

На правах рукописи

Бакаев Максим Александрович

РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ДЛЯ ПОДДЕРЖКИ  
ПРОЕКТИРОВАНИЯ ЧЕЛОВЕКО-КОМПЬЮТЕРНОГО  
ВЗАИМОДЕЙСТВИЯ В ВЕБ-ПРИЛОЖЕНИЯХ

Специальность 05.13.11 – Математическое и программное обеспечение  
вычислительных машин, комплексов и компьютерных сетей

АВТОРЕФЕРАТ  
диссертации на соискание ученой степени  
кандидата технических наук

Новосибирск-2012

Работа выполнена в Федеральном государственном бюджетном образовательном учреждении высшего профессионального образования «Новосибирский государственный технический университет»

Научный руководитель: доктор технических наук, профессор  
Авдеенко Татьяна Владимировна

Официальные оппоненты: Хабаров Валерий Иванович,  
доктор технических наук, профессор,  
ФГБОУ ВПО Сибирский государственный  
университет путей сообщения,  
зав. кафедрой «Информационные технологии на  
транспорте»

Тузовский Анатолий Федорович,  
доктор технических наук, профессор,  
ФГБОУ ВПО Национальный исследовательский  
Томский политехнический университет,  
проф. кафедры оптимизации систем управления

Ведущая организация: Федеральное государственное бюджетное учреждение науки Институт систем информатики им. А.П. Ершова Сибирского отделения РАН, г. Новосибирск.

Защита состоится 20 сентября 2012 г. в 16:00 на заседании диссертационного совета Д 212.173.06 при Федеральном государственном бюджетном образовательном учреждении высшего профессионального образования «Новосибирский государственный технический университет» по адресу: 630092, г. Новосибирск-92, пр. К. Маркса, 20.

С диссертацией можно ознакомиться в библиотеке Новосибирского государственного технического университета.

Автореферат разослан «03» июля 2012 г.

Ученый секретарь  
диссертационного совета

Чубич Владимир Михайлович

## ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

### Актуальность проблемы

За первое десятилетие XXI века количество интернет-пользователей в мире увеличилось более чем в 5 раз, достигнув 2 миллиардов человек, а число активных веб-приложений в 2011 г. превысило отметку в 150 миллионов. По результатам этого же года суммарный объем реализации крупнейших российских компаний в сфере информационных технологий (ИТ) достиг рекордной величины в 508,4 млрд руб., причём максимальным (46%) оказался прирост в сфере разработки программного обеспечения (ПО). При этом, согласно исследованиям, не менее 50% всего создаваемого программного кода посвящено пользовательским интерфейсам, а разработка средств эффективного взаимодействия человека с компьютером считается одним из приоритетных направлений развития искусственного интеллекта и информатики в целом<sup>1</sup>.

Интерфейс пользователя является основным предметом полидисциплинарного научного направления под названием «человеко-компьютерное взаимодействие» (ЧКВ), формирование которого началось в 1960-е годы, в том числе в нашей стране, на стыке информатики, эргономики, инженерной психологии и других областей. Согласно рекомендациям специалистов, на обеспечение качества интерфейса следует выделять не менее 10% от общего бюджета проекта по разработке ПО. При этом среднее улучшение основных бизнес-показателей веб-приложений составляет от 83% (США, 2008 г.), что позволяет сделать вывод о значительной экономической эффективности проектирования качественного взаимодействия. Тем не менее, применение методов проектирования взаимодействия на практике осуществляется далеко не во всех проектах, связанных с разработкой ПО. В результате значение даже такого базового показателя качества интерфейса как «процент успешного выполнения задач», для веб-приложений составляло в 2009 г. не более 81% (в России, предположительно, около 60%), а для отдельных категорий пользователей ещё в 1,5-2 раза ниже.

Одна из проблем, отмечаемых в сфере ЧКВ, заключается в том, что практическое знание в данной области характеризуется слабой степенью организации, – это приводит к существенным затратам времени разработчиков на поиск, интерпретацию и применение соответствующих рекомендаций или готовых «шаблонов проектирования» (типовых решений, используемых при проектировании интерфейсов). Интеллектуальные (экспертные) системы (ИС) для поддержки проектирования интерфейсов, ряд которых создавался с начала 2000-х годов, можно разделить на инструменты для организации рекомендаций (MetroWeb, BORE и др.) и для автоматизированной генерации кода интерфейса и его валидации<sup>2</sup>. Эффективное совмещение этих подходов и устранение их недостатков позволило бы сократить затраты времени проектировщиков на поиск существующих рекомендаций, снизить количество ошибок, связанных с их

---

<sup>1</sup> Ронжин, А.Л., Карпов, А.А., Ли, И.В. Речевой и многомодальный интерфейсы. М.: Наука, 2006.

<sup>2</sup> Грибова, В.В. Автоматизация проектирования, реализации и сопровождения пользовательского интерфейса на основе онтологического подхода. // Докторская диссертация – Владивосток, ИАПУ ДВО РАН, 2007.

применением, и, тем самым, повысить уровень качества создаваемых пользовательских интерфейсов.

### **Цель работы**

Целью диссертационной работы является разработка, с использованием методов инженерии знаний, средств интеллектуальной поддержки проектирования человеко-машинных интерфейсов в веб-приложениях. Разрабатываемая интеллектуальная система должна включать базу знаний (БЗ), максимально полно охватывать стадии процесса разработки ПО, а также учитывать специфику проектируемого взаимодействия (для предоставления практических знаний, соответствующих контексту конкретного проекта).

Для достижения поставленной цели в рамках диссертационной работы были поставлены и решены следующие задачи:

1. Проведение анализа структуры знаний в сфере ЧКВ и процесса проектирования взаимодействия, а также выбор адекватных моделей и средств представления знаний.

2. Разработка и экспериментальное исследование моделей взаимодействия в человеко-машинных интерфейсах для выявления характеристик пользователей, значимых для различных аспектов взаимодействия.

3. Построение базы знаний для предметной области проектирования ЧКВ в веб-приложениях, включающей механизмы организации хранимых знаний и оценки их сравнительной эффективности.

4. Создание интеллектуальной системы на основе разработанной онтологии проектирования ЧКВ, её применение для решения практических задач в предметной области, оценка качества полученных результатов.

Объектом исследования являются знания в сфере ЧКВ, а предметом исследования – процессы обработки, организации и практического применения знаний для повышения эффективности проектирования ЧКВ в веб-приложениях.

### **Методы исследования**

Среди использованных методов исследования: методы инженерии знаний (построение онтологий, применение фреймовой и продукционной модели, анализ текстологических источников), компьютерной лингвистики, статистического анализа. При разработке программного обеспечения использовались подходы логического, структурного и объектно-ориентированного программирования. В качестве методологической основы также использовались исследования отечественных и зарубежных учёных: Я. Нильсена, Т.А. Гавриловой и В.Ф. Хорошевского, Т. Грубера, Н.Г. Загоруйко, Б.Ф. Ломова, П. Фитса, У. Хика, Ю.А. Загоруйко, В.И. Хабарова, Е.Б. Цоя и М.Г. Грифа, А.С. Клещева и В.В. Грибовой, J. Vanderdonckt, N. Noy, S. MacKenzie, Д. Нормана и др.

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

1. Разработан новый подход к проектированию человеко-машинных интерфейсов для веб-приложений, комбинирующий организацию практических знаний и модели ориентированную автоматическую генерацию прототипа интерфейса исходя из специфики конкретного проекта.

2. Предложена гибридная модель представления знаний, комбинирующая онтологию, фреймовый и продукционный компоненты для эффективной обра-

ботки знаний посредством индексирования их базовыми терминами из контролируемого словаря.

3. Предложены расширенные модели человеко-машинного общения, описывающие поведение различных категорий пользователей, и выявлены значимые для взаимодействия характеристики пользователей для уточнения структуры БЗ.

4. Создана интеллектуальная система для поддержки проектирования человеко-машинных интерфейсов в веб-приложениях, на основе онтологической модели, базы знаний продукционной модели, а также механизмов их взаимодействия.

5. Разработан «портал знаний» для интеллектуальной поддержки процесса проектирования ЧКВ в веб-приложениях, интегрирующий компоненты гибридной модели с интерфейсом ИС и включающий механизм определения «эффективности» рекомендаций, основанный на композиции нечётких отношений.

