

на основе метода расчета динамики машинного агрегата с жесткими звеньями, разработанного М. З. Коловским.

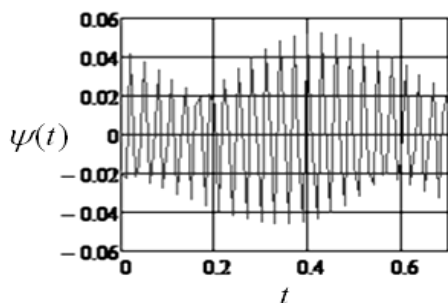


Рис. 5. Динамическая ошибка скорости выходного звена ВРП

В связи с линейной характеристикой скорости рабочего органа в возмущающем воздействии системы привода ВРП приведенный момент активных сил сопротивления $T_c(q, \dot{q})$ в зависимости (1) следует рассматривать как величину постоянную.

В приведенном значении момента инерции передачи следует учитывать его изменение в связи с цикличностью входа в зацепление очередного толкателя, а также частоту возмущающего воздействия от переменного момента инерции – в Z раз превышающую частоту вращения кулачкового вала, где Z – количество кулачковых механизмов в передаче.

Библиографические ссылки

1. Лимаренко Г. Н. Реечная передача, работающая по принципу бегущей волны // Теория машин металлургического и горного оборудования : межвуз. сб. Свердловск, 1985. Вып. 9. С. 94–99.
2. Коловский М. З. Динамика машин. Л. : Машиностроение, 1989.
3. Конструирование машин : справ.-метод. пособие : в 2 т. Т. 1 / К. Ф. Фролов, А. Ф. Крайнев, Г. В. Крейнин и др. ; под общ. ред. К. Ф. Фролова. М. : Машиностроение, 1994.
4. Механика машин : учеб. пособие для вузов / И. И. Вульфсон, М. Л. Ерихов, М. З. Коловский и др. ; под ред. Г. А. Смирнова. М. : Высш. шк., 1996.

G. N. Limarenko, M. V. Shevchugov, A. N. Schepin

TO DYNAMIC ANALYSIS OF STEADY MOTION OF AN OUTPUT ELEMENT IN WAVE RACK AND PINION DRIVE

The authors present the results of theoretical researches and method of analysis in forward drive with harmonic rack-and-pinion drive of irregularity of steady motion of its rolling strip, namely, outlet link, which interacts with multicam mechanism.

Keywords: pusher, diagram, phase shift, equivalent moment of inertia, velocity dynamic error.

© Лимаренко Г. Н., Шевчугов М. В., Щепин А. Н., 2012

УДК 519.6+004.4

А. Г. Матвеев, О. Э. Якубайлик

РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ОБРАБОТКИ И ПРЕДСТАВЛЕНИЯ ПРОСТРАНСТВЕННЫХ МЕТАДААННЫХ ГЕОПОРТАЛА

Анализируются технологии современного веб-программирования, основанные на использовании средств JavaScript/AJAX/CSS/PHP. Рассматриваются функциональные возможности ряда популярных фреймворков веб-приложений. Обсуждаются технические характеристики разработанного веб-приложения для работы с пространственными метаданными геопортала.

Ключевые слова: веб-приложение, геопортал, JavaScript-фреймворк, пространственные метаданные, AJAX, JSON.

Непрерывное развитие технологий и технических характеристик сети Интернет в течение двух десятилетий ее существования обеспечило плавный переход в способе представления информации – от тривиальной презентации простых статичных гипертекстовых документов до сложнейших интерактивных мультимедийных информационных систем. Методология

программирования для Интернета постоянно совершенствуется, нарастающими темпами возникают новые средства разработки.

В данной статье мы будем обсуждать архитектуру и методы построения распределенных приложений для сети Интернет, основанных на применении современных средств веб-программирования, методики

разработки приложений с использованием так называемых фреймворков для JavaScript/AJAX и других популярных сегодня средств разработки программного обеспечения.

