УДК 004.043

Анализ и хранение данных о состоянии окружающей среды♣

Ю.И.Молородов

Институт вычислительных технологий СО РАН
yumo@ict.nsc.ru

Рассмотрена архитектура и веб-интерфейс информационно-вычислительной системы, для построения и исполнения комплексных запросов к данным наблюдений за состоянием атмосферы промышленного центра Сибири. Представлена структура основных разделов расширяемой системы обеспечивающей выполнение множественных запросов и представляющей результаты в виде таблиц или графиков. Предлагается метод обработки временных рядов данных, на основе Вейвлет-функций [1]. Описывается интерфейс веб-приложения, реализованный на технологии Model–View–Controller (MVC).

Введение

Информационные технологии оказывают огромное влияние на все области человеческой деятельности, связанные с накоплением и обработкой информации. За относительно небольшое время существования информационно-коммуникационных технологий накоплен огромный объем разнообразных данных, представленных исключительно в электронной форме. Возникают задачи обеспечения доступа (в том числе и удаленного) пользователей к разнородным типам и форматов данных, обработки и интерпретации результатов наблюдений за состоянием атмосферы промышленного центра Западной Сибири.

Интеграция информационных ресурсов в единую информационную среду и организация доступа к вычислительным ресурсам – это одни из важнейших направлений развития современных информационных технологий. Решение проблем создания и интеграции информационных ресурсов и продуктов должно стать необходимым условием развития многих стран, в том числе и России. Стремительное развитие глобальных информационных и вычислительных сетей ведет к изменению фундаментальных парадигм обработки данных, которое можно охарактеризовать как переход к поддержке и развитию распределенных информационно-вычислительных ресурсов [2]. Технологии использования распределенных информационно-вычислительных ресурсов получают все больший приоритет в информационном обществе. При этом наблюдаются переход к исключительно распределенной схеме создания,

поддержания, хранения ресурсов[1] и стремление к виртуальному единству посредством предоставления свободного доступа к любым ресурсам сети через ограниченное число точек доступа. Постулируется принцип формирования в ресурсах сети единого, математически однородного поля компьютерной информации, которое способно стать универсальным и машинонезависимым носителем данных, унифицированных программ и глобально распределенных вычислительных процессов [3].

Предметная область

Атмосфера крупных городов подвержена интенсивному загрязнению продуктами жизнедеятельности человека. Это как выбросы в атмосферу предприятий находящихся вблизи городов, так и загрязнение со стороны автомобильного транспорта Уровень качественного и количественного содержания органических веществ в аэрозолях воздуха являются важным критерием для оценки загрязнения атмосферы и окружающей среды. Для этого на территории г. Новосибирска службами Гидрометеоцентра расположены посты наблюдения за состоянием атмосферы. На них производятся, в соответствии с рекомендациями ВОЗ, измерения в воздухе атмосферы содержания пыли, сажи, диоксида серы SO_2, озона, оксида углерода CO, диоксида азота NO_2, оксида азота NO, сероводорода H_2S, фенола CH, фтористого водорода HF, аммиака NH_3, формальдегида CH_2O и др. Объемы данных, полученных на постах наблюдений значительны по объемам и разнообразию. Это обстоятельство делает необходимым и актуальным использование распределенных информационно-вычислительных систем для задач, связанных с наблюдением за состоянием атмосферы. Использование электронных версий обеспечивает пользователю оперативный доступ к результатам наблюдений, хранение и математическую обработку.

Такую задачу решает специализированная проблемно-ориентированная Информационная Системы, которая позволяет решить проблему хранения этого эмпирического материала, его обработку с помощью современных математических алгоритмов. Это позволит перевести работу с этими данными на качественно более высокий уровень, открывающий перспективы для постановки и эффективного решения новых научных и практических задач.

Модель данных

Анализ данных наблюдений за состояние атмосферы позволяет сделать следующие выводы.

---

[1] Эффективная эксплуатация информационных ресурсов возможна только в том случае, когда они постоянно поддерживаются авторами, т.е. на основе технологий использования распределенных информационно-вычислительных ресурсов, которые получили название GRID-технологий.

Исходные данные наблюдений представляют собой временные ряды скалярных вещественнозначных функций, представленные в виде пар <***D, V***>, где ***D*** — дата и время проведения замера, ***V*** — результат измерения.

Каждому из рядов соответствует набор метаданных, адекватно описывающих временные ряды результатов измерений характеристик атмосферы:

- *Пост наблюдений*, на котором производились измерения.
- Параметрами этого свойства являются *координаты* поста наблюдений и его *тип* (например, пост наблюдений за загрязнением атмосферы, метеостанция)
- *Объект измерения*. Единственным параметром этого свойства является его название. Например, атмосферный воздух и т.п.
- *Измерявшаяся характеристика* объекта измерений. Это свойство обладает двумя параметрами: название измерявшейся характеристики (например, концентрация того или иного атмосферного аэрозоля), и ее размерность (например, мг/м^3)
- *Инструмент* (прибор), с помощью которого производились измерения.
- Единственной характеристикой этого свойства является название прибора.
- *Метод предварительной обработки* данных (например, почасовое усреднение). Единственной характеристикой этого свойства является название метода.

