

УДК 004.9

DOI: 10.15827/0236-235X.121.032-037

Дата подачи статьи: 12.12.17

2018. Т. 31. № 1. С. 032–037

ВЕБ-МОДЕЛЬ РАСПРЕДЕЛЕННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ

Н.А. Остроглазов¹, аспирант, nickitaost@gmail.com

А.И. Чугунов¹, аспирант, 4ugunoff@bk.ru

М.А. Кудрин¹, магистрант, mihail-mtl-kudrin@yandex.ru

В.В. Елизаров¹, студент, elizarov43@gmail.com

¹ Самарский национальный исследовательский университет им. академика С.П. Королева, Московское шоссе, 34, г. Самара, 443086, Россия

В статье описываются подходы к решению проблемы эффективного использования вычислительных ресурсов при проектировании сложноорганизованных веб-ориентированных распределенных систем, а также разработка веб-модели распределенной информационной системы реального времени проведения конкурсов научно-образовательных работ.

Система реализована на основе фреймворка Yii на языке PHP. Взаимодействие между сервером приложений и клиентским приложением организовано с использованием паттерна MVC (Model View Controller), где представлением является интерфейс пользователя с инструментами для работы в системе, а контроллер обеспечивает взаимодействие между моделью и представлением. Клиентское приложение спроектировано на основе паттерна проектирования «Модуль» за счет механизма замыканий (closures) на языке JavaScript.

Такой подход к организации структуры веб-системы создает условия для обеспечения централизованной обработки, хранения и доставки пространственных данных через сеть Интернет для удаленных пользователей. В системе реализовано разделение пользователей по ролям на «Администратор», «Проверяющий» и «Участник». В статье описаны диаграммы деятельности пользователей с ролями «Администратор» и «Проверяющий», построенные по методологии UML (Unified Modeling Language – унифицированный язык моделирования).

Разработанная система позволяет в режиме реального времени подавать заявки на участие в конкурсе научно-образовательных работ «Участнику конкурса», проверять, утверждать, отклонять и отправлять на доработку заявки, а также формировать документы с подведением итогов конкурса «Проверяющему», тем самым осуществляя и контролируя весь цикл деятельности по организации проведения конкурса научно-образовательных работ в веб-системе.

Ключевые слова: распределенная вычислительная система, программный комплекс, веб-приложение, сервер, клиент, паттерн, архитектура, управление.

Развитие современных информационных технологий, в том числе глобальной сети Интернет, и освоение пользователями ее возможностей оказывают влияние на различные области жизни и деятельности, что приводит к возникновению новых техно-социальных структур.

В архитектуре крупномасштабных распределенных систем в настоящее время преобладает синхронная платформа клиент-сервер (WWW, Corba, J2EE, COM+), которая формирует две роли: компонент действует как клиент, при этом требует данные/функционал от другого компонента, или действует как сервер, если отвечает на запрос клиента. Клиент блокируется в момент отправки запроса до тех пор, пока не придет надлежащий ответ.

Существуют приложения, которые не могут быть эффективно реализованы при использовании технологии запрос/ответ. Более того, при их реализации, когда информация, предоставляемая одной службой, зависит от информации, поставляемой другой службой, использование этой технологии приводит к решению, которое не масштабируется и предоставляет данные, которые могут быть неточными и неполными. В отличие от этого новая асинхронная платформа Публикация/Подписка непосредственно отражает поведение, свойственное информационно-управляемым приложениям, так как

такая коммуникация является косвенной, инициируемой производителями информации. Гибкость механизма выбора уведомлений, используемого потребителями, является важнейшим фактором при реализации службы уведомлений. Использование избирательного механизма выбора уведомлений по их содержанию в сравнении с выбором на основе канала или темы позволяет повысить эффективность функционирования, гибкость всей системы и в целом облегчает ее расширяемость и непрерывное изменение с целью адаптации к изменениям в объединяемых ею приложениях [1].

