

УДК 004.65

ИСПОЛЬЗОВАНИЕ RUBY ON RAILS ДЛЯ АВТОМАТИЗАЦИИ УЧЕТА ИПОТЕЧНЫХ КРЕДИТОВ

Аббакумов Андрей Александрович, Егунова Алла Ивановна,
Панкратов Михаил Викторович
ГОУВПО «Мордовский государственный университет им. Н. П. Огарева»,
Российская Федерация, г. Саранск,
E-mail: abbakumov_aa@mail.ru, тел. +7(927) 6427452,
430005, г.Саранск, ул. Богдана Хмельницкого, д. 39

Аннотация. В статье рассматриваются существующие автоматизированные системы выдачи ипотечных кредитов, а также производится выбор среды программирования для создания собственной программной системы, учитывающей достоинства и недостатки действующих АИС в области ипотеки.

Ключевые слова: ипотека, заявка, кредит, фреймворк, Ruby on Rails.

Тема ипотечного кредитования в настоящее время очень актуальна. Создание системы долгосрочного кредитования граждан на цели приобретения жилья позволяет увеличить их платёжеспособный спрос и сделать приобретение жилья доступным для основной части населения. Оно обеспечивает взаимосвязь между денежными ресурсами населения, банками, финансово-строительными компаниями и предприятиями стройиндустрии, направляя финансовые средства в реальный сектор экономики. Количество потенциальных заемщиков растет изо дня в день, предъявляя новые, более жесткие требования к качеству обслуживания клиентов. Поэтому очевидна необходимость создания автоматизированной системы выдачи и учета ипотечных кредитов, результатом применения которой будет повышение эффективности работы кредитных учреждений и упрощение процедуры получения кредитов.

В настоящее время существует несколько действующих автоматизированных систем в области ипотеки, в частности, Федеральное агентство по ипотечному жилищному кредитованию с 2005 года использует систему, разработанную компаниями EPAM и VDI. Они построили свою систему на базе собственного ПО с использованием технологий ActiveX, VB Script, а в качестве СУБД использовали MSSQL 2000. Национальная ипотечная компания закупила АИС фирмы ОАО "ЦМД-софт", построенную на базе Microsoft Dynamics CRM. Крупные российские банки используют свои автоматизированные системы, в основе которых, как правило, лежит промышленная СУБД. Но все вышеперечисленные автоматизированные системы имеют ряд недостатков, среди которых можно выделить следующие:

- высокая себестоимость, поскольку для разработки АИС применяется дорогостоящее программное обеспечение;
- ориентация систем исключительно на сотрудников банка, отсутствие клиентского интерфейса.

На основании вышеизложенного, было принято решение организовать разрабатываемую систему как веб-приложение, для предоставления доступа к системе не только сотрудникам банка, но и клиентам. Веб-приложение — клиент-серверное приложение, в котором клиентом выступает браузер, а сервером — веб-сервер. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются межплатформенными сервисами. Вместо того чтобы писать различные версии для различных операционных систем, приложение создается один раз для произвольно

выбранной платформы и на ней разворачивается. Для создания веб-приложений на стороне сервера используются разнообразные технологии и языки программирования.

Таким образом, все программное обеспечение, необходимое для создания системы в виде веб-приложения, можно разделить на следующие компоненты:

- Среда программирования;
- СУБД;
- Веб-сервер.

Наиболее оптимальным выбором среды программирования является Фреймворк Ruby on Rails. Это – программный каркас, написанный на языке программирования Ruby. Ruby on Rails предоставляет архитектурный образец MVC (Model-View-Controller — Модель-Представление-Контроллер). Такая архитектура подразумевает, что модель данных приложения, пользовательский интерфейс и управляющая логика разделены на три отдельных компонента, так, что модификация одного из компонентов оказывает минимальное воздействие на другие компоненты. Модель отвечает за поддержку состояния приложения. Иногда это состояние является кратковременным, продолжающимся только на время нескольких взаимодействий с пользователем. А иногда состояние является постоянным и сохраняется вне приложения, чаще всего в базе данных. В модели прописаны все бизнес-правила, применяемые к данным. Помещая реализацию бизнес-правил в модель, мы гарантируем, что ничто иное в приложении не может их испортить. Модель работает и сторожем, и хранителем данных. Представление отвечает за формирование пользовательского интерфейса, который обычно основан на данных модели. Хотя представление может преподносить пользователю различные способы ввода данных, оно никогда не занимается их непосредственной обработкой. Работа представления завершается, как только данные будут отображены. Допускается существование множества представлений, имеющих доступ к одним и тем же данным модели, часто с различными целями. Контроллеры организуют работу приложения. Они воспринимают события внешнего мира, взаимодействуют с моделью и отображают соответствующее представление для пользователя. На рисунке 1 показана MVC в абстрактном представлении.

