

*На правах рукописи*

**МАЛИКОВА Карина Эмильевна**

**ТЕХНОЛОГИЯ РАЗРАБОТКИ ИНТЕРНЕТ-ПРИЛОЖЕНИЙ  
НА ОСНОВЕ  
ВСТРОЕННЫХ ДИНАМИЧЕСКИХ МОДЕЛЕЙ И XML**

**Специальность 05.13.11**

**Математическое и программное обеспечение  
вычислительных машин, комплексов и компьютерных сетей**

**АВТОРЕФЕРАТ**

**диссертации на соискание ученой степени  
кандидата технических наук**

**Уфа – 2011**

Работа выполнена в ФГБОУ ВПО  
«Уфимский государственный авиационный технический университет»  
на кафедре автоматизированных систем управления

Научный руководитель      д-р техн. наук, проф.  
**МИРОНОВ Валерий Викторович**

Официальные оппоненты      д-р техн. наук, проф.  
**МАРТЫНОВ Виталий Владимирович**  
Уфимский государственный авиационный  
технический университет  
канд. техн. наук,  
**АЛИМБЕКОВА Софья Робертовна**  
НИИ технических систем «Пилот»

Ведущая организация      ОАО «Уфимское научно-производственное  
предприятие «Молния»

Защита диссертации состоится 28 октября 2011 г. в 14:00 часов  
на заседании диссертационного совета Д-212.288.07  
при Уфимском государственном авиационном техническом университете  
по адресу: 450000, Уфа-центр, ул. К. Маркса, 12

С диссертацией можно ознакомиться в библиотеке университета

Автореферат разослан 23 сентября 2011 г.

Ученый секретарь  
диссертационного совета  
д-р техн. наук, профессор



С. С. Валеев

## ОБЩИЕ ПОЛОЖЕНИЯ

### Актуальность темы исследования

Для обеспечения доступа к корпоративным информационным ресурсам широко применяются интернет-приложения, размещенные на веб-серверах и обслуживающие разнообразные информационные запросы клиентов через глобальную или локальную сеть на основе информации из серверных баз данных. За последние годы инфраструктура интернет-приложений расширилась за счет динамической обработки контента, динамического поведения и представления информационного наполнения приложений, соответствующего потребностям пользователей при управлении различными бизнес-процессами на основании гибких сценариев на сервере. Создание таких приложений зачастую включает достаточно сложные этапы проектирования функциональности приложений и последующего программирования в той или иной инструментальной среде. При этом интернет-приложения, обслуживающие сложные бизнес-процессы, изначально определяются на высоком уровне абстракции, абстрагируясь от ранней детализации архитектуры и специфики реализации, а полученная модель может быть транслирована в рабочее интернет-приложение. Проектирование функциональности интернет-приложения как информационной системы базируется на методологии структурного анализа, предполагающей разработку на этапе проектирования комплекса системных моделей функционирования, которые в дальнейшем служат основой для реализации, тестирования, установки и эксплуатации.

Функционирование интернет-приложения на концептуальном уровне абстракции может быть представлено в виде динамической модели с конечным множеством состояний, в которой заданы состояния функционирования, возможные переходы состояний, действия, ассоциированные с состояниями и переходами. Динамические модели такого рода (конечные автоматы, сети Петри, графы переходов, цепи Маркова и т. п.) достаточно хорошо исследованы как в плане свойств и возможностей, так и в плане применения для конкретных процессов<sup>1</sup>.

Что касается перехода от динамической модели к конечной реализации интернет-приложения, то сейчас он не имеет поддержки в форме научно обоснованных концепций, моделей, методов. Таким образом, возникает научно-техническая задача, актуальная для программной инженерии, — создание научно обоснованного подхода к проектированию программного обеспечения интернет-приложений исходя из динамических моделей их функционирования.

---

<sup>1</sup> Вопросы являются предметом многих исследований как в нашей стране, так и за рубежом Дж. Питерсон, В. Е. Котов (сети Петри), R. Hennicker, J. Conallen (UML). В УГАТУ вопросы использования динамических моделей исследовались в школах профессоров Г. Г. Куликова, Л. Р. Черняховской (сети Петри), В. В. Миронова и Н. И. Юсуповой (иерархические ситуационные модели), различные аспекты иерархических ситуационных моделей исследовались в рамках кандидатских диссертаций Ю. Б. Головкина, Р. А. Ярцева, Л. Е. Гончар, О. Н. Сметаниной, А. Н. Ситчихина, Р. Ф. Ахметшина, Т. А. Гарифуллина, Г. Р. Шакировой.

**Объект исследования** — процесс создания программного обеспечения для построения и функционирования интернет-приложений.

**Предмет исследования** — применение динамических моделей как основы для управления программным обеспечением интернет-приложений.

**Цель исследования** — уменьшение трудоемкости построения интернет-приложений на основе динамических моделей.

**Задачи**, решаемые для достижения цели, включают разработку:

- 1) концепции построения интернет-приложений на основе встроенных динамических моделей;
- 2) метамоделей для задания встроенных динамических моделей интернет-приложений;
- 3) метода интерпретации встроенной динамической модели на основе отслеживания текущих состояний множества клиентов;
- 4) исследовательского прототипа интернет-приложения со встроенной динамической моделью с целью программной реализации предложенных концепции, метамоделей, метода и проверки их работоспособности.

**Методика исследования.** В работе использовались технологии и методы построения интернет-приложений, объектно-ориентированного программирования, системного анализа, теории множеств, ситуационного управления, иерархических моделей, моделирования сложных процессов.

**Результаты, выносимые на защиту, их новизна и значение:**

На защиту выносятся:

- 1) концепция построения интернет-приложений, предполагающая встраивание динамической модели функционирования (в том числе — взаимодействия с пользователем) в серверный сценарий и интерпретацию ее в процессе функционирования приложения;
- 2) мета модель для задания встроенных динамических моделей в виде иерархии графов, определяющих множество возможных состояний и переходов между ними, ориентированная на интернет-приложения, взаимодействующие с множеством клиентов;
- 3) метод интерпретации встроенной динамической модели интернет-приложения на основе отслеживания текущих состояний множества клиентов;
- 4) исследовательский прототип интернет-приложения со встроенной динамической моделью, реализующий предложенные концепцию, мета модель и метод, подтверждающий их работоспособность.