Одним из направлений приложения такого рода технологий становится разработка дистанционных конкурсов в рамках дошкольного, школьного, университетского обучения, являющихся неотъемлемой частью образовательного процесса. Кроме мероприятий, проводимых в рамках образовательного учреждения, появилась возможность участия в международных конкурсах, при проведении которых присутствие самого участника не является необходимым. В настоящее время существует большое количество ресурсов, позволяющих принять участие в проводимых мероприятиях, разместить свою работу и получить заслуженный приз. При работе с ними возникают проблемы хранения и анализа работ. Поэтому появляется необходи-

мость создания системы, удобной для подачи заявок участников, отбора данных по различным критериям, удобной сортировки конкурсов, просмотра доступных положений конкурса и качественного анализа результатов проверки.

Возможности системы включают облачное хранение данных и автоматическое формирование дипломов участника или победителя в зависимости от результатов проверки.

Архитектура системы

При проектировании архитектуры веб-ориентированной информационной системы основным критерием выделения подсистем и их функциональности является их пространственная распределенность. Подсистемы взаимодействуют, обмениваясь сообщениями посредством граничных (интерфейсных) объектов особого типа. В свою очередь, параллельные задания, формирующие подсистему, представляются как активные объекты, которые могут взаимодействовать друг с другом в пределах одной подсистемы либо, преодолевая границы, в разных подсистемах различными способами: асинхронно, синхронно, при помощи групповых коммуникаций.

Основным преимуществом разработанного в описываемой системе метода является его ориентированность на проектирование системы реального времени, состоящей из распределенных параллельно выполняющихся компонентов, в то время как другие основанные на методологии UML методы применимы в основном для последовательных систем [2].

В связи с тем, что веб-ориентированная система является сложной, многокомпонентной, с распределенной бизнес-логикой, для облегчения ее разработки и поддержки использована современная программная платформа (фреймворк), удовлетворяющая требованиям архитектуры системы, – Yii framework [3].

При разработке алгоритмов взаимодействия сервера приложений с клиентской частью системы возникает необходимость разделения бизнес-логики и пользовательского интерфейса, так как их взаимное внедрение друг в друга может привести к значительным трудностям при дальнейшем расширении функциональных возможностей системы [4]. При организации взаимодействия сервера приложений и клиентской части используется паттерн Model-View-Controller (MVC), который предназначен для разделения бизнес-логики и пользовательского интерфейса, что позволяет вносить изменения в отдельные части системы, не затрагивая другие. Контроллер в паттерне MVC обеспечивает взаимодействие между моделью и представлением [5]. На рисунке 1 показана схема, отражающая структуру сервера приложений.

Скрипт инициализации (index.php) предназначен для запуска серверного приложения системы

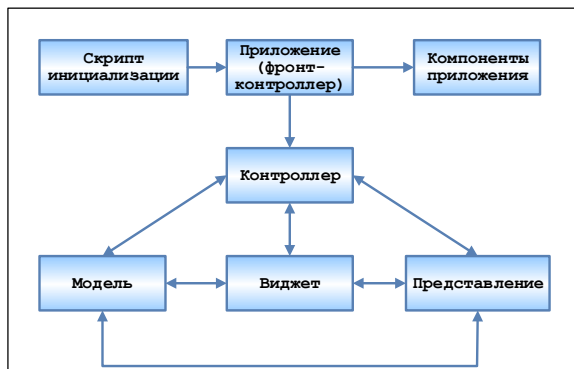


Рис. 1. Функциональная схема сервера приложений

Fig. 1. Application server functional diagram

на выполнение и является связующим звеном между пользователем и системой. Фронт-контроллер инкапсулирует контекст обработки запроса – собирает информацию о запросе и передает ее соответствующему контроллеру для дальнейшей обработки. Объект фронт-контроллера создается инициализационным скриптом в единственном экземпляре, реализуется через паттерн Singleton (одиночка) и доступен из любого места приложения по ссылке. На рисунке 2 представлена схема, отражающая этапы взаимодействия пользователя с сервером приложений системы.

Представим алгоритм взаимодействия пользователя с сервером приложений.

1. Посредством URL пользователь осуществляет запрос к серверу приложений. Сервер приложений обрабатывает запрос и запускает скрипт инициализации на выполнение.