Выбор JavaScript-фреймворка для веб-приложения. Разнообразии современных веб-приложений и необходимость разработки эффективных пользовательских интерфейсов, функциональные возможности которых растут с каждым днём, а также необходимость поддержки их стабильной работы не только в последних версиях популярных веб-браузеров, но и в устаревших (но при этом ещё используемых) версиях – все это делает разработку на JavaScript без использования вспомогательных средств чрезвычайно неэффективной и утомительной. До сих пор существует проблема различного поведения разных типов браузеров (так называемых движков – Trident, Gecko, WebKit, Presto) в области интерпретации JavaScript, поддержки объектной модели документа (DOM) и каскадных таблиц стилей (CSS).

Начиная с середины 2000-х гг. среди разработчиков клиентской части веб-приложений все популярнее становятся так называемые JavaScript-фреймворки. *JavaScript-фреймворк (или библиотека JavaScript)* – это набор функций и средств, облегчающих реализацию сценариев JavaScript, совместимых с большинством современных браузеров. Каждая из библиотек в процессе своего развития проходит активное тестирование в различных веб-браузерах, и это в определенной степени гарантирует одинаковую работу веб-приложений на основе этих библиотек в различных типах браузеров (кроссбраузерность) [1].

Рассмотрим основные функциональные возможности типичного современного JavaScript-фреймворка.

Операции с элементами объектной модели документа (DOM). Как правило, в основе библиотек содержится реализация функций поиска и выборки элементов DOM с использованием селекторов CSS (Cascading Style Sheets, каскадные таблицы стилей) или XPath (XML Path Language). Данный функционал позволяет получать ссылки на элементы модели документа согласно идентификаторам этих элементов, их классов, значений атрибутов, а также с использованием всевозможных комбинаций этих свойств.

Кроме выборки элементов с использованием селекторов, JavaScript-библиотеки зачастую предоставляют функционал поиска и выборки элементов DOM согласно их относительной позиции в дереве документа, т. е. с использованием следующих отношений: родительский элемент, дочерний элемент, элементы-предки, элементы-потомки и соседние элементы.

Естественным продолжением развития функционала фреймворков JavaScript, связанного с поиском и выборкой элементов, является реализация функций, предназначенных для манипуляций с моделью DOM. Это функции изменения значения элементов, их атрибутов, классов CSS, содержимого; их перемещения, копирования, удаления и т. д. Использование функций, связанных с объектной моделью документа,

является одним из основных неоспоримых преимуществ по сравнению с реализацией кода на «чистом» JavaScript, поскольку реализация DOM до сих пор не является полной в различных версиях и типах браузеров.

Поддержка асинхронных HTTP-запросов (AJAX). Использование концепции AJAX (Asynchronous JavaScript and XML) уже не раз доказало свою эффективность при реализации веб-приложений. Подобное объединение технологий динамического обращения к серверу (чаще всего при помощи объекта XMLHttpRequest) и DHTML (Dynamic HTML) для организации интерактивных веб-интерфейсов обладает следующими преимуществами: оно уменьшает нагрузку на сервер и дает возможность экономить трафик за счёт существенного уменьшения объёмов передаваемых данных и переноса части нагрузки на сторону веб-клиента, а также повышает интерактивность пользовательских интерфейсов и их ускорение.

Функционал для работы с AJAX в JavaScript-фреймворках представляет собой набор удобных в использовании методов, позволяющих выбрать тип ответа (XML, JSON, HTML или текст), назначить функции, запускаемые при успешном и неудачном запросе, выбрать метод запроса (GET или POST), настроить синхронную передачу данных или асинхронную и т. д. Возможности конфигурирования отличаются у различных библиотек, но все они облегчают работу с AJAX, существенно сокращая объёмы кода, увеличивая его читаемость и прозрачность, а также избавляют от необходимых проверок на совместимость с текущим браузером, поскольку, например, браузер Internet Explorer версии 6 использует ActiveX вместо объекта XMLHttpRequest. Кроме того, существуют различия в реализации XMLHttpRequest в других типах браузеров.

Обработка событий. Традиционный способ назначения событий браузера путём внедрения в HTML-код элемента специальных атрибутов является очень неэффективным, в особенности учитывая функциональную сложность интерфейсов современных веб-приложений. Кроме того, отчасти существует проблема кроссбраузерности. Практически во всех современных JavaScript-фреймворках реализован собственный функционал обработки событий, который является простой и удобной альтернативой стандартному способу.