**Новизна результатов в целом** состоит в применении встраиваемых динамических моделей в новой области — в интернет-приложениях — с учетом ее специфики — многопользовательского клиент-серверного взаимодействия. При этом *новизна по существу* выражается в следующем:

– предложенная концепция отличается тем, что с целью обеспечения многопользовательского режима на веб-сервере размещается динамическая модель, общая для всех клиентов, и множество клиентских экземпляров памяти текущего состояния, один из которых выбирается в сеансе интерпретации по результатам аутентификации клиента;

– предложенная метамодель отличается тем, что с целью обеспечения возможности задавать сценарии диалогового взаимодействия с пользователем в ней предусмотрены элементы-действия, ассоциированные с текущими состояниями и специфицирующие элементы управления интерфейса пользователя;

– предложенный метод интерпретации отличается тем, что с целью управления диалоговым взаимодействием с пользователем в дополнение к первому проходу интерпретации, на котором в соответствии с параметрами клиентского запроса контролируется текущее состояние, предусмотрен второй проход, на котором в результате обработки элементов-действий, ассоциированных с текущими состояниями, формируется код, отправляемый клиенту в ответ на запрос;

– разработанный исследовательский прототип интернет-приложения отличается тем, что с целью подтверждения работоспособности полученных результатов в нем реализованы метод интерпретации на языке серверных сценариев PHP, динамическая модель, соответствующая метамодели в форме документа XML, постоянно хранящегося на веб-сервере, память текущего состояния анонимных клиентов — в форме сессионных переменных, а аутентифицированных клиентов — в форме базы данных XML, сохраняемой между сеансами.

**Значение результатов для методологии** взаимодействия клиент / сервер в веб выражается в том, что идея использования встроенных динамических моделей способствует устранению разрыва между этапами логического проектирования функциональности и реализации приложения в той или иной инструментальной среде. Значение результатов для практики разработки интернет-приложений обусловлено тем, что использование встроенных динамических моделей высокого уровня абстракции снижает трудоемкость программирования серверных сценариев интернет-приложений.

**Внедрение результатов.** Практическая значимость результатов подтверждается их использованием в УГАТУ в учебном процессе и в научно-производственной фирме «РД-Технология» в качестве методики построения интернет-приложений на основе встроенных динамических моделей.

**Связь с плановыми исследованиями.** Работа выполнена в рамках плановых исследований кафедры АСУ УГАТУ по разработке информационных технологий веб-портала. Исследование поддержано грантом РФФИ № 10-07-00167-а «Электронные документы со встроенными динамическими моделями».

**Апробация и публикации.** Основные положения, представленные в диссертации, были заявлены и обсуждены на 6 конференциях всероссийского и международного уровня. Список публикаций автора по теме диссертации включает 11 публикаций, в том числе 3 статьи в рецензируемом научном журнале из списка ВАК (31 с.), 8 публикаций в трудах конференций всероссийского и международного уровней (27 с.), 3 публикации выполнены без соавторов (10 с.).

**Структура диссертации.** Диссертация состоит из введения, пяти глав, заключения, приложения, библиографического списка 115 наименований, всего на 181 листе.

## ОСНОВНОЕ СОДЕРЖАНИЕ

**Во введении** обсуждается актуальность решаемой научной задачи, формулируются цель работы, задачи, решаемые для достижения поставленной цели, результаты, выносимые на защиту. Обсуждается методика исследования, раскрывается научная новизна полученных в ходе исследования результатов, их значение для теории и практики. Рассматривается связь работы с плановыми исследованиями.

**В первой главе** анализируются существующие подходы к концепции построения интернет-приложений на основе моделей. Обсуждаются технологии построения интернет-приложений и методы обеспечения контроля пользовательских текущих состояний.

В настоящее время глобальное информационное пространство представлено множеством интернет-приложений, т. е. программных компонентов, функциональные возможности которых обеспечиваются сервером и доставляются конечным пользователям по распределенной сети интернет с помощью интернет-протоколов. Построение интернет-приложения, обслуживающего достаточно сложный бизнес-процесс, можно представить в виде двух этапов: проектирования концептуальной модели обработки и программирования серверного сценария. Сложность перехода от первого этапа ко второму связана с различными уровнями абстракций данных этапов: на первом этапе рассматриваются состояния и переходы, а на втором – конструкции языка программирования серверного сценария (PHP, ASP, ASP.NET и др.). Для снижения трудоемкости данного перехода целесообразно автоматизировать процесс построения серверного сценария на основе динамической модели соответствующего бизнес-процесса.

Как правило, в качестве фундамента связи между пользователем и сервером применяется протокол HTTP (HyperText Transfer Protocol), известным ограничением которого является отсутствие учета информации о текущем состоянии, в результате чего каждая транзакция изолирована от предыдущих и последующих. Для преодоления этого ограничения серверные платформы, базирующиеся на HTTP, используют механизмы, позволяющие сохранить данные пользователя между отдельными шагами сеанса. Вместе с тем, обеспечение контроля текущего состояния сложных бизнес-процессов без привязки к пользовательскому обозревателю или платформе реализации остается достаточно трудоемкой задачей при разработке интернет-приложений.

В дополнение в первой главе обсуждаются концепции разработки интернет-приложений на основе моделей и формулируются их недостатки. *Во-первых*, в настоящее время концепции не являются полностью проработанными, и предлагаемые инструменты не в состоянии осуществить доведение моделей до конечной реализации. *Во-вторых*, для построения функционирующего интернет-приложения необходимо создание нескольких моделей, между

которыми надлежит сохранять соответствие. *В-третьих*, децентрализация бизнес-логики интернет-приложения по моделям требует значительных временных и финансовых затрат при переходе на иную платформу реализации, среду программирования или внесении значительных корректив. *В-четвертых*, моделирование динамических аспектов затруднено в связи с отсутствием формального и однозначного описания правил интерпретации.

На основе результатов проведенного анализа в главе обосновывается необходимость и актуальность использования XML-технологий для построения интернет-приложений. Формулируются цели и задачи исследования, включающие разработку концепции построения интернет-приложений на основе встроенных динамических моделей, разработку метода интерпретации встроенной динамической модели на основе отслеживания текущих состояний и реализацию исследовательского прототипа интернет-приложения.

