

УДК 004.89

Архитектура интеллектуальной системы дистанционного обучения

М.А.Гаврилова¹, Фокин А.А.², П.И. Коняев³

*¹Студент, кафедра «Компьютерные системы и сети» МГТУ им.Н.Э.Баумана,
г. Москва, Россия*

*²Студент, кафедра «Компьютерные системы и сети» МГТУ им.Н.Э.Баумана,
г. Москва, Россия*

*³Студент, кафедра «Компьютерные системы и сети» МГТУ им.Н.Э.Баумана,
г. Москва, Россия*

*Научный руководитель: Ерёмин О.Ю., ассистент кафедры «Компьютерные системы и
сети » МГТУ им.Н.Э.Баумана, г. Москва, Россия*

МГТУ им.Н.Э.Баумана

Mariya.gavrilovaa@gmail.com

helblinde92@gmail.com

ruzurberg@gmail.com

Информатизация образования является одним из важнейших условий успешного развития процессов информатизации общества. Ведь именно в сфере образования подготавливаются и воспитываются те люди, которые не только формируют новую информационную среду общества, но которым предстоит самим жить и работать в этой новой среде. Одним из эффективных методов расширения и глобализации образовательного пространства в современном мире является развитие системы дистанционного образования.

В настоящее время системы дистанционного образования позволяют решить лишь ряд проблем, связанных с качеством подаваемого учебного материала, качеством педагогического состава, и с повышающимися потребностями населения всех возрастов в образовательных услугах[1].

Существует два принципиально разных способа (если брать с точки зрения архитектуры информационной системы) построения систем дистанционного образования:

1. Локальная версия, предполагающая использование специального приложения на компьютере пользователя.

2. Онлайн версия, предполагающая использование средств удаленного доступа к образовательным ресурсам.

Рассмотрим данные версии более подробно.

Достоинством локальной версии является возможность использования обучающей системы без доступа к глобальной сети интернет. Тем не менее, подключение к сети все равно требуется, так как обновление системы и отправка готовых заданий преподавателю предполагает временный выход в режим «онлайн». Также локальные версии имеют высокую производительность, и имеют возможность поставки базового комплекта учебных материалов на компакт-дисках.

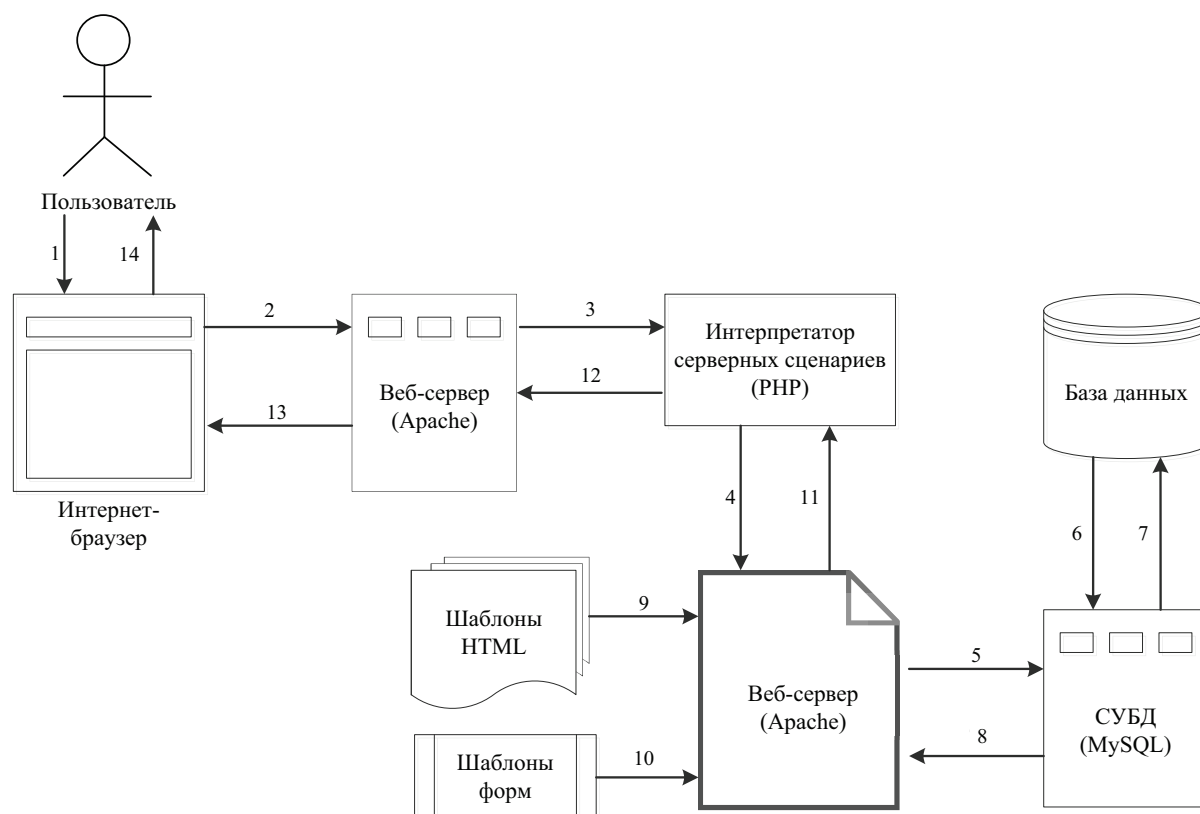
Основным недостатком локальной версии является необходимость создания приложения для аппаратных платформ различного типа (для PC, для Linux и т.п.), что требует дополнительных затрат на программирование. Также локальные версии занимают большой объем на жестком диске пользователя, и могут не иметь постоянной возможности для обновления[6].

