

УДК 004.42:519.688:332.05

И.Г. Митченков, М.И. Баумгартэн, В.Г. Михайлов,  
А.А. Тайлакова, Т.В. Сарапулова

## ИСПОЛЬЗОВАНИЕ ВЕБ-ТЕХНОЛОГИЙ ДЛЯ РЕАЛИЗАЦИИ МЕТОДИКИ ОЦЕНИВАНИЯ ЭКОЛОГИЧЕСКИХ ПРОБЛЕМ

В условиях теоретической неясности научное прогнозирование политики в сфере взаимодействия общества и окружающей среды (экологической политики) существенно усложняется и задача исследователей скорее сводится к тому, чтобы по возможности адекватно оценивать текущую (реальную) практику в сфере природоохранной деятельности и природопользования. Решение данной проблемы может быть связано с построением системы экологического мониторинга, наличие которой позволит, с одной стороны дать обобщающие характеристики и проанализировать тенденции (в том числе и долгосрочные) в данной сфере деятельности, а с другой стороны, отработать методику исследования.

Современное общество осознает необратимый и катастрофический характер экологической ситуации и пытается реализовать определенные мероприятия для предотвращения негативных последствий потребительского отношения к природе. При этом, чаще всего, разрешение экологических противоречий остается либо на уровне деклараций, либо реализуется в «режиме ручного управления» волей крупных руководителей или политических лидеров. Требуются многоуровневые, системные исследования, основанные на общемировых разработках и ориентированные на особенности конкретного региона.

Кузбасс является динамично развивающимся регионом, но традиционно на его территории существовала сложная экологическая ситуация. В последние годы региональные власти много внимания уделяют экологической безопасности, но все же меры по решению экосоциальных противоречий требуют более глубокой систематической концептуальной проработки.

Экологическая оценка территории проводится с целью идентификации и ранжирования основных экологических проблем, характерных для исследуемой территории. Коллективом авторов (Митченков И.Г., Баумгартэн М.И., Михайлов В.Г.) предложена методика для оценки экологических проблем. Её показатели несут разную смысловую нагрузку: одни являются результирующими, другие представляют собой переменные, объясняющими этот результат.

В качестве результирующих переменных используются:

- выбросы в атмосферу наиболее распространенных загрязняющих веществ, отходящих от стационарных источников (тыс. тонн);
- отходы производства и потребления (млн. тонн);

тонн);

- объем сброса сточных вод, имеющих загрязняющие вещества (млн. м<sup>3</sup>);
- заболеваемость населения (тыс. чел.).

Первые три показателя являются экологическими показателями, характеристиками состояния окружающей среды, четвертый – показатель-реципиент, показывающий воздействие окружающей среды на здоровье человека.

В качестве объясняющих переменных при анализе можно использовать экономические показатели, такие как:

- объем добычи угля (млн. тонн);
- объем производства металлургической продукции (тыс. тонн);
- потребление электроэнергии (млрд. кВт·час);
- наличие транспортных средств (тыс. шт.) и др.

Для описания взаимосвязи факторов естественно использовать многофакторные модели  $y=f(x_1 \dots x_n)$ . Так как данные для анализа представляют собой различные по своей природе показатели, нами выбрана мультиплексивная форма представления функции (произведение степенных функций от переменных). Для оценки параметров модели используется метод регрессионного анализа.

В первую очередь исследуются зависимости между переменными 1-го уровня:

- объем добычи угля (млн. тонн);
  - наличие транспортных средств (тыс. шт.);
  - потребление электроэнергии (млрд. кВт·час);
  - объем производства металла (тыс. тонн);
  - численность населения (тыс. чел.);
  - объем с/х производства (млн. руб.);
  - использование свежей воды на хозяйственно-питьевые нужды (млн. м<sup>3</sup>);
  - использование свежей воды на производственные нужды (млн. м<sup>3</sup>);
- и затем 2-го уровня.
- выбросы в атмосферу (тыс. тонн);
  - отходы производства и обращения (млн. тонн);
  - объем сброса сточных вод (млн. м<sup>3</sup>).

Созданная модель является условием и началом разработки комплексной системы регионального экологического мониторинга (КСЭМ). Для создания КСЭМ, в качестве задач на следующие этапы работы, требуется создать критерии оценки региональных экологических проблем и ситуаций; выбрать и адаптировать модель к экологической и климатической ситуации Кемеровской области;

выявить основные тенденции правового регулирования в сфере природоохранной деятельности и природопользования. Реализация КСЭМ позволит определить основные принципы инновационной экологической политики и механизмы ее реализации в Кемеровской области.

Сложность расчетов, большой объем справочной информации требуют применения средств автоматизации, а отсутствие доступных специализированных программ вызывает необходимость разработки собственного приложения, учитывающего специфику поставленной задачи.