Помимо непосредственного назначения обработчиков событий клиента (таких как клик мышью по элементу, двойной клик, наведение курсора, перемещение мыши и т. д.), элементам, уже созданным в модели документа, современные фреймворки позволяют осуществлять делегирование событий. Делегирование событий в JavaScript – это способ назначения элементу DOM обработки события путём передачи управления от обработчика родительского элемента. Иными словами, функции-обработчики назначаются не самим элементам, а их родителям, что приводит к благоприятным последствиям:

- в коде становится меньше функций, которыми нужно оперировать;
- занимает меньше памяти;
- уменьшается количество связей между JavaScript-кодом и DOM;
- исчезает необходимость удаления ненужных событий при изменении DOM;
- становится возможным сквозное добавление событий к элементам DOM, создаваемым на лету.

Визуальные эффекты. Сочетание работы с таймерами JavaScript и стилями CSS позволяет реализовывать множество визуальных эффектов, таких как анимированное движение элементов, изменение их пропорций, смена цвета, поддержка прозрачности и т. д. Такого рода функционал играет важную роль при реализации интерактивных пользовательских интерфейсов веб-приложений.

Служебные функции. Практически каждый JavaScript-фреймворк обладает набором вспомогательных (служебных) функций, которые призваны облегчить разработку. Часто среди таких функций можно увидеть:

- определение «движка» и версии браузера;
- обход массива с выполнением функции на каждой итерации;
- поиск значения в массиве или объекте;
- объединение массивов или объектов;
- обработку строк (синтаксический анализ и пр.) и др.

Помимо описанных выше возможностей, библиотеки JavaScript часто имеют возможность расширения посредством модулей или плагинов. Наряду с расширениями от разработчиков библиотек, существует огромное количество сторонних проектов. Поскольку большинство из JavaScript-фреймворков поставляется с открытым исходным кодом по лицензии свободного/открытого ПО (BSD, GPL или MIT), с течением времени вокруг этих продуктов обычно возникают сообщества активных веб-разработчиков, многие из которых создают собственные модули и выкладывают их в свободный доступ.

В целом использование библиотек JavaScript дает огромные преимущества при разработке веб-приложений. Среди очевидных плюсов – уменьшение объема кода, повышение удобочитаемости, поддерж-

ка кроссбраузерности. В дополнение к перечисленному, разработчик получает возможность правильно структурировать программный код, так как необходимость физически смешивать HTML и JavaScript практически исчезает.

Рассмотрим наиболее популярные на сегодняшний день JavaScript-фреймворки. К ним относятся Dojo, Mootools, YUI, ExtJS, JQuery (см. таблицу).

Каждый из фреймворков является мощным инструментом для веб-разработчика, со своими положительными и отрицательными сторонами. Все они обладают поддержкой рассмотренного набора функциональных возможностей – асинхронной передачи данных AJAX, обработки событий и пр. Помимо базовой функциональности, которая в общих чертах совпадает у всех перечисленных библиотек, рассматриваемые JavaScript-фреймворки отличаются наличием и возможностями подключаемых модулей и элементов веб-интерфейса (виджетов). Например, ExtJS обладает мощным набором функций для создания графического пользовательского интерфейса веб-приложения, так называемого оконного интерфейса на веб-странице, в стиле настольной операционной системы.

Самым популярным фреймворком, согласно статистике Google Trends, является JQuery, причем его отрыв от конкурентов все больше растет, начиная с 2008 г. Популярность JQuery определяется тем, что, с одной стороны, данный фреймворк является превосходным средством для решения повседневных задач, связанных с манипуляциями с объектами DOM, обработкой событий, реализацией визуальных эффектов, работой с AJAX и т. д., и, с другой стороны, обладает довольно высокой производительностью, малым размером, хорошей документацией, и как следствие – низким порогом вхождения. Возможность расширения при помощи официальных и авторских модулей в совокупности с активным сообществом разработчиков, интеграция данного фреймворка в популярные системы управления веб-контентом (Drupal, Wordpress, TikiWiki), а также библиотека JQuery UI, содержащая набор примитивов графического интерфейса пользователя – все это сделало данный продукт очень качественным универсальным инструментом веб-разработчика [2].

