

Resonances science

PROCEEDINGS OF ARTICLES II INTERNATIONAL SCIENTIFIC CONFERENCE
CZECH REPUBLIC, KARLOVY VARY - RUSSIA, MOSCOW, AUGUST 29-30, 2017



Resonances science

Proceedings of articles II International scientific conference

Czech Republic, Karlovy Vary - Russia, Moscow, August 29-30, 2017

Czech Republic, Karlovy Vary - Russia, Kirov, 2017

UDC 001
BBK 72
P345

Scientific editors:

Fatuev Viktor Aleksandrovich, Doctor of Technical Sciences, Professor, Tula State University
Vasil'eva Elena Nikolaevna, Doctor of Sociological Sciences, Professor, Tyumen State Institute of Culture
Kade Azamat Halidovich, Doctor of Medical Sciences, Professor, Kuban State Medical University
Baranova Irina Vjacheslavovna, PHD, associate professor of the department of management and informatics in technical systems, Moscow State Technological University "STANKIN"

P345 Resonances science: Proceedings of articles II International scientific conference. Czech Republic, Karlovy Vary - Russia, Moscow, August 29-30, 2017 [Electronic resource] / Editors prof. V.A.Fatuev, E.N.Vasil'eva, Az.H.Kade, Ir.V. Baranova. – Electron. txt. d. (1 file 2.9 MB). – Czech Republic, Karlovy Vary: Skleněný Můstek – Russia, Kirov: MCNIP, 2017. –ISBN 978-80-7534-149-5 + ISBN 978-5-00090-126-7.

Proceedings includes materials of the international scientific conference «Resonances science», held in Czech Republic, Karlovy Vary-Russia, Moscow, August 29-30, 2017. The main objective of the conference - the development community of scholars and practitioners in various fields of science. Conference was attended by scientists and experts from Russia, Kazakhstan, Kyrgyzstan.

ISBN 978-80-7534-149-5 (Skleněný Můstek, Karlovy Vary, Czech Republic)
ISBN 978-5-00090-126-7 (MCNIP LLC, Kirov, Russian Federation)

Articles are published in author's edition. Editorial opinion may not coincide with the views of the authors

Reproduction of any materials collection is carried out to resolve the editorial board

© Skleněný Můstek, 2017
© MCNIP LLC, 2017

Table of Contents

Section 1. Technology	7
Баранова И.В., Синьюэ Л., Чжао К. Поддержка жизненного цикла программного обеспечения информационных систем цифрового производства	8
Митасов В.М., Стаценко Н.В. Оценка трещиностойкости железобетонных балок	23
Фатуев В.А. Оптимальная идентификация динамических систем в реальном масштабе времени.....	29
Section 2. Agriculture	39
Кудрин М.Р. Влияние нормативных показателей при размещении молодняка крупного рогатого скота на их рост.....	40
Сухопаров А.И., Попов В.Д., Ерохин И.В. Многокритериальная система оценки технологий производства кормов из трав	45
Section 3. Economics.....	50
Исенова Г.К. Проблемы внедрения проектного менеджмента в государственном секторе постсоветских стран	51
Калкабаева Г.М. Приоритетные направления в обеспечении финансовой стабильности Казахстана.....	67
Майструк Е.В. Проект создания макета эталонной модели в рамках механизма «единое окно»	75
Огородников П.И., Матвеева О.Б., Гусева Е.П., Спешилова И.В. Научно - методические аспекты формализации биотехнических систем в сельскохозяйственном производстве	79