В процессе разработки программного комплекса необходимо решить следующие задачи.

1) Разработать интерфейс программы, обеспечивающий диалог при удобном вводе исходных данных.

2) Реализовать оценку эколого-экономических взаимодействий на основе разработанной модели, которая включает оценку зависимости:

- выбросов в атмосферу от экономических факторов;
- отходов производства и потребления от экономических факторов;
- объема сброса загрязненных сточных вод от экономических факторов;
- здоровья населения региона от экологических факторов.

3) Организовать вывод результатов вычислений на экран в виде таблиц.

4) Организовать экспорт результатов расчетов в среду MS Excel.

5) Визуализировать результаты вычислений в виде графиков.

6) Организовать хранение результатов вычислений в базе данных.

В качестве технологии разработки программного комплекса предлагается выбрать веб-технологии.

Важное значение в динамичном развитии информационных технологий имеют веб-приложения. Сегодня приложения этого вида стали такими же сложными программными продуктами, как и обычные десктоп-приложения [1].

Веб-приложения – это программы, предназначенные для автоматизированного выполнения каких-либо задач на веб-серверах и использующие в качестве программы клиента интернет – браузеры. Данные хранятся на сервере, а обмен информацией между сервером и клиентом происходит по сети.

Веб-приложения не требуют установки на компьютер заказчика объемного программного обеспечения (для полноценной работы нужен только браузер и доступ в интернет).

Веб-разработки не требуют специальной настройки и администрирования, их администраторами являются разработчики.

Обновление веб-приложений происходит автоматически. Они обеспечивают высокую мо-

бильность при условии доступа в интернет [2].

Существуют инструменты для создания таких приложений, шаблоны проектирования, освоенные технологии.

Представление передаваемой по сети информации в виде контента осуществляется средствами языка гипертекстовой разметки HTML и технологии каскадных таблиц стилей CSS. Новый стандарт HTML5 и анимация CSS3 позволяют делать приложение динамическим, реагирующими на действия пользователя. HTML5, появившийся не так давно, представляет веб-разработчикам отличные инструменты для интересного отображения как текстовой, так и графической информации. Однако, он еще не настолько распространен и поддерживается даже самыми современными браузерами не полностью. Для создания интерактивных веб-страниц часто применяют язык программирования JavaScript, который дает возможность значительно снизить нагрузку на веб-сервер за счёт того, что основные сценарии для работы со страницей будут выполняться на стороне пользователя. Такие скрипты применяют для создания динамических страниц, для написания программ, например, легкой браузерной игры или калькулятора. Большинство сервисов и сайтов применяют JavaScript для определения браузера, операционной системы и т. д., чтобы более корректно отобразить страницу на конкретном устройстве. Особенно это актуально для обозревателя Internet Explorer, который иначе интерпретирует стандарты HTML и CSS [3].

Существует большое количество подключаемых библиотек JavaScript. Популярная библиотека jQuery фокусируется на взаимодействии JavaScript, HTML и CSS. Она позволяет обращаться к любому элементу DOM (объектной модели документа) и манипулировать им; работать с событиями; легко осуществлять различные визуальные эффекты; работать с AJAX (технология, позволяющая общаться с сервером без перезагрузки страницы); имеет огромное количество JavaScript плагинов, предназначенных для создания элементов пользовательских интерфейсов [4].

Для разработки сценариев, выполняемых на стороне сервера, можно использовать язык сценариев общего назначения с открытым исходным кодом PHP. PHP позволяет создавать качественные веб-приложения за очень короткие сроки, получая продукты, легко модифицируемые и поддерживаемые в будущем [5].

Решением проблемы хранения данных являются реляционные системы управления базами данных. MySQL – компактный многопоточный сервер баз данных. MySQL характеризуется большой скоростью, устойчивостью и легкостью в использовании. MySQL поддерживает язык запросов SQL в стандарте ANSI 92, и кроме этого имеет множество расширений к этому стандарту, которых нет ни в одной другой СУБД. MySQL под-

держивает неограниченное количество пользователей, одновременно работающих с базой данных, допускает до 50 млн строк в таблицах, обеспечивает быстрое выполнение команд и имеет простую и эффективную систему безопасности [6].

Использование реляционной базы данных для хранения объектно-ориентированных данных приводит к семантическому провалу, заставляя программистов писать программное обеспечение, которое должно обрабатывать данные в объектно-ориентированном виде и уметь сохранить эти данные в реляционной форме. Для решения этой задачи применяют технологию ORM. ORM — технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных» [7].

На сегодняшний день наиболее часто используемым шаблоном проектирования веб-приложений является Модель-Вид-Контроллер (MVC). Сейчас MVC является практически синонимом веб-разработки среди всех платформ.

