

# ИСПОЛЬЗОВАНИЕ МЕТАСТРУКТУР ПРИ ПРОЕКТИРОВАНИИ ВЕБ-ИНТЕРФЕЙСА ВИРТУАЛЬНОЙ ЛАБОРАТОРИИ

В.В. Воробьев

## Введение

Виртуальные лаборатории представляют собой современный способ организации совместной исследовательской или образовательной деятельности. В общем случае с помощью виртуальной лаборатории организуется работа географически удаленных друг от друга исследователей и учащихся. При этом отдельные пользователи и группы пользователей являются работниками различных учреждений, университетов, институтов, лабораторий. Каждое из таких учреждений, как правило, обладает своими особыми ограничениями на установку и использование стороннего ПО. В таких условиях оптимальным способом организовать работу в виртуальной лаборатории является использование веб-интерфейса. Виртуальная лаборатория при этом представляет собой программный комплекс, состоящий из набора сервисов и позволяющий организовать для всех пользователей единое информационное пространство. В данной работе под виртуальной лабораторией понимается именно такой программный комплекс, позволяющий организовать удаленную совместную исследовательскую деятельность с помощью набора сервисов и веб-интерфейса. Ключевой набор сервисов зависит от области проводимых с помощью лаборатории исследований. В общем случае в лаборатории присутствует набор сервисов, упрощающих организацию исследовательской деятельности и обмен информацией между пользователями.

В литературе, как правило, не уделяется достаточно внимания веб-интерфейсу подобных программных комплексов [3, 4]. Веб-интерфейс в таких случаях формируется определенным набором слабо связанных документов, основным назначением которых является представление результатов проведения экспериментов удаленному пользователю. В данной работе под веб-интерфейсом виртуальной лаборатории понимается полноценный компонент программного комплекса, предназначенный не только для отображения информации об исследовании, но и для управления работой самого комплекса, анализа его состояния, выполнения задач, связанных не только с исследовательскими, но и с учебными процессами, а также координации действий различных пользователей лаборатории. Несмотря на очевидные преимущества использования такого рода унифицированного интерфейса, а также востребованность в реализации виртуальных лабораторий с такого рода интерфейсами, в литературе и интернете описано немного подобных решений [5].

Специфика исследовательской и образовательной деятельности во многом определяет способы взаимодействия с сервисами лаборатории. Прежде всего, набор пользователей виртуальной лаборатории постоянно меняется. Если определенное фундаментальное исследование может объединить усилия ученых на продолжительные сроки (более 1-2 лет), то активное использование виртуальной лаборатории в образовательном процессе приведет к постоянной ротации пользователей. Доступ к виртуальной лаборатории может предоставляться определенным группам учащихся на непродолжительное время в рамках отдельных курсов для выполнения лабораторных работ, либо небольших проектов. Даже при наличии группы постоянных пользователей, принимающих активное участие в организации исследовательских и образовательных процессов средствами виртуальной лаборатории, определенный процент пользователей, возможно достаточно высокий, будет постоянно меняться. Интенсивность использования тех или иных сервисов лаборатории может зависеть от конкретных научных направлений, проектов, реализуемых с помощью лаборатории. Все это предъявляет определенные требования к веб-интерфейсу виртуальной лаборатории как к основному инструменту организации взаимодействия пользователей с системой.

В данной работе описывается архитектура веб-интерфейса виртуальной лаборатории, основанного на использовании метаструктур, объединяющих в себе информацию различных уровней приложения. В статье описывается метод использования в рамках одного приложения нескольких СУБД: реляционной – для хранения предметной информации, относящейся непосредственно к виртуальной лаборатории, и нереляционной – для хранения метаданных. Представлен алгоритм формирования страниц, составляющих веб-интерфейс. Описывается функционал системы управления содержимым веб-интерфейса через метаструктуры. При проектировании архитектуры учитывалась специфика области применения сервисов виртуальной лаборатории, предназначенной для исследования информационно-коммуникационных систем. При этом описанная архитектура является универсальной и может использоваться для организации веб-интерфейсов аналогичных околонуниверситетских программных комплексов. Именно поэтому, в отличие от представленного в [1] «интернет-сайта», в архитектуре программного прототипа веб-интерфейс является логически независимым компонентом, допускающим описание без приведения деталей предметной области.

Работа поддержана грантом РФФИ № 13-07-96506.

### Используемые технологии

При разработке соответствующих прототипу программных систем предлагается использовать свободное программное обеспечение, что позволяет существенно снизить финансовые затраты. В описанных условиях постоянного изменения состава пользователей и разработчиков системы немаловажным является технологическая независимость от закрытых проприетарных решений, для поддержки которых, как правило, требуется подготовленная и стабильная команда разработчиков. Напротив, при использовании открытых технологий велика вероятность того, что новому разработчику не потребуется дополнительного обучения и ознакомления с системой.