Сравнительные характеристики JavaScript-фреймворков

Фреймворки	Dojo	Mootools	YUI	ExtJS	JQuery
Версия	1.6.0 (15.03.2011)	1.3.1 (24.02.2011)	3.0 (29.09.2009)	4 (26.04.2011)	1.6 (3.05.2011)
Размер	28 КБ (gzip), 123 КБ (исх.)	7.3...65 КБ 101 КБ (исх.)	Ядро ≈ 31 КБ	84...502 КБ	24 КБ (gzip), 155 КБ (исх.)
Лицензия	BSD и AFL	MIT	BSD	Коммерческая и GPL v3	MIT и GPL
Поддержка браузеров					
IE	6+	6+	6+	6+	6+
Firefox	3+	2+	3+	1,5+	2+
Safari	4	3+	4,0	3+	3+
Opera	10	9+	10,0+	9+	9+
Chrome	3	1+	Нет данных	1+	1+

Однако стоит также отметить, что выбор JavaScript-фреймворка – сложный вопрос, который должен быть рассмотрен в контексте решаемых задач. Иногда целесообразно использовать сразу несколько подобных инструментов.

Веб-приложение для работы с пространственными метаданными геопортала. Рассмотренный выше анализ технологий разработки веб-приложений стал составной частью исследований, направленных на формирование программно-технологической основы геопортала ИВМ СО РАН – распределенной геоинформационной системы, предназначенной для организации хранения, доступа к географической информации, навигации и поиска по геопространственным данным, связанным с ними географическим сервисам (для доступа, редактирования, анализа и т. д.) посредством сети Интернет. Предполагается, что основной задачей геопортала ИВМ СО РАН является поддержка научных исследований, обеспечение информационного взаимодействия специалистов, занимающихся обработкой и формированием тематических картографических, пространственных данных [3].

Будем рассматривать конкретную задачу – разработку клиентской части каталога пространственных метаданных геопортала (подсистема «Каталог метаданных»). Предполагается создание технологической платформы веб-приложения, которая должна быть достаточно гибкой, расширяемой в процессе эксплуатации и при этом обладающей рядом весьма специфичных характеристик. Проведенный анализ возможного использования различных фреймворков показал, что такие популярные системы, как Joomla, Drupal и Wordpress нецелесообразно рассматривать в качестве базовых средств для решения данной задачи – они по своему основному целевому назначению далеки от систем каталогизации, и, соответственно, излишне громоздки, так как перегружены не нужным в нашем случае функционалом. Основные требования, предъявляемые к платформе веб-приложения, – скорость, гибкость, расширяемость, простота установки, независимость представления от содержания, поддержка асинхронных запросов. Указанному перечню вполне соответствует JavaScript-фреймворк JQuery. В дополнение к нему необходимо выполнить разработку компонентов программного обеспечения с помощью таких языков и технологий программирования, как PHP, XHTML, CSS, JavaScript, AJAX, JSON и пр.

Применение в основе серверной части приложения шаблона проектирования MVC (модель–представление–контроллер) предоставляет широкие возможности для решения поставленных задач. Использование этой архитектуры предполагает разделение данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: модель, представление и контроллер. Модификация каждого компонента может осуществляться независимо. Помимо стандартных элементов MVC, ключевыми блоками рассматриваемой системы также являются шаблоны представления страниц и AJAX-

обработчики. Рассмотрим эти элементы веб-приложения.

Модель (Model). Каждый раздел интернет-сайта, построенного на основе описываемой платформы, связан с одной или несколькими, иерархично расположенными, моделями, основной задачей которых является реализация бизнес-логики приложения. Модели, в зависимости от параметров запроса (которые предоставляются контроллером системы), осуществляют следующие действия:

- получение или изменение данных предметной области;
- определение представления (View) и шаблона представления страницы, соответствующего параметрам запроса;
- подготовка выходного набора данных для использования в представлении (View).

Представление (View). Отвечает за отображение данных предметной области (предоставляемых моделью). Один раздел сайта может иметь несколько различных отображений, изменяемых в зависимости от контекста. Например, представление list содержит список записей, содержащихся в таблице, а представление edit содержит форму для создания и (или) редактирования этих записей. Разделение логики предметной области и отображения данных является безусловным преимуществом шаблона MVC и практически необходимым условием при реализации многофункциональных веб-приложений.