**Вторая глава** посвящена разработке концепции построения интернет-приложений на основе встроенных динамических моделей и XML. Обсуждаются вопросы обеспечения обслуживания клиентов в многопользовательском режиме. Разрабатывается архитектура предлагаемого класса интернет-приложений. Анализируется проблема распознавания критических ситуаций в результате нарушения структуры моделей.

Идея встраиваемых динамических моделей в интернет-приложениях состоит в том, что в составе серверного сценария предусматривается специальный объект – динамическая модель, в котором заданы набор возможных состояний, переходов из состояния в состояние, правил, определяющих условия переходов, погружений, а также прикладных фрагментов кода и элементов управления интерфейса пользователя, ассоциированных состояниями и переходами.

Основное отличие предлагаемой технологии заключается в следующем (рис. 1). Серверный сценарий содержит динамическую модель взаимодействия с клиентом, с элементами которой ассоциированы прикладные элемент-фрагменты формирования разметки изображения ( $Im_n$ ). На каждом шаге взаимодействия с клиентом отслеживаются переходы текущего состояния динамической модели ( $S_n$ ), и на основе соответствующих элемент-фрагментов прикладного типа собирается HTML-разметка изображения интернет-страницы для следующего шага взаимодействия ( $Pg_n$ ).

При реализации интернет-приложений на основе встроенных динамических моделей предлагается руководствоваться принципом соответствия структуры изображения интернет-страниц структуре динамической модели. В соответствии с этим принципом каждому состоянию, входящему в состав динамической модели, на экране пользовательского обозревателя соответствует неко-

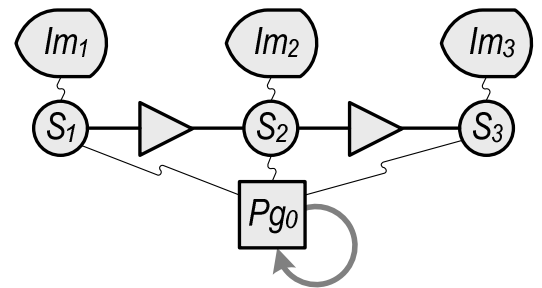


Рисунок 1 – Состояния динамической модели и изображения на экране клиента

торая область изображения, которая заполняется контентом в соответствии с текущим состоянием.

Таким образом, следующим специфичным элементом для данного класса интернет-приложений является модель памяти текущего состояния. Данный элемент строится для каждого пользователя интернет-приложения при первом погружении в динамическую модель и обеспечивает хранение информации обо всех пройденных состояниях и их источниках. Для поддержания многопользовательского режима, при котором с одним приложением могут взаимодействовать как анонимные, так и аутентифицированные пользовательские группы, было определено, что динамическая модель является общей для всех пользователей, а ее текущие состояния отслеживаются независимо для каждого пользователя. При этом для обслуживания как анонимных, так и аутентифицированных пользовательских групп, встроенная динамическая модель может содержать субмодели, доступные только для первых, только для вторых, а также для тех и других пользовательских групп.

Обеспечение хранения пользовательских сведений о текущем состоянии динамической модели многопользовательского интернет-приложения осуществляется за счет поддержки экземпляров памяти текущего состояния для анонимных (в пределах одного сеанса) и аутентифицированных пользователей (для множества сеансов). Таким образом, каждый пользователь интернет-приложения на основе встроенной динамической модели, получая доступ к приложению, имеет свой сеанс и пользовательскую модель памяти текущих состояний. В соответствии с концепцией архитектура интернет-приложений на основе встроенных динамических моделей дополняется компонентами: исходной динамической моделью взаимодействия с клиентом (*HSM*), экземплярами памяти текущего состояния для анонимных пользователей (*CSMU*), экземплярами памяти текущего состояния для аутентифицированных пользователей (*CSMA*), программными DOM-объектами и интерпретатором динамической модели (рис. 2).

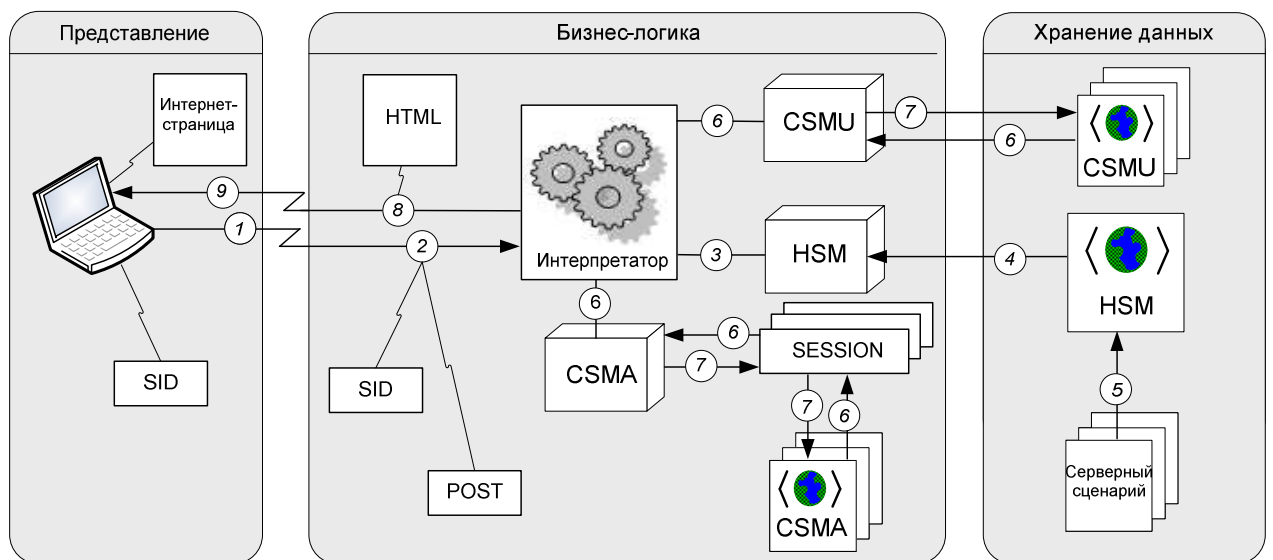


Рисунок 2 – Архитектура интернет-приложений на основе встроенных динамических моделей



На основе предлагаемого подхода могут быть реализованы любые интернет-приложения, обслуживающие бизнес-процессы ситуационного характера. Иными словами, это такие интернет-приложения, для которых могут быть выделены группы ситуаций различной степени сложности, в которых могут находиться пользователи и которые связаны друг с другом переходами. В этой связи актуальным аспектом функционирования интернет-приложений на основе встроенных моделей является задача обеспечения распознавания критических ситуаций. Выбранная XML-спецификация в качестве ключевой для построения динамических моделей позволяет реализовать автоматически простую проверку структуры моделей, а также проверку на основании дополнительных правил (например, в форме XSD-схемы).