Технологической платформой для реализации компонентов виртуальной лаборатории являются операционные системы семейства Linux, веб-сервер Apache и языки программирования C/C++ для реализации серверной составляющей системы. Для хранения информации в системе предлагается использовать подход «Polyglot Persistence», т.е. использование нескольких СУБД для различных задач в рамках одного приложения [2]. Для хранения предметной информации, связанной с исследовательскими и образовательными процессами, используется реляционная СУБД MySQL. Для хранения вспомогательной метаданных, используемой для управления компонентами веб-интерфейса, используется NoSQL хранилище MongoDB. Для взаимодействия клиентской составляющей веб-интерфейса с серверной составляющей используется технология асинхронного обмена данными AJAX. Для упрощения процессов разработки веб-интерфейса используется программный JavaScript каркас jQuery, предоставляющий, прежде всего, инструменты для манипулирования DOM, а также для отправки и приема AJAX сообщений.

### Основные структуры данных

Одним из вариантов объединения различных СУБД в рамках одного приложения является использование дополнительного уровня абстракции. Ключевым компонентом такого уровня являются метаданные. Метаданные могут содержать в себе атрибуты, описывающие компоненты уровня данных, уровня пользовательского интерфейса и уровня приложения. На уровне приложения метаданные могут быть представлены различными типами данных, но для определенности будем считать такие данные метаструктурами. Далее описываются различные группы атрибутов метаструктур, представляющих информацию об основных уровнях системы: уровне данных, уровне приложения и уровне пользовательского веб-интерфейса (рисунк 1).



Рис. 1. Взаимодействие метаструктур с различными уровнями системы

Основными метаструктурами являются «Поле» и «Таблица». В качестве хранилища метаструктур предлагается использовать документоориентированную NoSQL СУБД MongoDB. Хранение метаструктур в документах позволяет реализовать произвольный набор атрибутов с возможностью поиска по ним. Для управления уровнем предметной информации в метаструктурах содержатся атрибуты, позволяющие описать элементы схемы базы данных. Для метаструктуры «Поле» такими атрибутами являются «Ключ», «Имя» и «Имя таблицы», для метаструктуры «Таблица» - «Ключ», «Имя». Атрибут «Ключ» предназначен для индексирования и поиска документов, представляющих описанные структуры. Атрибут «Имя таблицы» связывает документы, соответствующие полям, с документами, представляющими таблицы предметной БД. Такого набора атрибутов для описанных метаструктур достаточно для того, чтобы представить на уровне приложения информацию обо всех таблицах предметной БД.

Для обеспечения связи уровня данных с уровнем веб-интерфейса метаструктуры должны содержать дополнительные наборы атрибутов, определяющих представление предметной информации на странице. Т.к. на уровне веб-интерфейса метаструктура «Поле» в общем случае представляет собой текстовое поле, соответственно группа атрибутов этой метаструктуры, относящаяся к веб-интерфейсу, определяется свойствами отображаемых текстовых полей. К таким атрибутам могут относиться свойства стилей, относящиеся непосредственно к внешнему виду полей. Также могут использоваться специальные управляющие свойства, определяющие поведение элементов веб-интерфейса. Например, наличие у поля атрибута «Автозаполнение» позволит автоматически на основе метаданных сгенерировать SQL запрос для формирования списка возможных значений, а атрибут «Дата» указывает на необходимость контролировать соответствие вводимой пользователем информации определенному формату, например, «dd.mm.yy».

### Основные алгоритмы

Для формирования страниц веб-интерфейса на основе метаструктур, представляющих одновременно информацию как уровня данных, так и уровня пользовательского интерфейса, необходим особый алгоритм взаимодействия сервера и клиента. Ниже представлен один из способов формирования динамического содержимого веб-страницы, учитывающий специфику используемой метаинформации (рисунок 2).

