

оборота, учитывающая организационную структуру института и реальные процессы движения документов.

При этом система электронного документооборота института должна обеспечивать выполнение следующих функций: регистрация входящих, исходящих и внутренних документов (регистрационная карточка документа), контроль исполнения документов; осуществление совместной работы пользователей над документами; полнотекстовый и атрибутный поиск документов; ведение и передача дел в архив с возможностями поиска; обеспечение доступа к современным средствам телекоммуникаций.

Система электронного документооборота создается на основе использования архитектуры клиент-сервер, благодаря которой обеспечивается физическое разделение ее пользовательской части и базы данных. Использование web-интерфейса позволит получить доступ к документации ВУЗа с учетом системы идентификации пользователей с любой рабочей станции института, подключенной к сети, без установки дополнительного программного обеспечения на стороне клиента. В качестве СУБД используется Oracle 10g. Серверная часть системы предполагается реализовывать с помощью языка программирования Java.

Особенности разработки веб-интерфейса геоинформационной системы

М.Б. Пайсон

ЗАО «Энтерра», г. Барнаул

Работа посвящена созданию клиентской части геоинформационной системы для Web.

При реализации приложения для Internet необходимо учитывать достаточно низкую скорость и высокую стоимость передачи данных. Вследствие этого система должна обеспечивать:

- начальную передачу лишь минимально необходимого количества данных;

- загрузку дополнительных данных по запросу пользователя;

- кэширование данных;

Кроме того, клиентская часть системы должна иметь:

- современный дизайн;

- интуитивно понятный интерфейс.

Для разработки выбрана среда Macromedia Flash и язык Action Script 2.0. Для передачи данных между сервером и клиентом использу-

ется формат XML. Получение XML документа происходит через http запрос.

Важной особенностью является независимость клиентской части от содержания предоставляемой сервером информации. Присутствует возможность получения данных из заранее сформированных файлов без обращения к серверу.

Данные состоят из объектов, связанных в иерархическую структуру. Например, объект типа «страна» содержит несколько объектов типа «область», каждый из которых, в свою очередь, содержит несколько объектов типа «район».

Каждый объект состоит из нескольких информационных слоев. Слой может состоять из данных одного из трех типов: точечных (например, города), линейных (дороги, реки) или полигональных (озера, границы областей). Слои отображаются независимо. Каждый из них может иметь легенду, расшифровывающую условные обозначения и цвета. Пользователь имеет возможность «включать» и «выключать» слои, а также выбирать активный слой. Информация об объектах и легенда активного слоя отображается в отдельном окне.

В каждый момент времени система отображает один объект. Пользователь может вернуться к родительскому объекту, а также перейти к любому из дочерних, указав соответствующий элемент карты или выбрав его из списка объектов активного слоя. При переходе между объектами осуществляется запрос информации с сервера.

В ответ клиент получает список слоев запрашиваемого объекта. После этого загружаются слои, помеченные в списке как «главные». Главные слои являются «включенными» по умолчанию, остальные слои устанавливаются «выключенными» и загружаются при первом обращении к ним пользователем. Все полученные слои сохраняются в памяти до закрытия приложения.

Кроме непосредственного отображения информации поддерживаются такие функции, как изменение масштаба данных, выделение одного или нескольких объектов с последующей передачей данных по указанному адресу. Реализован механизм синхронизации Flash приложения с остальной частью сайта. Система может сохранять просматриваемый объект и автоматически загружать его при следующем запуске. Выделение объектов также может быть сохранено и восстановлено при следующем обращении к приложению.

Клиентская часть имеет объектно-ориентированную структуру. Выделяется четыре основных подсистемы:

– «пользовательский интерфейс», обеспечивающий визуализацию объектов и реакцию на действия пользователя;

- «модель данных», отвечающая за хранение и кэширование данных;
- «менеджер соединений», выполняющий соединение с сервером, формирование запросов и обработку результатов;
- «менеджер взаимодействий», обеспечивающий взаимодействие между подсистемами и определяющий логику работы приложения.

Кроме того, можно выделить подсистему навигации и визуализации данных.

Ряд стандартных компонент Flash заменен на более эффективные с точки зрения данной задачи, разработано несколько дополнительных компонент.

Приложение является универсальным и может быть использовано для решения разнообразных задач. В данный момент система используется для выбора интересующего региона при поиске объектов недвижимости в штате Нью-Йорк. Система разработана в рамках проекта компании ЗАО «Энтерра».

Ядровая модель данных

А.С. Покидов

ЗАО «Энтерра», г. Барнаул

Особенность многих современных систем хранения данных заключается в том, что они не всегда могут полностью решить задачу по целостности данных, по отношениям между данными, перекладывая часть решения на внешние системы. Это означает необходимость для каждой внешней системы, работающей с системой хранения данных реализовывать одно и тоже решение. Внешние системы могут быть самыми разнообразными и написанными на разных языках программирования, что не позволяет использовать одни и те же библиотеки.

Несмотря на то, что реляционные СУБД существуют много времени, они остаются наиболее популярными для решения прикладных задач. Большой интерес представляют объектно-ориентированные СУБД. Однако объектно-ориентированная модель имеет существенный недостаток – ООСУБД требуют больше вычислительных ресурсов, чем реляционные, что не позволяет применять их там, где требования по расходуемым ресурсам играют существенную роль.

Работа направлена на создание эффективной модели данных позволяющей решать широкий спектр прикладных задач. Ее основой является проведенное автором математическое исследование хранения данных в табличном виде, которое показало, что реляционная модель