

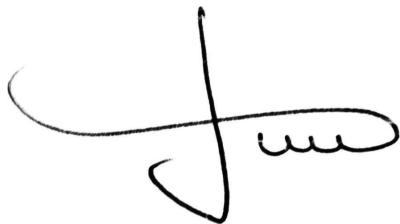
Шапкин Павел Александрович

**МОДЕЛИ И МЕТОДЫ АППЛИКАТИВНОГО МОДЕЛИРОВАНИЯ
КОНЦЕПТУАЛЬНЫХ ЗАВИСИМОСТЕЙ**

Специальность 05.13.11 — математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

Автореферат
диссертации на соискание ученой степени
кандидата технических наук

Автор:



Работа выполнена в Национальном исследовательском ядерном университете «МИФИ».

Научный руководитель: доктор технических наук, профессор
Вольфенгаген Вячеслав Эрнстович

Официальные оппоненты: доктор технических наук, профессор
Петров Андрей Евгеньевич

кандидат технических наук
Четвериков Василий Николаевич

Ведущая организация: Учреждение Российской академии наук
Институт системного анализа РАН
(ИСА РАН)

Защита состоится “21” февраля 2011 г. в 15 часов на заседании диссертационного совета Д 212.130.03 при Национальном исследовательском ядерном университете «МИФИ» по адресу: 115409, г. Москва, Каширское ш., 31.

С диссертацией можно ознакомиться в библиотеке НИЯУ МИФИ.

Отзывы на автореферат, заверенные гербовой печатью организации, просьба направлять по указанному адресу в двух экземплярах не позднее, чем за две недели до защиты.

Автореферат разослан “28” декабря 2010 г.

Ученый секретарь
диссертационного совета



Леонова Н. М.

Общая характеристика работы

Актуальность темы исследований. В настоящее время особенно остро стоит вопрос разработки моделей, методов и программных средств, нацеленных на работу с резко возрастающим объемом цифровой информации, который в 2010 г. составил 1,2 зеттабайт, и, согласно прогнозам компании IDC, возрастет в 44 раза до 2020 г. В таких условиях требуются модели данных, способные устанавливать и поддерживать структуризацию информации, которая имеет тенденцию к изменению в зависимости от динамики предметной области. Известные решения, основанные на статичной классификации, стандартные онтологии, имеющиеся предложения по семантическому Вебу в таких условиях теряют эффективность и требуют пересмотра заложенных в их основу моделей данных, программного обеспечения и информационных технологий.

В связи с рассматриваемым кругом вопросов в области аппликативного моделирования разработаны и используются чистые и прикладные теории программирования (Р. Хиндли, М. Гордон, Т. Хоар и др.), использующие формализмы лямбда-абстракций. Техника абстрагирования зависимостей получила закрепление в методе концептуального моделирования (М. Броди, Р. Шмидт, Л. А. Калиниченко и др.). Соединение обоих подходов было исследовано под руководством В. Э. Вольфенгагена и реализовано в проекте ЛАМБДА, выполненном в МИФИ. В области собственно онтологий консорциумом W3 даны рекомендации по моделям данных, предложены стандарты, носящие рекомендательный характер, и все еще не получившие реализации в виде общепринятого коммерческого программного обеспечения. Как показывает практика, модели онтологий могут рассматриваться как частный случай концептуальных зависимостей (Р. Шенк и др.). Это породило попытки установления типовой формализации, в частности различные языки дескрипций (А. Боргида, Д. МакГиннесс и др.). Тем не менее, исчерпывающее решение все еще не достигнуто.