Resonances science

II Международная научная конференция

Чехия, Карловы Вары - Россия, Москва, 29-30 августа 2017

Pshunetlev A.A. Regional sustainable development: fostering social principles and organizational innovations	86
Section 4. Philology	111
Трегубов А.Н. Мотивационное поле и семантическая деривация	112
Section 5. Legal Studies	122
Бакытова М.Е. Лица, осуществляющие давностное владение	123
Section 6. Pedagogy	129
Малолетко Н.Е., Воробьева Т.И. Образовательные ресурсы как средство повышения качества профессионального обучения в вузах МВД России	130
Панфилова О.И. Результаты исследования профессиональной компетентности педагогов в воспитательной деятельности общеобразовательных организаций.....	136
Section 7. Medicine.....	145
Жирнова А.С., Курлаев П.П., Абрамзон О.М., Залошков А.В. Хирургическое лечение фиброаденом молочных желез с использованием топической локализации	146
Залошков А.В., Абрамзон О.М., Курлаев П.П., Жирнова А.С. Миниторакотомный доступ: пути и возможности оптимизации	150
Каде А.Х., Поляков П.П., Липатова А.С., Сотниченко А.С., Богданова Ю.А., Куевда Е.В., Губарева Е.А. Влияние ТЭС-терапии на характер стресс-индуцированной экспрессии c-fos нейронами медиальной префронтальной коры.....	154
Лисняк М.А., Горбач Н.А. Организация системы охраны здоровья профессорско-преподавательского состава вузов (комплексный подход).....	159
Сопуев А.А., Султакеев М.З., Акматов Т.А. Многообразие причин спонтанного гемоторакса.....	168

Resonances science

II Международная научная конференция

Чехия, Карловы Вары - Россия, Москва, 29-30 августа 2017

Тазина С.Я., Семенов Н.А., Трухин И.В. Особенности течения первичного инфекционного эндокардита 182

Section 8. Psychology185

Акимова М.К., Персиянцева С.В. Влияние средовых условий на уровень гражданской идентичности молодых россиян 186

Section 9. Sociology205

Васильева Е.Н. Концептуализация социального обслуживания семей с детьми в социологическом дискурсе 206

Section 10. Earth Sciences217

Варежкина Л.С. Экологический мониторинг водоканала города Дмитрова 218

Resonances science

II Международная научная конференция

Чехия, Карловы Вары - Россия, Москва, 29-30 августа 2017

SECTION 1. TECHNOLOGY

ПОДДЕРЖКА ЖИЗНЕННОГО ЦИКЛА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ ЦИФРОВОГО ПРОИЗВОДСТВА

БАРАНОВА И.В., Синьюэ Л., Чжао К.

Россия, МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ
«СТАНКИН»

Аннотация. Рассмотрена роль Web-приложений, как инструмента повышения эффективности создания и функционирования цифрового производства. Исследована структура жизненного цикла Web-приложения. Дан анализ целесообразности использования архитектурного шаблона на основе «тонкого» клиента» при проектировании Web-приложений. На примере создания Web-приложений исследован процесс моделирования разработки программного обеспечения цифрового производства.

Ключевые слова: Программное обеспечение. Информационная система. Цифровое производство. Web-приложение. Жизненный цикл. Архитектурный шаблон. Модель «тонкого клиента».

Abstract. The role of Web-applications as a tool for increasing the efficiency of the creation and operation of digital production is considered. The structure of the life cycle of a Web application has been studied. The analysis of the expediency of using an architectural template based on the "thin" client is presented in the design of Web applications. On the example of creating Web applications, the process of modeling the development of digital production software has been studied.

Keywords: Software. Information system. Digital production. Web application. Life cycle. Architectural template. The thin client model.

Введение

Мировые тенденции повышения эффективности реализации бизнес-процессов связаны с переходом к цифровому производству. Подобная ситуация характерна и для России. Цифровизация кардинальным образом изменит ситуацию в российской экономике. Приоритет получают новые формы представления информации, контента и знаний. Совершенствование принципов организации вычислений будет связано с использованием суперкомпьютеров, созданием новых интерфейсов «человек – цифровая среда», конвергенцией информационных платформ.

Использование цифровых технологий выдвигает более высокие требования к информационным системам и сервисам. В этой связи возрастает актуальность решения таких задач, как совершенствование программного обеспечения, формирования стратегии поддержки его жизненного цикла. Подобная стратегия наиболее актуальна для высокотехнологичных предприятий, ориентированных на цифровизацию производства. Такие предприятия для обеспечения эффективности своей деятельности формируют единое информационное пространство, выполняют инновационные разработки и используют их результаты, применяют высокоавтоматизированные организационно-производственные структуры.