#### **Практическая значимость**

Использование разработанной интеллектуальной системы может позволить повысить эффективность работы проектировщика человеко-машинных интерфейсов (снизить требования к квалификации проектировщиков, затраты времени и количество ошибок) и повысить качество взаимодействия для всех категорий пользователей. Онтологическая модель предметной области проектирования ЧКВ, созданная в данной работе, была внедрена в учебный процесс в рамках дисциплины «Интеллектуальные информационные системы» на кафедре Экономической информатики Новосибирского государственного технического университета (НГТУ). Разработанная на основе онтологической модели интеллектуальная система была успешно использована для поддержки проектирования ЧКВ при создании веб-приложений для Народного факультета НГТУ и ОАО «Трест Сибэнергомонтаж», внутреннего веб-приложения для швейцарской компании «Комакс АГ», веб-интерфейса системы «АИС Электронный инспектор МЧС», входящей в состав системы автоматизации функционирования органов надзорной деятельности МЧС России (государственный контракт №16/2.2.4.1-0314 от 22.11.2011, договор кафедры ЭИ НГТУ № 2011/11-07 от 07.11.2011). Интеллектуальная система прошла государственную регистрацию как программа для ЭВМ, о чём Федеральной службой по интеллектуальной собственности выдано свидетельство № 2011615212 от 01 июля 2011 г.

#### **Апробация работы**

Основные выводы и научные результаты диссертационной работы докладывались на ряде научно-практических конференций, из которых 13 являлись международными: The 10th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems и International Association of Societies of Design Research 2009 Conference, г. Сеул, Ю. Корея, 2007 и 2009 гг.; The Eighth Pan-Pacific Conference on Occupational Ergonomics, г. Бангкок, Таиланд, 2007 г.; VIII международная конференция «Информатика: проблемы, методология, технологии», г. Воронеж, 2008 г.; VIII Brazilian Symposium on Human Factors in Computing Systems, г. Порту-Алегри, Бразилия, 2008 г.; IEEE Region 8 International Conference on Computational Technologies in Electrical and Electronics Engineering, IASTED Automation, Control, and Information Technology ACIT'2010, и Ershov Informatics Conference (PSI 11), г. Новосибирск, 2008, 2010

и 2011 гг.; DST-RFBR Sponsored Indo-Russian Joint Workshop on Computational Intelligence and Modern Heuristics in Automation and Robotics, г. Сурат, Индия, 2010 г.; International Conference on Information and Multimedia Technology и International Conference on Business Intelligence and Financial Engineering, г. Гонконг, 2010 и 2011 гг.; IADIS International Conference Applied Computing, г. Рио-де-Жанейро, Бразилия, 2011 г.; The 17th International Conference on Database Systems for Advanced Applications, г. Пусан, Ю. Корея, 2012 г.

### **Публикации**

По теме диссертации автором опубликовано 17 работ (из них 11 на английском языке), в том числе 3 публикации в ведущих рецензируемых научных журналах и изданиях, входящих в перечень рекомендованных ВАК, 1 публикация в зарубежном международном научном периодическом издании, 1 свидетельство о регистрации программы для ЭВМ, 12 публикаций в сборниках материалов международных научно-практических конференций.

### **Структура и объем**

Диссертационная работа состоит из введения, 4 глав, заключения, списка литературы, содержащего 208 наименований, и приложений. Общий объем работы – 265 страниц, включая 32 таблицы и 25 рисунков.

### **СОДЕРЖАНИЕ РАБОТЫ**

Во **введении** обосновывается актуальность темы исследования и формулируются задачи диссертационной работы.

В **первой главе** рассматриваются современные подходы к созданию пользовательских интерфейсов, их место в индустрии разработки программного обеспечения в целом, а также даётся обзор инструментов, способных поддерживать разработчика в данной деятельности. Анализируется структура знаний, накопленных и применяемых в сфере ЧКВ, формулируются принципы построения ИС для поддержки проектирования взаимодействия.

Основным предметом человеко-компьютерного взаимодействия является интерфейс («пользовательский интерфейс», «человеко-машинный интерфейс») – совокупность средств и правил, согласно которым происходит взаимодействие пользователя и компьютера. В рамках диссертационного исследования рассматривается в основном графический интерфейс пользователя, причём используются следующие количественно измеримые показатели его качества (на основе существующих стандартов, например ISO/IEC 25062:2006, и работ Я. Нильсена):

- 1) процент успешного выполнения задач пользователем;
- 2) затраченное на выполнение задач время;
- 3) уровень ошибок, допускаемых в процессе выполнения;
- 4) субъективная удовлетворенность пользователя.

Важность применения методов проектирования взаимодействия в процессе разработки ПО широко признана, а вложения в качество интерфейса создаваемого программного продукта, как правило, имеют высокий экономический эффект. Тем не менее, далеко не все команды разработчиков (особенно в средних и небольших проектах по разработке веб-приложений) включают в себя квалифицированных специалистов в данной области. В то же время, проектирование ЧКВ

не имеет и комплексной поддержки в виде инструментальных средств, которые бы содержали доступные и качественно организованные экспертные знания.

Произведён обзор знания, накопленного в сфере ЧКВ, с его условным разделением на три группы: «законы» (высокоуровневые теоретические построения), «принципы» (более или менее универсальные правила), а также рекомендации (практические советы или напоминания, как избежать ошибок) или «шаблоны проектирования». Отмечено, что рекомендации, которые в наибольшей степени используются в практике проектирования взаимодействия и составляют самый обширный из уровней знания в сфере ЧКВ, могут дублироваться или противоречить друг другу, не содержать явной связи с теоретическим обоснованием, не описывать контекст своей применимости. J. Vanderdonckt, который совместно с коллективом последователей занимался исследованием проблем и методов организации рекомендаций с конца 1990-х годов, отмечал следующие барьеры для применения рекомендаций при проектировании интерфейсов:

1. Значительное количество рекомендаций при недостатке средств их организации (в среднем затраты времени на применение одной рекомендации составляли 15 мин).

2. Сложность интерпретации (проектировщики испытывали сложности с интерпретацией 30% найденных рекомендаций).

3. Пониженная применимость в связи с оторванностью рекомендаций от контекста проектирования.

Инструменты, поддерживающие создание программного кода интерфейса или его проекта (прототипа, модели), классифицированы следующим образом:

1. Универсальные и специализированные визуальные редакторы: графики (Adobe Photoshop), веб-страниц (Adobe Dreamweaver, MS Expression Web), интерактивных приложений (Adobe Flash), прототипов интерфейсов (Balsamiq Mockups и др.) – в рамках так называемой дизайнерской парадигмы. Данные инструменты воплощают методы высокоуровневого проектирования и способны существенно снизить трудоемкость разработки визуального компонента интерфейса, однако они не содержат знаний о принципах организации взаимодействия.

2. Средства автоматизации создания кода интерфейса и его валидации в рамках модели ориентированной парадигмы. Так, система, разработанная в ИАПУ ДВО РАН коллективом под руководством В.В. Грибовой и А.С. Клещева и базирующаяся на онтологическом подходе, позволяет фиксировать решения на этапе проектирования в модели интерфейса, проводить автоматическую оценку качества интерфейса и т.д. Существуют также инструменты, способные автоматически генерировать интерфейсы для относительно простых классов задач или для особых условий взаимодействия. Так, применение системы PUC (предназначенной для создания стандартизованных интерфейсов для управления различной бытовой техникой) позволило увеличить показатели качества взаимодействия в 2-4 раза<sup>3</sup>. Система SUPPLE создаёт альтернативные интерфейсы для пользователей, чьи потребности не были учтены в основном интер-

---

<sup>3</sup> Nichols, J., Myers, B.A. Automatically generating high-quality user interfaces for appliances // Doctoral dissertation, Carnegie Mellon University, Pittsburgh, PA, 2006. – 358 p.

фейсе продукта или устройства, рассматривая генерацию интерфейса как задачу дискретной оптимизации. Принципиальным ограничением моделиориентированной парадигмы автоматизации создания интерфейсов является необходимость предварительного задания подробной модели взаимодействия – задач пользователя, обрабатываемых данных и т.д. В случае, если результатом должен являться сложный интерфейс с высоким уровнем детализации, трудоемкость задания входных параметров для его генерации может превысить затраты на его проектирование обычными методами.

