УДК 004.6

А.А. Козинский, В.А. Козлов

МАТЕМАТИЧЕСКИЕ МОДЕЛИ ВЕБ-СИСТЕМЫ «ТРЕНАЖЕР РУССКОГО ЯЗЫКА ДЛЯ ИНОСТРАННЫХ СТУДЕНТОВ»

Статья посвящена описанию некоторых математических моделей, созданных в рамках проектирования и внедрения веб-системы «Тренажер русского языка для иностранных студентов». Для описания в настоящей статье выбрана модель данных, использованная для хранения информации о пользователях, учебном содержании, статистических показателях результатов тестирования и другие. Второй математической моделью, описанной авторами, является модель организации видеоконференцсвязи. Модель видеоконференцсвязи отражает сущность сетевого взаимодействия пользователей. При ее создании использованы технологии: Ajax, HTML, Ghostscript, ActiveMQ, Flex, Grails, Tomcat, Radis и др. Математические компоненты модели видеоконференцсвязи используют теорию защиты сигнала методами криптографического анализа, шифрования пользовательских данных, сжатия видеосигнала с использованием кодека MPEG-4 FFMPEG.

Результаты проектирования используются в обучении, проведении вебинаров, видеоконференций.

Введение

В условиях осуществления основной деятельности Брестским государственным университетом имени А.С. Пушкина активизирована работа по привлечению иностранных студентов. Данное обстоятельство является основанием разработки автоматизированных систем обучения. Остро стоит вопрос обучения русскому языку иностранных студентов университета. Перед одним из авторов статьи была поставлена задача создания такой системы. Причем одной из важнейших задач проектирования была разработка и реализация ряда математических моделей с учетом особенностей учебного процесса для иностранных студентов. При создании модели учтены следующие особенности: значительный объем преподаваемого материала, особенности взаимодействия обучаемых с преподавателем, необходимость мониторинга эффективности обучения.

Некоторые из реализованных моделей описаны в настоящей работе.

Применение метода декомпозиции для построения логической модели приложения RLT

Применение метода декомпозиции для решения сложной задачи проектирования веб-приложения позволило создать логическую модель проекта. Схема логической модели представлена на рисунке 1.

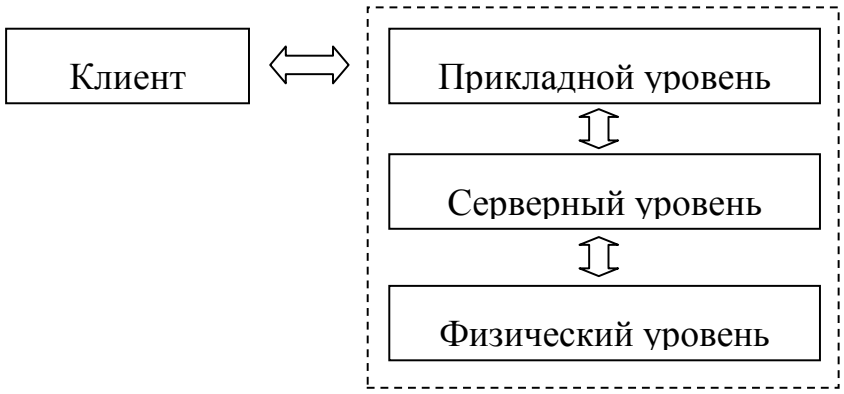

Рисунок 1 – Логическая модель реализации веб-приложения RLT

Физический уровень – это самый нижний уровень системы. На нем, как на фундаменте, расположены все остальные уровни RLT.

Физический уровень представляет собой выделенный персональный компьютер, называемый сервером, подключенный к источнику питания. Он выполняет базовые низкоуровневые действия по вычислению переданных ему данных, после чего возвращает результат верхнему уровню. Задачи физического уровня решены средствами сети.

Серверный уровень отвечает за выполнение инструкций и сценариев тренажера. Слово «серверный» здесь обозначает логический сервер – программный комплекс, функции которого реализованы автором в приложении. Основным компонентом серверного уровня является база данных, фрагменты модели и реализация которой представлены ниже.

Прикладной уровень представлен файлами скриптов для обработки контента. С данным уровнем системы непосредственно работает пользователь программы. Этот уровень формально можно назвать «направляющим» или «устанавливающим». Он является фактическим генератором команд для других уровней. К прикладному уровню относится пользовательский интерфейс приложения (*GUI*).

