пониманию особенностей толкования смысла отображенного содержания и более эффективно организовать рынок геоинформационных и картографических услуг и продукции.

Литература:

1. Божилина Е.А. Обучающее значение легенд электронных карт природы //Вестник МГУ. Сер. 5. География. 2004. № 4. – С.41-44.

2. ИнтерКарто-ИнтерГИС-18: Устойчивое развитие территорий: теория ГИС и практический опыт. Материалы Международной конференции. /Редколлегия: С.П. Евдокимов (отв. Ред.), В.С. Тикунов, Т.В. Ватлина. – Смоленск, 26-28 июня, 2012 г. – Смоленск, 2012. – 532 с.

3. Лютый А.А. Язык карты: сущность, система, функции. – Изд. 2-е, испр. – М.: ИГ РАН, 2002. – 327 с.

4. Суворов А. К. Географические информационные системы, дистанционное зондирование и картографическая культура: принципы интеграции и формирования субкультур // ИнтерКарто-ИнтерГИС-18: Устойчивое развитие территорий: теория ГИС и практический опыт. Материалы Международной конференции. /Редколлегия: С.П. Евдокимов (отв. Ред.), В.С. Тикунов, Т.В. Ватлина. – Смоленск, 26-28 июня, 2012 г. – Смоленск, 2012. – С. 167-176.

5. Суворов А.К. Язык карты как семиотико-топологическая основа интеграции информационных потоков // Геодезия и картография. - 2013. - № 8. – С. 23-32.

6. URL: http://digit.ru/opinion/20130711/403246794.html.

7. URL: http://www.marhi.ru/AMIT/2012/1kvart12/goldin/goldin.pdf.

8. URL: http://old.mosmap.ru/stat/proektirovanie-i-oformlenie-kart-dlya-web-servisov.shtml.

9. URL: http://roscartography.ru/images/doc/strategiya_v13_10_10.pdf.

ПРОГРАММНО-ТЕХНОЛОГИЧЕСКОЕ ОБЕСПЕЧЕНИЕ СБОРА И ХРАНЕНИЯ ДАННЫХ СЕРИЙНЫХ НАБЛЮДЕНИЙ ЗА СОСТОЯНИЕМ ОКРУЖАЮЩЕЙ СРЕДЫ

А.В. Токарев
Институт вычислительного моделирования СО РАН
г. Красноярск, Россия

SOFTWARE AND TECHNOLOGY FOR COLLECTING AND STORAGE SERIAL OBSERVATIONS OF THE ENVIRONMENT

A.V. Tokarev
Institute of Computational Modeling SB RAS
Krasnoyarsk, Russia Federation

Abstract. Some aspects of the collecting and storage serial observations of the environment is considered. The paper considers the creation of a storage subsystem and climate data collection, which

will enable users to perform data collection, integration, storage and analysis. Developed server data collector prototype implement basic functions of the program interface (API) and administrative web-interface. Several data flows from different data sources is created – public sources of climate data, autonomous weather stations, information systems of monitoring. Work is underway to establish a means for effective access to observational data, services, reporting and analysis of information.

Введение

Решение задач информационного обеспечения наблюдений за состоянием окружающей природной среды невозможно без применения современных средств измерения и связи, новых компьютерных технологий. Интегрирование всех составных частей мониторинга в единой технологии минимизирует затраты на их стыковку, сокращает время обмена и преобразования данных, исключает потери информации, повышая тем самым надежность и эффективность создаваемых систем [Якубайлик, 2012]. Исследования на данном этапе были посвящены анализу существующих систем экологического мониторинга, проблемам и особенностям реализации в них технологических цепочек для получения, передачи, хранения и использования данных.

Предполагается разработка эффективных информационных моделей для сопровождения регулярных наблюдений за окружающей средой, разработка соответствующего математического аппарата и создание прототипа распределенной информационно-вычислительной системы обеспечивающих возможности:

- сбора, интеграции и хранения данных;
- поиска метаданных, включая запросы по временным, пространственным и содержательным характеристикам;
- наполнения и предоставления данных из хранилища через веб-сервисы;
- интерактивного анализа и визуального представления данных.