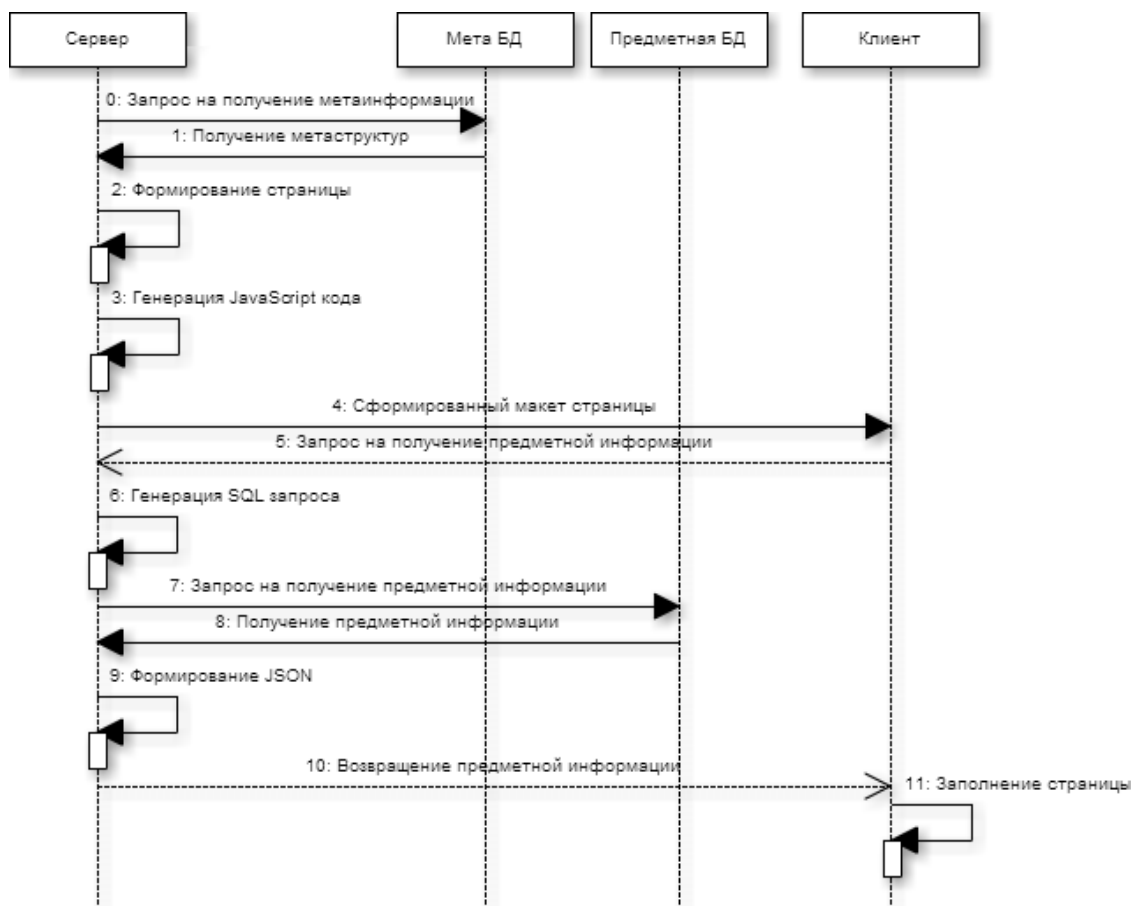


Рис. 2. Алгоритм формирования страницы веб-интерфейса

На начальном этапе клиент запрашивает у сервера определенную страницу для отображения. Клиентом в данном случае выступает браузер, а сервером – программный комплекс, представляющий виртуальную лабораторию. Иницирующим событием начального этапа в веб-интерфейсе может служить, например, нажатие пользователем кнопки, либо переход по ссылке. После получения запроса сервер загружает

запрашиваемую страницу, представляющую из себя макет будущей страницы со статичной информацией. Для подготовки запрашиваемой страницы серверу, прежде всего, необходимо получить метainформацию, описывающую содержимое страницы и определяющую формат ее представления. Для этого сервер обращается к мета базе данных (MongoDB), формируя из полученных документов метаструктуры классов «Поле» и «Таблица». По этим метаструктурам генерируется код на языке JavaScript, который встраивается непосредственно в страницу. Этот код позволяет создавать аналоги метаструктур на стороне клиента, генерировать по ним HTML код, предназначенный для отображения на странице динамической информации, а затем асинхронно запрашивать у сервера предметные данные. После этого макет страницы вместе с соответствующим кодом возвращается клиенту. Предполагается, что взаимодействие клиента и сервера осуществляется по защищенным каналам связи, но для обеспечения дополнительной безопасности отправляемый вместе со страницей код может быть предварительно обфусцирован.

Сгенерированный на сервере JavaScript активируется на стороне клиента после загрузки страницы, определяемой соответствующим событием браузера «onload». Подготовленный на сервере код локально создает структуры данных, описывающие динамические компоненты страницы. Для генерации HTML кода, который бы представлял на странице эти компоненты, клиентом асинхронно запрашиваются предметные данные. Для асинхронного взаимодействия клиента и сервера используется технология AJAX. На сервере снова происходит обращение к мета базе данных. По сформированным метаструктурам генерируются SQL запросы к предметной БД (MySQL). После получения всей необходимой предметной информации сервер создает JSON массив, структура которого повторяет структуру соответствующих метаданных. Наконец подготовленный JSON массив асинхронно возвращается клиенту.