Общая схема взаимодействия пользователя с системой отражена на рисунке 1.

Опишем подробнее некоторые элементы логической модели.

Основным элементом серверного уровня является база данных. Инфологическая модель базы реализована средствами PHP MyAdmin и представлена на рисунке 2.

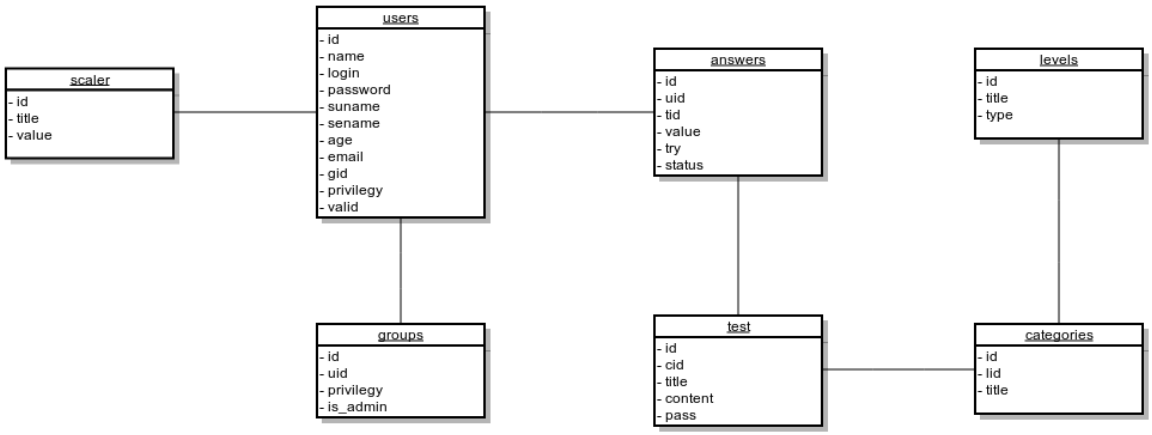

Рисунок 2 – Схема базы данных системы RLT

Фрагмент реализации модели приведен в листинге дампа данных.

Листинг фрагмента дампа данных

```
DROP TABLE IF EXISTS `users`;
CREATE TABLE IF NOT EXISTS `users` (
  `id` smallint(5) unsigned NOT NULL auto_increment,
  `login` varchar(15) NOT NULL,
  `password` varchar(32) default NULL,
  `name` varchar(60) NOT NULL,
  `suname` varchar(60) NOT NULL,
  `sename` varchar(60) NOT NULL,
```

```
  `age` tinyint(3) unsigned NOT NULL,
  `email` varchar(40) NOT NULL,
  `gid` smallint(5) unsigned NOT NULL,
  `privilegy` tinyint(3) unsigned NOT NULL default '3',
  `valid` tinyint(3) unsigned NOT NULL default '1',
  PRIMARY KEY  (`id`),
  UNIQUE KEY `login` (`login`,`email`)
) ENGINE=MyISAM AUTO_INCREMENT=4 DEFAULT CHARSET=cp1251 AUTO_INCREMENT=4;
```

Прикладной уровень представлен компонентами, реализующими проведение видеоконференций, статистический мониторинг, интерфейс пользователей, анализ звуковых данных [1] и др. Перечисленные компоненты для RLT являются наиболее интересными и сложными в реализации. Остановимся на их описании подробнее.

Статистический мониторинг предназначен для анализа результатов обучения отдельных студентов и агрегированных данных групп. Внешний вид окна отчета о результатах обучения отдельного студента представлен на рисунке 3.