Были проанализированы существующие системы экологического мониторинга, проблемы и особенности реализации технологических цепочек для получения, передачи, хранения и использования данных [Токарев, 2013]. Примеры сервисов сбора и обработки:

- **Pachube** (http://xively.com) – сервис, позволяющий собирать, транслировать и визуализировать потоки различных данных. Пользователь системы может, к примеру, подключить электронный термометр к сервису, и он будет транслировать измеренную температуру, в интернет пространство. Проект несколько раз переименовывался (Pachube → Cosm → Xively), менялась лицензионная политика, с уклоном в коммерческую сторону.
- **Народный мониторинг** (http://narodmon.ru) **–** молодой проект по сбору и отображению на карте мира показаний различных датчиков среды (температура, давление, влажность и т.п.) практически в реальном времени по фактическому состоянию.
- **OpenWeatherMap** (http://openweathermap.org). Собираются данные от профессиональных и частных погодных станций (>40 тыс.). Набор показателей фиксирован (температура, влажность, давление, осадки, освещенность, и др., всего около 15 шт.). На основе собранных данных и математических моделей формируется прогноз погоды. С помощью API можно получать текущую погоду, прогноз, и исходные данные с метеостанций.

Консорциум OGC разработал набор спецификаций моделей, интерфейсов и протоколов, предназначенных для описания сенсоров и сенсорных сетей с ориентацией на использование в веб-приложениях. В частности:

• ***Sensor Model Language (SensorML)*** – спецификация информационной модели и XML-схем, которые позволяют описывать сенсоры, искать их в сети, а также использовать результаты наблюдений с датчиков.

• ***Sensor Observation Service (SOS)*** – стандарт интерфейса для сервиса, предоставляющего данные результатов наблюдений с датчиков, включая удалённые, локальные, фиксированные и мобильные датчики.

Архитектура

В работе рассмотрено создание подсистемы хранения и сбора климатических данных, которая даст возможность пользователям выполнять сбор, интеграцию, хранение и анализ данных (рисунок 1).

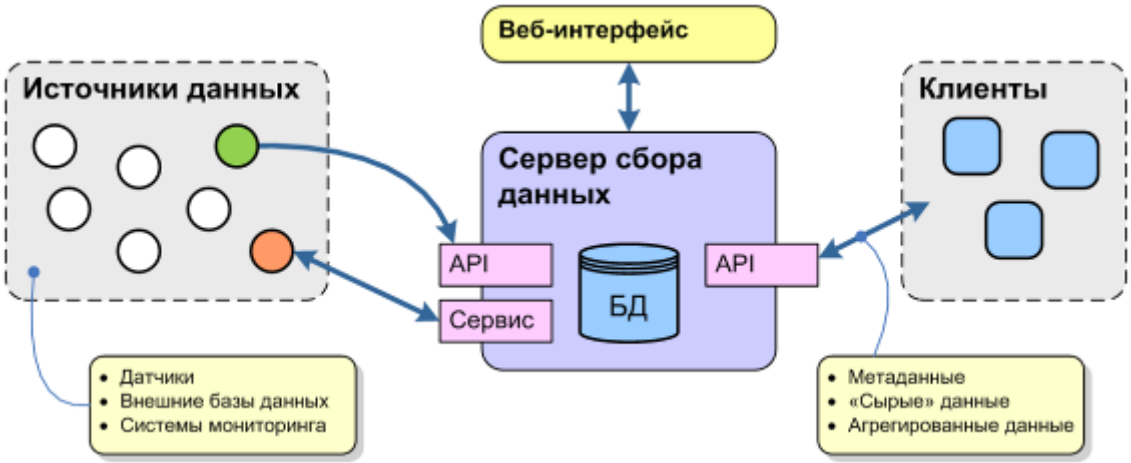

Рисунок 1 – Общая архитектура системы

Показатели с внешних источников данных собираются на сервере сбора. Источниками данных могут быть отдельные датчики, внешние базы данных или информационные системы через дополнительные адаптеры. Предусмотрено несколько способов наполнения: активного, выполняющего периодический опрос и загрузку данных из источника и пассивного, обеспечивающего только прием данных по инициативе источника через предоставленный программный интерфейс (API) в виде веб-сервиса. Извлечение данных возможно как в «сыром», так и в агрегированном по времени виде.