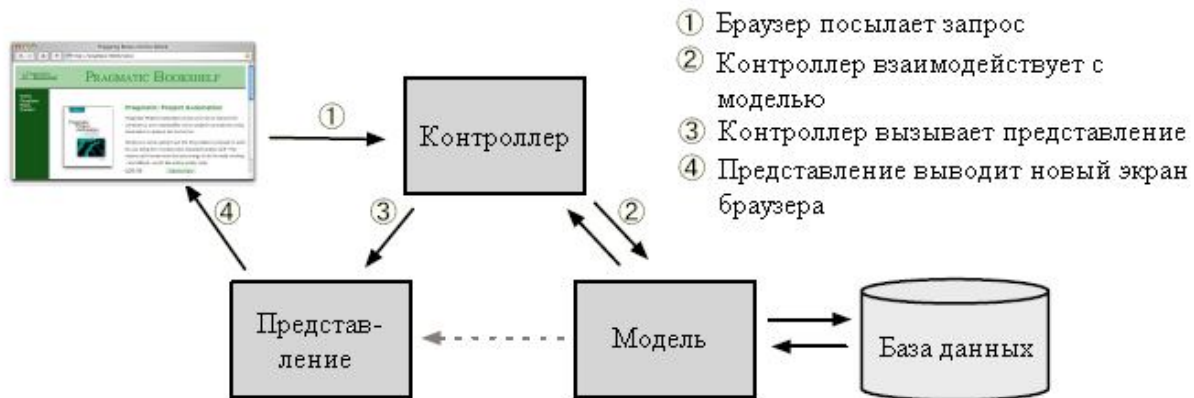


Рис. 1. Архитектура Модель-Представление-Контроллер

Таким образом, модели, представления и контроллеры разрабатываются как различающиеся по функциональным возможностям фрагменты, а среда Rails связывает их вместе при выполнении программы. Одно из положительных свойств Rails заключается в том, что связывание процессов основано на использовании разумных установок по умолчанию, поэтому для того, чтобы все заработало, обычно не приходится создавать какие-либо внешние конфигурационные метаданные.

Достоинства данной среды разработки:

- Объектно-ориентированный подход, лежащий в основе Фреймворка, значительно

упрощает и ускоряет процесс создания приложения. То есть для выполнения конкретных задач на Ruby on Rails требуется гораздо меньше времени по сравнению с использованием других веб-инструментов: PHP, Perl, Java.

- Rails предоставляет механизмы повторного использования, позволяющие минимизировать дублирование кода в приложениях (принцип Don't Repeat Yourself).
- Ruby содержит автоматический сборщик мусора, что облегчает работу с памятью.
- Кроссплатформенность, гарантирующая работу приложения на любой операционной системе независимо от аппаратных характеристик.

К недостаткам данной среды можно отнести невысокую скорость выполнения скриптов. Данная проблема объясняется относительно молодым возрастом Фреймворка. Но с выходом последней версии этот недостаток был практически устранен.

Важной составляющей разрабатываемой информационной системы является СУБД.

Rails способна взаимодействовать со следующими серверами БД: MySQL, Firebird, PostgreSQL, IBM DB2, Oracle и Microsoft SQL Server. Выбор пал на реляционную СУБД MySQL, поддерживающую SQL (структурированный язык запросов) и может применяться в качестве SQL-сервера. Это означает, что общаться с сервером можно на языке SQL: клиент посылает серверу запрос, тот его обрабатывает и отдает клиенту только те данные, которые были получены в результате этого запроса. MySQL - это ПО с открытым кодом, т.е. его можно свободно изучать и изменять. Распространяется на условиях GPL (General Public License). Существуют и коммерческие версии MySQL Enterprise. С появлением Интернет-технологий, позволяющих создавать динамичные Web-страницы, необычайно возрос спрос и на СУБД, которые наиболее полно подходили бы для этого по быстрдействию, надежности и стабильности. И здесь хорошо проявил себя пакет MySQL, который получился быстрым, простым и надежным, но, правда, за счет ухудшения функциональности. По большому счету, отсутствие некоторых функций, которые были принесены в жертву быстрдействию и надежности, не создает больших хлопот пользователям. Перечислим преимущества MySQL:

- Быстрдействие. Благодаря внутреннему механизму многопоточности быстрдействие MySQL весьма высоко.
- Безопасность. Довольно высокий уровень безопасности обеспечивается благодаря базе данных mysql, создающейся при установке пакета и содержащей пять таблиц. При помощи этих таблиц можно описать, какой пользователь из какого домена с какой таблицей может работать и какие команды он может применять. Пароли, хранящиеся в базе данных, можно зашифровать при помощи встроенной в MySQL функции password();
- Лицензия. Раньше лицензирование MySQL было немного запутанным, сейчас эта программа для некоммерческих целей распространяется бесплатно;
- Открытость кода. Возможно добавление в пакет нужных функций, расширяя его функциональность.
- Сообщество. Как следствие открытости кода, бесплатности программы, стабильной и надежной ее работы образовалось сообщество людей, которые не просто лояльны к MySQL, но и всячески участвуют как в развитии самого пакета, так и в обучении менее опытных людей работе с ним. Существует огромное количество листов рассылки и конференций, где можно получить бесплатную помощь в любое время суток;
- Переносимость. В настоящее время существуют версии программы для большинства распространенных компьютерных платформ. Это говорит о том, что определенная операционная система не навязывается. Можно самому выбрать, с чем работать, например с Linux или Windows, но даже в случае замены ОС данные не потеряются и даже не понадобятся дополнительные инструменты для их переноса.

