay et al., The NCEP/NCAR 40-year reanalysis project, Bull. Amer. Meteor. Soc., 77, 437-470, 1996.
- [2]. Unidata Common Data Model. Сайт проекта:
<http://www.unidata.ucar.edu/software/netcdf-java/CDM/>
- [3]. Протокол доступа к данным OPeNDAP. Сайт проекта, URL: <http://opendap.org/>
- [4]. Network Common Data Form (NetCDF). Сайт проекта:
<http://www.unidata.ucar.edu/software/netcdf/>
- [5]. Сравнение моделей NetCDF-3 и NetCDF-4 на сайте разработчика. Электронный ресурс. URL: <http://www.unidata.ucar.edu/projects/THREDDS/GALEON/Reports/CDM-ISO/CDM-ISO-DataModels.htm>
- [6]. D. Crockford "The application/json Media Type for JavaScript Object Notation (JSON)" Technical report, IETF Network Working Group, RFC4627 (2006)
- [7]. К. Зур "A JSON Media Type for Describing the Structure and Meaning of JSON Documents" Technical report, IETF Network Working Group (2010)

[8]. Годовые очищенные композитные изображения по данным DMSP-OLS.
Национальный геофизический центр данных США (NGDC NOAA). Сайт проекта:
<http://www.ngdc.noaa.gov/dmsp/downloadV4composites.html>

**Александр Павлович Крюков
Михаил Николаевич Жижин
Алексей Анатольевич Пойда
Дмитрий Юрьевич Мишин
Дмитрий Петрович Медведев
Андрей Павлович Демичев
Дмитрий Сергеевич Коковин**

**РАЗРАБОТКА RESTFUL-ВЕБ-СЕРВИСОВ УДАЛЕННОГО ДОСТУПА К
РЕСУРСАМ ХРАНЕНИЯ ДАННЫХ ДЛЯ РАСПРЕДЕЛЕННЫХ СИСТЕМ В
ОБЛАСТИ НАУК О ЗЕМЛЕ**

Препринт НИИЯФ МГУ № 2012-4/882

Работа поступила в ОНТИ 20.07.2012