Доступ к серверу сбора предоставляется только после аутентификации. Для этого все пользователи и взаимодействующие информационные системы должны пройти регистрацию на сервере, после которой выдается уникальный ключ для работы с API сервера. Поддерживается разграничение доступа к объектам системы в зависимости от прав пользователя. Веб-интерфейс обеспечивает навигацию объектам системы, базовые инструменты для отображения данных, управления пользователями, проектами, показателями и другими сущностями системы.

Для хранения метаинформации и собираемых данных разработана концептуальная модель базы данных (рисунок 2). Центральной сущностью является ***проект***, который определяет тип собираемых данных, например, метеорологические данные, загрязненность воздуха, или пожарная обстановка. Проекты могут быть как приватные – доступные только владельцу, так и публичные. Администратор проекта может задавать список пользователей, которые могут наполнять проект или получать доступ к накопленным данным.

В рамках проекта собираются данные с различных ***площадок***, с фиксированным расположением либо мобильных. На площадках размещены ***датчики***, которые генерируют значения отдельных ***показателей***, такие как: температура воздуха, относительная влажность, концентрация мышьяка в почве, концентрация фенола в воздухе, и др. Сами показатели разбиты по тематическим разделам, у каждого показателя заданы единицы измерения, ПДК, и тип данных (целое, число, строка).

Собираемые показатели с датчиков записываются в таблице значений (рисунок 3). Это таблица фактов с такими измерениями, как дата-время, показатель и площадка. На основе первичных значений рассчитывается набор агрегированных значений. Агрегация выполняется по нескольким временным интервалами: 5 мин, 1 час, 1 день. Данные за промежуточные интервалы рассчитываются уже «налету» посредством динамических SQL-запросов. Кроме этого автоматически вычисляется и кэшируется набор показателей, данные по которым были собраны в хранилище.