В онлайн версии отсутствуют указанные недостатки, но она требует постоянного подключения к сети интернет. Благодаря тому, что доступ к широкополосному интернету в наши дни имеют все больше и больше людей, то влияние этого недостатка постоянно уменьшается.

Основным достоинством онлайн версии является то, что пользователь может обучаться в любом удобном для него месте (например, дома, на работе, на отдыхе). Проще говоря, пользователь не привязан к определенному месту. В случае необходимости он может связаться со своим куратором в режиме реального времени.

Поэтому, на сегодняшний день онлайн версию наиболее удобно реализовать с использованием web-технологий. При этом на компьютере учащихся не надо будет устанавливать дополнительное программное обеспечение, так как будет достаточно наличия программы интернет-обозревателя, или браузера (который, как правило, есть во всех современных операционных системах). Также преимуществом Web-технологий является то, что они позволяют использовать различные средства для разработки приложения и базы данных. Благодаря этому интеллектуальная система не будет зависеть от практической реализации системы обработки и хранения данных[4].

Приложение взаимодействует также с пользователем, получая от него команды, и запускает соответствующие процессы по обработке данных, которые содержат обращения к интеллектуальной подсистеме. Приложение взаимодействует с интеллектуальной подсистемой, когда необходимо выполнить действие, связанное с принятием определённого решения, или необходимо получить информацию, хранящуюся в базе знаний.



Архитектура интеллектуальной системы дистанционного образования показана на рисунке.

Данное приложение реализуется по технологии «клиент-сервер». Приложение взаимодействует с пользователем. При получении от него команды, оно запускает соответствующие процессы по обработке данных, которые, в свою очередь, содержат обращения к интеллектуальной системе. Взаимодействие происходит тогда, когда необходимо выполнить действие, связанное с принятием определённого решения, или необходимо получить информацию, хранящуюся в базе данных этой системы[5,7,9,13].

Архитектура строится на основе схемы MVC (model-view-controler, модель-вид-контроллер). MVC предполагает разделение приложения на три части: модель данных приложения, пользовательский интерфейс и бизнес-логику приложения. Причем так, что изменение одного из компонентов оказывает минимальное влияние на другие компоненты.

Модель в приложении представляет собой модель предметной области, или знания. Она реализуется на основе системы управления базами данных (при использовании *активной модели* СУБД реализует полностью модель предметной области, в том числе и реакцию на вносимые изменения), а сами знания хранятся в базе данных.

Вид отвечает за отображение данных пользователю. В веб-приложениях визуальная часть чаще всего реализуется с использованием одной из разновидностей языка разметки гипертекста HTML и каскадных таблиц стилей CSS, а некоторые динамические элементы приложения с помощью языка программирования JavaScript, исполняющегося на стороне клиента (в браузере).

Контроллер реализует бизнес-логику приложения, то есть обеспечивает взаимодействие между пользователем и *моделью* данных, он контролирует вводимые пользователем данные, а также с использованием шаблонов *вида* формирует ответные реакции системы на действия пользователя. Для реализации контроллера в веб-приложении в основном используются языки серверных сценариев, например, Java, PHP. При реализации всей логики приложения в виде серверных сценариев возникает ситуация, что на СУБД при использовании модели является лишь интерфейсом для доступа к БД, тогда говорят о *пассивной модели*. Использование современных мощных и быстрых СУБД позволяет получать эффективную работу приложения при использовании активной модели, при этом реализация контроллера максимально упрощается.