Шаблон MVC описывает простой способ построения структуры приложения, целью которого является отделение бизнес-логики от пользовательского интерфейса. В результате, приложение лучше масштабируется, тестируется и является более простым в сопровождении и реализации. В архитектуре MVC модель предоставляет данные и правила бизнес-логики, представление отвечает за пользовательский интерфейс, а контроллер обеспечивает взаимодействие модели и представления. Типичную последовательность работы MVC-приложения можно описать следующим образом:

- при заходе пользователя на веб-ресурс, скрипт инициализации создает экземпляр приложения и запускает его на выполнение (при этом отображается вид главной страницы сайта);
- приложение получает запрос от пользователя и определяет запрошенные контроллер и действие. В случае главной страницы, выполняется действие по умолчанию (index);
- приложение создает экземпляр контроллера и запускает метод действия, в котором, к примеру, содержаться вызовы модели, считывающие информацию из базы данных;
- после этого, действие формирует представление с данными, полученными из модели, и выводит результат пользователю.

Модель - содержит бизнес-логику приложения и включает методы выборки (это могут быть методы ORM), обработки (например, правила валидации) и предоставления конкретных данных. Одна и та же модель, например: модель аутентификации пользователей может использоваться как в пользовательской, так и в административной части приложения. В таком случае можно вынести общий код в отдельный класс и наследоваться от него, определяя в наследниках специфичные для подприложений методы. Вид – используется для

задания внешнего отображения данных, полученных из контроллера и модели. Виды содержат HTML-разметку и небольшие вставки PHP-кода для обхода, форматирования и отображения данных. Виды обычно разделяют на общий шаблон, содержащий разметку, общую для всех страниц (например, шапку и подвал) и части шаблона, которые используются для отображения данных выводимых из модели или отображения форм ввода данных. Контроллер – связующее звено, соединяющее модели, виды и другие компоненты в рабочее приложение. В хорошо спроектированном MVC-приложении контроллеры обычно содержат только несколько десятков строк кода [7].

Для разработки веб-приложения на языке PHP целесообразно использовать PHP-фреймворк. Фреймворк – это набор всевозможных библиотек (инструментов) для быстрой разработки приложения. Главная цель фреймворка, предоставить программисту удобную среду для проекта с большим и хорошо расширяемым функционалом [8].

Одним из популярных PHP-фреймворков является Yii. Yii – это высокоеффективный, основанный на компонентной структуре MVC-фреймворк для быстрой разработки крупных веб-приложений. Он позволяет максимально применить концепцию повторного использования кода и может существенно ускорить процесс веб-разработки [9].

CodeIgniter является мощным, высокопроизводительным, открытым PHP фреймворком, который поможет быстро создать PHP-приложение. CodeIgniter известен своей легковесностью, минимально загружая сервер.

Zend-фреймворк представляет собой чрезвычайно мощную структуру. Разработанный Zend Technologies, Zend Framework распространяется под лицензией New BSD license, как и CodeIgniter. Zend базируется на простоте, передовом опыте объектно-ориентированного программирования, корпоративно-дружественном лицензировании и тщательном тестировании гибкости кода. Zend Framework ориентирован на построение более безопасных, надежных и современных Web 2.0 приложений и веб-сервисов, использующих широко доступные интерфейсы от ведущих производителей, таких как: Google, Amazon, Yahoo [10].

Для формирования отчетов MS Excel можно воспользоваться библиотекой PHPExcel. Она содержит огромный функционал для работы с форматами xls,xlsx, позволяет считывать данные из файла Excel, создавать и записывать файлы в формате xls,xlsx, pdf, менять форматирование, задавать формулы, а также работать с изображениями [11].

Для визуализации итогов вычислений можно использовать Envision – библиотеки для создания быстрых динамических и интерактивных визуали-

заций данных на HTML5. Возможности библиотеки - визуализация в реальном времени, временная шкала, визуализация валют, работа с фракталами, поддержка Ajax. Поддерживается основными браузерами, в том числе и Internet Explorer 6-8 через FlashCanvas, а также мобильными браузерами, использующими WebKit. Envision основана на библиотеке Flotr2 [12].

Flotr2 - библиотека с открытым кодом для построения HTML5 графиков и диаграмм. Возможности библиотеки Flotr2 - поддержка мобильных устройств, фреймворк-независимость, расширяемость с помощью плагинов, пользовательские типы диаграмм, поддержка браузерами на уровне FF, Chrome, Internet Explorer 6+, Android, iO, поддерживает линии, бары, свечи, пироги, пузыри, возможна работа с заголовками, подзаголовками, цветами [13].