3. Инструменты для работы с рекомендациями: такие системы как MetroWeb, BORE и др., разрабатывались с начала 2000-х годов и для некоторых из них на экспериментальных данных был показан положительный эффект при проектировании интерфейсов. Тем не менее, данные экспертные системы не получили широкого распространения, причина чего видится в том, что они не помогают соотносить рекомендации с контекстом проекта и выбирать наиболее подходящие, не учитывают характеристики различных категорий пользователей, не охватывают все стадии процесса разработки. Кроме того, в построенных моделях представления знаний предполагается, что все отношения между элементами знаний (например, в BORE – «требует», «является альтернативой», «конфликтует с» и т.п.) устанавливаются и поддерживаются вручную, что весьма затруднило бы сопровождение базы знаний, содержащей значительное количество объектов. Представляется, что устранение этого недостатка возможно при использовании более сложных моделей представления знаний, чем стандартная онтологическая модель предметной области.

В диссертации предлагается разработка интеллектуальной системы, вмещающей основные принципы описанных подходов, для повышения эффективности работы проектировщиков человеко-машинных интерфейсов и качества создаваемых веб-приложений. Система охватывает все выделенные уровни знаний в сфере ЧКВ и способна учитывать контекст проектирования, осуществляя вывод модели интерфейса из входной информации на базе онтологического подхода. Исходя из особенностей современных веб-приложений, построение модели (логического представления интерфейса) и генерация кода осуществляются на основе правил вывода, а итоговый веб-интерфейс является прототипом с гибкими физическими характеристиками. Система имеет средства для автоматизированного определения сравнительной значимости экспертных знаний на основе показателей качества созданных веб-интерфейсов, валидация которых может производиться с использованием широко применяемых на практике методов проектирования взаимодействия.

Во **второй главе** определяется архитектура интеллектуальной системы, структура входной и выходной информации (представлены на Рис. 1), рассматриваются адекватные модели представления знаний и предлагаются средства для порождения системой новых знаний в предметной области.

Работа системы предполагает наличие входной информации – характеристик пользователей, являющихся целевыми для разрабатываемого программного продукта, и требований к нему. В ходе работы системы механизм вывода (решатель) формирует знание о конкретном проекте, сопоставляя входные дан-

ные и информацию, хранимую в БЗ, а затем вырабатывает выходные данные, которыми являются релевантный контексту проекта упорядоченный набор рекомендаций, а также автоматически создаваемый прототип интерфейса разрабатываемого веб-приложения. При этом окончательные решения относительно проекта интерфейса принимаются специалистом, использующим систему, на основе сгенерированного прототипа и выходного множества рекомендаций.

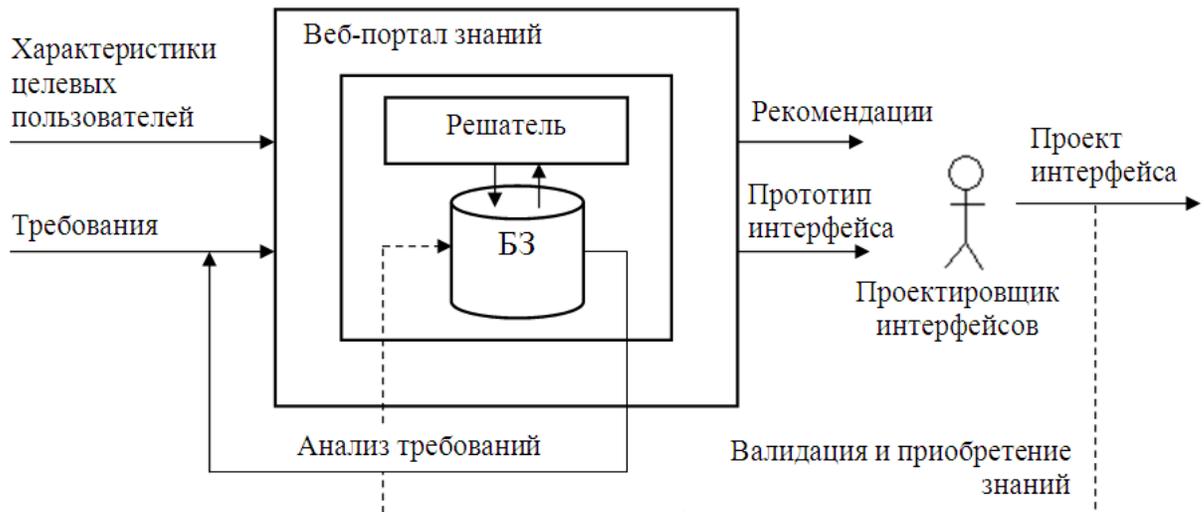


Рисунок 1. Архитектура ИС поддержки проектирования ЧКВ в веб-приложениях.

Предложена гибридная модель для представления знаний в системе, включающая фреймовую онтологию, максимально использующую возможности объектно-ориентированного подхода, а также продукционную модель, позволяющую формулировать знания в виде импликаций (правил «если-то»). Ключевым компонентом базы знаний ИС является онтология. Онтологии – формальные явные описания терминов предметной области и отношений между ними. Под это определение онтологии, данное Т. Грубером, попадают многие модели представления знаний: фреймы, семантические сети, концептуальные карты и т.д. Идея фрейма (frame – остов, скелет, каркас) для представления знаний, предложенная М. Минским, заключается в концентрации данных (знаний) об объекте в единой сложной структуре, отображающей взаимосвязь объектов этой области, в отличие от их распределения между множеством более мелких структур. Формально фрейм представляет собой статическую структуру данных, состоящую из слотов и их заполнителей. Характерной особенностью фреймовой модели является то, что первоначально слоты фреймов могут быть заполнены «заданиями отсутствия», которые в процессе приспособления начальной общей модели знаний к конкретной ситуации постепенно заполняются реальными данными. Данное свойство позволило эффективно реализовать подсистему ввода исходной информации путем ее встраивания в соответствующие слоты фреймов и последующего распространения по сети фреймов. Построенную в работе фреймовую онтологию можно формально представить следующим кортежем:

$$O_F = \langle C, R, S, G, T, D_S, D_G, E \rangle, \quad (1)$$

где  $C = \{c_i \mid i = 1, \dots, n\}$  – конечное непустое множество фреймов-классов, описывающих понятия предметной области;

$R = \{r_i \mid i = 1, \dots, m\}$  – конечное множество бинарных отношений, заданных на классах,  $R \subseteq C \times C$ ,  $R = \{R_{ISA}\} \cup R_{ASS}$ , где  $R_{ISA}$  – антисимметричное, транзитивное, неререфлексивное отношение иерархии «класс-подкласс», задающее частичный порядок на множестве классов;  $R_{ASS}$  – конечное множество ассоциативных отношений;

$S = \{s_i \mid i = 1, \dots, k\}$  – конечное множество слотов (атрибутов класса);

$G = \{gs_i \mid i = 1, \dots, l\}$  – конечное множество фасетов (атрибутов слота);

$E = \{e_i \mid i = 1, \dots, u\}$  – конечное множество экземпляров классов;

$T$  – конечное непустое множество, определяющее контролируемый словарь терминов предметной области, построенное на множестве базовых терминов

$B = \{b_i \mid i = 1, \dots, n\}$ , составляющих множество имен классов онтологии:

$$T = \bigcup_{i=1}^n T_i, T_i = \{b_i\} \cup Eq(b_i), \bigcap_{i=1}^n T_i = \emptyset;$$

$Eq(b_i)$  – множество синонимичных терминов, каждый из которых связан с базовым термином  $b_i \in B$ ,  $D_S$  – конечное множество типов слотов,  $D_G$  – конечное множество типов фасетов.

Структура фрейма-класса определена следующим образом:

$$c = \langle Name_c, (isa \ c_{parent}), (s_1, s_2, \dots, s_{n(c)}) \rangle,$$

где  $c, c_{parent} \in C$  – классы онтологии, связанные отношением иерархии  $R_{ISA}$ ,  $s_i \in S$  – слоты фрейма,  $Name_c \in B$  – имя класса, являющееся базовым термином контролируемого словаря  $T$ . Иерархии фреймов-классов образуются посредством указания в подчиненном фрейме связи «isa» и имени фрейма-родителя  $c_{parent}$ .