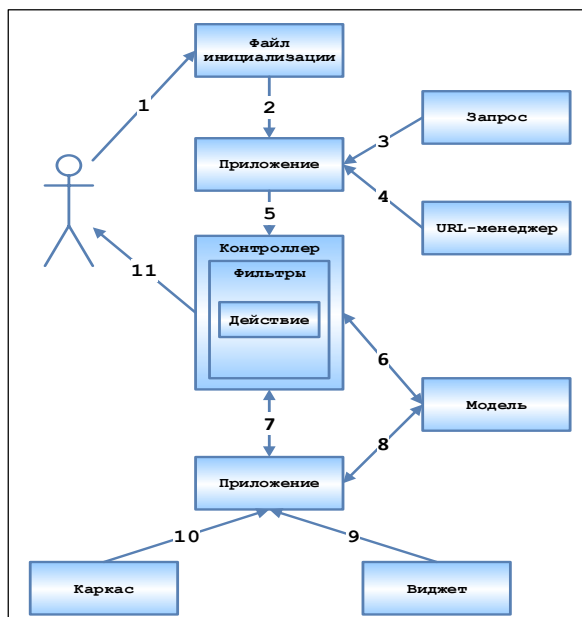


Рис. 2. Схема взаимодействия клиента с сервером приложений

Fig. 2. Client-server interaction scheme

2. Скрипт инициализации создает экземпляр фронт-контроллера и запускает его на выполнение.

3. Фронт-контроллер инкапсулирует контекст обработки запроса, получая подробную информацию о нем через компонент приложения request.

4. Фронт-контроллер определяет запрошенный контроллер приложения и действие, которое должен выполнить контроллер при помощи компонента URL-менеджера. Суть каждого действия определяется внутри соответствующего контроллера.

5. Фронт-контроллер создает экземпляр запрашиваемого контроллера для дальнейшей обработки запроса пользователя. Контроллер определяет соответствие действия пользователя в его классе. Далее создаются и применяются фильтры, связанные с данным действием.

6. Если в действии прописано обращение к БД, оно считывает из нее указанную модель через ее менеджера.

7. Действие подключает указанное представление и передает в него извлеченную модель.

8. Представление получает и отображает атрибуты переданной модели.

9. Если необходимы виджеты, представление подключает их.

10. Формируется представление с переданной моделью и передается в представление макета страницы.

11. Завершается формирование представления, и выводится результат пользователю.

Структурная модель системы

При проектировании сложноорганизованной распределенной системы на первом этапе выявлены объекты предметной области, отнесены к классам с соблюдением разумной степени детализации, определены интерфейсы классов и иерархия наследования, установлен регламент отношений между классами [4, 6]. Функциональная схема системы разделяется на основные подсистемы, между которыми указываются информационные связи и/или связи по управлению, описывается основное назначение подсистем. Клиентская и серверная части осуществляют передачу данных при помощи протокола HTTP. Основой HTTP является технология «клиент-сервер», то есть предполагается существование потребителей (клиентов), которые иницируют соединение и посылают запрос, и поставщиков (серверов), которые ожидают соединения для получения запроса, производят необходимые действия и возвращают обратно сообщение с результатом.

В составе серверной части осуществляется следующий функционал:

- авторизация, отвечающая за регистрацию нового пользователя, аутентификацию и авторизацию уже существующего;

- взаимодействие с БД: конвертируются запросы от сервера в SQL-запросы к БД;

- ведение БД: контролируется корректность записей и их актуальность;

- справочная функция: содержит сведения о системе (руководство пользователю) и о разработчиках;

- прием/отправка сообщений: принимает сообщения от сервера и отправляет сообщения от клиента, проверяет корректность формата сообщения.

В состав клиентской части входят подсистемы:

- визуализации: представляет в удобном графическом виде информацию для пользователя системы;

- приема/отправки сообщений: принимает сообщения от сервера и отправляет сообщения от клиента, проверяет корректность формата передачи данных.

Такой подход к организации структуры веб-ориентированной системы создает условия для обеспечения централизованной обработки, хранения и доставки пространственных данных через сеть Интернет для удаленных пользователей, решающих задачи справочно-информационного и аналитического обслуживания [5].