При разработке программных систем на базе веб-онтологий определенные трудности обуславливаются тем, что языки их описания, такие как OWL (Ontology Web Language, Язык веб-онтологий), сами по себе предназначены только для описания схем данных, т. е. концептуального аспекта предметной области, и не предоставляют возможности описания методов обработки данных — операционного аспекта. Существенное значение при построении программных систем на основе веб-онтологий представляет решение задачи погружения язы-

ка дескрипций концептов в вычислительную среду языка программирования. Предлагаемые на сегодня подходы к встраиванию веб-онтологий в программные системы не позволяют в полной мере использовать как возможности основной вычислительной среды, так и возможности средств концептуального моделирования.

Для преодоления такого дисбаланса требуется соединение новейших вычислительных моделей, усовершенствованных языков описания объектов данных и связей между ними, приемов построения поддерживающего их программного обеспечения, что свидетельствует об **актуальности** темы исследования.

Цель исследования. Целью диссертационного исследования является построение аппликативной модели концептуальных зависимостей, поддерживающей настройку вычислений на фактические параметры с использованием частично упорядоченных структур концептов, а также разработка на ее основе программного обеспечения, предназначенного для создания веб-ориентированных приложений, настраиваемых на предметную область при помощи ее веб-онтологии.

Достижение поставленной цели потребовало решить следующие **задачи**:

- 1) выполнить анализ существующих средств представления концептуальных моделей (веб-онтологий) и используемых в них формализмов;
- 2) разработать модель концептов, пригодную для внедрения в аппликативную среду вычислений;
- 3) разработать формальную модель перегрузки функций на основе концептов;
- 4) разработать алгоритм диспетчеризации функций, перегруженных на основе концептов;
- 5) разработать оснащающее программное обеспечение, включая его архитектуру и интерфейсы;
- 6) разработать программный инструментарий для создания веб-приложений, основанный на разработанных принципах;
- 7) экспериментально проверить работоспособность разработанных моделей, алгоритмов и программных средств.

Объектом исследования являются веб-онтологии и веб-ориентированные программные системы.

Предмет исследования — построение частично упорядоченных систем объектов метаданных, выражаемых иерархиями в рамках концептуальных зависимостей, и связанных с программными модулями.

Методы исследования. При проведении исследований и разработок использованы методы лямбда-конверсий, частично упорядоченных систем, семантики языков программирования, концептуального моделирования, объектно-ориентированного проектирования, функционального программирования.

Научная новизна работы. В отличие от модели данных, основанной на классических онтологиях, предлагается их абстрагированный аналог, ведущий к установлению и поддержанию концептуальных зависимостей объектов программных модулей. Новизну составляет использование метода аппликативного моделирования, подразумевающего систематическое применение метаоперации аппликации при формировании связной онтологии, построенной из концептов, и для настройки шаблонированных конструкций на фактические параметры с использованием концептуальных зависимостей.

Основные научные результаты, полученные автором, состоят в следующем:

1. Предложена абстрагированная модель описаний концептов аппликативного типа, позволяющая проводить классификацию значений в функциональном языке программирования.
2. Предложена модель определения перегруженных функций, метод их верификации и алгоритм диспетчеризации на основе концептов. Разработанная модель позволяет проводить выбор конкретных реализаций для настройки программной системы на основе связных онтологий, построенных из концептов.
3. Проведена реализация разработанных моделей на объектно-ориентированном функциональном языке программирования. Реализованная библиотека позволяет проводить классификацию объектов среды Java в соответствии с иерархиями концептов, а также описывать функции, перегруженные на основе концептов, и осуществлять их диспетчеризацию.
4. Разработан набор программных инструментов создания веб-ориентированных приложений, поддерживающих настройку на предметную область при помощи веб-онтологий.
5. На основе модели перегруженных функций и метода диспетчеризации на концептах введено понятие семантических шаблонов и разработан основанный на XML язык преобразования RDF-данных с использованием веб-онтологий OWL.
6. Реализована и исследована программная система, реализующая базовые разновидности концептуальных зависимостей предметной области.