Фрагмент иерархии классов рассматриваемой прикладной области проектирования веб-интерфейсов представлен на Рис. 2. В данной иерархии выделяются мета-класс *THING* («Объект»); классы *Interface quality metric* («Показатель качества интерфейса») с подклассами *Success rate* («Процент успешности»), *Subjective satisfaction* («Субъективная удовлетворенность пользователя»), *Error rate* («Уровень ошибок») и *Performance time* («Время выполнения»); *Design solution* («Проектное решение»), *Interface design* («Проект интерфейса»); *Interface element* («Элемент интерфейса»); *Website element* («Элемент веб-приложения»).

Все множество классов  $C$  разбивается на два непересекающихся подмножества  $C = C_{abstract} \cup C_{concrete}$ . Для классов подмножества  $C_{concrete}$  возможно определять экземпляры класса (конкретные объекты)  $e \in E$ . Структура фрейма-экземпляра аналогична структуре класса  $c$ , для которого построен экземпляр:

$$e(c) = \langle Name_e, (s_1^e, s_2^e, \dots, s_{n(c)}^e) \rangle,$$

где  $s_1^e, s_2^e, \dots, s_{n(c)}^e$  – экземпляры слотов класса  $c$ , заполненные конкретными значениями атрибутов.

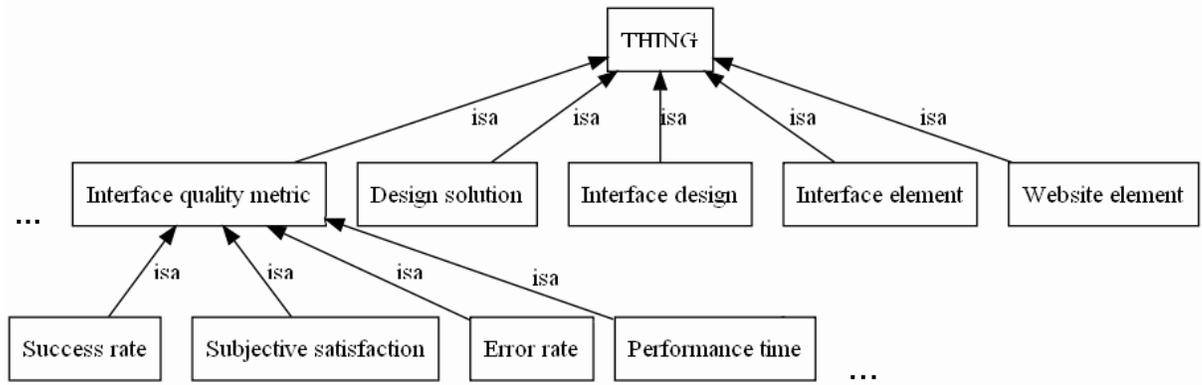


Рисунок 2. Фрагмент иерархии классов фреймовой структуры.

Возможности фреймовой модели существенно усиливаются с использованием на её основе объектно-ориентированного подхода. Основным преимуществом такого подхода применительно к рассматриваемой базе знаний является реализация механизма наследования значений слотов (свойств) классов-родителей классами-потомками и экземплярами в качестве составляющей механизма вывода на знаниях, реализованного в системе.

Описание рассматриваемой проблемной области не ограничивается лишь иерархическими отношениями  $R_{ISA}$ . Определение ассоциативных отношений, составляющих множество  $R_{ASS}$ , осуществляется путем явного указания в качестве значения слота некоторого фрейма имени связанного с ним фрейма, а также типа связи, существующей между этими фреймами. Для реализации ассоциативных связей среди элементов множества типов слотов  $D_S$  используются также типы  $D_{class}$  (тип «Класс») и  $D_{instance}$  (тип «Экземпляр»):

$$D_S = D_{SS} \cup \{D_{class}\} \cup \{D_{instance}\},$$

где  $D_{SS}$  – подмножество стандартных типов (symbol, string, float, ...).

Задание типов  $D_{class}$  и  $D_{instance}$  предполагает указание дополнительного аргумента – ассоциированного класса. Если один из слотов класса  $c_1$  имеет тип  $D_{instance}$  с ассоциированным классом  $c_2$ , то в качестве значений слота при создании экземпляров класса  $c_1$  могут быть использованы экземпляры классов множества  $Tr(c_2)$  – транзитивного замыкания  $c_2$  по отношению  $R_{ISA}$ , включающего класс  $c_2$  и все его подклассы ниже по иерархии:

$$Tr(c_2) = \{c_2\} \cup \{c_i \in C \mid \exists R_{ISA}(c_i, c_2)\}.$$

В этом случае классы  $c_1$  и  $c_2$  связаны ассоциативным отношением, т.е.  $\exists R_{ASS}(c_1, c_2)$ . Если один из слотов класса  $c_1$  имеет тип  $D_{class}$  с ассоциированным классом  $c_2$ , то в качестве значений слота при создании экземпляров класса  $c_1$  могут быть использованы классы множества  $Tr(c_2)$ . В этом случае классы  $c_1$  и  $c_2$  также связаны ассоциативным отношением, т.е.  $\exists R_{ASS}(c_1, c_2)$ . Таким образом, значением слота может становиться не только экземпляр ассоциированного класса, но и базовый термин, что было использовано для описания сложных объектов предметной области терминами контролируемого словаря. Так, начальный класс онтологии *HCI engineering task* («Задача проектирования ЧКВ»)

имеет слот *project tags* («теги проекта»), позволяющий задавать контекст проекта – значением слота могут быть элементы множества  $B$  (т.е.  $c_2 = THING$ ).

В построенной онтологической модели фреймами являются не только классы онтологии, но также и атрибуты классов, и отношения между ними. Структура фрейма-слота определяется следующим образом:

$$s_c = \langle Name_{s,c}, (gs_1, gs_2, \dots, gs_{k(s,c)}) \rangle,$$

где  $s_c \in S$  – слот класса  $c$ ,  $gs_i \in G$  – фасеты слота,  $Name_{s,c}$  – имя слота. Такой подход можно считать развитием идеи атрибутивных отношений (т.е. обладающих набором собственных атрибутов, в отличие от обычных отношений, имеющих только название), позволяющих осуществлять более качественное моделирование предметных областей за счёт расширения выразительных свойств модели.

Реализация вывода во фреймовой модели, связанная с эффектом передачи данных во фрейм или извлечения данных из него, предполагает, наряду с использованием наследования свойств от других фреймов, широкое применение процедурного подхода: через вызов функции, указанной в слоте, через присоединенную к слоту процедуру-демон, из диалога с пользователем, из базы данных и т.д. В работе рассматривается подход, при котором часть процедур системы реализуется с использованием продукционной модели, заменяя, таким образом, процедурный код декларативными утверждениями в виде правил «если-то».

Продукционная модель является одним из самых распространённых средств представления знаний благодаря своей наглядности, модульности и удобству для осуществления вывода. Однако в чистом виде системы на основе правил являются довольно поверхностными и не обеспечивают более глубоких рассуждений. Сочетание продукционной модели с фреймовой позволяет, с одной стороны, дополнить поверхностные эвристические правила детализированными структурными описаниями фреймов и связей между ними. С другой стороны, такая комбинация позволяет описать часть процедурного компонента фреймовой модели в виде правил, повысив наглядность описания и сделав возможным применить механизм логического вывода.

Таким образом, общая гибридная модель интеллектуальной системы представлена следующим образом:

$$Model = \langle O_F, F, P, M \rangle, \quad (2)$$

где  $O_F$  – вышеописанная онтология (1),  $F = \{f_i \mid i = 1, \dots, n_f\}$  – конечное множество фреймов-фактов, образуемых в процессе работы системы,  $P = \{p_i \mid i = 1, \dots, n_p\}$  – конечное множество правил, описывающих в декларативном виде процедурный компонент системы,  $M$  – гибридный механизм вывода, объединяющий механизмы, действующие в различных компонентах системы.