В работе в качестве сквозного примера рассмотрено приложение предложенной концепции к построению интернет-приложения, обеспечивающего обслуживание как анонимных, так и аутентифицированных пользователей. Задача проектировщика такого приложения сводится к разработке динамической модели и определению механизма соответствия между состояниями модели и разметкой изображения. На первом этапе разрабатывается концептуальное представление динамической модели бизнес-процесса, задаются существенные состояния, переходы между ними и действия, ассоциированные с состояниями и переходами.

**В третьей главе** разрабатывается метамодель для задания встроенных динамических моделей, предусматриваются способы оформления содержания элементов-действий и элементов управления интерфейса пользователя.

Встроенная динамическая модель реализуется в классе иерархических ситуационных моделей (ИСМ), относящихся к классу дискретно-событийных. Важное обязательное синтаксическое требование ИСМ заключается в том, что динамическая модель обязательно содержит только одно корневое, сложное состояние, содержащее множество субмоделей, переходов, внешних или внутренних погружений, курсоров погружений прикладных фрагментов-действий и элементов управления интерфейса пользователя. При этом каждый элемент динамической модели обладает уникальным именем и набором именованных атрибутов (рис. 3).

Количество уровней логического разбиения исходной модели ограничено только аспектами предметной области. Наличие субмоделей в составе состояний обуславливает многоуровневую иерархию встроенной динамической модели. Субмодели, входящие в состояние верхнего уровня, могут содержать состояния, субмодели которых, в свою очередь, содержат другие состояния для последующего погружения, и т.д. За исключением начального состояния, которое становится текущим после выполнения погружения в динамическую модель, все состояния связаны друг с другом при помощи переходов. При этом один переход может определять несколько состояний-целей для погружения. Срабатывание перехода связано с выполнением входящих в его состав действий.

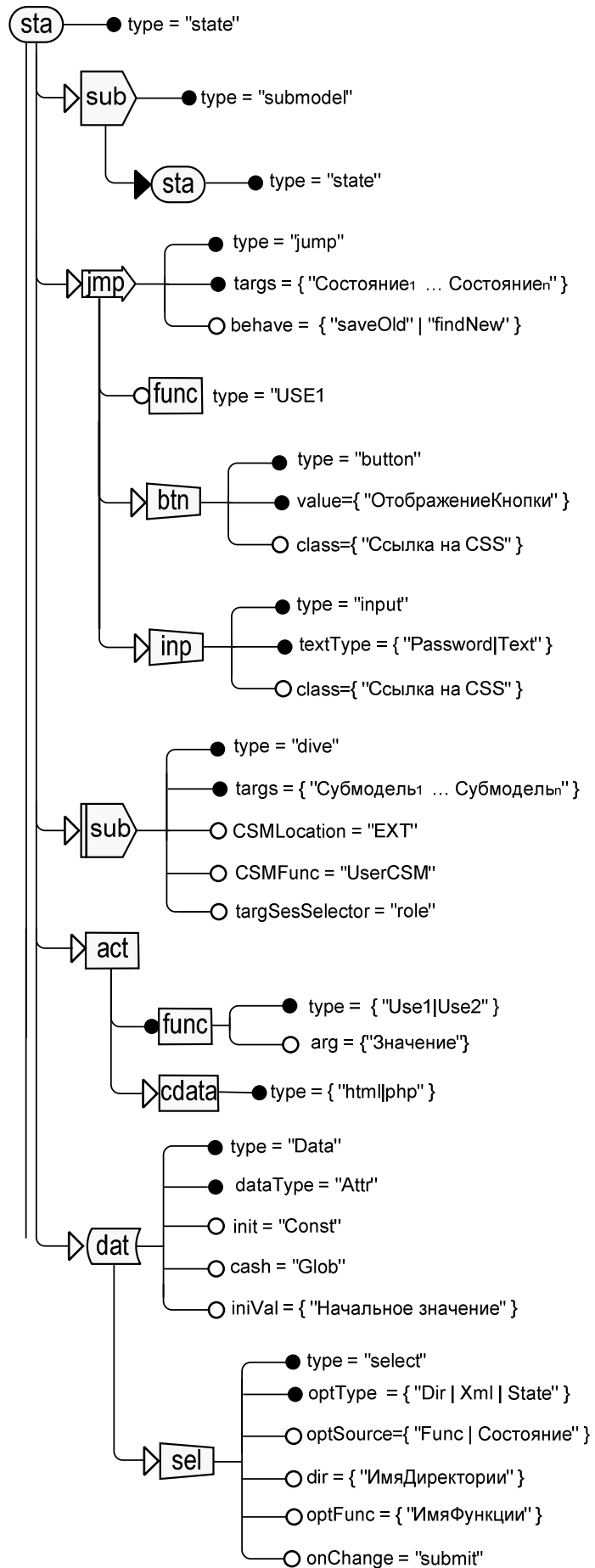


Рисунок 3 – Концептуальная схема встроенной динамической модели

Согласно концепции с каждым состоянием встроенной модели ассоциированы один или несколько фрагментов-действий, основное содержание которых составляет HTML-разметка или серверный сценарий. В динамической модели предусмотрены три варианта оформления содержания фрагментов-действий. Для статических интернет-приложений предлагается оформление содержимого в соответствии со спецификацией XHTML, что предоставляет доступ к их содержанию без обращения к дополнительным механизмам обработки. В качестве второго варианта предлагается оформление содержимого прикладных фрагментов в специальных разделах внутри динамической модели. Несмотря на необходимость применения дополнительных средств их извлечения, данный вариант позволяет создавать обозримые динамические модели, обслуживающие несложные бизнес-процессы динамических интернет-приложений. Для динамических моделей, обслуживающих достаточно сложные бизнес-процессы, оптимален третий вариант оформления прикладных фрагментов. Для уменьшения размера исходной динамической модели предполагается оформлять прикладные фрагменты как отдельные пользовательские функции и обращаться к ним из состояний встроенной динамической модели.