1. Web-приложения как инструмент повышения эффективности создания и функционирования цифрового производства

Web-приложение представляет собой клиент-серверное приложение, в котором клиентом выступает браузер, а сервером – web-сервер. При этом хранение данных осуществляется, преимущественно, на сервере, а обмен данными происходит по сети. Web-приложения обеспечивают управление массивами сложно-структурированной информации, повышая эффективность взаимодействия с пользователями через Internet [5, 7, 9].

Реализуя стратегию поддержки жизненного цикла программного обеспечения информационных систем (например, Web-приложений), высокотехнологичные предприятия повышают качество управления бизнес-процессами, протекающими в различных сферах деятельности предприятий. Используя Web-приложения, пользователь имеет возможность дополнять массивы информации новыми данными, а также осуществлять их модификацию.

Реализация стратегии поддержки жизненного цикла программного обеспечения информационных систем обеспечивает высокотехнологичному предприятию ряд преимуществ, в частности, устойчивое функционирование различных подсистем предприятия в едином информационном пространстве. Это достигается за счет [1]:

- возможности информационного наполнения приложения текстовой, графической, анимационной и мультимедийной информацией;
- удобного дизайна пользовательского интерфейса (Web-интерфейса), чему способствует применение каскадных таблиц стилей (Cascading Style Sheets – CSS)¹, различных графических элементов и т.д. Причем изменения пользовательских предпочтений требуют непрерывного совершенствования дизайна и эргономики интерфейса.
- включения в систему навигации различных разделов, меню и т.д. Необходимость постоянного совершенствования системы навигации является следствием мировых тенденций развития Web-приложений. Эти тенденции заключаются в динамичном росте уровня гибкости навигационных систем и минимизации затрат на их модификацию.

¹CSS (Cascading Style Sheets – каскадные таблицы стилей) – формальный язык описания внешнего вида документа, созданного с использованием языка разметки. Может применяться как средство описания, оформления внешнего вида веб-страниц, созданных с помощью языков разметки HTML и XHTML, а также к XML-документам.

Resonances science

II Международная научная конференция

Чехия, Карловы Вары - Россия, Москва, 29-30 августа 2017

Для пользователя важна ценность информационного наполнения, определяемая полнотой, оперативностью и достоверностью информации, содержащейся в Web-приложении. Ценность информационного наполнения определяется такими параметрами, как скорость ориентирования по сайту (понятность, структурированность и наглядность сайта), уровень дизайна сайта (узнаваемый стиль, красота, эргономичные цвета, шрифты и т.д.), величина платы за доступ к сайту, защищенность информации на сайте и т.д. Кроме того, для пользователя существенное значение имеет функциональность Web-приложений [5, 7, 9].

Поэтому при проектировании Web-приложения необходимо учитывать ряд их особенностей. Во-первых, особенность реализации клиент-серверной архитектуры. Реализация подобной архитектуры предполагает высокий уровень абстракции и централизации, предполагающей передачу серверу выполнение всех основных функций. Во-вторых, наличие фактора гетерогенности, т.е. разнообразия используемых программных и аппаратных платформ. Причем появление новых информационных технологий приводит к быстрому их обновлению. В-третьих, территориальная распределенность, достигаемая средствами Internet. Эта особенность обуславливает многопользовательский режим работы Web-приложений, значительные колебания нагрузки и наличие различных категорий пользователей.

Участие в проектировании Web-приложений специалистов разных предметных областей (дизайнер, верстальщик, Web-программист, администратор сети, администратор базы данных и т.д.) требует не только координирования их деятельности, но и создания соответствующей методологии проектирования, отвечающей особенностям цифровой среды. В условиях цифровизации подобная методология должна учитывать особенности реализации этапов жизненного цикла Web-приложений. Это предполагает, что все этапы проектирования и эксплуатации Web-приложений реализуются на единой методической платформе,

формирование которой осуществляется с использованием CALS-технологий.