Факты и правила являются компонентами, на основе которых строится работа механизма логического вывода продукционной модели. Факты имеют структуру фреймов и формируются на основе конкретных классов и экземпляров онтологии:

$$f = \langle Name_f, (g_1^f, g_2^f, \dots, g_{n(f)}^f) \rangle,$$

где  $f \in F$  – фреймы-факты,  $Name_f$  – имя факта,  $g_i^f$  – слоты факта, сформированные из слотов соответствующего класса или экземпляра с использованием механизма заполнения или вытеснения их значений на основе входной информации. Таким образом, при формировании факта происходит объединение условно-постоянной информации, содержащейся в онтологии, с входной информацией об особенностях конкретного проекта, создаваемого с использованием ИС.

Правила  $p \in P$  представляют собой следующую структуру:

$$p = \langle Name_p, (a_1 \wedge a_2 \wedge \dots \wedge a_h \Rightarrow d) \rangle,$$

где  $Name_p$  – имя правила, используемое для возможного частичного управления порядком выбора правил при реализации логического вывода. Импликация  $a_1 \wedge a_2 \wedge \dots \wedge a_h \Rightarrow d$  состоит из левой части  $a_1 \wedge a_2 \wedge \dots \wedge a_h$ , представляющей собой конъюнкцию предпосылок (предусловий) правила, и правой части  $d$  – следствия, представляющего собой набор действий, выполняемых в случае истинности всех предпосылок левой части правила. Предпосылками правила могут быть логические условия, предикаты, шаблоны и другие конструкции. Так, проверка условия при помощи шаблона представляет особый интерес – это реализация алгоритма сопоставления (унификации) шаблона, заданного в теле правила с использованием различных синтаксических конструкций, с имеющимися в системе фактами или объектами онтологии. В случае нахождения факта или объекта, унифицируемого с заданным шаблоном, соответствующая предпосылка считается истинной. Особую гибкость данному виду условий придает возможность использования переменных внутри шаблона. В этом случае переменные, получившие значения при реализации алгоритма унификации для предпосылки  $a_i$ , могут использоваться либо в оставшихся предпосылках  $a_{i+1}$ ,  $a_{i+2}$ , ..., либо в процедурах, определяющих следствие  $d$ .

Создана онтологическая модель поддержки проектирования ЧКВ в веб-приложениях, после нескольких итераций разработки включившая в себя более 150 фреймов-классов. Например, в соответствии с уровнями знания в сфере ЧКВ, в онтологии имеются классы: *Law* («Закон»), *Principle* («Принцип») и *Guideline* («Рекомендация»), а также *Finding* («Сведение»), *Source* («Источник») и *Reference* («Ссылка»), что позволяет реализовать объясняющий компонент (обоснование, ссылки на источники и т.д.). Особый интерес представляет *tag* («тег»), слот класса *Guideline*, – его значением могут являться любые классы онтологии (термины предметной области), что позволяет производить индексацию рекомендаций, сформулированных на естественном языке.

Разработанная для системы фреймовая модель предметной области может содержать как связи с типом  $D_{instance}$ : *supported by* («иметь в качестве обоснования»), *reference* («ссылаться») и т.д., так и с типом  $D_{class}$ : *tag* («иметь в качестве области применения»). Так, например, хранимой в БЗ системы рекомендации со следующим текстом: «С логотипа сайта должна быть ссылка на главную страницу (кроме как на самой главной странице)», инженером по знаниям сопоставлены фреймы-классы *Logo* («Логотип»), *Homepage* («Главная страница»), *Hyperlink*

(«Гиперссылка»). Контекст конкретного проекта описывается аналогичным образом, однако набор классов формируется системой автоматически, исходя из характеристик целевого пользователя и требований, задаваемых в качестве входной информации. Сопоставляя два этих набора, ИС способна определять степень релевантности («близости») каждой из рекомендаций к условиям конкретной задачи по проектированию взаимодействия. Каждая рекомендация, таким образом, оценивается как потенциальная составляющая «решения задачи» – выходного перечня рекомендаций.

Слот *efficiency* класса *Guideline* предназначен для отражения показателей сравнительной эффективности использования рекомендаций  $\{g_1, g_2, \dots, g_N\}$  в проектировании ЧКВ, которые рассчитываются согласно следующей модели. Если  $In = \{w_1, w_2, \dots, w_V\}$  – множество проектов веб-интерфейсов, хранящихся в БЗ (экземпляров класса *Web interface design*), то можно определить бинарное нечёткое отношение *GI*:

$$GI = \{ \langle g_i, w_j \rangle, \mu_{GI}(\langle g_i, w_j \rangle) \},$$

где  $\mu_{GI}(\langle g_i, w_j \rangle)$  – функция принадлежности, которая содержательно обозначает оценку степени, в которой проект веб-интерфейса  $w_j$  следует рекомендации  $g_i$ . Далее, если  $Q = \{q_1, q_2, \dots, q_M\}$  – множество возможных показателей качества для веб-интерфейсов (экземпляров и подклассов класса *Interface quality metric*), то можно определить бинарное нечёткое отношение *IQ*:

$$IQ = \{ \langle w_j, q_k \rangle, \mu_{IQ}(\langle w_j, q_k \rangle) \},$$

где  $\mu_{IQ}(\langle w_j, q_k \rangle)$  – функция принадлежности данного нечёткого отношения, содержательно соответствующая оценке проекта веб-интерфейса  $w_j$  по показателю качества  $q_k$  (значения показателей качества, как правило, получаются в ходе применения различных методов проектирования юзабилити).

В итоге на основе *GI* и *IQ* определено нечёткое отношение *GQ*, содержательно определяющее эффективность рекомендации  $g_i$  по критерию качества  $q_k$ :

$$GQ = GI \otimes IQ = \{ \langle g_i, q_k \rangle, \mu_{GQ}(\langle g_i, q_k \rangle) \}. \quad (3)$$

В работе было рассмотрено несколько видов свёртки (в том числе максимумная и *max-prod* композиции) и, исходя из максимального разнообразия получаемых значений эффективностей, предложено определять функцию принадлежности  $\mu_{GQ}(\langle g_i, q_k \rangle)$  с использованием усредняемой *prod*-свёртки.

Тогда, полагая сравнительную значимость всех показателей качества одинаковой, можно рассчитать значение эффективности для каждой из  $N$  рекомендаций:

$$\text{efficiency}_i = \frac{1}{M} \sum_{q_k \in Q} \{ \mu_{GQ}(\langle g_i, q_k \rangle) \} \quad (4)$$

В **третьей главе** диссертационной работы предлагаются расширенные модели для различных аспектов взаимодействия в такой актуальной области ЧКВ как обеспечение общедоступности для всех категорий пользователей. В рамках исследования были поставлены два количественных эксперимента, методологической основой которых явились законы Фиттса, Хика и методика «эмоциональ-

ной инженерии». С использованием полученных результатов также уточняется структура онтологии в части классов, связанных с характеристиками пользователей, и формулируются дополнительные знания, расширяющие базу знаний ИС.

Закон Фиттса базируется на 17-й теореме Шеннона (также называемой теоремой Шеннона-Хартли) и устанавливает взаимосвязь между временем, затрачиваемым на выполнение быстрого прицельного движения ( $MT$ ), расстоянием, на которое производится движение ( $A$ ), и допустимым отклонением ( $W$ ):

$$MT = a + b \cdot ID = a + b \cdot \log_2(A/W + 1), \quad (5)$$

где  $ID$  – так называемый индекс сложности движения,  $a$  и  $b$  – оцениваемые по результатам эксперимента параметры.

Очевидно, что время, затрачиваемое на движение, не должно изучаться в отрыве от точности, с которой движение совершается, и для закона Фиттса была предложена методика учета этого фактора. Согласно ей, вместо использования «номинального» размера цели ( $W$ ), вычисляется «эффективный» размер ( $W_e$ ), который определяется исходя из предположения нормальности распределения конечных точек движения:

$$W_e = (2\pi \cdot e)^{1/2} * \sigma,$$

где  $\sigma$  – среднее квадратичное отклонение.