Контроллер (Controller). Является связующим звеном между прочими компонентами системы. Практически все HTTP-запросы перенаправляются для обработки контроллеру. Основными задачами контроллера являются:

- подключение настроек, внутренних и внешних библиотек;
- приём и анализ HTTP-запросов;
- выбор на основе параметров запроса необходимой модели и передача ей управления;
- подключение необходимого шаблона представления страницы;
- формирование заголовков HTTP и вывод данных.

Шаблон представления страницы. Формально является составной частью представления (View), но фактически это самостоятельный элемент системы. Средний по функциональности веб-сайт имеет 1–3 шаблона страницы, каждый из которых представляет собой подключаемый к контроллеру PHP-сценарий, содержащий общие для используемых представлений элементы (X)HTML-кода страницы – заголовочную секцию, ссылки на необходимые файлы каскадных таблиц стилей и сценариев JavaScript, метаполя и общие элементы DOM. Выбираемый моделью файл представления затем встраивается в указанный шаблон и в результате генерируется полный (X)HTML-код веб-страницы.

AJAX-обработчик. Представляет собой модель, предназначенную для работы с асинхронными запросами.

Модели обработки HTTP-запросов веб-приложения. Большинство HTTP-запросов, идущих от браузера к основному URI системы или любому из его дочерних путей, средствами веб-сервера перенаправляются для обработки контроллеру, например, при помощи модуля `mod_rewrite` сервера Apache. Исключение составляют запросы файлов изображений, стилей CSS и JavaScript-сценариев. Контроллер при запуске подключает внешние и внутренние библиотеки, файлы конфигурации, осуществляет запуск сессии. Затем при помощи одной из библиотек производится анализ запроса, определение его типа (стандартный или асинхронный), поиск запрашиваемой модели и передача ей управления.

Модель, используя уже проанализированный запрос, производит запрашиваемые в параметрах запроса действия с данными. В клиентской части каталога метаданных геопортала в основном производятся операции поиска и фильтрации ресурсов. При этом промежуточные вычисления, полученные в результате разнообразных выборок, подсчёта количества ресурсов и т. д., сохраняются во временном кэше (на время текущей сессии пользователя) и восстанавливаются для повторного использования, что значительно снижает нагрузку на сервер и увеличивает быстроедействие клиентской части.

Помимо получения данных, модель определяет, какое представление и какой шаблон представления страницы должны использоваться при визуализации результатов. Кроме того, модель выполняет перенаправления на другие разделы веб-сайта, если этого требует бизнес-логика приложения. Если же перена-

правления на другой раздел не произошло, то управление опять возвращается к контроллеру, который подключает выбранный шаблон представления (либо шаблон по умолчанию), а тот в свою очередь подключает выбранный сценарий представления. Результатом этой цепочки событий является полный XHTML-код страницы (рис. 1).

В качестве формата сообщений, передаваемых от сервера клиенту, используется также хорошо зарекомендовавший себя стандарт JSON (JavaScript Object Notation) – текстовый формат представления структурированных данных, базирующийся на подмножестве стандарта ECMA-262. Преимущество использования данного стандарта перед XML в случае клиентской части веб-приложения связано с тем, что код в JSON-формате представляет собой текст на JavaScript, т. е. не требуется дополнительных усилий для его трансляции. Поддержка данной концепции в архитектуре веб-приложения «Каталог метаданных» осуществляется через библиотеку JQuery и вспомогательную авторскую библиотеку функций, упрощающую работу с асинхронной передачей данных в технологии AJAX [4].

При помощи асинхронных запросов производятся следующие операции:

- подсчёт количества ресурсов в категориях фильтрации;
- подсчёт количества ожидаемых результатов поиска;
- перемещение ресурса в корзину;
- загрузка метаданных ресурса и пр.