Программный комплекс, разрабатываемый на основе архитектуры MVC, должен включать

1) модели: (проверки корректности введенных данных, оценки зависимости выбросов в атмосферу, отходов производства и потребления,

объема сброса сточных вод, имеющих загрязняющие вещества, от экономических факторов, оценки зависимости здоровья населения региона от экологических факторов, формирования отчетов MS Excel, записи результатов расчетов в БД и их поиска в ней);

2) контроллеры (оценки эколого-экономических взаимодействий, работы с БД, представления результатов расчетов пользователю);

3) виды (ввода исходных данных, вывода результатов расчетов в виде таблиц, графиков и диаграмм).

Для организации хранения данных рекомендуется использовать СУБД MySQL. Для реализации серверной части приложения (модели и контроллеры) язык программирования PHP и библиотеку PHPExcel. Для реализации интерфейса (вид) технологии HTML, CSS, JavaScript, библиотеки jQuery, Envision.

Использование веб-технологий обеспечит мобильность приложения, а также возможность удаленной работы с данными одновременно нескольким пользователям.

## СПИСОК ЛИТЕРАТУРЫ

1. Материалы сайта «Web-Dev. Заметки web-мастера» [Электронный ресурс]. – Режим доступа: <http://lamp-dev.ru/web-dev/desktop-vs-web-applications/>, свободный.
2. Материалы сайта «QScо. Информационные системы для вашего бизнеса» [Электронный ресурс]. – Режим доступа: <http://www.q-sco.ru/oblasti-ekspertitzi/napravlenija/web-prilozhenija.html>, свободный.
3. Материалы сайта «MyBlaze.ru. Портал о высоких технологиях и Интернете» [Электронный ресурс]. – Режим доступа: <http://myblaze.ru/porassuzhdaem-zachem-nuzhen-javascript-plyusyi-i-minusyi/>, свободный.
4. Материалы сайта «Site-do.ru» [Электронный ресурс]. – Режим доступа: <http://www.site-do.ru/js/jquery1.php>, свободный.
5. Материалы сайта «Php.su. Все о PHP» [Электронный ресурс]. – Режим доступа: <http://www.php.su/php/?php>, свободный.
6. Материалы сайта «Mysql.ru» [Электронный ресурс]. – Режим доступа: <http://www.mysql.ru/docs/tkachenko/>, свободный.
7. Материалы сайта «habrahabr.ru. Аналитические статьи, мысли, связанные с информационными технологиями, бизнесом и Интернетом» [Электронный ресурс]. – Режим доступа: <http://habrahabr.ru/post/150267/>, свободный.
8. Материалы сайта «dbhelp.ru. Маленький Yii блог» [Электронный ресурс]. – Режим доступа: <http://www.dbhelp.ru/what-is-framework/page/>, свободный.
9. Материалы сайта «Yiiframework» [Электронный ресурс]. – Режим доступа: <http://www.yiiframework.com/doc/guide/1.1/ru/quickstart.what-is-yii>, свободный.
10. Материалы сайта «ZView.ru — журнал для профессионалов веб-разработки: программистов, дизайнеров, специалистов по юзабилити» [Электронный ресурс]. – Режим доступа: <http://www.zview.ru/news/102>, свободный.
11. Материалы сайта «habrahabr.ru. Аналитические статьи, мысли, связанные с информационными технологиями, бизнесом и Интернетом» [Электронный ресурс]. – Режим доступа: <http://habrahabr.ru/post/148203/>, свободный.
12. Материалы сайта «habrahabr.ru. Аналитические статьи, мысли, связанные с информационными технологиями, бизнесом и Интернетом» [Электронный ресурс]. – Режим доступа: <http://habrahabr.ru/post/141346/>, свободный.
13. Материалы сайта «habrahabr.ru. Аналитические статьи, мысли, связанные с информационными технологиями, бизнесом и Интернетом» [Электронный ресурс]. – Режим доступа: <http://habrahabr.ru/post/137744/>, свободный.

□ Авторы статьи:

Митченков  
Игорь Григорьевич,  
докт.филос.наук, проф., зав. каф.  
философии КузГТУ  
E-mail: mig.phil@kuzstu.ru

Тайлакова  
Анна Александровна,  
аспирант каф.прикладных информа-  
ционных технологий КузГТУ  
E-mail: knopka.anya@mail.ru

Баумгартэн  
Михаил Ицекович,  
канд.физ.-мат.наук, доцент каф.  
философии КузГТУ  
E-mail: bmi45@mail.ru

Михайлов  
Владимир Геннадьевич,  
канд.техн.наук, доцент каф. отрас-  
левой экономики КузГТУ  
E-mail: mvg.eohp@kuzstu.ru

Сарапулова  
Татьяна Викторовна,  
аспирант каф.прикладных информа-  
ционных технологий КузГТУ  
E-mail: sarapulova\_t@mail.ru