Данное архитектурное решение обладает рядом преимуществ:

- выполнение независимо от операционной системы;

- возможность использования на мобильных устройствах;

- максимально быстрое распространение среди клиентов;

- минимальная аппаратная платформа;

- автоматическое обновление версий.

Визуальное моделирование системы

Диаграммы деятельности – это один из пяти видов диаграмм, применяемых в UML для моделирования динамических аспектов поведения системы. Диаграмма деятельности – это по существу блок-схема, которая показывает, как поток управления переходит от одной деятельности к другой. Как правило, они применяются, чтобы промоделировать последовательные или параллельные шаги вычислительного процесса.

На рисунках 3, 4 представлены диаграммы деятельности пользователей системы с ролями «Администратор» и «Проверяющий».

Клиентская часть системы

Логика работы клиентской части системы реализована с применением паттерна проектирования «Модуль». Паттерн «Модуль» осуществляет инкапсуляцию приватной информации, состояния

или структуры за счет встроенного в JavaScript механизма замыкания. Реализация паттерна «Модуль» в системе позволяет оборачивать методы и переменные в программные конструкции особого вида, предотвращая попадание методов и переменных в глобальный контекст. Паттерн «Модуль» возвращает только общедоступную часть через механизм API, оставляя внутреннюю реализацию доступной только в пределах модуля [2, 6, 7].

В разработанной системе взаимодействие пользователя с системой осуществляется посредством визуального графического SDI-интерфейса.

Для пользователя с ролью «Администратор» доступны функции создания, удаления, редактирования конкурсов, пользователей, заявок. Для пользователя с ролью «Проверяющий» (интерфейс страницы пользователя с ролью «Проверяющий» показан на рисунке http://www.swsys.ru/uploaded/image/2018_1/2018-1-dop/13.jpg) возможны функции:

- просмотр работ участников конкурсов;
- изменение статуса заявок участников конкурсов;
- формирование дипломов участников конкурсов.

Для пользователя с ролью «Участник» возможны функции:

- создание, удаление, редактирование заявок на участие в конкурсах (интерфейс страницы создания заявок показан на рисунке http://www.swsys.ru/uploaded/image/2018_1/2018-1-dop/14.jpg);
- изменение данных своего профиля;
- просмотр результатов и скачивание диплома.