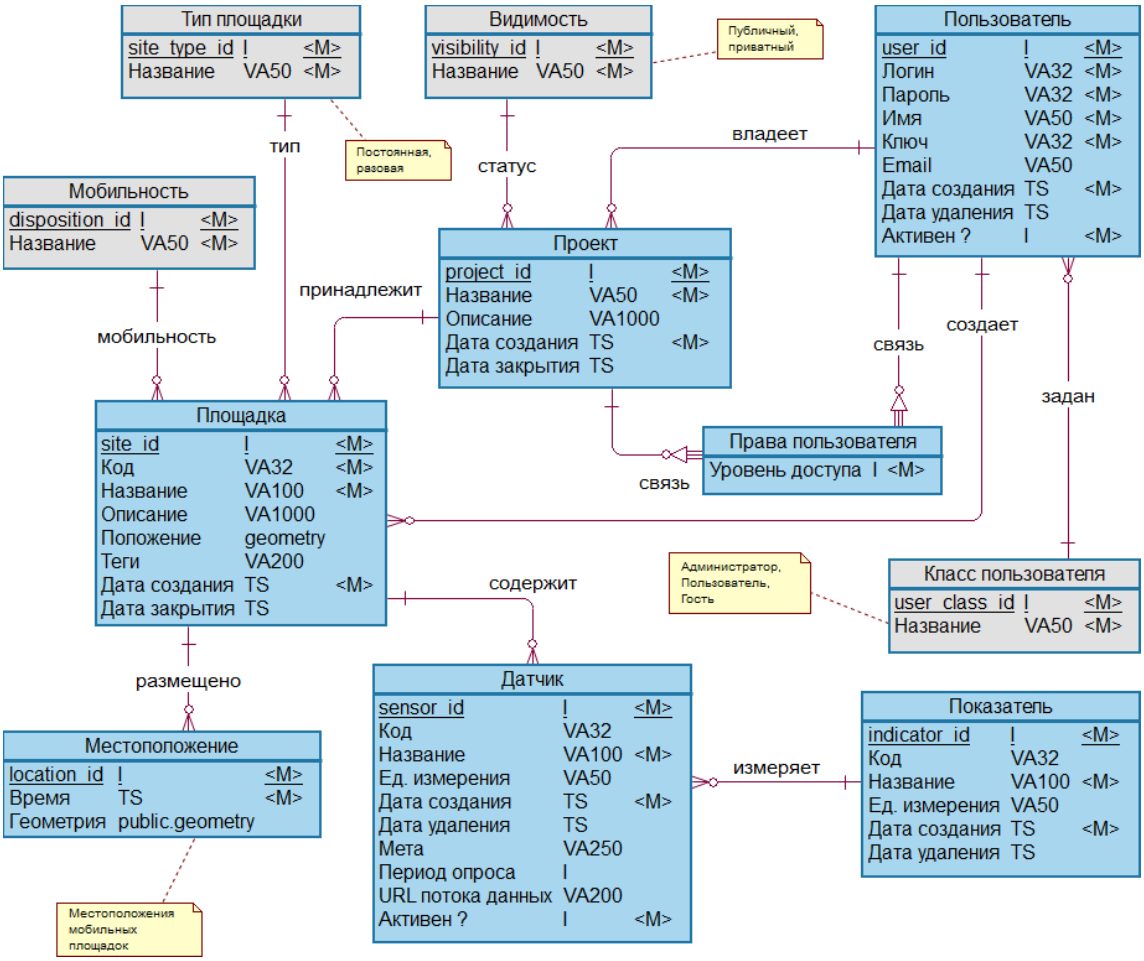

Рисунок 2 – Концептуальная модель базы данных (ядро)

Технологии

Для построения веб-сервисов наибольшее распространение получили следующие протоколы и подходы в реализации: XML-RPC, SOAP, REST [Pautasso, 2008].

В начале своего развития веб-сервисы часто использовали в качестве языка запросов простые GET/POST запросы по HTTP протоколу, аналогичные запросам веб-браузера, а в

качестве ответов обычно использовались XML документы. В дальнейшем этот подход сформировался в виде архитектурного стиля построения веб-сервисов REST (Representational State Transfer) [Fielding, 2002], выстроенный на хорошо известных стандартах консорциума W3C, таких как: HTTP, URI, XML, RDF. Это набор общих принципов построения веб-сервисов с определенными приоритетами: масштабируемость, независимость от платформы, расширяемость. Сущность подхода заключается в нескольких ключевых аспектах:

- система представляется отдельными ресурсами со своим состоянием;
- ресурсы идентифицируются собственными URI и связываются только через явное указание ссылок и форм в их представлениях для обеспечения интеграции между сервисами;
- для доступа к ресурсу используется универсальный интерфейс в виде небольшого количества методов HTTP (GET, PUT, DELETE, POST) в соответствии с их изначальным смыслом;
- ресурсы представляются стандартными MIME-типами для независимости от платформы и самоописываемости;
- состояние клиентского процесса не сохраняется для масштабируемости.

Самой известной системой, построенной в значительной степени по архитектуре REST, является современная Всемирная паутина (WWW).

Необходимость стандартизировать способ представления запросов и ответов для веб-сервисов привела к созданию стандарта SOAP (Simple Object Access Protocol), а так же ряда сопутствующих стандартов – языка описания веб-сервисов WSDL, распределенной системы регистрации веб-служб UDDI, и др. SOAP изначально разрабатывался как замена протоколу XML-RPC, но позже был значительно доработан, и стал протоколом-стандартом для построения веб-сервисов уровня предприятия. SOAP активно использует XML для кодирования запросов и ответов, а также строгую типизацию данных, гарантирующую их целостность при передаче между клиентом и сервером. Обратной стороной медали является большая сложность (по сравнению с REST) стандарта, повышенные накладные расходы, невозможность работать без программного инструментария.

Как правило, веб-сервисы, которые в основном предназначены для поиска и получения информации, лучше приспособлен REST подход. Он легковесный, гибкий, мультиформатный и более приспособлен для решения подобных задач. Поэтому мы будем ориентироваться на него.