С использованием «эффективного» размер цели должен быть вычислен «эффективный» индекс сложности  $ID_e$ , используемый вместо «номинального»  $ID$  (5), который может включать также реально покрытое расстояние  $A_e$ :

$$ID_e = \log_2(A_e/W_e + 1). \quad (6)$$

Важно, что на основе  $ID_e$  может быть рассчитана «производительность» выполнения движений ( $TP$ ), которая является объективной и полной мерой для заданных условий эксперимента (состоящего из  $n$  исходов), учитывающей как скорость, так и точность выполнения движения:

$$TP = \frac{1}{n} \sum_{i=1}^n ID_{e_i} / MT_i. \quad (7)$$

В работе выделены две типичные для использования человеко-машинных интерфейсов операции, для которых произведено исследование влияния характеристик пользователей (с представителями различных возрастных, половых и профессиональных групп):

- 1) движение интерфейсным устройством к заранее известной пользователю цели (элементу интерфейса) – моделируемое при помощи закона Фиттса;
- 2) выбор при помощи интерфейсного устройства заранее не известной цели (примером может являться вызов контекстного или ниспадающего меню) – моделируемый законами Фиттса и Хика в сочетании.

Получены результаты, которые свидетельствуют, что наибольшее влияние в рассмотренных аспектах взаимодействия оказывают такие характеристики пользователей как возраст ( $T$ ), опыт ( $LE$  – наличие низкого опыта) и, в меньшей степени, пол. Для производительности (7) построена следующая модель:

$$TP = 6,308 - 0,998 * LE - 0,053 * (T - 18) \quad (8)$$

Показана низкая пригодность закона Хика в моделировании когнитивного аспекта взаимодействия, т.к. время выбора объектов не имело существенной зависимости от количества альтернатив. Взамен в диссертационной работе предложено, по аналогии с индексом сложности движения (5), понятие индекса сложности (*IDS*) для задач выбора:

$$IDS = ID_e + c/W, \quad (9)$$

где *c* – оцениваемый параметр – характеризует влияние сложности принятия решения о верной цели относительно индекса сложности движения (Фиттса), зависит от индивидуальных характеристик пользователей, условий эксперимента, используемых значений независимых переменных (размеров целей). Соответственно, производительность для задач по выбору объектов (*TPS*) предлагается рассчитываться исходя из *IDS* и времени, затрачиваемого на выбор (*ST*):

$$TPS = \frac{1}{n} \sum_{i=1}^n \frac{IDS_i}{ST_i}. \quad (10)$$

Построена модель, связывающая производительность выбора и характеристики пользователей, имеющая следующий вид:

$$TPS = 7,499 - 1,014 * LE - 0,049 * (T - 18) \quad (11)$$

При помощи *TPS* может производиться сравнение различных веб-интерфейсов или интерфейсных устройств с точки зрения качества взаимодействия (показателей времени, затрачиваемого на выполнение типовых задач выбора, и уровня ошибок допускаемых в процессе выполнения).

С учётом полученных результатов доработана структура онтологической модели предметной области проектирования ЧКВ: класс *Target user* («Целевой пользователь») связан с такими подклассами класса *User attribute* («Характеристика пользователя») как *User experience* («Опыт пользователя»), *User education level* («Уровень образования пользователя»), *User gender* («Пол пользователя»), *User age range* («Возраст пользователя»), *User nationality (culture)* («Национальная (культурная) принадлежность пользователя»), и т.д. (см. Рис. 3).

В четвёртой главе описывается реализация интеллектуальной системы, использование системы для решения задач предметной области, а также производится оценка качества полученного результата.

Реализована онтология для интеллектуальной системы, с использованием редактора фреймовых онтологий Protégé-Frames – открытого и бесплатного программного продукта, разработанного Стэнфордским университетом (США). Затем онтология преобразована (при помощи специального модуля CLIPSTab) в объектно-ориентированную структуру данных компонента COOL, являющегося одним из расширений языка разработки экспертных систем CLIPS (C Language Integrated Production System). Среда CLIPS (в настоящее время также находится в свободном доступе) позволяет реализовывать интеллектуальные системы на базе продукционной модели и имеет встроенный механизм вывода (решатель) с управлением очередности выполнения правил, т.е. способна поддерживать предложенную гибридную модель представления знаний. Кроме то-

го, механизм CLIPS может запускаться через веб-сервер, работающий под управлением ОС Unix, что позволило реализовать веб-интерфейс для интеллектуальной системы в каркасной системе управления содержимым Drupal и использовать язык сценариев PHP для расширения функциональности ИС, реализовав в совокупности «портал знаний».

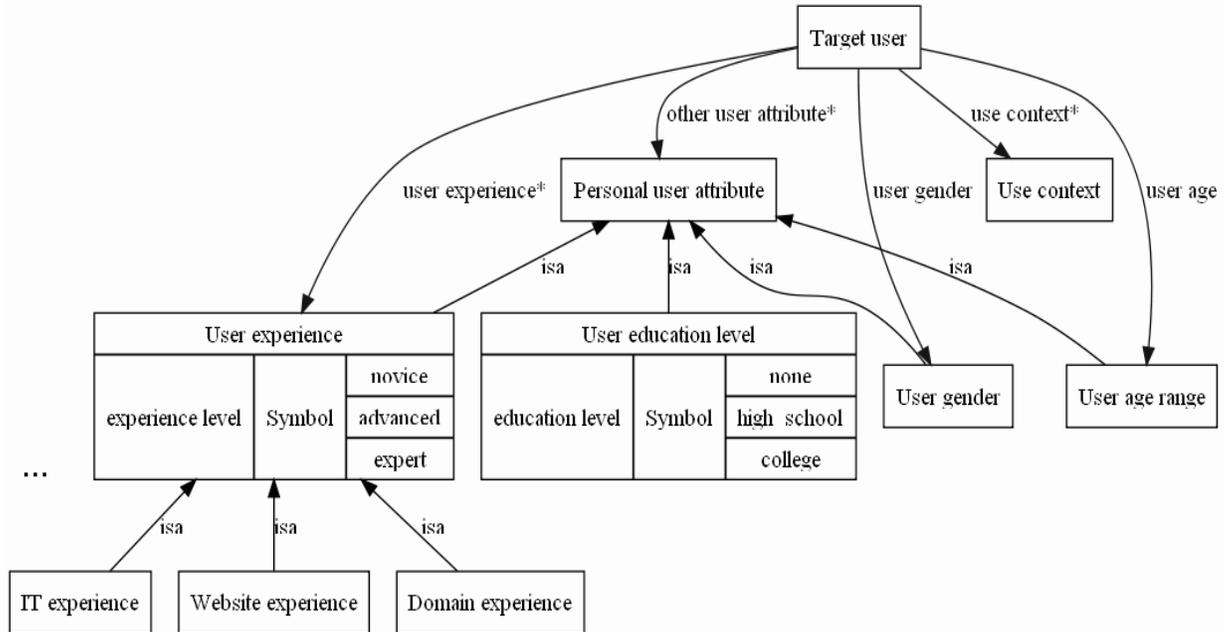


Рисунок 3. Уточненная структура классов *Target user* и *User attribute*.

Гибридный механизм вывода на знаниях во фреймовой модели включает:

1. Механизм создания структуры фреймов-фактов и конечных элементов множества  $F$ . Для обеспечения возможности использования фреймов-фактов  $f_i$ , представляющих текущую информацию в системе, был реализован механизм, осуществляющий копирование структуры онтологии (фреймов-классов из  $C$  и фреймов-слотов из  $S$ ) в прототипы фактов. При этом в структуру каждого такого прототипа, за исключением тех, которые соответствуют классам подмножества  $C_{abstract}$ , предусмотрено добавление слота с именем  $Name_{f_i}$ , значение которого служит идентификатором для факта и используется при навигации по сети фреймов. Механизм создания фреймов-фактов из прототипов, задействованный в ходе объединения входной информации и знаний, хранящихся в БЗ, по принципу работы аналогичен созданию фреймов-экземпляров из классов онтологии.

2. Механизм наследования, в том числе множественного, и создания фреймов-экземпляров – реализован средствами решателя, в котором поддерживается объектная модель, позволяющая классам-потомкам и фреймам-экземплярам из  $E$  наследовать у класса-родителя  $c_{parent}$  структуру слотов, а также методы. Наследуются не только типы слотов, но и значения по умолчанию (соответствуют «заданиям отсутствия» фреймовой модели), а решатель при создании факта  $f$  или экземпляра  $e$  способен сам подобрать начальное значение для слота.