Согласно структуре встроенной модели состояние может содержать элементы управления пользовательского интерфейса. Каждое состояние встроенной модели проверяется на наличие соответствующих элементов управления. Преимуществом расширения структуры динамической модели пользовательскими элементами управления является то, что данные элементы можно многократно ассоциировать с различными состояниями. Это позволяет устранить дублирование программного кода, упростить процессы программирования, модификации, а также абстрагироваться от конструкций серверных сценариев и HTML-разметки для их формирования.

Следующим специфичным элементом для данного класса интернет-приложений, структуру которого необходимо однозначно зафиксировать, является модель памяти текущего состояния. На рис. 4 изображена концептуальная схема модели памяти текущего состояния. Данный компонент демонстрирует динамическое поведение, поскольку его структура и содержание корректируются в процессе работы пользователя с интернет-приложением. Структурное содержание пользовательских моделей текущих состояний повторяет структуру встроенной динамической модели, но содержит несколько дополнительных атрибутов, раскрывающих свойства пользовательской модели на отображение интернет-страниц в соответствии с пользовательскими предпочтениями. Структура и содержание данного компонента формируются динамически, и для возможности определения времени и даты погружения в субмодель и ее состояния, фиксируется дата и время их занесения (в атрибут *data*). Для устранения повторного определения текущих состояний для каждой субмодели, текущее состояние сохраняется для каждой субмодели (в атрибут *current*). Для обеспечения сохранения информации для последующего использования и настройки интернет-страницы в соответствии с действиями, которые были выполнены пользователем на предыдущих сеансовых шагах, предполагается формирование курсора погружения (в атрибут *cursor*).

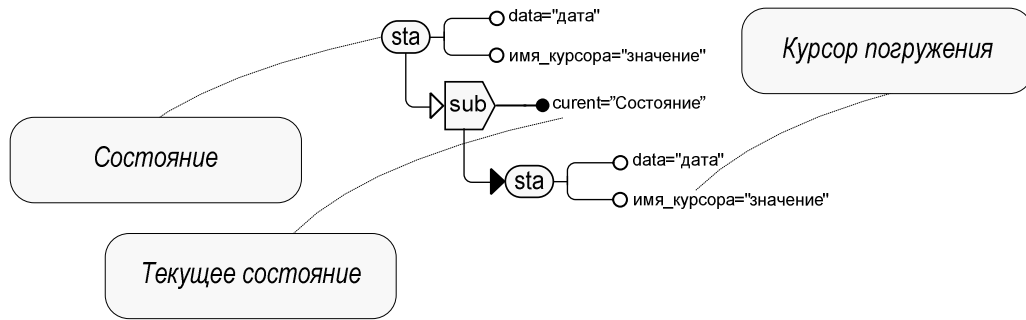


Рисунок 4 – Концептуальная схема пользовательских моделей памяти текущего состояния

Подробное описание структуры пользовательских моделей памяти текущих состояний в соответствии с нотацией форм Бэкуса–Наура приведено на рис. 5.

```

CSM ::= HSMState
HSMState ::= '<' Name cursor? data? S? ' >' HSMSubmodel* '</' Name S? '>'
S ::= (#x20)+
cursor ::= cursorName Eq cursorValue
cursorName ::= Name
Eq ::= S? '=' S?
cursorValue ::= '"' [0-9] '\ "'
data ::= dataName Eq dataValue
dataName ::= 'data'
dataValue ::= '"' [0-9] '\.' [0-9] '\.' [0-9] '\ "'
HSMSubmodel ::= '<' Name current S? ' >' HSMState+ '</' Name S? '>'
Name ::= [a-zA-Za-яA-Я0-9]
current ::= currentName Eq currentValue
currentName ::= 'current'
currentValue ::= '"' Name '\ "'

```

Рисунок 5 – Структура пользовательских моделей памяти текущих состояний

Поскольку пользовательские модели текущих состояний для аутентифицированных пользователей хранятся на стороне сервера, то нельзя не рассмотреть недостаток присущий серверным механизмам – снижение производительности сервера. Для решения задачи данного характера предусмотрен режим «работа без предыстории» в рамках субмоделей, что позволяет сохранять меньшее количество пользовательской информации, которую разработчик почитает «критической» для надлежащего функционирования интернет-приложения.

В дополнение к третьей главе выполняется XML-реализация динамической модели в соответствии с предложенной метамоделью, в основу которой положена концептуальная схема представленная в предыдущей главе. Демонстрируется, что применение XML-технологий для реализации динамических моделей целесообразно по следующим причинам: независимость динамической модели от платформы реализации; возможность описания сложных структур данных с неограниченной вложенностью элементов; невысокие требования

к системным ресурсам; возможность отделения ситуационной составляющей от прикладной; обработка модели стандартными средствами.

**В четвертой главе** предлагаются унифицированные алгоритмы интерпретации встроенной динамической модели с учетом XML-специфики. Описываются приемы построения интернет-приложений на основе динамических моделей, а также предлагаются прикладные аспекты практической реализации построения таких интернет-приложений.

Процесс интерпретации встроенной динамической модели взаимодействия с клиентом является серверным и ориентирован на двухпроходный обход модели. На первом этапе (первый проход) осуществляется контроль пользовательских текущих состояний на основании обработки элементов управляющих типов – состояний, переходов, погружения, курсоров. Основной задачей на данном проходе является формирование пользовательской модели текущих состояний (*CSM*). Интерпретатор определяет доступные управляющие элементы в контексте текущего состояния и последовательно сохраняет их в пользовательскую модель. На втором этапе (второй проход) выполняется обработка прикладных фрагментов и формирование результирующего контента, соответствующего HTML-спецификации на основании пользовательской модели памяти текущих состояний, полученной на первом проходе (рис. 6).