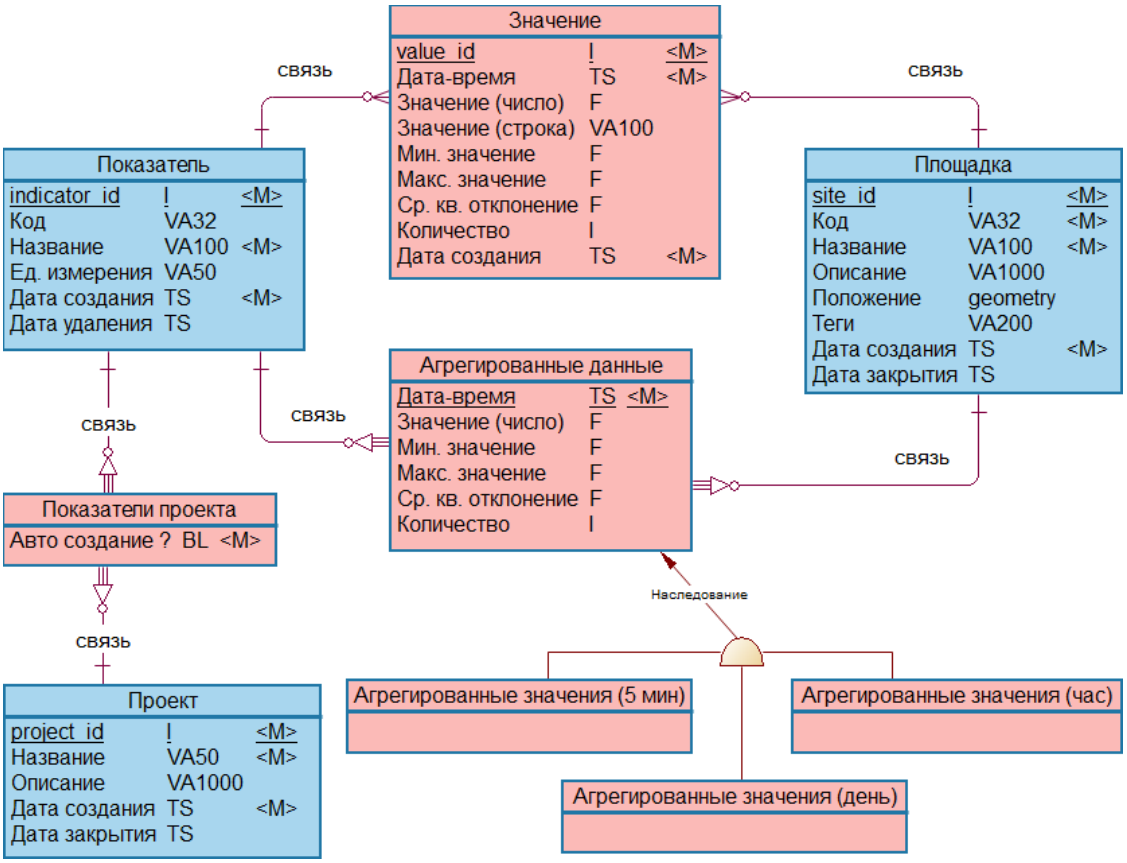

Рисунок 3 – Концептуальная модель базы данных (хранения данных)

Разработка выполнялась на языке сценариев PHP 5 с использованием фреймворка Yii. Это высокоэффективный PHP-фреймворк для разработки веб-приложений. Он основан на компонентной структуре и позволяет максимально применить концепцию повторного использования кода, существенно ускоряя процесс веб-разработки.

Для хранения данных выбрана СУБД PostgreSQL – свободно распространяемая объектно-реляционная система управления базами данных, наиболее развитая из открытых СУБД в мире и являющаяся реальной альтернативой коммерческим базам данных. Немаловажным преимуществом является наличие дополнительных модулей, в том числе модуля PostGIS, который обеспечивают удобную работу с пространственными данными внутри СУБД. Поддерживается более 300 различных функций для работы с векторными данными, пространственные индексы, популярные обменные форматы и многое другое.