Практическую значимость и ценность проведенных исследований составляет реализованный программный инструментарий, предназначенный для создания веб-ориентированных приложений, позволяющий использовать для настройки системы концептуальную модель предметной области, представленную в декларативном виде. Использование предложенных инструментальных средств позволяет повысить эффективность обработки данных за счет возможности частичного осуществления проверок принадлежности индивидов концептам на этапе компиляции, а также снижения трудоемкости модернизации системы. Наличие алгоритма верификации перегруженных функций и валидации шаблонных систем позволяет статически выявлять многие ошибки и, таким образом, повышает надежность процессов обработки данных. Получаемая унифицированная архитектура дает свойство адаптивности, делая программные компоненты инвариантными к предметной области — как по составу объектов, так и по сложности их описаний. Основной сферой применения разработок является обработка данных семантического Веба.

Результаты, полученные в работе, могут быть использованы:

- для построения веб-ориентированных приложений в предметных областях со сложной, часто изменяемой концептуальной моделью;
- для построения инструментальных средств работы с данными семантического Веба;
- при разработке языков программирования, поддерживающих концептуальное моделирование предметных областей и функциональный стиль.

На защиту выносятся:

1. Модель данных, поддерживающая описание связанных концептов, и алгоритм диспетчеризации функций на основе концептов, предназначенные для проектирования программных систем, настраиваемых на концептуальную модель предметной области.
2. Набор программных инструментов для организации взаимодействия компонентов веб-ориентированных программных систем, управляемых веб-онтологией предметной области.
3. Язык шаблонного преобразования данных в формате RDF на основе OWL-онтологий.

Апробация работы. Теоретические и практические результаты работы доложены на следующих научных конференциях:

- III международная конференция «Web Information Systems and Technologies» (Барселона, 2007);
- IX, X и XI международная конференция «Computer Science and Information Technologies» (Уфа, 2007; Анталия, 2008; Крит, 2009);
- VII международная конференция «Научно-техническая информация» (Москва, 2007);
- Научные сессии МИФИ-2008, 2010;
- конференции «Аппликативные вычислительные системы» (Москва, 2008, 2010);
- научно-практический семинар «Фундаментальные основы информационных технологий» (Москва, 2009–2010).

По результатам выполненных исследований опубликовано 15 печатных работ, в том числе в трех журналах, включенных ВАК РФ в перечень ведущих рецензируемых научных журналов и изданий.

Программные системы, основанные на предложенных автором моделях и методах, внедрены во Всероссийском институте научной и технической информации Российской академии наук (ВИНИТИ РАН) и в Российском государственном архиве научно-технической документации (РГАНТД), что подтверждается соответствующими актами.

Результаты внедрения и эксплуатации разработанных программных средств позволяют сделать вывод о достоверности и обоснованности научных и практических результатов работы.

Структура и объем работы. Диссертация состоит из введения, четырех глав, заключения и списка литературы. Общий объем основного текста, без учета приложений — 122 страницы, с учетом приложений — 175 страниц. Диссертация содержит 34 рисунка, 8 таблиц и 11 листингов программного кода. Список литературы включает 124 источника.

Основное содержание работы

Во введении обосновывается актуальность работы, определяются цели и задачи работы.

В первой главе проведен анализ особенностей использования концептуальных моделей для создания программных систем. Рассмотрены методы при-

менения веб-онтологий при разработке программного обеспечения. Проведена оценка моделей, методов и средств построения программных систем на основе технологий семантического Веба.

Как показывают результаты проведенного анализа, в настоящее время большой интерес привлекает к себе концепция семантического Веба. Данная инициатива направлена на то, чтобы обеспечить возможность программной интерпретации данных, доступных на веб-ресурсах. Для достижения данной цели требуется наличие общедоступных спецификаций концептуализаций предметных областей (ПрОбл), которые представляются в виде веб-онтологий — совокупностей дескрипций концептов и связей между ними, или концептуальных зависимостей. Концепты представляют собой абстрактные сущности, обладающие интенционалом — правилом, в соответствии с которым вычисляется принадлежность индивидов данному концепту, и экстенционалом — множеством индивидов, удовлетворяющих данному правилу, — экземпляров концепта. На концептах задается отношение вложенности, образующее частичный порядок.