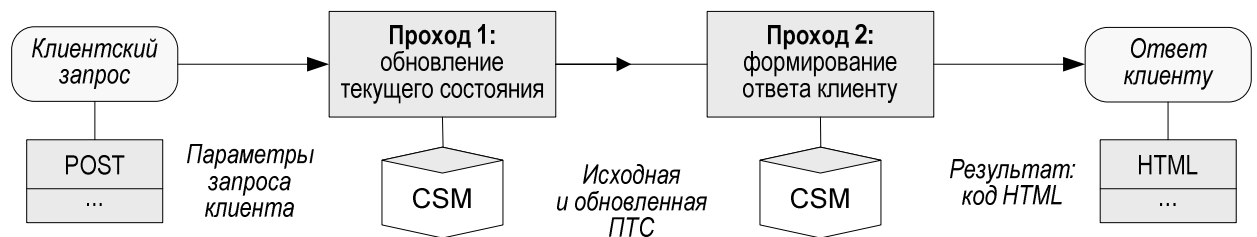


Рисунок 6 – Построение интернет-приложения на основе интерпретации динамической модели

Непосредственная работа с содержимым динамической модели и моделями текущих состояний начинается с создания экземпляров DOM-объектов, на основании которых осуществляется представление моделей в целом. На начальном этапе взаимодействия пользователя с интернет-приложением выполняется проверка наличия модели текущих состояний для данного пользователя. Если пользовательская модель найдена, осуществляется погружение и дальнейшая обработка пользовательской модели. В случае если пользовательская модель не была обнаружена, осуществляется ее формирование. После погружения в корневое состояние динамической модели осуществляется погружение в его дочерние элементы. Таким образом, на начальном этапе при получении головного события интерпретатор выбирает все исходящие из текущего состояния управляющие дочерние элементы, осуществляет проверку типа данных элементов и, при необходимости, выполняет последующие погружения в динамическую модель.

В случае если дочерний управляющий элемент был определен как субмодель, запускается обработка данного элемента и формирование модели текущих состояний путем добавления в нее данной субмодели и ее текущего состояния.

Согласно структуре встроенной модели состояние может содержать одну или несколько субмоделей, которые, в свою очередь, могут содержать в своем составе свои дочерние состояния. Поэтому вносимое в пользовательскую модель состояние проверяется на наличие дочерних субмоделей. Если при обработке динамической модели интерпретатор обнаруживает активный переход, выполняется перевод модели в новое текущее состояние в рамках той субмодели, которой принадлежит выполненный переход. Если активность перехода не установлена, то продолжается дальнейшее погружение. Дочерний элемент также может быть представлен внешней или внутренней областью погружения, в этом случае происходит погружение в субмодель, соответствующую заданной области, и последующая ее обработка. Если дочерний элемент состояния определен как курсор погружения, то в результате выполнения ассоциированных с данным элементом действий к текущему состоянию в пользовательской модели добавляется полученное значение курсора. В качестве результата первого прохода ожидается сформированная пользовательская модель текущих состояний. Экземпляр данной модели сохраняется на стороне сервера для аутентифицированных или в переменную сессионного массива для анонимных пользователей.

На втором проходе интерпретации также предусмотрена возможность последовательного обхода динамической модели, но уже с учетом информации, полученной из пользовательской модели текущих состояний, сформированной в результате первого интерпретационного прохода. Основной целью обработки данной модели является получение содержимого каждого прикладного фрагмента-действия и элемента управления интерфейса пользователя, ассоциированных с состоянием, и отправка HTML-результата их обработки клиенту. В любой соответствующей XML-спецификации динамической модели обязательно определен корневой элемент. Если текущий узел является корневым, то запускается формирование верхнего уровня интернет-страницы. В случае если текущий узел не является корневым, интерпретатор рекурсивно совершает полный обход модели на основании пользовательской модели памяти текущих состояний и последовательно генерирует HTML-фрагменты интернет-страницы для каждого текущего состояния. В процессе второго интерпретационного прохода содержимое прикладных фрагментов последовательно возвращается клиенту. В результате браузер формирует соответствующее текущему состоянию изображение интернет-страницы. При этом при переходе в конечное состояние интерпретатор заканчивает обход динамической модели и отправляет завершающую HTML-часть интернет-страницы пользователю.

Принципы и методы интерпретации встроенных динамических моделей отражены в разработанном программном обеспечении и не привязаны к конкретной предметной области или практической задаче. В нем выделены программные модули для обработки: состояний, субмоделей, переходов, погружений, прикладных фрагментов-действия и элементов управления интерфейса пользователя. Каждый из этих модулей имеет сложную внутреннюю структуру, представленную набором функциональных подмодулей, отвечающих за реализацию подзадач модуля.

Практический пример подтверждает работоспособность полученных результатов и реализует предложенные концепцию, метамодели и метод интерпретации на языке серверных сценариев.

**В пятой главе** обсуждаются вопросы применения концепции (гл. 2) построения интернет-приложений, метамодели (гл. 3), метода интерпретации (гл. 4) к практическим задачам.

Разработанный исследовательский прототип интернет-приложения для диссертационных советов УГАТУ отличается тем, что бизнес-логика приложения сосредоточена как в динамической модели (рис. 7), так и в пользовательских моделях памяти текущих состояний (рис. 8). При этом результирующая HTML-разметка (рис. 9) формируется на основе двухпроходной интерпретации динамической модели.