Программный интерфейс

Веб-сервис реализован на основе REST подхода. Запрос передается HTTP методами GET/POST/DELETE с параметрами, результат формируется в формате XML/JSON. Базовая структура запроса следующая:

http://<сервер>/api/<версия API>/<объект>.<формат>?key=<key>& <дополнительные параметры>, где:

– *версия API* – версия интерфейса, параметр предназначен для развития сервиса и поддержки устаревших версий программного интерфейса;

– *объект* – составной идентификатор объекта URI, например, «project/1/metadata» - метаданные проекта с внутренним идентификатором = 1;

– *формат* – формат выдаваемого ответа, поддерживаются форматы XML и JSON.

– *key* – ключ пользователя для доступа к сервису. Может указываться в параметрах, либо заголовке HTTP запроса.

– *дополнительные параметры* – дополнительные параметры для конкретного запроса.

Основные запросы:

1. Получить описание проекта. GET http://host/api/1.0/project/<project_id>/metadata.xml.

Возвращается полное описание проекта и список собираемых показателей.

2. Получить площадки проекта. GET http://host/api/1.0/project/<project_id>/sites.xml. Возвращается список площадок, которые зарегистрированы в проекте (рисунок 4).

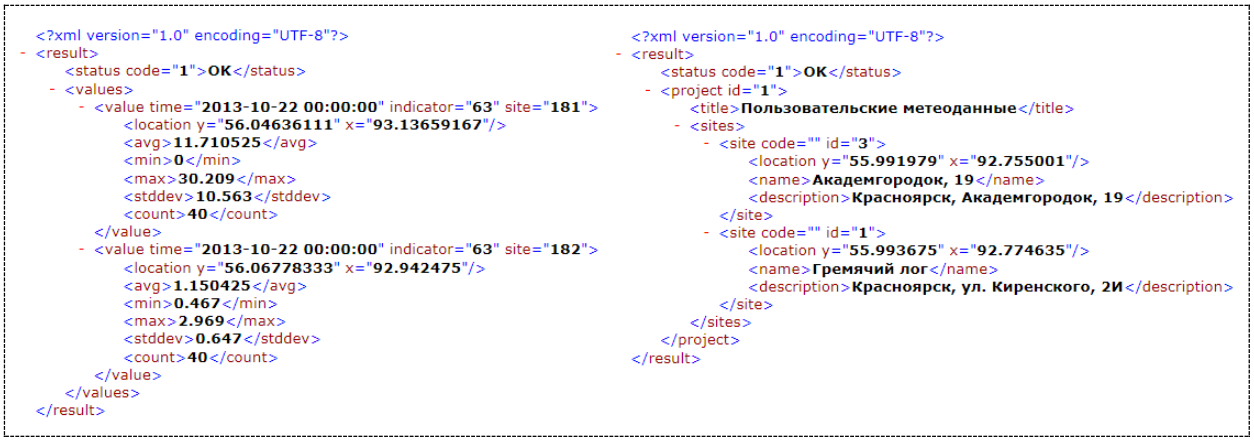

Рисунок 4 – Примеры ответов веб-сервиса

3. Получить «сырые» данные измерений. GET http://host/api/1.0/project/<project_id>/values.xml?<filter>. Параметры определяют дополнительные фильтры на данные:

– time_begin – дата начала выборки (YYYY-MM-DD HH:mm:ss);

– time_end – дата окончания выборки;

– extent – область выборки (xmin,ymin, xmax, ymax);

– sites – площадки (site_id, site_id,…);

– indicators – показатели (indicator_code, indicator_code, …);

– limit – максимальное число возвращаемых записей (по-умолчанию, 1000);

– offset – смещение в возвращаемых записях (по-умолчанию, 0);

– srid – код проекции (по-умолчанию, 4326)

– fields - список необходимых полей в возвращаемом массиве (по-умолчанию, все);

4. Получить агрегированные данные. GET http://host/api/1.0/project/<project_id>/aggvalues.xml?<filter>. Параметры практически аналогичны параметрам предыдущего запроса, добавляется только: time_interval – интервал агрегации данных (1week, 1day, 1hour, 5min).