2. Структура жизненного цикла Web-приложения

В процессе создания и использования Web-приложения реализуется последовательность этапов. Эти этапы формируют структуру жизненного цикла Web-приложения. Подобная структура охватывает такие этапы, как определение требований к Web-приложению, его проектирование, реализация, тестирование, эксплуатация и сопровождение Web-приложения [9, 12]. С точки зрения разработчика подход, ориентированный на применение при создании и использовании Web-приложения CALS-технологий, является целесообразным, т.к. позволяет решить задачу повышения эффективности управления процессом информационного обеспечения цифрового производства.

Каждый из перечисленных этапов жизненного цикла имеет свои особенности реализации, которые в значительной степени определяются ограничениями, накладываемыми на процесс создания и использования Web-приложения. Так на этапе определения требований к Web-приложению, как правило, отсутствуют четко сформулированные требования к функциональности, производительности, информационной структуре и навигации [2]. Кроме того, на других этапах жизненного цикла происходит постоянная модификация этих требований. Кроме того, существует фактор неопределенности при выборе дизайнерских и эргономических решений пользовательского интерфейса.

Поэтому в составе данного этапа выделяют две составляющие, во-первых, составляющая, предполагающая определение начальных требований, а во-вторых, составляющая, в рамках которой производится разработка прототипов и уточнение требований.

На этапе определения начальных требований аналитики группы разработки совместно с заказчиком, формируют упрощенную спецификацию

функциональных требований к Web-приложению. Кроме того, на этом этапе также составляется спецификация нефункциональных требований. Подобная спецификация включает требования к производительности и надежности приложения, переносимости его на разные компьютерные платформы, требования к эксплуатации и т.д. Результатом данного этапа является начальная спецификация Web-приложения. Как правило, результат оформляется в виде технического задания, согласно которому в дальнейшем осуществляется процесс разработки Web-приложения.

На этапе разработки прототипов и уточнения требований аналитики, проектировщики и разработчики пользовательского интерфейса, активно взаимодействуя с заказчиком, формируют ряд прототипов Web-приложения. Изучая функционирование прототипов, заказчик уточняет и дополняет свои первоначальные требования, а также предоставляет информацию для составления более подробной спецификации функциональных требований. Кроме того, на этом этапе уточняются требования к пользовательскому интерфейсу.

Результатом выполнения данного этапа являются спецификации функциональных требований и требований к предметной области, а также программные модули каркаса приложения. Спецификация функциональных требований охватывает подробные описания вариантов использования Web-приложения, планы его системного тестирования и аттестации. На основе спецификации требований к предметной области осуществляется программная реализация исполнительных модулей информационной системы. Программные модули каркаса приложения реализуют управляющую логику Web-приложения в соответствии со спецификацией функциональных требований.

На этапе проектирования необходимо обеспечить, во-первых, высокий уровень гибкости и возможность быстрого наращивания функциональности Web-приложения, во-вторых, совместимость с различными внешними клиентскими системами, а в-третьих, интеграцию с имеющейся

инфраструктурой, включая системы управления базами данных, корпоративные приложения, системы защиты информации и т.д. Однако отсутствие методик проектирования Web-приложений, адекватных требованиям цифровой среды высокотехнологичного предприятия, затрудняет решение подобных задач. Кроме того, на эффективность реализации этапа проектирования существенное влияние оказывает высокая скорость обновления программных и аппаратных платформ.

Поэтому данный этап реализуется, как совокупность ряда составляющих жизненного цикла Web-приложений, включающих, во-первых, кодирование и тестирование исполнительных модулей, а во-вторых, сборку, системное тестирование и аттестацию Web-приложения.

Кодирование и тестирование исполнительных модулей Web-приложения выполняется после получения спецификации требований к предметной области. Кодированию и тестированию подлежат такие модули, как объекты доступа к данным, команды, генераторы представления [8]. Объекты доступа к данным обеспечивают интерфейс к хранилищам данных и выполняют операции по добавлению, удалению, изменению данных и составлению их выборки по заданным параметрам. Команды реализуют функции предметной области по управлению данными, а генераторы представления обеспечивают заполнение полей и слотов во фрагментах HTML, созданных группой разработки пользовательского интерфейса. Результатом выполнения работ будут являться перечисленные нами программные модули, а также их описания.