Разберем подробнее следующие компоненты системы дистанционного обучения (см. рис.):

- интернет-браузер – позволяет пользователю получить доступ к системе дистанционного образования и результаты на отправленные запросы;
- веб-сервер – принимает запросы от пользователя, вносит в них изменения (например, с использованием `mod_rewrite`) и передает данные запроса интерпретатору серверных сценариев;
- интерпретатор серверных сценариев – предназначен для обработки серверных сценариев;
- серверные сценарии – реализуют основную логику приложения, обрабатывают поступающие от пользователя данные, а также возвращают визуальное представление для пользователя;
- СУБД – реализует основное поведение для модели MVC;
- база данных – хранит знания о модели предметной области.

Между компонентами системы осуществляются следующие типы запросов:

1 – пользователь взаимодействует с элементами управления на странице в своем интернет - браузере;

2 – запрос от браузера по протоколу HTTP поступает на веб-сервер, где осуществляется его предобработка;

3 – обработанный запрос от веб-сервера поступает на интерпретатор серверных сценариев;

4 – интерпретатор серверных сценариев вызывает необходимый ему сценарий для реализации требуемой функции;

5 – серверный сценарий обращается к СУБД для получения необходимых данных или для внесения изменения в модель данных;

6 – СУБД выполняет запрос к базе данных за необходимыми данными;

7 – от базы данных возвращается результат запроса;

8 – СУБД возвращает результат изменения или состояния модели к серверному сценарию;

9 и 10 – серверные сценарии подключают шаблоны HTML и шаблоны форм для визуального отображения данных для пользователя;

11 – ссылка на готовую страницу возвращается интерпретатору серверных сценариев;

12 – готовая страница передается веб-серверу;

13 – веб-сервер по протоколу HTTP передает страницу веб-браузеру пользователя;

14 – пользователь получает на экране страницу, которая отвечает его запросу.

Выделение указанных компонентов приложения позволяет реализовывать каждый из них независимо от других, а также при изменении одного не вносить (или вносить минимальные) изменения в другом компонент.

Заключение

В данной статье рассмотрена архитектура интеллектуальной системы дистанционного обучения. Основным элементом такой системы является база знаний, которая взаимодействует с подсистемой обработки и хранения данных. Преимуществом данной архитектуры является то, что при редактировании одного из компонентов приложения, влияние на остальные компоненты минимально. Благодаря этому система становится более гибкой и способной адаптироваться к специфике работы пользователей.

Таким образом, использование данной архитектуры интеллектуальной системы упростит взаимодействие между компонентами приложения, взаимодействие между приложением и пользовательским интерфейсом.

Список литературы

1. Полат Е.С. Педагогические технологии дистанционного обучения [Электронный ресурс]//<http://www.distant.ioso.ru/seminary/09-02-6/tezped.htm> (дата доступа: 18.09.2012).
2. Джонс М.Т. Программирование искусственного интеллекта в приложениях/ М.Тим Джонс.- М.: ДМК Пресс, 2006.
3. Гроссберг С. Внимательный мозг [Электронный ресурс]// Открытые системы, №4, 1997. – URL: <http://www.osp.ru/os/1997/04/179198/>(дата обращения:06.04.2011).
4. Комарцова Л.Г. Нейрокомпьютеры: Учеб. Пособие для вузов. /Л.Г.Комарцова, А.В.Максимов. – 2-е изд., перераб. и доп. –М.: изд-во МГТУ им.Н.Э.Баумана, 2004
5. Норенков И.П., Зимин А.М. Информационные технологии в образовании. – М.: Изд-во МГТУ им. Н.Э.Баумана, 2004.
6. Федоров И.Б., Норенков И.П. Критерии качества дистанционного обучения и структура электронных учебников. - http://portal.ntf.ru/BolonskProcess/NFPK-MONI/ko-ob_r_stat_sbor.doc.
7. Норенков И.П., Уваров М.Ю. База и генератор образовательных ресурсов // Информационные технологии, 2005, № 9.
8. Хайкин С. Нейронные сети: полный курс, 2-е изд., испр. – М.: ООО «И.Д.Вильямс», 2006.
9. Carpenter G.A. A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine / G.Carpenter, S. Grossberg //Computing Vision, Graphics, and Image Processing, 37. pp. 54-115, 1987
10. Grossberg S. Adaptive Pattern Classification and Universal Recording: I. Parallel Development and Coding of Neural Feature Detectors // Biological Cybernetics, No. 23, 1976.
11. Barbara M. ART and Pattern Clustering. - Proceedings of the 1988 Connectionist Model Summer. - Published by M.Kaufmann, San Mateo, CA, pp. 174-185.
12. Devaux S. Classification hybride ART-CS: Apprentissage par renforcement, vision et robotique. Rapport de projet de fin d'etudes [Электронный ресурс] / Tours, [1996]. - URL: <http://sde.eduvax.net/artcs/> (дата обращения: 06.04.2011).
13. Hagan M.T. Neural Network Design / M.T. Hagan, H.B. Demuth, M.H. Beale. - Boston, MA: PWS Publishing, 1996.