Язык описания веб-онтологий OWL DL имеет в своей основе формальную систему описания концептов — дескрипционную логику (ДЛ) *SHOIN*, основная особенность которой заключается в вычислимости вложенности концептов, что позволяет выводить не указанные в явном виде зависимости между концептами ПрОбл. Анализ доступных сегодня средств разработки программных систем с использованием веб-онтологий показал, что только часть из них позволяет использовать возможности вычисления концептуальных зависимостей, и при этом теряются возможности языка программирования, такие как статическая типизация.

При построении программной системы на основе концептуальной модели основную сложность вызывает связывание функций программной системы с концептами модели ПрОбл: необходима возможность спецификации различных способов выполнения одних и тех же задач для различных концептов, т. е. определения перегруженных функций с использованием концептов. Вопросы построения модели перегрузки и однозначной диспетчеризации функций применительно к ДЛ и, в частности, к веб-онтологиям, остаются относительно неизученными.

Во второй главе описаны результаты проведенного погружения языка дескрипций концептов в аппликативную среду вычислений, которая рассматривается в как модель языка программирования. Также описано построение модели перегруженных функций, реализации которых связываются с концептами, обес-

печивающее возможность создания программных систем, настраиваемых на концептуальную модель ПрОбл.

В качестве языка описаний концептов использована ДЛ *SHOIN*. В качестве аппликативной вычислительной среды было выбрано лямбда-исчисление второго порядка с подтипизацией (система $F_{<}$), моделирующее такие особенности объектно-ориентированных языков программирования, как наследование и обобщенные (generic) функции. Система $F_{<}$ была расширена типами-пересечениями и конструкцией *case* для проверки типов во время выполнения. В системе $F_{<}$ при помощи кодирования по Черчу были определены значения истинности *tru* и *fls*, операции над ними, а также конструкции списков с соответствующими операциями.

Погружение языка описаний концептов в систему $F_{<}$ проведено путем построения двух отображений:

- typ_C , ставящего в соответствие каждой описанию концепта C тип системы $F_{<}$;
- int_C , ставящего в соответствие каждой описанию концепта C терм системы $F_{<}$, который соответствует функции интенционала и имеет тип $typ_C \rightarrow Bool$.

Функция интенционала рассматривается как функция принадлежности индивидов концептам: индивид i считается экземпляром концепта C т. и т. т., когда выполняется условие $int_C(i) \rightarrow tru$. За счет использования типов достигается возможность частичного вычисления принадлежности индивидов концептам при помощи типизации: если выражение $int_C(i)$ не имеет типа, то индивид i не является экземпляром концепта C .

Аналогичным образом проведено погружение описаний ролей — бинарных отношений между индивидами: отображения $domTyp_R$ и $codTyp_R$ ставят в соответствие роли R типы областей определения и значений, int_R — ее интенционал.

Значения функций typ и int для различных описаний концептов приведены в табл. 1. Построенное отображение составляет модель ДЛ *SHOIN* в системе $F_{<}$, т. к. сохраняет условия принадлежности индивидов концептам, эквивалентные интерпретациям концептов в ДЛ. Из вычислимости отношений в ДЛ *SHOIN* следует вычислимость отношений, и, в частности, вложенности, для построенной аппликативной модели концептов.

Табл. 1: Отображение концептов ДЛ \mathcal{SHOIN} на типы и термы $F_{<}$.