Здесь могут возникнуть проблемы, которые обусловлены рядом причин. Во-первых, различием программно-аппаратных баз среды разработки и среды эксплуатации. Во-вторых, сложностью отладки, которая обусловлена распределенной структурой среды разработки. В-третьих, участием на этом этапе специалистов различного профиля, координирование деятельности которых бывает трудно обеспечить. Особенности тестирования часто обусловлены отсутствием специализированных методик

автоматизированного системного тестирования таких параметров Web-приложения, как функциональность, производительность и т.д.

Далее производится сборка Web-приложения из функционального прототипа и исполнительных модулей системы, а затем осуществляется системное тестирование приложения. При успешном тестировании принимается решение об успешной аттестации разработанного Web-приложения. В противном случае формируются отчеты об ошибках или не соответствиях Web-приложения спецификации. Тогда цикл разработки Web-приложения повторяется, начиная с этапа прототипирования. Прошедшее аттестацию Web-приложение передается для использования заказчику.

На этапе эксплуатации может быть организовано сопровождение Web-приложения, предполагающее, например, модификацию пользовательского представления путем изменения фрагментов HTML. На этом этапе эксплуатации может возникнуть потребность оперативного изменения дизайна пользовательского интерфейса, добавления новой функциональности или изменения информационной структуры и навигации. Кроме того, на этом этапе могут появиться «пиковые» нагрузки на серверах, следствием чего выступает потребность в оперативном масштабировании.

3. Моделирование процесса разработки программного обеспечения цифрового производства

Процесс создания программного обеспечения цифрового производства может быть описан с использованием различных моделей. Наиболее часто используются такие модели, как каскадная, эволюционная, модель формальной разработки, модель разработки на основе ранее созданных компонентов программного обеспечения.

На этапе определения требований к программному обеспечению используется эволюционная модель (модель прототипирования). В этом

случае предполагается, что разработчики Web-приложений используют инструменты автоматического построения прототипов по схемам функционирования приложения. Для этапа проектирования программного обеспечения характерно применение эволюционной модели, основанной на подходе пробных разработок, и модели, разработка, которой осуществляется на основе ранее созданных компонентов. Условием эффективного применения подобных моделей является наличие инфраструктуры для накопления, поиска и использования типовых компонентов Web-приложения.

Этап реализации характеризуется либо продолжением использования эволюционной модели, созданной на предыдущем этапе жизненного цикла программного обеспечения, либо модели формальной разработки. Для использования подобных моделей на этапе реализации в распоряжении создателей программного обеспечения должны быть инструменты автоматической генерации программных модулей на целевых языках по формальным спецификациям.

Этап тестирования программного обеспечения ориентирован на каскадную модель, реализация которой осуществляется с помощью инструментов автоматического тестирования компонентов программного обеспечения, объединенных в подсистемы. При этом составляются планы тестирования, в основе которых лежат варианты использования программного обеспечения. На этапе эксплуатации и сопровождения программного обеспечения используется каскадная модель. Целесообразность ее применения на этом этапе обусловлена тем, что развитие системы происходит итерационно и расширение возможностей требует проведения полного цикла разработки.

Преимущества перечисленных нами моделей интегрирует гибридная модель создания программного обеспечения, представленная на рисунке 1. Информационное обеспечение цифрового производства целесообразно ориентировать на эту модель.