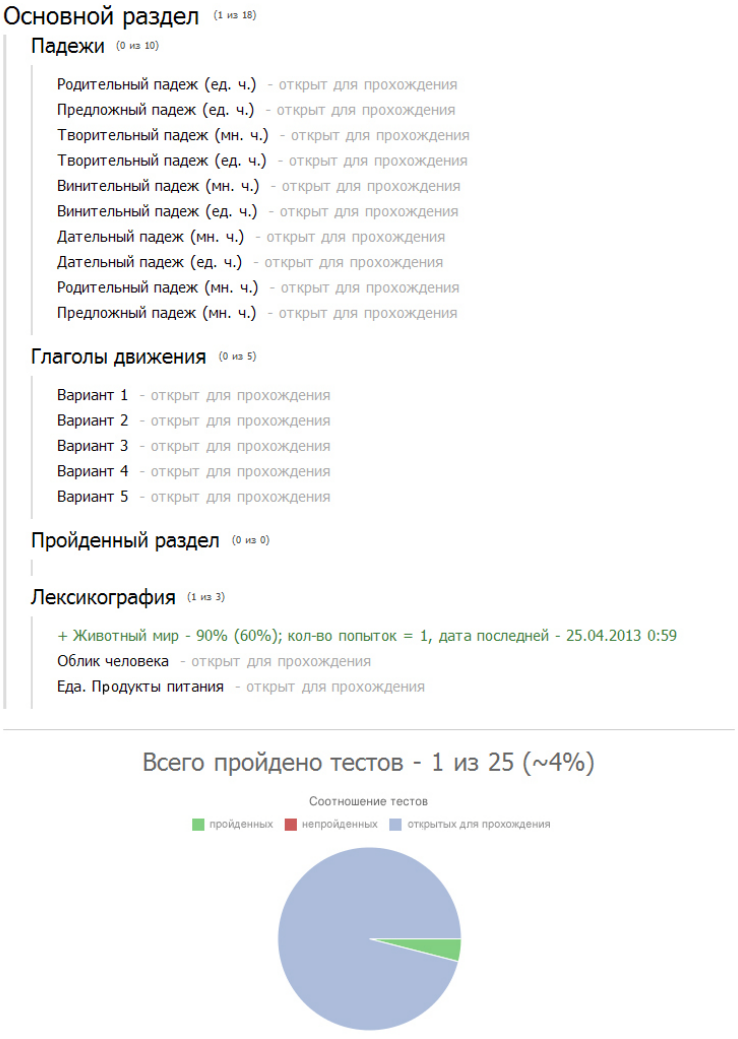

Рисунок 3 – Вид отчета по статистическим показателям результатов тестирования

Система видеоконференцсвязи

Опыт использования RLT показал необходимость разработки дополнительного компонента для видеоконференцсвязи пользователей. Клиентская часть представлена браузером. Для создания серверной части использованы технологии Ajax, HTML, Ghostscript, ActiveMQ, Flex, Grails, Tomcat, Radis и другие.

Основную трудность в реализации серверной части составляет передача потока видеоданных. Для решения задачи минимизации трафика применены алгоритмы сжатия сигнала Хаффмана [3] и PPM (Prediction by Partial Matching) [4].

Фрагмент программного кода алгоритма сжатия видеоданных реализован средствами Java (листинг 2).

Листинг фрагмента программного кода алгоритма сжатия видеоданных

```
public static String postURL(String targetURL, String urlParameters)
{
        return postURL(targetURL, urlParameters, "text/xml");
}

public static String postURL(String targetURL, String urlParameters, String
contentType)
{
        URL url;
        HttpURLConnection connection = null;
        int responseCode = 0;
        try {
                //Create connection
                url = new URL(targetURL);
                connection = (HttpURLConnection)url.openConnection();
                connection.setRequestMethod("POST");
                connection.setRequestProperty("Content-Type", contentType);

                connection.setRequestProperty("Content-Length", "" +
                Integer.toString(urlParameters.getBytes().length));
                connection.setRequestProperty("Content-Language", "en-US");

                connection.setUseCaches (false);
                connection.setDoInput(true);
                connection.setDoOutput(true);

                //Send request
                DataOutputStream wr = new DataOutputStream (
                connection.getOutputStream ());
                wr.writeBytes (urlParameters);
                wr.flush ();
                wr.close ();

                //Get Response
                InputStream is = connection.getInputStream();
                BufferedReader rd = new BufferedReader(new InputStreamReader(is));
                String line;
                StringBuffer response = new StringBuffer();
                while((line = rd.readLine()) != null) {
                        response.append(line);
                        response.append('\r');
                }
                rd.close();
                return response.toString();

        } catch (Exception e) {

                e.printStackTrace();
                return null;
```

```
        } finally {

                if(connection != null) {
                        connection.disconnect();
                }
        }
}

//
// parseXml() -- return a DOM of the XML
//
public static Document parseXml(String xml)
throws ParserConfigurationException, IOException, SAXException {
        DocumentBuilderFactory docFactory = DocumentBuilderFactory
        .newInstance();
        DocumentBuilder docBuilder = docFactory.newDocumentBuilder();
        Document doc = docBuilder.parse(new InputSource(new StringReader(xml)));
        return doc;
}

//
// urlEncode() -- URL encode the string
//
public static String urlEncode(String s) {
        try {
                return URLEncoder.encode(s, "UTF-8");
        } catch (Exception e) {
                e.printStackTrace();
        }
        return "";
}

//
//encodeURIComponent() -- Java encoding similiar to JavaScript encodeURIComponent
//
public static String encodeURIComponent(String component)   {
        String result = null;

        try {
                result = URLEncoder.encode(component, "UTF-8")
                           .replaceAll("\\%28", "(")
                           .replaceAll("\\%29", ")")
                           .replaceAll("\\+", "%20")
                           .replaceAll("\\%27", "'")
                           .replaceAll("\\%21", "!")
                           .replaceAll("\\%7E", "~");
        } catch (UnsupportedEncodingException e) {
                result = component;
        }

        return result;
}
```