Логическая структура данных, представлена в виде логической модели на рис.1.

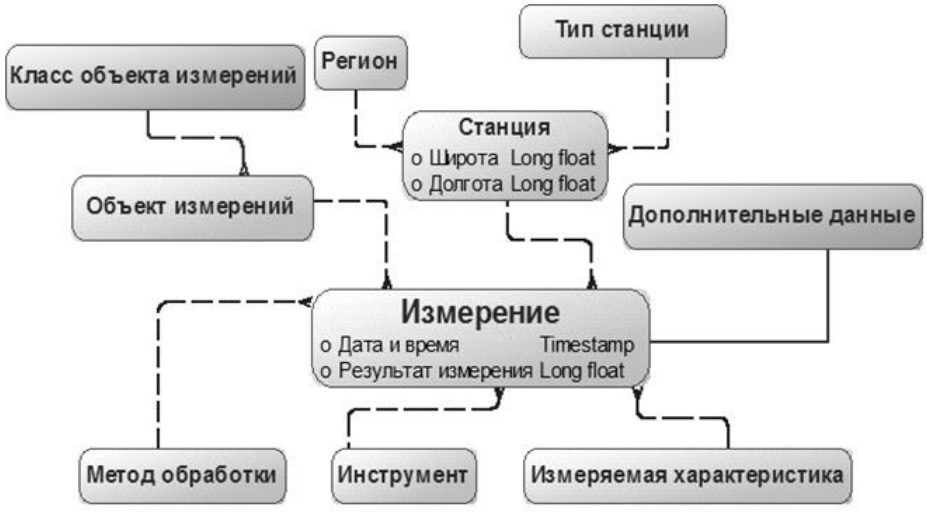

Рис.1 Структура логической модели данных

Реализация логической модели данных

Каждому из типов метаданных создается отдельная таблица реляционной СУБД со следующей структурой:

$<MPK_k, DESCR_k, C^1_k, \ldots, C^i_k, \ldots>$, где MPK_k – первичные ключи, C^i_k – значения специфичных для каждого типа метаданных характеристик, $DESCR_k$ – необязательное текстовое описание.

Данные измерений хранятся в отдельной таблице, в виде единичных значений, с каждым из которых ассоциирован набор метаданных ряда и отметка о времени замера значения. Этой таблице соответствует следующая структура:

$<PK, D, V, MPK_1, \ldots, MPK_k>$, где D – дата и время проведения замера, V – результат измерения, PK – первичный ключ, MPK_j – значения первичных ключей для записей таблиц, соответствующих базовому набору метаданных и задающих связь типа многие-к-одному между результатами измерений и метаданными.

Для задания связей между записями в таблице результатов измерений и дополнительными метаданными, а также для иерархической организации метаданных используются таблицы со структурой вида $<MPK_k, TPK_k>$, где MPK_j – значение первичного ключа из одной из таблиц, хранящей дополнительные метаданные, TPK_j – значение первичного ключа из таблицы, хранящей целевые записи, с которыми должна быть связана запись из таблицы метаданных.

Были введены служебные метаданные “регион” и “предметная область”, относящиеся к постам наблюдения и измерениям, соответственно. Свойство “регион” предназначено для задания параметров ассоциированной с постом наблюдения карты, “предметная область” – для задания предметной области, к которой имеет отношение тот или иной временной ряд.

Модульная платформа ИС

При выборе инструментов для разработки были проанализированы различные веб-фреймворки: Tapestry, Apache Strut для языка Java, Ruby on Rails для языка Ruby, Pylons, Django и TurboGears для языка Python. Эти программные каркасы веб-приложений, позволяют оптимизировать процесс разработки приложения за счет повторного использования программного кода. Наиболее подходящим из них оказался веб-фреймворк Pylons для языка Python.

Для придания гибкости создаваемой архитектуры и, повышения скорости и эффективности разработки, было принято решение о разработке модульной (плагинной, pluggable) архитектуры. Она предоставляет разработчику следующие возможности:

- Зависимости модулей и их разрешение;
- Реестр модулей;
- Ловушки событий (hooks), и обратные вызовы (callbacks);
- Генератор иерархических меню;

- Стек глобальных текстовых конвертеров;
- Подстройка используемых Object’s Relation Manager (ORM) типов под диалекты различных СУБД.

Рассмотрим некоторые из них.

Зависимости модулей и их разрешение

Каждый модуль имеет свой уникальный строковый идентификатор и список модулей, которые необходимы для правильного функционирования данного модуля (зависимостей), представленный набором аналогичных идентификаторов.

При запуске приложения, компонент, называемый реестром модулей, перечисляет имеющиеся в наличии плагины и проводит процедуру разрешения зависимостей.