3. Механизм логического вывода является основным механизмом решателя. Данный механизм представляет собой реализацию алгоритма прямого ло-

гического вывода (от фактов к заключениям) на основе обобщенного правила Modus Ponens, применяемого к хорновским базам знаний:

$$\frac{a_1', a_2', \dots, a_n', (a_1, a_2, \dots, a_n \Rightarrow d)}{Subst(\theta, d)},$$

где  $\theta$  – унификатор – множество подстановок значений переменных, присутствующих в левой части правила, получаемый из условия  $Subst(\theta, a_i') = Subst(\theta, a_i), i = 1, \dots, n$ . Результатом применения обобщенного правила Modus Ponens является  $Subst(\theta, d)$  – оператор применения подстановок  $\theta$  к выражению  $d$ . На каждом шаге алгоритма прямого логического вывода, решатель пытается применить каждое правило из множества  $P$  к текущему состоянию системы, – а именно, пытаясь унифицировать конъюнкты его левой части с каждым элементом множеств  $F$  и  $E$  (за исключением экземпляров классов, имеющих встроенное свойство неучастия в логическом выводе). Для обеспечения навигации по сети фреймов, используются как уникальные идентификаторы  $Name_e$  и  $Name_f$  каждого фрейма-факта  $F_F$ ), так и специальные служебные классы (*instance-address* и *fact-address*). Существует возможность управления порядком применения правил, исходя из явно заданного значения для соответствующего свойства правила, или через настройки механизма вывода (например, стратегии разрешения конфликтов). Работа механизма логического вывода завершается, когда алгоритм достиг «фиксированной точки», т.е. ни одного из ранее неиспользованных правил не может быть применено.

4. Интерфейс взаимодействия с пользователем интеллектуальной системы (ввод информации, объясняющий компонент и т.д.) вынесен в портал знаний, а, соответственно в «ядре» системы реализован интерфейс взаимодействия с порталом, которое осуществляется через файлы установленной структуры (в формате CLIPS). На входе механизм вывода получает файлы с фактами  $F$ , соответствующими характеристикам целевого пользователя и терминами из множества  $B$ , описывающим требования к разрабатываемому продукту. На выходе создается файл проекта, содержащий итоговые характеристики проекта (например, набор сервисов веб-приложения), перечень терминов из  $B$ , описывающих контекста проекта, набор соответствующих им рекомендаций, а также логическое представление прототипа веб-интерфейса.

5. Механизм расширения функционирования решателя имеет три составляющих: логическую (дополнительные правила вывода  $P$ ), процедурную (пользовательские функции) и объектно-ориентированную (механизм отправки «сообщений» фреймам-экземплярам, обеспечивающий инкапсуляцию). Соответственно, в интеллектуальной системе были реализованы правила, функции и «сообщения» для автоматического: доопределения характеристик целевого пользователя (подмножество  $P_U \subset P$ ), определения перечня типичных сервисов веб-приложения ( $P_W \subset P$ ), формирования контекста проекта ( $P_{Pr} \subset P$ ), порождения упорядоченного набора рекомендаций ( $P_G \subset P$ ) и логического представления прототипа веб-интерфейса ( $P_I \subset P$ ).

Описанные выше компоненты объединены в «портал знаний» – специализированное веб-приложение, отражающее основную (не полную) функциональность интеллектуальной системы, созданное на основе каркасной системы управления содержимым Drupal и размещённое для доступа в сети Интернет. Основным компонентом в базе знаний портала являются рекомендации из области проектирования ЧКВ для веб-приложений для: а) всех типов веб-приложений; б) веб-приложений электронной коммерции; в) веб-приложений электронного правительства и государственных услуг; г) веб-сайтов образовательных учреждений. Помимо создания нового проекта с вводом входной и получением выходной информации, портал предоставляет возможность упорядоченного (по терминам из базового подмножества  $B$  и эффективности) просмотра базы знаний рекомендаций, ввода показателей для расчёта эффективности согласно предложенному алгоритму и др.

Разработан процесс создания нового проекта, который описывается следующей последовательностью шагов:

1. Создаётся экземпляр нового проекта (класса *HCI engineering task*), для которого входной информацией должны быть заполнены значения слотов, реализующих ассоциативные отношения с классами *Target user* и *Requirement*, а выходной – с классами *Guideline* и *Web interface design*.

2. Проектировщиком через интерфейс портала знаний задаются персональные характеристики целевого пользователя продукта и контекст использования (заполняются соответствующие слоты класса *Target user*, связывающие его с подклассами классов *User attribute* и *Use context*), которые в качестве фреймов-фактов фиксируются в файле *user\*.clp*, являющимся входным для «ядра» системы. Вводимые проектировщиком требования к продукту обрабатываются механизмом анализа текста, который записывает найденные термины в виде фреймов-фактов в файл *req-tags\*.clp*, также поступающий на вход «ядра» системы.

3. В системе запускается механизм логического вывода (подмножества правил  $P_U$  и  $P_W$ ), который уточняет характеристики пользователей и проекта, заполняя слоты фрейма-экземпляра *Website*, имеющего ассоциативную связь с классом *Web interface design*. В частности, заполняются значения ассоциативного слота *website content* и формируется набор типовых сервисов веб-приложения, выбираемых среди подклассов класса *Website service (section)*.

4. Механизм логического вывода, применяя подмножество правил  $P_{Pr}$ , заполняет значения слота *protect tags* экземпляра класса *HCI engineering task*, формируя тем самым набор терминов, описывающих контекст проекта ( $Pr$ ):

$$Pr = D_{pr} \cup U_{pr} \cup R_{pr} \cup W_{pr} \cup C_{pr}, \quad (12)$$

где  $D_{pr}$  – множество терминов, добавляемых по умолчанию;  $U_{pr}$  – множество терминов, добавляемых исходя из характеристик целевых пользователей;  $R_{pr}$  – множество терминов, добавляемых по результатам анализа текста требований;  $W_{pr}$  – множество терминов, добавляемых в зависимости от типа веб-приложения (например, электронная коммерция, электронное правительство и

т.д.);  $C_{pr}$  – множество терминов, добавляемых исходя из содержимого (разделов и сервисов) веб-приложения.

5. Механизм логического вывода при помощи набора правил  $P_G$  определяет выходной набор рекомендаций (заполняя значения ассоциативного слота *design guidelines* класса *HCI engineering task*), посредством сопоставления терминов, соответствующих рекомендации, и множества  $Pr$ , сформированного на предыдущем шаге. Далее, система, применяя набор правил  $P_I$ , формирует логическое представление прототипа интерфейса (класс *Web page*, связанный с классом *Web interface design*), как упорядоченную иерархическую последовательность узлов веб-страницы (слот *web page content*, связанный с классом *Web page node*) и соответствующих им стилей оформления (ассоциативный слот, связанный с классом *CSS declaration*). Все фреймы-факты, созданные на данном и предыдущих шагах, сохраняются в файле проекта, который связан через уникальное имя проекта с экземпляром *HCI engineering task*, и механизм логического вывода завершает свою работу.

6. На основе файла проекта, специальный компонент портала знаний формирует выходной прототип интерфейса в виде кода на языке HTML и каскадной таблицы стилей CSS. Генерация кода прототипа интерфейса однозначно определяется его логическим представлением и конфигурационными значениями, хранящимися в базе знаний портала. Проектировщик получает доступ к выходной информации системы – упорядоченному набору рекомендаций для проекта (отображаемому в объясняющем компоненте системы) и прототипу интерфейса.

Разработанная интеллектуальная система использовалась при реализации ряда проектов:

1) разработке веб-интерфейса системы «АИС Электронный инспектор МЧС», входящей в состав системы автоматизации функционирования органов надзорной деятельности МЧС России – сразу для нескольких групп целевых пользователей: государственных инспекторов МЧС, сотрудников управлений и руководства МЧС, граждан и представителей юридических лиц;

2) создании внутреннего веб-приложения для швейцарской компании «Комакс АГ» (<http://komaxused.com>) – общая трудоемкость проекта составила около 750 человеко-часов, были предусмотрены версии веб-интерфейса для мобильных устройств;

3) создании веб-приложения для ОАО «Трест Сибэнергомонтаж» ([www.tsem.ru](http://www.tsem.ru)) – целевыми посетителями являются потенциальные клиенты и инвесторы, представители контролирующих государственных органов, зарубежные контрагенты;

4) созданию веб-приложения для Народного факультета (НФ) НГТУ (ход проекта и процесс оценки качества полученного решения подробно описаны в диссертационной работе).