Дескрипция	typ	int
\top	Top	$\lambda x : Top.true$
\perp	Top	$\lambda x : Top.false$
$\neg C$	Top	$\lambda x : Top.case[typ_A](\neg int_A(x))true$
$C \sqcap D$	$typ_C \cap typ_D$	$\lambda x : (typ_C \cap typ_D).(int_C(x)) \wedge (int_D(x))$
$C \sqcup D$	Top	$\lambda x : Top.case[typ_C \cap typ_D]$ $(int_C(x)) \vee int_D(x)$ $(case[typ_C](int_C(x)))$ $(case[typ_C](int_C(x)))fls$
$\exists R.C$	$domTyp_R$	$\lambda x : domTyp_R.exists int_C (int_R(x))$
$\forall R.C$	$domTyp_R$	$\lambda x : domTyp_R.forall int_C (int_R(x))$
$\geq nR.C$	$domTyp_R$	$\lambda x : domTyp_R.exists_n int_C (int_R(x))$

Перегруженные функции представлены термами типа $\mathcal{OF}(TIn, TOut)$, где TIn — тип области определения, $TOut$ — тип области значений. Рассмотрены перегруженные функции двух типов: первый тип — индивидуальные, принимающие на вход индивиды, второй — концептуальные, принимающие на вход концепты. Каждой перегруженной функции соответствует список ее *реализаций*, или *методов*, представляющих собой пару из концепта и функции-тела метода.

Определение 1 Пусть Cs — множество концептов, связанных с реализациями некоторой перегруженной функции. Данная функция называется *правильно сформированной*, если $Cs \cup \{\perp\}$ является нижней полурешеткой — множеством концептов, в которое включены все пересечения его элементов.

Определение 2 Глубиной концепта τ в наборе концептов T называется значение функции $depth(\tau, T)$, вычисляемой следующим образом:

- 1) если $\nexists C \in T. \tau \sqsubset C$, то $depth(\tau, T) = 0$;
- 2) если $\exists Cs. Cs \subset T \wedge (\forall c \in Cs. \tau \sqsubset c)$, то $depth(\tau, T) = \max_{c \in Cs}(depth(c, T)) + 1$.

Алгоритм диспетчеризации перегруженных функций построен в виде функции

$$choose(i, T) = \max_{\lambda t. depth(t, T)} \{t \mid i \in concept(t) \wedge t \in methods(T)\}.$$

Однозначность диспетчеризации при использовании данной функции показана путем доказательства теоремы 1.

Теорема 1 Если в правильно сформированной перегруженной функции F для индивида i для реализации r выполняется условие $r = \text{choose}(i, F)$, то:

- 1) данная реализация единственна;
- 2) данная реализация является наиболее специфичной для индивида i , т. е. не существует реализации, концепт которой более специфичен для i , чем у τ , и при этом содержит i :

$$\forall r' \in \text{methods}(F). i \in \text{concept}(r') \Rightarrow \tau \sqsubset \text{concept}(r) \vee r' = r.$$

В третьей главе описывается разработка программных инструментов для определения в среде Java концептов и функций, перегруженных на основе концептов. Кроме того, описан реализованный язык преобразования данных в формате RDF (Resource Description Framework, формат данных семантического Веба) на основе шаблонов.

В качестве языка реализации выбран язык Scala. Типизация в Scala основана на использованной в данной работе системе $F_{<}$: с типами-пересечениями. Наличие интероперабельности между языком Scala и платформой Java позволяет использовать имеющиеся средства вычислений в ДЛ, такие как Pellet.

Проведена реализация аппликативной модели концептов и ролей, позволяющая осуществлять классификацию объектов среды Java (рис. 1). Разработанная библиотека (Concepts.jar) предоставляет компактный синтаксис для работы с данными в формате RDF и OWL. При проведении классификации вложенность концептов вычисляется с использованием логического вывода в ДЛ, осуществляемого библиотекой Pellet. Для доступа к Pellet, а также анализа RDF и OWL документов используется библиотека Jena.