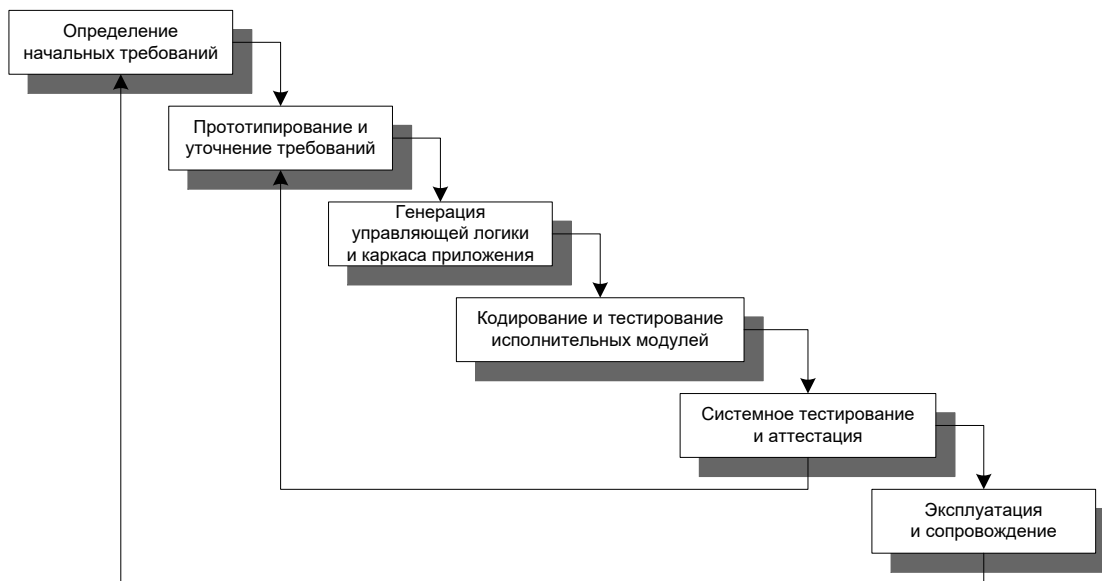


Рис. 1. Гибридная модель процесса разработки программного обеспечения

Архитектурные шаблоны Web-приложений отражают фундаментальную структурно-организационную схему программных систем цифрового производства [6, 11]. В настоящее время существует большое количество информационных технологий и программных продуктов, влияющих на выбор архитектуры Web-приложений. Поэтому в современных условиях эта архитектура может быть различной.

Разнообразие архитектур Web-приложений также во многом обуславливается тем, что интегрируемые в единое информационное пространство высокотехнологичного предприятия ранее созданные информационные продукты и технологии могут выступать, как часть архитектуры Web-приложений.

Например, архитектурный шаблон на основе «тонкого» клиента» (Thin Web Client) представляет собой конструкцию «Модель-Представление-Контроллер» (Model-View-Controller – MVC). Архитектурный шаблон на основе «тонкого» клиента» целесообразно использовать при создании гибких, т.е. адаптируемых к различным ситуациям элементов Web-

приложения [6, 11]. При этом гибкость может рассматриваться, как со статической, так динамической точек зрения. Под статической гибкостью подразумевается возможность добавления в приложение нового класса представления или контроллера, а под динамической – возможность замены объекта представления или контроллера во время работы Web-приложения.

С точки зрения архитектурного шаблона «тонкого клиента» Web-приложения относятся к классу так называемых «реактивных систем». Состояние подобных систем изменяется под влиянием внешних воздействий. Наличие этих воздействий обуславливает выполнение системой определенных операций. Такая ситуация обусловлена тем, что Web-приложения, созданные на основе архитектурного шаблона «тонкого клиента», базируются на протоколе HTTP², который не поддерживает соединения и состояния, а является процедурой оперативного реагирования на внешние запросы. В архитектурном шаблоне «тонкого клиента», использующем процедуру «Модель-Представление-Контроллер» (Model-View-Controller – MVC), сложный элемент описывается с помощью следующих терминов:

- модель (model), представляющая совокупность данных и отражающая состояние элемента, а также средства, обеспечивающие изменение состояния;
- представление (view) элемента, которое может быть как визуальное, так и не визуальное;
- контроллер (controller), отражающий управляющую функциональность элемента, которая обеспечивает соответствие

² Протокол HTTP (HyperText Transfer Protocol – «протокол передачи гипертекста») – протокол прикладного уровня передачи данных, в основе которого лежит клиент-серверная технология.

между операциями, выполняемыми с его представлением, и состоянием, образуемым моделью.