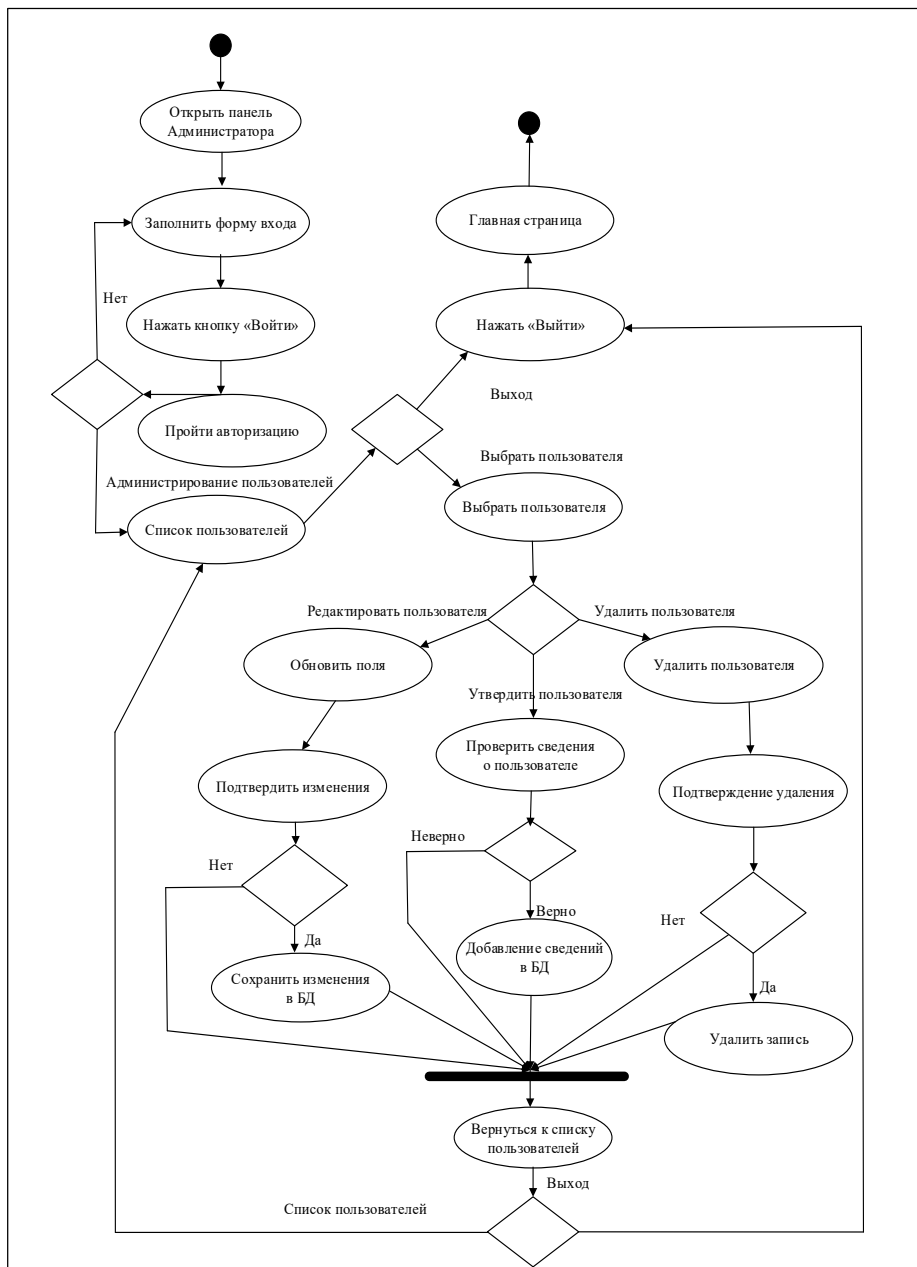


Рис. 3. Диаграмма деятельности пользователя с ролью «Администратор»

Fig. 3. Administrator activity diagram

Заключение

В результате проделанной работы создана модель веб-системы на основе паттернов, на которой построена распределенная веб-система проведения конкурса научно-исследовательских работ «IT & TRANSPORT», позволяющая в режиме реального времени подавать заявки на участие в конкурсе научно-образовательных работ участнику конкурса, проверять, утверждать, отклонять и отправлять на доработку заявки, а также формировать документы с подведением итогов конкурса