Разработанный программный инструментарий включает реализацию алгоритмов диспетчеризации и верификации перегруженных функций, т. е. проверки их правильной сформированности. На основе реализованных средств работы с перегруженными функциями разработан язык преобразования данных в формате RDF с использованием веб-онтологий в формате OWL. Основу языка составляют системы шаблонов, схожие с таблицами стилей XSL (Extensible Stylesheet Language, Расширяемый язык таблиц стилей), применяемыми для преобразования данных в формате XML. При интерпретации шаблоны в разработанном языке понимаются как реализации перегруженных функций. В дополнение к верифи-

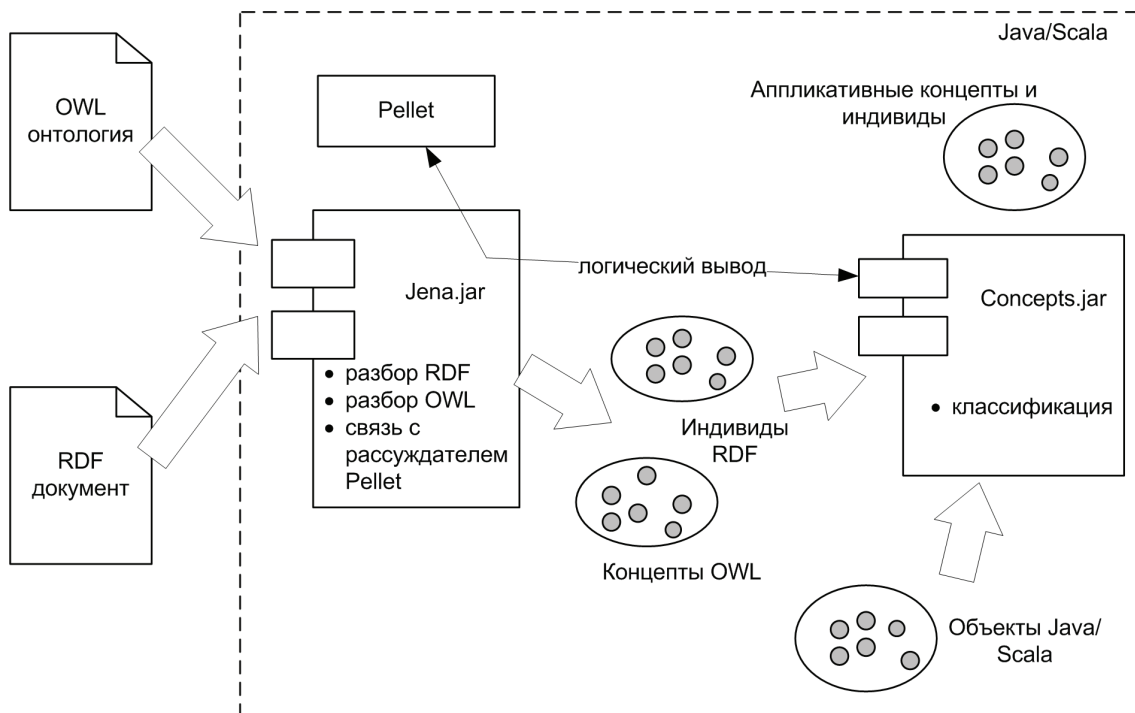


Рис. 1: Взаимодействие библиотек Concepts.jar, Jena.jar и среды Scala

кации, поддерживается возможность валидации шаблонов — проверки того, что в шаблонах используются только свойства связанных с ними классов веб-онтологии. Валидация шаблонных систем позволяет повысить надежность процесса преобразования данных, гарантируя отсутствие ошибок, вызываемых обращением к отсутствующим свойствам индивидов во время выполнения.

В четвертой главе рассмотрена реализация Электронного каталога научно-технической информации (ЭК-НТИ), выполненная для ВИНТИ РАН на основе разработанных инструментальных средств.

