

## **ПРОБЛЕМЫ ИНТЕГРАЦИИ СИСТЕМЫ УПРАВЛЕНИЯ ОБУЧЕНИЕМ MOODLE И ПРИКЛАДНЫХ ПРОГРАММ**

**Бабаков А.В.**

**Научный руководитель – доцент Редькина А.В.**

*Сибирский федеральный университет*

Тенденция к интеграции наблюдается во многих, если не во всех, сферах жизни – от экономической до духовной. Размытие национальных границ, переход к нескольким и даже одному мировому языку, объединение экономических мощностей, обмен ресурсами – все это проявление интеграции в жизни мирового сообщества и, как следствие, в нашей жизни.

Тенденция к интеграции наблюдается и в области разработки программного обеспечения. Как показывает анализ современных информационных систем, необходимость интеграции приложений чаще возникает не на уровне проектирования этих приложений, а уже в момент их использования. Перед компаниями, использующими приложения для автоматизации бизнес-процессов, в результате расширения их сферы деятельности и объединения с другими компаниями появляется проблема взаимодействия бизнес-процессов и, соответственно, приложений, их автоматизирующих. Именно в этот момент перед разработчиками возникает задача интеграции приложений, и решение данной задачи во многом зависит от того, насколько эти приложения к интеграции подготовлены. С возникновением проблемы взаимодействия появилось такое понятие, как интеграция корпоративных приложений, связанное непосредственно с трудностями установления взаимосвязей между приложениями и методами преодоления этих трудностей.

Система управления обучением – это программное обеспечение, автоматизирующее администрирование и отслеживание процесса дистанционного обучения, а также составление отчетов о событиях, связанных с этим процессом. На сегодняшний день существует огромное количество таких систем обучения. В основном их можно поделить на две категории: свободные (aTutor, eFront, ILIAS, Moodle и др.) и коммерческие (Blackboard Learning System, Oracle iLearning, JoomlaLMS и др.).

Основное назначение систем управления обучением – поддерживать прикладные функции, которые подходят для любых дисциплин. Эти функции предназначены для управления информацией по дисциплине (редактирование статей и заданий) и для контроля знаний студентов (ответы в виде текста, файлов; тесты). Однако в них отсутствует функциональность, отражающая специфические особенности дисциплин. Например, в системе Moodle, которая применяется в данной работе, отсутствует возможность подготовки блок-схем алгоритмов. Таким образом, для расширения функциональности системы Moodle предлагается соединить с ней редактор блок-схем, разработанный на языке Java-script с помощью открытой библиотеки рисования `wz_jsgraphics.js`.

Рассмотрим эти приложения в плане интегрируемости. Прежде всего, следует обратить внимание на несколько важных свойств: наличие прикладного программного интерфейса, его полнота, открытость программного интерфейса, наличие документации для этого интерфейса, а также его актуальность. Система Moodle обладает всеми этими свойствами, а значит, уровень интегрируемости является достаточно высоким.

Одной из возможностей добавления собственных приложений в систему Moodle является разработка нового вида заданий (assignment). По умолчанию имеется четыре

типа заданий, которые делятся в зависимости от ответа: текст (online), один файл (uploadsingle), несколько файлов (upload) и вне сайта (offline). Каждое из заданий представлено своим классом, наследуемым от базового класса (assignment\_base), в котором описаны основные функции для взаимодействия с базой данных, а также построения стандартной веб-страницы для Moodle. Так как блок-схему удобнее всего хранить в формате XML, то рациональнее использовать как прототип задание с ответом в виде текста. Таким образом, взаимодействие было построено на передаче XML-сообщения.

Итак, была произведена интеграция редактора блок-схем и системы Moodle. Архитектура данной системы относится к типу «точка-точка», поскольку имеет место непосредственная связь между редактором и системой Moodle через общую базу данных и файловую систему сервера, где располагается созданный пользователем XML-файл.

Рассматривая достоинства и недостатки выполненной интеграции, следует обратить внимание на то, что идеология системы Moodle заключается в работе без дополнительных технологий, таких как JavaScript, ActionScript и пр. Однако, если такие технологии имеют место в дополнениях, то разработчики предлагают набор инструментальных средств. Но смысл данной работы заключался не в том, чтобы разработать дополнение к Moodle, используя предлагаемые средства, а в том, чтобы интегрировать два независимых приложения, что и было успешно произведено. Таким образом, с нарушенной некоторым образом идеологией Moodle поставленная задача решена.

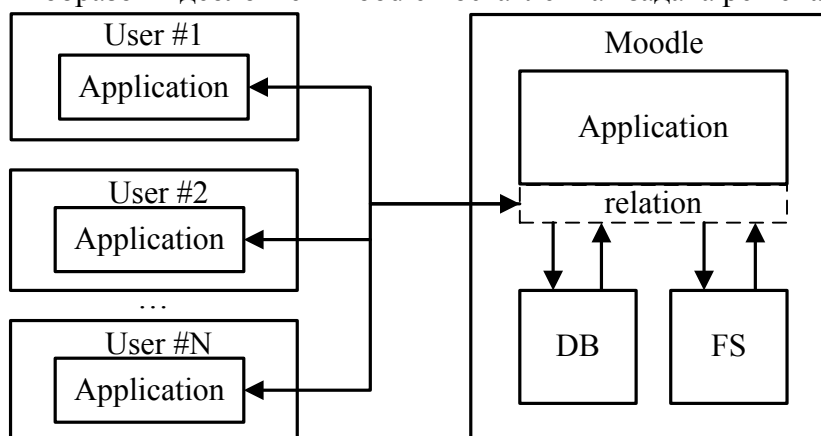


Рис. 1.

Структурная схема полученной системы представлена на рис. 1. Как видно из схемы разработанный модуль находится внутри системы Moodle. При обращении пользователя к соответствующему заданию редактор загружается на клиентскую машину. При расширении функциональности самого редактора, например, анализа блок-схем, можно переложить нагрузку на клиентскую часть, что является преимуществом, однако за этим последует увеличение размера самого приложения. Интернет является большой распределенной системой, а значит, в данном случае можно говорить о простоте масштабирования, поскольку сервер занимается хранением данных, а не их анализом, обработка данных происходит на стороне клиента.

В заключение хотелось бы отметить, что в системах управления обучением отсутствует поддержка, которая бы отражала специфические особенности различных дисциплин. В связи с этим возникает необходимость разработки проблемно-ориентированных приложений. Выполненная работа, являясь прототипом такого приложения, позволит упростить разработку программного обеспечения за счет выявления общих принципов взаимодействия и построения подобных систем.