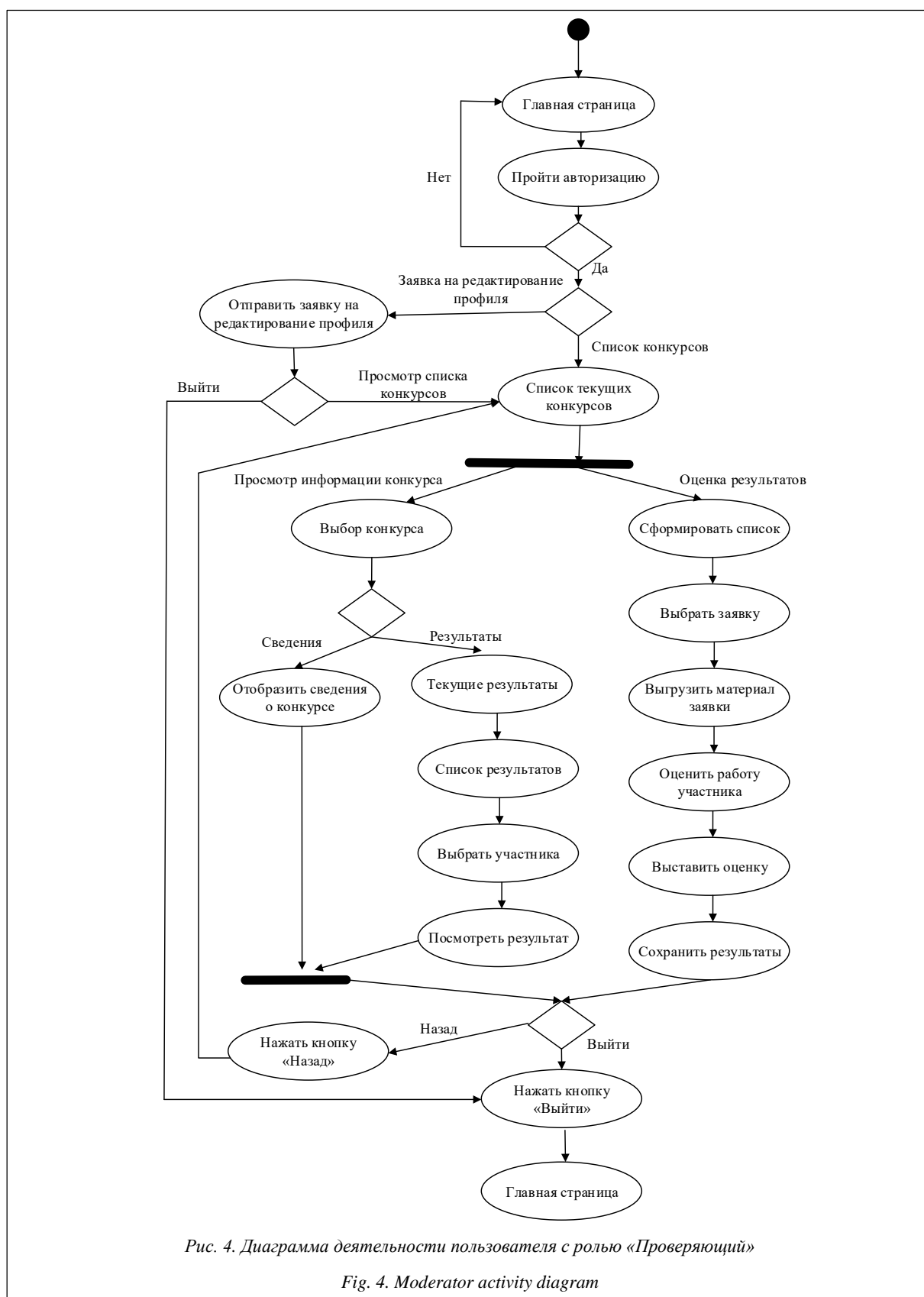


Рис. 4. Диаграмма деятельности пользователя с ролью «Проверяющий»

Fig. 4. Moderator activity diagram

проверяющему, тем самым контролируя весь цикл деятельности по организации и проведению конкурса научно-исследовательских работ.

Литература

1. Антипов В.А., Антипов О.В., Пылькин А.Н. Интеграция распределенных программных приложений на основе ком-

муникационной парадигмы // Математические и компьютерные методы в технических, гуманитарных и общественных науках: коллективная монография. Вып. 3. Пенза–М., 2013. С. 36–67.

2. Михеева Т.И., Головнин О.К., Федосеев А.А. Паттерновое проектирование интеллектуальных транспортных систем // Современные проблемы науки и образования. 2012. № 6. URL: <http://www.science-education.ru/106-7967> (дата обращения: 11.12.2017).

3. Макаров А. Yii. Сборник рецептов. М.: ДМК Пресс, 2013. 372 с.

4. Липаев В.В. Программная инженерия. Методологические основы. М.: ТЕИС, 2006. 609 с.

5. Имамудинов А.Н., Остроглазов Н.А., Головнин О.К. Веб-ориентированная информационная система дислокации объектов транспортной инфраструктуры // Изв. Самарского научного центра РАН. 2016. Т. 17. № 4. С. 739–743.

6. Михеева Т.И. Использование принципов объектно-ориентированного проектирования интеллектуальной транспортной системы // Вестн. Самарского гос. технич. ун-та. Сер.: Физ.-матнауки. 2004. № 34. С. 141–148.

7. Михеева Т.И., Головнин О.К. Паттерны поддержки принятия решений по дислокации технических средств организации дорожного движения // ПИТ-2013: тр. Междунар. науч.-технич. конф. Самара: Изд-во ЦНЦ РАН, 2013. С. 267–272.