Тогда, рассматривая архитектурный шаблон «тонкого» Web-клиента как реактивную систему, функциональность подсистемы «контроллер» можно представить, используя модель конечного автомата³ [3, 4]. Подобные модели применяются в различных нотациях, используемых при разработке программного обеспечения, например, в унифицированном языке моделирования – UML (Unified Modeling Language) [10, 12]. Этот язык используется для объектного моделирования процессов разработки программного обеспечения и бизнес-процессов, системного проектирования и т.д. Следовательно, «контроллер» Web-приложения можно представить в виде совокупности конечных автоматов. При этом каждому элементу подсистемы «представление» будет соответствовать один автомат подсистемы «контроллер».

На рисунке 2 представлена структурная схема Web-приложения, построенная на основе модели «тонкого клиента».

Представленная на рисунке 2 схема включает в себя следующие элементы:

- пользователь, который использует Web-приложение для достижения поставленных целей;
- рабочая станция, представляющая собой персональный компьютер пользователя, который содержит браузер-программу просмотра текста HTML и навигации в сети Internet;
- Web-сервер, выступающий в виде программно-аппаратного комплекса, в котором хранятся Web-приложения;
- Web-приложение, являющееся программной системой, которая предоставляет пользователю функциональность, необходимую для

³ Модель конечного автомата – способ описания системы как абстрактной математической машины. В модели переменные состояния представляют состояния машины, а функции перехода описывают способ изменения переменных.

решения бизнес-задач. Web-приложение охватывает ряд компонентов, включая такие подсистемы, как «Представление», «Контроллер», «Модель», а также систему управления базами данных, унаследованные и смежные системы.

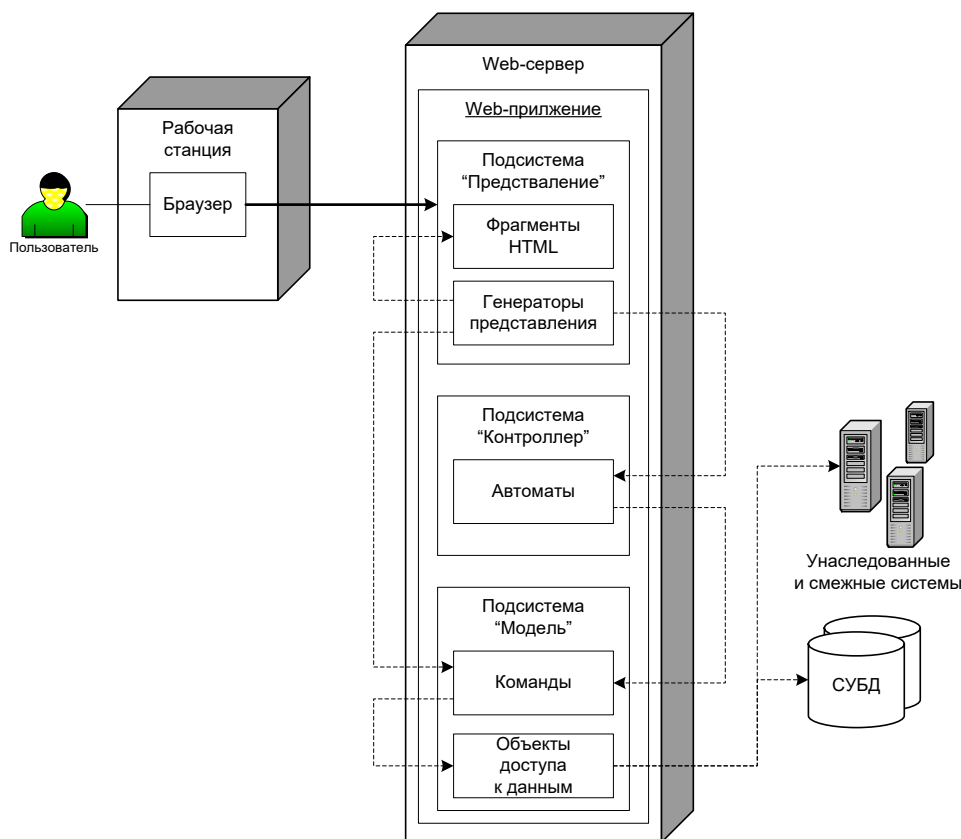


Рис. 2. Структурная схема Web-приложения, построенная на основе модели «тонкого клиента»

Подсистема «Представление», обеспечивая сборку пользовательского представления на языке HTML, объединяет фрагменты этого языка и генераторы представления. Фрагменты HTML охватывают коллекцию фрагментов представления на языке HTML, комбинации которых образуют пользовательское представление в зависимости от состояния системы. Фрагменты содержат слоты для вставки представления других компонентов, а также поля для вставки динамических данных. Генераторы

представления представляют собой программные модули, заполняющие поля фрагментов HTML динамическими данными.