ЭК-НТИ предоставляет доступ через веб-интерфейс к массивам данных ВИНТИ, накапливающим большие объемы сведений о разнообразных объектах, вовлеченных в сферу научно-технической информации. ПрОбл характеризуется большим числом разнородных объектов (более 6 млн.) и связей между ними (более 14 млн.), а также устойчивой тенденцией к расширению по мере появления новых источников информации.

Семантическая характеристика объектов (тематика литературы или научных мероприятий, профили организаций и т. п.) формализуется через указание рубрик различных классификаторов, таких как Универсальная десятичная классификация, Государственный классификатор НТИ России и др. Доступ к иерархиям рубрик классификаторов осуществляется через Систему классификационных схем (СКС). СКС ВИНТИ поддерживает доступ к отображениям рубрик меж-

ду различными классификационными схемами, что позволяет автоматизировать перевод тематического описания документов на языки различных классификационных схем. В настоящее время в СКС представлено 15 классификаторов с суммарным количеством рубрик 275 270 шт.

ПрОбл каталога научной литературы представлена в виде веб-онтологии на языке OWL. Разработанная программная система имеет возможность настройки на ПрОбл благодаря набору predefined концептов: концепты ПрОбл должны быть связаны с этими внутрисистемными концептами для того, чтобы система могла их верно интерпретировать. Данные от СКС получаются в формате RDF. Данные из внутренних и внешних по отношению к ВИНТИ документальных баз данных получаются в различных форматах, однако для внутреннего представления данных в системе используется формат RDF. Функционал системы описывается перегруженными функциями, при построении интерфейса используются шаблоны, преобразующие RDF-представление данных в код HTML-страниц, возвращаемых пользователям.

Пример обработки запроса в ЭК-НТИ приведен на рис. 2. Запрос пользователя содержит идентификатор действия — получение списка экземпляров (index), и концепт — журнальные статьи, связанные с рубрикой Рубрикатора ВИНТИ «271.17». При обработке запроса проводится обращение к различным источникам данных: системе классификационных схем, локальной базе данных (БД) ВИНТИ и к внешней БД международного центра Zentralblatt. Использование СКС расширяет заданное пользователем описание концепта, давая возможность поиска во внешней БД. Процесс обработки запроса управляется концептуальной моделью ПрОбл, с концептами которой связаны реализации используемых при обработке запроса перегруженных функций. Составление HTML-ответа на запрос проводится с использованием XML-шаблонов преобразования RDF-данных.

За счет использования метаданных ПрОбл достигнута адаптивность системы и инвариантность ее программных компонентов к ПрОбл — как по составу объектов, так и по сложности их описаний. Использование функций, перегруженных на основе концептов, позволило связать функционал разработанной системы с концептуальной моделью ПрОбл, таким образом делая систему настраиваемой на данную модель. Наличие возможности настройки системы на концептуальную модель ПрОбл позволило снизить трудозатраты на модернизацию системы. Многие виды изменений осуществляются путем изменения метаданных, т. е.