Software & Systems

DOI: 10.15827/0236-235X.121.032-037

Received 12.12.17

2018, vol. 31, no. 1, pp. 032–037

A WEB-MODEL OF A DISTRIBUTED REAL-TIME INFORMATION SYSTEM

*N.A. Ostroglazov*¹, *Postgraduate Student, nickitaost@gmail.com*

*A.I. Chugunov*¹, *Postgraduate Student, ugunoff@bk.ru*

*M.A. Kudrin*¹, *Graduate Student, mihail-mtl-kudrin@yandex.ru*

*V.V. Elizarov*¹, *Student, elizarov43@gmail.com*

¹ *Samara National Research University, Moskovskoe Highway 34, Samara, 443086, Russian Federation*

Abstract. The article describes the approaches to solving the problem of efficient use of computing resources in the design of complex web-oriented distributed systems. It also describes the development of a web real-time model of a distributed information system for competitions of academic works.

The system is implemented in Yii framework in PHP. The MVC design pattern (Model View Controller) provides the interaction between the application server and the client application. The “representation” is a user interface with tools for working in the system, and a controller enables interaction between a model and a representation. The client application is based on a design pattern Module using the mechanism of closures in JavaScript.

Such approach to a web system structure creates the conditions to enable centralized processing, storage and delivery of spatial data via the Internet for remote users. The system includes the separation of users by roles: Administrator, Reviewer and Participant. The article describes user activity diagrams with the roles of an Administrator and a Reviewer. Activity diagrams are built on the UML methodology (Unified Modeling Language).

The developed system allows real-time applying for participation in the contest of scientific and educational works as a Contestant, reviewing, approving, rejecting and sending applications for revision, as well as generating documents that summarize the results as a Reviewer. Thus, the described system provides and controls the entire cycle of activities for organizing a contest of academic works in a web system.

Keywords: distributed computing system, software package, web application, server, client, pattern, architecture, management.

References

1. Antipov V.A., Antipov O.V., Pylkin A.N. Integration of distributed software applications based on the communication paradigm. *Matematicheskie i kompyuternye metody v tekhnicheskikh, gumanitarnykh i obshchestvennykh naukakh* [Mathematical and Computer Methods in Technical, Humanitarian and Social Sciences]. Collective Monograph. Moscow, Privolzhskiy Dom Znany Publ., 2013, pp. 36–67 (in Russ.).

2. Mikheeva T.I., Golovnin O.K., Fedoseev A.A. Pattern designing of intelligent transportation systems. *Sovremennye problemy nauki i obrazovaniya* [Modern Problems of Science and Education]. 2012, no. 6. Available at: <http://www.science-education.ru/106-7967> (accessed December 11, 2017).

3. Makarov A. *Yii. Sbornik retseptov* [Yii. Recipe Book]. DMK Press, 2013, 372 p.

4. Lipaev V.V. *Programmnaya inzheneriya. Metodologicheskie osnovy* [Software Engineering. Methodological Basics]. TEIS Publ., 2006, 609 p.

5. Ostroglazov N.A., Imamutdinov A.N., Golovnin O.K. Web centric information system of transportation facilities location. *Izvestiya SNTs RAN* [Izvestia RAS SamSC]. Samara, 2016, vol. 17, no. 4 (4), pp. 267–273 (in Russ.).

6. Mikheeva T.I. Using the principles of object-oriented design of an intelligent transport system. *Vestnik SamGTU. Seriya: Fiziko-matematicheskie nauki* [Jour. of Samara State Tech. Univ., Ser. Physical and Mathematical Sciences]. 2004, no. 34, pp. 141–148 (in Russ.).

7. Mikheeva T.I., Golovnin O.K. Decision support patterns for the deployment of traffic management technical means. *PIТ-2014: tr. Mezhdunar. nauch.-tekhnich. konf.* [PIТ 2014: Trans. Int. Scientific and Tech. Conf.]. Samara, RAS SamSC Publ., 2013, pp. 267–273 (in Russ.).