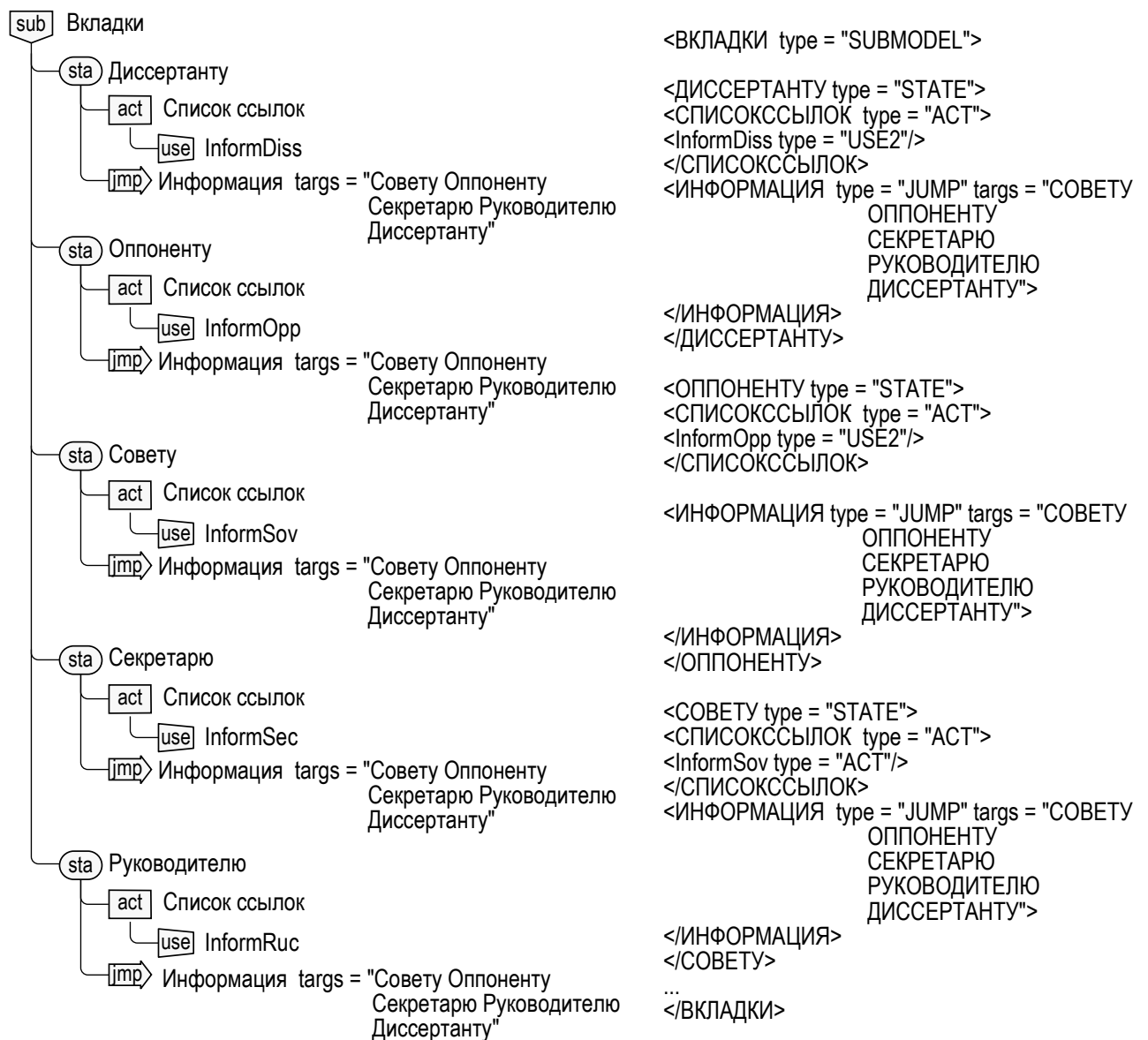


Рисунок 7 – Фрагмент динамической модели «Диспроцесс»:  
 а – концептуальная схема субмодели «Вкладки»; б – XML-реализация субмодели «Вкладки»

<pre> 1 &lt;?xml version="1.0" encoding="utf-8"?&gt; 2 &lt;ДИСПРОЦЕСС&gt; 3   &lt;СЛУЖВХОД current = "АНОНИМ"&gt; 4     &lt;АНОНИМ&gt; 5       &lt;ВКЛАДКИ current = "ДИССЕРТАНТУ"&gt; 6         &lt;ДИССЕРТАНТУ/&gt; 7       &lt;/ВКЛАДКИ&gt; 8     &lt;КОНТЕНТ current = "НОВОСТИ"&gt; 9       &lt;НОВОСТИ/&gt; 10    &lt;/КОНТЕНТ &gt; 11  &lt;/АНОНИМ&gt; 12 &lt;/СЛУЖВХОД&gt; 13 &lt;/ДИСПРОЦЕСС&gt; </pre>	<pre> 1 &lt;?xml version="1.0" encoding="utf-8"?&gt; 2 &lt;ДИСПРОЦЕСС&gt; 3   &lt;СЛУЖВХОД current = "АНОНИМ"&gt; 4     &lt;АНОНИМ&gt; 5       &lt;ВКЛАДКИ current = "ОППОНЕНТУ"&gt; 6         &lt;ОППОНЕНТУ/&gt; 7         &lt;СЕКРЕТАРЮ/&gt; 8       &lt;ДИССЕРТАНТУ/&gt; 9     &lt;/ВКЛАДКИ&gt; 10    &lt;КОНТЕНТ current = "НОВОСТИ"&gt; 11      &lt;НОВОСТИ/&gt; 12    &lt;/КОНТЕНТ &gt; 13  &lt;/АНОНИМ&gt; 14 &lt;/СЛУЖВХОД&gt; 15 &lt;/ДИСПРОЦЕСС&gt; </pre>
--	--

Рисунок 8 – Пример изменения текущего состояния пользователя в результате: *a* – первого погружения в динамическую модель в режиме «без предыстории»; *б* – выполнения перехода к состоянию «Оппоненту»

The screenshot shows the website of the Ufa State Aviation Technical University (UGATU) for its Dissertation Councils. The main navigation bar includes 'С СЫЛКИ', 'СРЕДА, 14.09.2011', 'В СЕАНСЕ: Гость', 'СПУЖЕБНЫЙ ВХОД:', and 'ВОЙТИ'. The main content area is divided into sections for 'ДИССЕРТАНТУ', 'ОППОНЕНТУ', 'СОВЕТУ', 'СЕКРЕТАРЮ', and 'РУКОВОДИТЕЛЮ'. The 'ОППОНЕНТУ' section is active, displaying instructions for opponents. Below this is a 'ЛЕНТА НОВОСТЕЙ' (News Feed) with several items, including announcements about diploma readiness and council activities. A sidebar on the right lists various sessions (e.g., Д-212.288.01) with dates and times.

Рисунок 9 – Отображение результирующей HTML-разметки для текущего состояния «Оппоненту»

Экспериментальные исследования показали, что разработка и функционирование интернет-приложений на основе встроенных динамических моделей позволяют существенно сократить трудоемкость программирования конструкций серверных сценариев (в среднем 1,6 раза). Кроме того, количество файлов серверных сценариев, которые необходимо подготовить разработчику для обеспечения функционирования интернет-приложения, сокращается в 17 раз (в рамках исследовательского прототипа), что централизует бизнес-логику интернет-приложения, обуславливает легкость последующего сопровождения и возможность реализации на любой аппаратной или программной платформе.