Связь Rails с СУБД имеет свои особенности, в виду использования объектно-реляционного отображения. Для связи кода приложения с базой данных понадобятся два уровня программного обеспечения. Первый уровень — это драйвер базы данных,

представляющий собой Ruby-библиотеку, подключающую низкоуровневый API базы данных к высокоуровневому миру программирования на языке Ruby. Второй уровень программного обеспечения представляет собой Rail-адаптер базы данных. Ему отводится место между библиотекой Ruby и приложением. Каждая библиотека базы данных будет иметь свой собственный API, зависящий от конкретной базы. Rails-адаптеры баз данных скрывают эти различия, поэтому Rails-приложения не нуждаются в сведениях о том, с какой именно базой данных они работают. Возможно использование одного и того же программного кода для работы с разными базами данных. В случае смены базы данных, нужно только поменять драйвер базы данных.

Ruby on Rails может работать с любым веб-сервером, поддерживающим FastCGI: Apache, Lighttpd и т.д. В язык Ruby уже встроены 2 веб-сервера: Mongrel и Webrick. Rails отлично взаимодействует с любым из них. Для построения системы было предложено использование веб-сервера Mongrel.

Таким образом, были проанализированы особенности программного каркаса Ruby on Rails, предназначенного для создания веб-приложений. Он был выбран в качестве средства для реализации системы в совокупности с реляционной СУБД MySQL и веб-сервером Mongrel. Данная архитектура позволит построить автоматизированную информационную систему, которая позволила бы решить проблемы документооборота, обмена информацией между клиентами и сотрудниками банка, генерировать отчеты по выданным кредитам и т.д.

Литература

1. Томас Д., Хэнссон Д. Х. Гибкая разработка веб-приложений в среде Rails. – СПб.: Питер, 2008. – 716 с.
2. Фернандес О. Путь Rails. Подробное руководство по созданию приложений в среде Ruby on Rails. – Пер. с англ. – СПб: Символ-Плюс, 2009. – 768 с.
3. Фултон Х. Программирование на языке Ruby. – М.: ДМК Пресс, 2007. – 688 с.

USE RUBY ON RAILS FOR AUTOMATION OF THE ACCOUNT OF HYPOTHECARY CREDITS

Abbakumov A.A., Egunova A.I., Pankratov M.V.
N. P. Ogarev's Mordovian State University
Russian Federation, Saransk city,
E-mail: abbakumov_aa@mail.ru, tel.+7(927) 6427452,
39, Bogdana Hmel'nitskogo str, room 502, 430005, Saransk, Mordovia, Russia

Annotation. This article considers the existing automated mortgage providing systems, as well as chooses the programming environment for creating as particular programming system, which takes into account both advantages and disadvantages of the present AIS in the hypothecary area.

Keywords: mortgage, application, credit, framework, Ruby on Rails.

Сведения об авторах

Аббакумов Андрей Александрович
Россия, РМ, г. Саранск

Мордовский государственный университет им. Н. П. Огарёва
E-mail: abbakumov_aa@mail.ru
Тел. +7(927) 6427452

Окончил Мордовский государственный университет им. Н. П. Огарёва в 2001 г. по специальности автоматизированные системы обработки информации и управления. В 2006 г. защитил кандидатскую диссертацию и работает на должности доцента кафедры автоматизированные системы обработки информации и управления в Мордовском государственном университете им. Н. П. Огарёва.

Егунова Алла Ивановна
Россия, РМ, г. Саранск

Мордовский государственный университет им. Н. П. Огарёва
E-mail: vystasalla@gmail.com
Тел. +7(834)2290602

Окончила филиал «Восход», Московского авиационного института им. С. Орджоникидзе в 1991 г. по специальности «Электронные вычислительные системы, комплексы и сети». В 1999 году защитила кандидатскую диссертацию и работает на должности доцента кафедры автоматизированные системы обработки информации и управления в Мордовском государственном университете им. Н. П. Огарёва.

Панкратов Михаил Викторович
Россия, РМ, г. Саранск

Мордовский государственный университет им. Н. П. Огарёва
E-mail: mike@mrsu.ru
Тел. +7(927) 1924757

Окончил Мордовский государственный университет им. Н. П. Огарёва в 2005 г. по специальности автоматизированные системы обработки информации и управления. В 2008 году закончил аспирантуру. В настоящее время подготовил диссертацию на соискание степени кандидата технических наук и работает преподавателем кафедры автоматики в Мордовском государственном университете им. Н. П. Огарёва.