Система кодирования и шифрования данных

Для сокрытия данных пользователей системы и предотвращения несанкционированного доступа к веб-приложению применяется компонент кодирования и шифрования данных. Шифрование данных обеспечивает двойной алгоритм MD5 [5] хеширования с дополнительным разделителем. В качестве основы для сжатия и кодирования потока данных между пользователями применяется алгоритм сжатия сигнала Хаффмана, а также дополнительный алгоритм PPM (Prediction by Partial Matching).

Заключение

Практика реализации web-приложения RLT показала, что важными задачами проектирования является реализация математических моделей. Основными этапами проектирования автоматизированной системы управления предприятием

общественного питания являются: изучение предметной области; разработка объектной модели; реализация объектов, интерфейса, связей; тестирование и внедрение.

В настоящей статье представлено описание некоторых моделей и основные особенности их реализации. Сложность проекта вызвала необходимость применения многоуровневого подхода для создания сложной системы, которой является «Система обучения русскому языку иностранных студентов». Реализованными уровнями системы являются: физический, серверный, прикладной.

Приложение внедрено в учреждениях образования и коммерческих предприятиях, что отражено актами внедрения. В настоящее время выполняется сопровождение системы RLT и проектирование модулей прикладного уровня.

СПИСОК ЛИТЕРАТУРЫ

1. Козлов, В.А. Применение быстрого преобразования Фурье для анализа звуковых данных / А.А. Козинский, В.А. Козлов // Веснік Брэсцкага універсітэта. Серыя 4. Фізіка, Матэматыка. – 2013. – № 1. – С. 66–73.

2. BigBlueButton: открытое решение организации конференций [Электронный ресурс] – Режим доступа : http://habrahabr.ru/post/112066/. – Дата доступа : 01.10.2013.

3. Код Хаффмана – Википедия [Электронный ресурс] – Режим доступа : http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4_%D0%A5%D0%B0%D1%84%D1%84%D0%BC%D0%B0%D0%BD%D0%B0. – Дата доступа : 01.10.2013.

4. Алгоритм сжатия PPM – Википедия [Электронный ресурс] – Режим доступа: http://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D1%81%D0%B6%D0%B0%D1%82%D0%B8%D1%8F_PPM. – Дата доступа : 01.10.2013.

5. MD5 – Википедия [Электронный ресурс] – Режим доступа : http://ru.wikipedia.org/wiki/MD5. – Дата доступа : 01.10.2013.

A. Kazinski, V. Kozlov **Mathematical Models of Web-based «Russian language Trainer for Foreign SStudents»**

The article describes some mathematical models created in the design and implementation of web-based "Russian language trainer for foreign students." For descriptions in this article the data model was selected to store information about users, educational content, statistics of the test results, and others. The second mathematical model, described by the authors is the model of video conferencing. The model captures the essence of videoconferencing networking users. At its creation technology used: Ajax, HTML, Ghostscript, ActiveMQ, Flex, Grails, Tomcat, Radis etc. Mathematical model components use videoconferencing theory of signal protection methods of cryptanalysis, encryption of user data, video compression codec using the MPEG-4 FFMPEG.

The results of the design used in training, webinars, video conferencing.

Рукапіс паступіў у рэдакцыю 17.10.13