## ЗАКЛЮЧЕНИЕ

Таким образом, в работе решена научно-техническая задача актуальная для программной инженерии, — создание научно обоснованной технологии разработки программного обеспечения интернет-приложений исходя из динамических моделей их функционирования. При этом получены следующие результаты, значимые для методологии взаимодействия клиент/сервер в веб и для практики разработки интернет-приложений:

1) Концепция построения интернет-приложений на основе встроенных динамических моделей, отличающаяся тем, что с целью обеспечения многопользовательского режима на веб-сервере размещается динамическая модель, общая для всех клиентов, и множество клиентских экземпляров памяти текущего состояния, один из которых выбирается в сеансе интерпретации по результатам аутентификации клиента.

2) Мета модель для задания встроенных динамических моделей интернет-приложений, отличающаяся тем, что с целью возможности задавать сценарии диалогового взаимодействия с пользователем в ней предусмотрены элементы-действия, ассоциированные с текущими состояниями и специфицирующие элементы управления интерфейса пользователя.

3) Метод интерпретации встроенной динамической модели на основе отслеживания текущих состояний множества клиентов, отличающийся тем, что с целью управления диалоговым взаимодействием с пользователем в дополнение к первому проходу интерпретации, на котором в соответствии с параметрами клиентского запроса контролируется текущее состояние, предусмотрен второй проход, на котором в результате обработки элементов-действий, ассоциированных с текущими состояниями, формируется код, отправляемый клиенту в ответ на запрос.

4) Исследовательский прототип интернет-приложения со встроенной динамической моделью, отличающийся тем, что с целью подтверждения работоспособности полученных результатов в нем реализованы метод интерпретации на языке серверных сценариев РНР, динамическая модель, соответствующая метамодели в форме документа XML, постоянно хранящегося на веб-сервере, память текущего состояния анонимных клиентов — в форме сессионных переменных, а аутентифицированных — в форме базы данных XML, сохраняемой между сеансами.

## ОСНОВНЫЕ ПОЛОЖЕНИЯ ДИССЕРТАЦИИ ОПУБЛИКОВАНЫ В РАБОТАХ

### *В рецензируемых журналах из списка ВАК*

1. Интернет-приложения на основе встроенных динамических моделей: концепция, архитектура, безопасность / В. В. Миронов, К. Э. Маликова // Вестник УГАТУ: науч. журн. Уфимск. гос. авиац. техн. ун-та. Серия «Управление, вычислительная техника и информатика». 2009. Т. 13, № 2 (35). С. 167–180.

2. Интернет-приложения на основе встроенных динамических моделей: архитектура, структура данных, интерпретация / В. В. Миронов, К. Э. Маликова // Вестник УГАТУ: науч. журн. Уфимск. гос. авиац. техн. ун-та. Серия «Управление, вычислительная техника и информатика». 2010. Т. 14, № 1 (36). С. 154–164.

3. Интернет-приложения на основе встроенных динамических моделей: элементы управления пользовательского интерфейса / В. В. Миронов, К. Э. Маликова // Вестник УГАТУ: науч. журн. Уфимск. гос. авиац. техн. ун-та. Серия «Управление, вычислительная техника и информатика». 2010. Т. 15, № 1 (37). С. 151–156.

### *В других изданиях*

4. Интернет-приложения на основе динамических моделей / В. В. Миронов, К. Э. Маликова // Тр. междунар. конф. компьют. наук и информ. техн. (CSIT'2008). Крит, Греция, 2009. Т. 2. С. 142–145. (Ст. на англ. яз.).

5. Принципы разработки интернет-приложений с использованием встроенных динамических моделей / К. Э. Маликова // Мавлютовские чтения : Всерос. молодеж. конф : сб. тр. Уфа : УГАТУ, 2009. Т. 3. С.112–114.

6. Интернет-приложения на основе XML и встроенных динамических моделей / К. Э. Маликова // Интеллектуальные системы обработки информации и управления : сб. ст. регион. зимней шк.-сем. аспирантов и молодых ученых, 2009 г. Уфа : Технология, 2009. Т. 2. С. 353–357.

7. Интернет-приложения на основе встроенных динамических моделей: описание программной реализации / В. В. Миронов, К. Э. Маликова // Тр. междунар. конф. компьют. наук и информ. техн. (CSIT'2010). М.–СПб, 2010. Т. 1. С. 142–145. (Ст. на англ. яз.).

8. Организация процесса функционирования интернет-приложений на основе встроенных динамических моделей / К. Э. Маликова // XXXVI Гагаринские чтения: науч. тр. Междунар. молодежн. науч. конф. М.: МАТИ, 2010. Т. 4. С. 270–271.

9. Интернет-приложения на основе встроенных динамических моделей: исследовательский прототип / В. В. Миронов К. Э. Маликова // Управление в сложных системах: научн. изд. Уфимск. гос. авиац. техн. ун-та. 2011. С. 1–6.

10. Интернет-приложения на основе встроенных динамических моделей и XML: динамическая модель, модели памяти текущих состояний и интерпретатор / В. В. Миронов, К. Э. Маликова // Тр. междунар. конф. инновац. инфор. техн. теория и практика. Уфа, Россия : Диалог, 2011. С. 66–68. (Ст. на англ. яз.).

11. Этапы построения интернет-приложений на основе встроенных динамических моделей и XML /В. В. Миронов, К. Э. Маликова // Труды междунар. конф. комп. наук и информ. техн. (CSIT'2011). Гармиш-Партенкирхен, Германия, 2011. Уфа : Диалог, 2011. Т. 1. С. 140–142. (Статья на англ. яз.).

МАЛИКОВА Карина Эмильевна

ТЕХНОЛОГИЯ РАЗРАБОТКИ ИНТЕРНЕТ-ПРИЛОЖЕНИЙ  
НА ОСНОВЕ  
ВСТРОЕННЫХ ДИНАМИЧЕСКИХ МОДЕЛЕЙ И XML

Специальность 05.13.11

Математическое и программное обеспечение вычислительных машин,  
комплексов и компьютерных сетей

АВТОРЕФЕРАТ

диссертации на соискание ученой степени  
кандидата технических наук

Подписано к печати 21.09.2011 г. Формат 60×84 1/16.

Бумага офсетная. Печать плоская. Гарнитура Times New Roman.

Усл. печ.л.1,0. Усл. кр.-отт. 1,0. Уч.-изд.л. 0,9.

Тираж 100 экз. Заказ № 269.

ФГБОУ ВПО «Уфимский государственный авиационный технический университет»

Центр оперативной полиграфии

450000, Уфа-центр, ул. К.Маркса, 12