Применение интеллектуальной системы позволило обеспечить эффективную работу проектировщика интерфейсов, что подтверждается высокими значениями показателей качества («процент успешного выполнения задач» и «субъективная оценка уровня доверия») для разработанных веб-интерфейсов.

В **заключении** сформулированы основные результаты, полученные в ходе диссертационного исследования:

1. Предложен подход к разработке инструментального средства (интеллектуальной системы) для поддержки проектирования человеко-машинных интерфейсов для веб-приложений, сочетающий организацию знаний и автоматическую генерацию прототипа интерфейса на основе моделиориентированного подхода.

2. Для реализации БЗ разработана гибридная модель представления знаний в области ЧКВ, интегрирующая онтологический подход, фреймовую и продукционную модели. Модель позволяет осуществлять как спецификацию предметной области, так и частичное замещение процедурного компонента фреймов логическими правилами вывода.

3. Осуществлено моделирование взаимодействия в человеко-машинных интерфейсах с использованием законов Фиттса и Хика, что позволило предложить расширенные модели для учёта различных характеристик пользователей.

4. Предложены расширенные модели для различных аспектов взаимодействия, а также введены понятия индекса сложности выбора *IDS* и производительности для задач выбора *TPS*, которые могут использоваться при сравнения различных человеко-машинных интерфейсов или интерфейсных устройств с точки зрения качества взаимодействия.

5. Создана действующая интеллектуальная система для поддержки проектирования ЧКВ, осуществляющая порождение упорядоченного набора рекомендаций и генерацию прототипа веб-интерфейса в зависимости от автоматически определяемого контекста проекта.

6. Реализован «портал знаний», база знаний которого содержит рекомендации по проектированию взаимодействия для различных видов веб-приложений, индексированные с использованием базовых терминов словаря и упорядоченные на основе предложенной нечёткой модели расчёта сравнительной эффективности, а также накапливает статистику реализованных проектов веб-интерфейсов и оценок их качества.

7. Созданная интеллектуальная система апробирована на ряде практических задач предметной области, причём экспериментальное тестирование полученных решений показало высокое качество взаимодействия по сравнению с аналогами из контрольной группы.

#### **ОСНОВНЫЕ ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ**

1. Бакаев М.А., Авдеенко Т.В. Разработка онтологии для поддержки проектирования человеко-компьютерного взаимодействия в сфере электронной коммерции // Вестник компьютерных и информационных технологий, №4, 2011. – С. 36-40.
2. Авдеенко Т.В., Бакаев М.А. Моделирование движения при использовании двумерных интерфейсов в человеко-компьютерном взаимодействии // Программные продукты и системы, №1, 2011. – С. 53-56.
3. Бакаев М.А. Об ограниченной применимости некоторых базовых законов в сфере человеко-машинного взаимодействия для пожилых пользователей. // Научный вестник Новосибирского государственного технического университета, №1, 2008. – С. 11-25.

4. Bakaev, M., Avdeenko, T. User Interface Design Guidelines Arrangement in a Recommender System with Frame Ontology // *Lecture Notes in Computer Science*, 2012, V. 7240, Database Systems for Advanced Applications, (Springer, 2012). – P. 311-322. [Организация рекомендаций по проектированию пользовательских интерфейсов в экспертной системе, основанной на фреймовой онтологии]
5. Bakaev M., Avdeenko T. Knowledge-Based System for Web Interface Design // *Proceedings of 2nd International Conference on Information and Multimedia Technology (ICIMT 2010)*, Hong Kong, China, Dec 2010. – Vol. 3, P.262-266. [Система, основанная на знаниях, для проектирования веб-интерфейсов]
6. Pustovalova N., Bakaev M., Avdeenko T. Knowledge-Based System for Software Requirements Analysis and Management. // *Proceedings of Knowledge and Ontology Elsewhere*, at Ershov Informatics Conference (PSI 11), Novosibirsk, July 2011. – P. 15-20. [Система, основанная на знаниях, для анализа и управления требованиями]
7. Bakaev M., Avdeenko T. Ontology to Support Web Design Activities in E-Commerce Software Development Process. // *Proceedings of IASTED Automation, Control, and Information Technology (ACIT'2010)*. June 15-18, 2010, Novosibirsk, Russia. ACTA Press, 2010. – P. 241-248. [Онтология для поддержки проектирования веб-интерфейсов в процессе разработки приложений электронной коммерции]
8. Avdeenko T., Bakaev M. Improving the Efficiency of E-government on the Basis of Intelligent Technologies // *Proceedings of International Conference on Business Intelligence and Financial Engineering*, Hong Kong, China, Dec 2011. – 8 p. [Повышение эффективности электронного правительства на основе интеллектуальных технологий]
9. Bakaev M., Avdeenko T. Rationalizing HCI Integration in E-Commerce Software Development // *Proceedings of DST-RFBR Sponsored Indo-Russian Joint Workshop on Computational Intelligence and Modern Heuristics in Automation and Robotics*, Surat, India, Sep 2010. – P. 144-155. [Интеллектуализация интеграции человеко-компьютерного взаимодействия в процесс разработки приложений электронной коммерции]
10. Bakaev M., Avdeenko T., Cheng H.I. Modelling Selection Tasks and Assessing Performance in Web Interaction // *Proceedings of IADIS International Conference Applied Computing 2011*, Rio de Janeiro, Brazil, Nov 2011. – P. 107-114. [Моделирование задач по выбору объектов и оценка производительности при взаимодействии в веб-интерфейсах]
11. Bakaev M. Fitts' law for older adults: considering a factor of age. // *Proceedings of the VIII Brazilian Symposium on Human Factors in Computing Systems*. Sociedade Brasileira de Computação, Porto Alegre, RS, Brazil, 2008. – P. 260-263. [Закон Фиттса для пожилых людей: учёт фактора возраста]
12. Bakaev M., Avdeenko T. A formal research of older adults' physical and cognitive traits in movement and selection tasks for interface design. // *Proceedings of International Association of Societies of Design Research (IASDR 2009) Conference*, Seoul, Korea, Oct 2009. – P. 255-264. [Формальное исследова-

- ние физических и когнитивных особенностей пожилых людей для проектирования интерфейсов: выполнение движений и выбор объектов]
13. Bakaev M., Lee K.H., Cheng H.I. The emotional gap between the elderly and the non-elderly. // Proceedings of the 10th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems, Seoul, Korea, Sep 2007. – Vol.10, part 1. – 7 p. [Различия в эмоциональном аспекте взаимодействия между пожилыми и обычными людьми]
  14. Bakaev M., Lee K.H., Cheng H.I. The aesthetic and emotional preferences of the elderly and the design factors for e-business web sites. Proceedings of the Eighth Pan-Pacific Conference on Occupational Ergonomics (PPCOE 2007), Bangkok, Thailand, Oct 2007. – 11 p. [Эстетические и эмоциональные предпочтения пожилых людей и факторы проектирования приложений электронного бизнеса]
  15. Bakaev M., Ponomarev V. and Prokhorova L. E-learning and Elder People: Barriers and Benefits // Proceedings of IEEE Region 8 International Conference on Computational Technologies in Electrical and Electronics Engineering (SIBIRCON 2008), Novosibirsk, Russia, 2008. – P. 110-113. [Электронное обучение и пожилые люди: барьеры и преимущества]
  16. Бакаев, М.А., Пономарев, В.Б., Прохорова, Л.В. Учет эстетических вкусов пожилых людей при дизайне сайтов электронного бизнеса // Материалы VIII международной конференции «Информатика: проблемы, методология, технологии», г. Воронеж, 7-8 февр. 2008, ИПЦ ВГУ. – Т. 1, С. 34-39.
  17. Бакаев М.А. Интеллектуальная система поддержки проектирования веб-интерфейсов (ИС ППВИ) / Бакаев М.А., Авдеенко Т.В. // Свидетельство о государственной регистрации программы для ЭВМ № 2011615212. М.: Федеральная служба по интеллектуальной собственности (Роспатент). – 2011.

Подписано в печать 28.06.2012.

Объем 1,5 п.л. Формат 60x84 1/16. Тираж 100 экз. Заказ № 806.

Отпечатано в типографии Новосибирского государственного технического университета  
630092, г. Новосибирск, пр. К. Маркса, 20, тел. +7 (383) 346-08-57