Те плагины, для которых разрешение зависимостей прошло успешно вносятся в реестр плагинов, другие не подключаются к системе.

Реализована поддержка кольцевых зависимостей (например, если модуль ***A*** зависит от модуля ***B***, зависящего от модуля ***C***, в свою очередь имеющего среди зависимостей модуль ***A***, то такая цепочка модулей будет успешно загружена).

Реестр модулей

Реестр модулей представляет из себя синглтон – порождающий шаблон проектирования, который позволяет обратиться к пространствам имен модулей из произвольной точки кода и получить полное описание любого модуля. Данный объект позволяет обращаться к плагину двумя способами: либо по его идентификатору, либо по его порядковому номеру в списке модулей, которые представляют собой упорядоченный список (кортеж) дескрипторов плагинов. Здесь каждый следующий плагин может иметь среди зависимостей только предшествующие ему плагины (за исключением случая с кольцевыми зависимостями.

Ловушки событий, и обратные вызовы}

Взаимодействие модулей осуществляется с помощью обратных вызовов (callbacks). В момент инициализации модуль может зарегистрировать обработчики заранее определенного набора событий (например, событием является поступление запроса от пользователя), при наступлении которых вызовы выполняются в порядке, соответствующем порядку следования плагинов в реестре модулей.

Для некоторых событий обратный вызов является не только уведомлением о наступлении этих событий, но и ловушкой (hook) – некие данные, связанные с этим событием могут быть изменены внутри обработчика, а сам процесс вызова обработчиков может быть остановлен (путем возбуждения исключения, либо возвращения заранее определенного значения).

Поскольку реестр модулей доступен всем модулям, он может использоваться для организации взаимодействия между модулями напрямую: один модуль может предоставлять возможность зарегистрировать обработчик событий, а другой модуль может, используя реестр модулей, зарегистрировать какой-то из своих методов в качестве обработчика.

Генератор иерархических меню

Модуль в момент инициализации может зарегистрировать произвольное число пунктов меню, имеющих следующие свойства:

- Уникальный идентификатор;
- Название;
- Вес при сортировке;
- Идентификатор родителя (при отсутствии это будет элемент верхнего уровня);
- Целевой URL.

Д для проверки их доступности модулей должен быть зарегистрирован обратный вызов. Во время регистрации модулей формируется внутреннее иерархическое представление меню. В момент генерации ответа на запрос пользователя выполняются обратные вызовы к модулям, позволяющие исключить из меню те пункты, которые не должны быть доступны в данный момент.

После этого генерируется HTML-представление меню, которое и помещается в отправляемый пользователю HTML-документ.

Заключение

Предложена модульная архитектура для построения веб-сайтов. На ее основе созданы модули информационно-вычислительной системы, предназначенный для хранения и обработки результатов мониторинга атмосферы промышленного центра Западной Сибири. Разработана расширяемая модель данных, предназначенная для хранения скалярных временных рядов. Эта модель может использоваться и в аналогичных компонентах систем, предназначенных для хранения и обработки временных рядов данных. На основе предложенной архитектуры создана расширяемая подсистема импорта данных наблюдений за состоянием атмосферы.

Создан модуль, позволяющий формировать отчеты по хранимым данным за произвольный период. Предоставлены возможности для математической обработки хранимых данных, и создания других компонентов системы. В качестве демонстрации возможностей API обработки реализованы модули построения графиков и экспорта отчетов в XML.

ЛИТЕРАТУРА

[1] Torrence, G. A practical guide to wavelet analysis / G. Torrence, G.P. Compo // Bull.Amer. Meteorol. Soc. — 1998. — Vol. 79, No. 1. — P. 61–78.

[2] *Жижимов О.Л., Федотов А.М., Чубаров Л.Б., Шокин Ю.И.* Технология создания распределенных информационно-вычислительных ресурсов СО РАН // Тр. Первой Межд. конф. САИТ-2005. 12–16 сент. 2005 г., Переславль-Залесский. Т. 2. “Системный анализ и информационные технологии". М., 2005. С. 161–165.

[3] The Grid: Blueprint for a New Computing Infrastructure Ed. by I. Foster, C. Kesselman. Morgan Kaufmann Pub., San Francisco, CA. 1999.

[4] *Федотов А.М., Барахнин В.Б., Гуськов А.Е., Молородов Ю.И.* Распределенная информационно-вычислительная среда для исследования экологических систем. // Вычислительные технологии. – 2006. – Т. 11. Ч. I. – С. 113-125.

[5] Мониторинг качества атмосферного воздуха для оценки воздействия на здоровье человека. Копенгаген: Региональные публикации ВОЗ // Европейская серия. 2001. №5. С. 293..

[6] ГОСТ 17.2.1.03-84. Охрана природы. Атмосфера. Термины и определения контроля загрязнения

♣ Работа финансировалась Президентской программой НШ 931.2008.9, РФФИ (грант № 09-07-00277), и Интеграционным проектом СО РАН № 50.