5. *Отправить текущие данные*. POST http://host/api/1.0/project/<project_id>/current_values.xml. За один запрос можно передать текущие значения показателей для разных площадок проекта.

6. *Отправить архивные данные.* POST http://host/api/1.0/project/<project_id>/values.xml. Функция позволяет загрузить массив данных по нескольким площадкам за прошлые периоды времени. Для каждого замера явно указывается дата-время. Если в базе уже есть данные с указанными параметрами (site, indicator, date), то они заменяются новыми.

7. *Удалить массив данных.* DELETE http://host/api/1.0/project/<project_id>/values.xml?<filter>. Параметры аналогичны параметрам запроса на выборку данных. Удаляются все данные, попавшие в выборку.

Реализация

На текущий момент создан прототип сервера сбора и хранения данных мониторинга окружающей среды. Реализованы основные функции программного интерфейса (API). Веб-интерфейс находится в стации разработки.

Созданы несколько потоков данных из разных типов источников – открытых источников климатических данных, автономных метеостанций, систем мониторинга за состоянием окружающей природной среды.

1. Реализована простая метеостанция на основе аппаратно-вычислительной платформы Arduino. Использованы такие компоненты, как Arduino Uno, Ethernet Shield W5100, сенсор BMP085. На языке программирования Processing написана программа для микроконтроллера ATmega328, обеспечивающая опрос сенсоров и передачу данных на сервер по сети Ethernet. На текущий момент измеряется температура и атмосферное давление, в дальнейшем количество измеряемых параметров можно расширять.

2. Выполнена загрузка ежедневных сводок глобального покрытия (ftp://ftp.ncdc.noaa.gov/pub/data/gsod/) от Национального климатического центра США. GSOD ежедневно пополняется данными с метеорологических стаций всего мира, в том числе – с российских метеостанций. Эти данные представляют собой отчеты, которые содержат информацию о значении температуры, влажности воздуха, скорости и направлении ветра, атмосферного давления, типах осадков и т.п.

3. Выполнена загрузка данных из ГИС мониторинга состояния окружающей природной среды в зоне действия предприятий нефтегазовой отрасли [4], где собраны результаты проведения КХА за 2008-2010 годы по отобранным пробам в районах размещения объектов НГО (94 показателя, 7 площадок, ~15 тыс. замеров).

4. Реализован механизм подключения к «Краевой системе наблюдений за состоянием окружающей среды» поддерживаемой Центром реализации мероприятий по природопользованию и охране окружающей среды Красноярского края (КГБУ «ЦРМПиООС»). С помощью адаптера выполняется периодический сбор и загрузка данных по загрязнению атмосферного воздуха (6 постов наблюдений) и поверхностных вод (21 пост) в Красноярском крае.

Ведутся работы по созданию средств эффективного доступа к данным наблюдений, сервисов представления и анализа информации. Планируется интеграция с геоинформационным порталом СО РАН (http://gis.krasn.ru).

Литература.

1. Токарев А.В. Разработка подсистемы сбора и хранения данных мониторинга окружающей среды // Тезисы докладов II Российско-монгольской конференции молодых

ученых по математическому моделированию, вычислительно-информационным технологиям и управлению. – Иркутск: РИО ИДСТУ СО РАН, 2013. – С. 58.

2. О.Э. Якубайлик, А.А. Гостева, М.Г. Ерунова, А.А. Кадочников, А.Г. Матвеев, А.С. Пятаев, А.В. Токарев. Разработка средств информационной поддержки наблюдений за состоянием окружающей природной среды // Вестник КемГУ. - 2012. - № 4(52) Т. 2 - С. 135-141.

3. Pautasso, C.; Zimmermann, O.; Leymann, F. RESTful Web Services vs. Big Web Services: Making the Right Architectural Decision // 17th International World Wide Web Conference (WWW2008). – Beijing, China, 2008. – P. 805-814.

4. Roy T. Fielding and Richard N. Taylor. Principled design of the modern Web architecture. – ACM Transactions on Internet Technology (TOIT), 2(2), May 2002. – pp. 115-150.