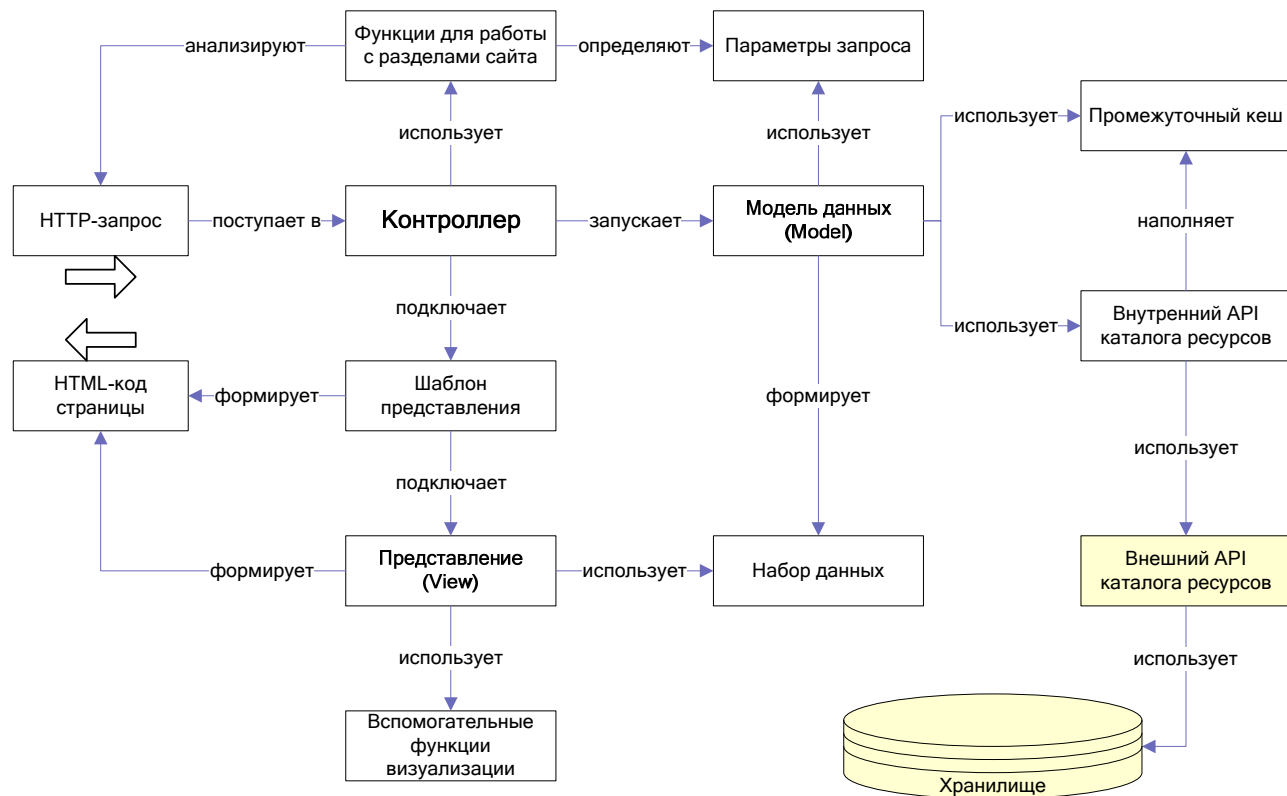


Рис. 1. Модель обработки стандартного HTTP-запроса.

Модель обработки асинхронного запроса (рис. 2) похожа на модель обработки рассмотренного выше стандартного запроса HTTP (см. рис. 1), за исключением того, что вместо компонента MVC Модель (Model) используется AJAX-обработчик, а для вывода не используется представление (View) и шаблон представления страниц, поскольку данные конвертируются в JSON-формат средствами PHP.

Рассмотренные технологии легли в основу выполненной разработки – веб-интерфейса клиентского приложения системы управления пространственными метаданными «Каталог метаданных» геопортала ИВМ СО РАН (рис. 3). Задачей этого веб-приложения является навигация по каталогу пространственных метаданных, с поиском и фильтрацией данных по различным критериям.