Подсистема «Контроллер» обеспечивает корректное взаимодействие пользователя с системой, является основным диспетчером бизнес-процесса и состоит из компонентов, называемых автоматами, т.е. программных модулей, реализующих модель расширенного конечного автомата. В рамках этой подсистемы также осуществляется управление потоком событий, компоновкой пользовательского представления, а также происходит выполнение команд над моделью.

Подсистема «Модель» отражает предметную область и содержит информационные сущности, а также методы управления ими. Подсистема охватывает такие компоненты, как команды и объекты доступа к данным. Команды и объекты доступа к данным являются программными модулями. С помощью команд реализуются бизнес-правила и функции предметной области, а объекты доступа к данным отражают информационные сущности и реализуются посредством использования систем управления базами данных, унаследованных и смежных систем.

Заключение

Таким образом, Web-приложения играют важную роль в цифровизации производственных процессов высокотехнологичных предприятий, выступая инструментом повышения эффективности использования информационных систем в рамках единого информационного пространства цифрового производства. В этой ситуации возрастает актуализация задачи создания оптимальной структуры жизненного цикла Web-приложений, а также стратегии управления этой структурой. Формирование подобной стратегии целесообразно ориентировать на использование архитектурного шаблона на основе «тонкого» клиента», моделируя различные процессы, входящие в структуру жизненного цикла Web-приложений. Это позволит минимизировать затраты на создание

Resonances science

II Международная научная конференция

Чехия, Карловы Вары - Россия, Москва, 29-30 августа 2017

единого информационного пространства в процессе цифровизации деятельности высокотехнологического предприятия.

Список литературы:

1. *Гудов А.М., Завозкин С.Ю., Трофимов С.Н.* Технологии разработки программного обеспечения: Учебное пособие. Кемерово: изд. КГУ, 2009. – 138 с.
2. *Вигерс К.* Разработка требований к программному обеспечению / 3-е изд., доп. / Пер. с англ.. С-Пб.: БХВ-Петербург, 2017. – 736 с.
3. *Кабылова Д.А., Когай Г.Д.* Расширенный конечный автомат для тестирования мобильных приложений // Молодой ученый, № 21 (125), 2016. – с. 141 – 144.
4. *Карпов Ю.Г.* Теория автоматов: Учебник для вузов. С-Пб.: Питер, 2003. – 206 с.
5. *Макдональд М.* Веб-разработка: Исчерпывающее руководство / Пер. с англ.. С-Пб.: Питер, 2017. – 640 с.
6. *Паван В.* Шаблоны проектирования веб-приложений / Пер. с англ. М.: Эксмо, 2011. – 870 с.
7. *Пьюривал С.* Основы разработки веб-приложений / Пер. с англ.. С-Пб.: Питер, 2015. – 272 с.
8. *Скотлмайер Д.* Тестирование Web-приложений / Пер. с англ.. М.: КУДИЦ-образ, 2003. – 240 с.
9. *Скотт Б., Нейл Т.* Проектирование веб-интерфейсов / Пер. с англ. М.: Символ-Плюс, 2010. – 352 с.
10. *Скотт К.* UML. Основные концепции / Пер. с англ. М.: ИД «Вильямс», 2002. – 144 с.
11. *Шкляр Л., Розен Р.* Архитектура веб-приложений / Пер. с англ. М.: Эксмо, 2011. – 640 с.
12. *Коналлен Дж.* Разработка Web-приложений с использованием UML / Пер. с англ. М.: ИД «Вильямс», 2001. – 288 с.