Полученный на стороне сервера массив данных разбирается с помощью встроенной в страницу ранее метainформации. Далее генерируется HTML код для представления полученной предметной информации. В общем случае локально средствами JavaScript и программного каркаса jQuery создаются текстовые поля, которые заполняются полученными асинхронно данными. Важным шагом алгоритма при этом является закрепление на странице дополнительных скрытых полей, полностью повторяющих сгенерированные видимые поля. При сохранении страницы для определения того, какая информация была изменена пользователем, значения видимых полей сравниваются со значениями скрытых полей. Это позволяет локально контролировать изменения содержимого страниц пользователем, а также динамически формировать SQL запросы на обновление только той информации, которая была изменена.

Благодаря использованию такого алгоритма формирования страниц веб-интерфейса удастся, по крайней мере частично, автоматизировать ряд операций, а именно: написание SQL запросов для получения предметной информации, написание JavaScript и HTML кода, управляющего отображением динамической информации на странице. Использование NoSQL СУБД позволяет не учитывать потерю производительности при повторном формировании метаструктур, при этом за счет применения подхода «Polyglot Persistence» происходит не только логическое, но и физическое разделение первичных данных и метаданных. Это особенно важно в рассматриваемой схеме организации веб-интерфейса, т.к. документы больше подходят для хранения метаструктур с нефиксированным набором атрибутов, объединяющих предметную информацию с информацией о пользовательском интерфейсе. При этом описанный алгоритм опирается на дополнительный уровень абстракции, управление которым можно осуществлять вне предметной логики работы системы.

Сохранение метаданных в БД позволяет организовать дополнительный «административный» веб-интерфейс, предназначенный для поиска, добавления и удаления метаструктур и их атрибутов, а также для создания связей между такими структурами. Наличие такого веб-интерфейса упрощает управление отображением динамического содержимого на страницах, а значит и процесс поддержки веб-интерфейса.

### **Заключение**

Разработанный в соответствии с представленной архитектурой веб-интерфейс виртуальной лаборатории позволит решить несколько проблем, неизбежно связанных с созданием подобных систем в научной среде. Прежде всего, финансовые затраты на разработку могут быть снижены до минимума, что достигается использованием свободного программного обеспечения. Введение дополнительного уровня абстракции позволяет определять механизмы управления динамически формируемым содержимым веб-интерфейса. Логическое и физическое разделение хранилищ предметной информации и метаданных позволяет вести разработку сервисов виртуальной лаборатории независимо от веб-интерфейса, обеспечивающего управление сервисами и представляющего результаты их работы. Этому способствует универсальный алгоритм формирования динамического содержимого страниц. Использование этого алгоритма совместно с системой управления метаструктурами может значительно упростить развитие и поддержку виртуальной лаборатории, что особенно актуально в условиях непостоянства групп пользователей и разработчиков. Специфика использования виртуальной лаборатории, предназначенной для исследования информационно-коммуникационных систем, в образовательном процессе такова, что инструмент одновременно является и объектом исследования. Реализация описанных алгоритмов в частности позволяет рассмотреть преимущества и недостатки совместного использования нескольких СУБД в рамках одного приложения.

Описанный веб-интерфейс разрабатывается в рамках проекта, поддержанного грантом РФФИ № 13-07-96506.

#### ЛИТЕРАТУРА:

1. М.С. Герман Интернет сайт сетевой компьютерной лаборатории // Научный сервис в сети Интернет: экзафлопсное будущее: Труды международной суперкомпьютерной конференции (19-24 сентября 2011 г., г. Новороссийск) — М.: Изд-во МГУ, 2011. - 643 с.
2. А. Посконин Web-приложения и данные: проблемы абстракции и масштабируемости // Труды института системного программирования РАН. – 2012. – Т. 23. – С. 159-172.
3. H. Bähring, J. Keller, W. Schiffmann Web-based support for computer engineering // Электронный ресурс: <http://www.fernuni-hagen.de/imperia/md/content/rechnerarchitektur/publikationen/vu05.pdf>
4. B. Eynard, S. Lienard, S. Charles, A. Odinet Web-based collaborative engineering support system: applications in mechanical design and structural analysis // Concurrent Engineering. – 2005. – Т. 13, № 2. – С. 145-153.
5. P. Le Couteur Review of literature on remote & web-based science labs // Электронный ресурс: [http://rws1.nic.bc.ca/Docs/Review\\_of\\_Literature\\_on\\_Remote\\_and\\_Web-based\\_Science\\_Labs.pdf](http://rws1.nic.bc.ca/Docs/Review_of_Literature_on_Remote_and_Web-based_Science_Labs.pdf)