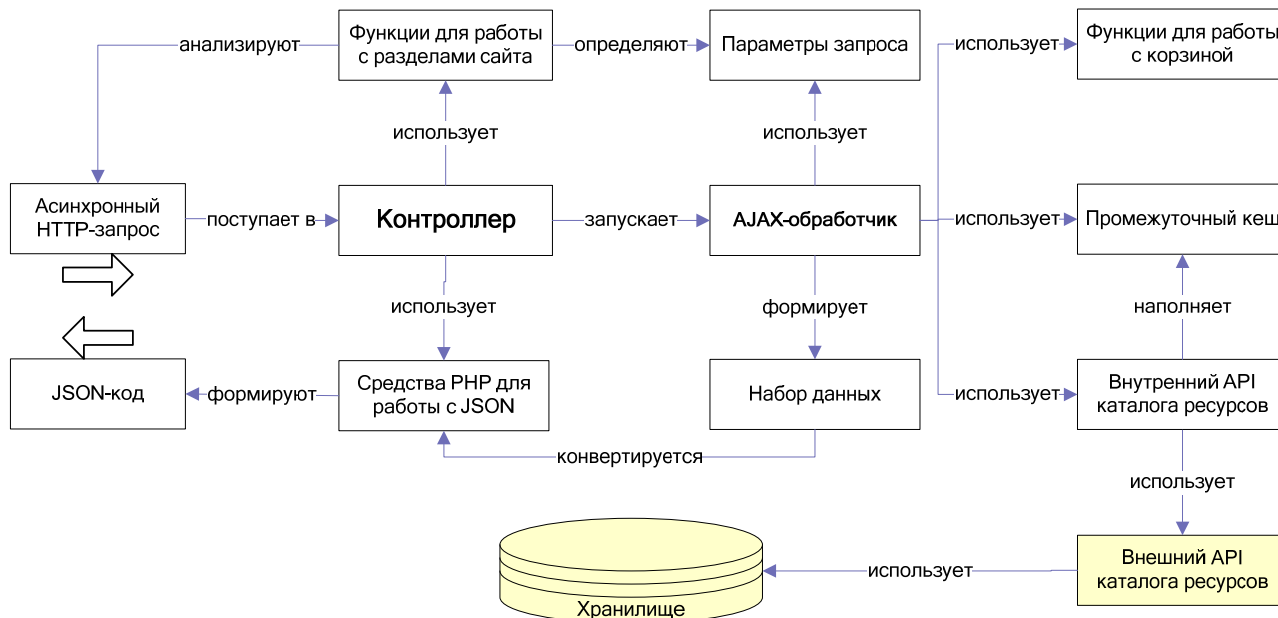


Рис. 2. Модель обработки асинхронного HTTP-запроса

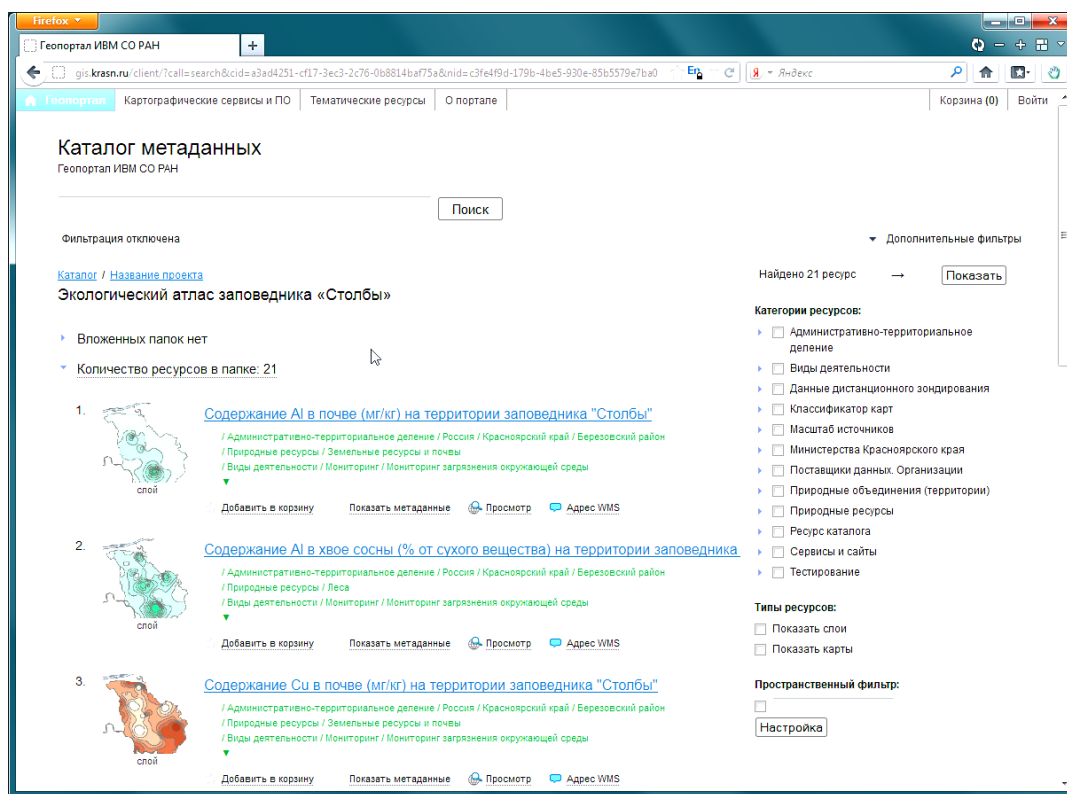


Рис. 3. Интерфейс веб-приложения «Каталог метаданных» геопортала ИВМ СО РАН

По принципам организации и внешнему виду выполненная разработка похожа на многие популярные системы каталогизации в Интернете – Яндекс.Каталог, Товары@Mail.Ru, Yahoo! Directory и пр. Отметим некоторые ее отличительные характеристики:

– основанность на принципах множественной (фасетной) классификации, благодаря которой ресурсы каталога группируются по разным логическим иерархиям, с динамическим подсчетом различных характеристик ресурсов, сгруппированных по выбранным разделам и подразделам;

– развитый пользовательский интерфейс для представления метаданных – динамически формируемые сведения о выбранном разделе каталога (количество ресурсов, категории фильтрации данных, режимы отображения информации из вложенных разделов и т. д.) и конкретном ресурсе (информация структурируется по подразделам через систему информационных страниц, размещаемых на разных раскрывающихся вкладках), набор ссылок для прямого доступа к данным (WMS -сервисы, zip-файлы с архивами и т. п.);

– средства полнотекстового поиска и фильтрации ресурсов по метаданным (названию, описанию, автором и пр.), а также по пространственным локализациям данных (выделением области поиска на интерактивной карте);

– просмотр картографических ресурсов каталога двумя способами – либо при помощи встроенной системы картографической веб-визуализации, либо

с помощью сторонних приложений, через WMS-протокол;

– средства персонализации интерфейса пользователя – возможность выбора и переноса ресурсов каталога метаданных во временное хранилище («Корзину») или сохранения в постоянное хранилище («Мои документы»).

В планах разработчиков – дальнейшее развитие веб-приложения «Каталог метаданных».

Библиографические ссылки

1. Якубайлик О. Э., Попов В. Г. Технологии для геоинформационных интернет-систем // Вычислит. технологии. 2009. Т. 14. № 6. С. 116–126.

2. Модель геоинформационной аналитической интернет-системы для анализа состояния и презентации региона / О. Э. Якубайлик, А. А. Кадочников, В. Г. Попов, А. В. Токарев // Вестник СибГАУ. 2009. Вып. 4. С. 61–66.

3. Формирование геоинформационного интернет-портала для задач мониторинга состояния природной среды и ресурсов / А. А. Кадочников, В. Г. Попов, А. А. Токарев, О. Э. Якубайлик // Журн. Сиб. федер. ун-та. Сер. Техника и технологии. 2008. Т. 1. № 4. С. 377–386.

4. Токарев А. В., Якубайлик О. Э. Каталог ресурсов для ГИС мониторинга состояния окружающей природной среды в зоне действия предприятий нефтегазовой отрасли // Горный информ.-аналит. бюл. 2009. Т. 18. № 12. С. 215–219.

A. G. Matveev, O. E. Yakubailik

DEVELOPMENT OF WEB-APPLICATIONS FOR THE PROCESSING AND PRESENTATION OF GEOPORTAL SPATIAL METADATA

The paper analyzes the technologies of modern web programming, based on the use of JavaScript/AJAX/CSS/PHP. The functional capabilities of a number of popular web application frameworks are considered. The specifications of the developed web application for spatial metadata processing are discussed.

Keywords: web application, geoportal, JavaScript framework, spatial metadata, AJAX, JSON.

© Матвеев А. Г., Якубайлик О. Э., 2012