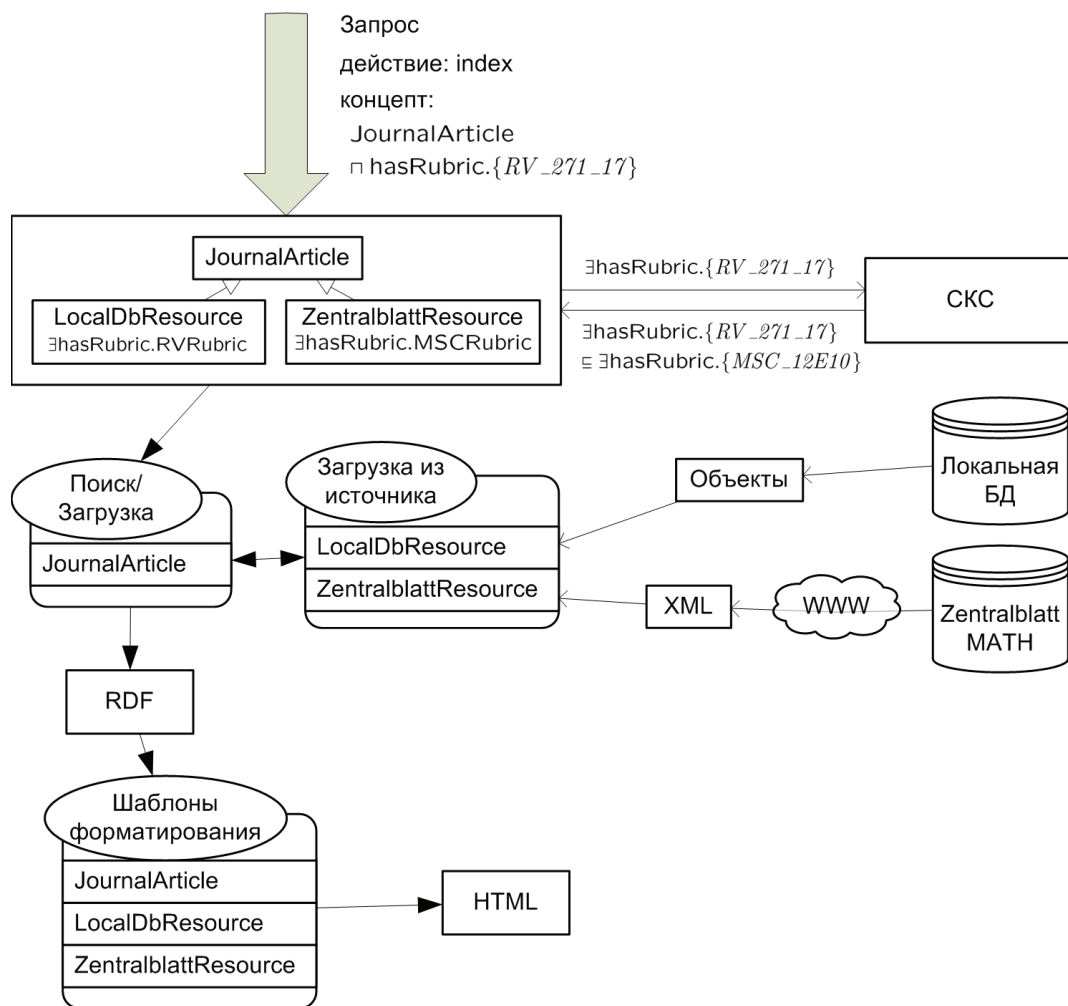


Рис. 2: Пример обработки запроса в ЭК-НТИ

концептуальной модели; при этом не требуется вмешательство программиста, требуется только труд специалиста по ПрОбл.

Для оценки эффективности использования метода разработки веб-ориентированных приложений, представленного в диссертации, был проведен анализ изменений ЭК НТИ и СКС. Производимые изменения были разделены на следующие виды: изменение функционала, добавление однотипных сущностей, добавление новых свойств сущностей, добавление связей между сущностями, изменение интерфейса. Была собрана статистика изменений ЭК НТИ и СКС за последние 5 лет. Был оценен объем трудозатрат, необходимых для внесения изменений рассматриваемых видов в предыдущую версию ЭК НТИ и в новое приложение, основанное на веб-онтологии ПрОбл. Сравнение показывает, что внедрение предложенного инструментария позволяет почти в два раза снизить трудозатраты при внесении изменений в систему, что способствует повышению эффективности процесса обработки данных в целом.

В заключении сформулированы основные результаты диссертации.

В приложениях приведены копии актов о внедрении, состав реализованного программного комплекса, код библиотеки перегруженных функций и веб-онтологии ЭК-НТИ.

Основные результаты и выводы

В представленном исследовании решена задача разработки моделей и методов аппликативного моделирования концептуальных зависимостей. В результате работы:

1. Проведено исследование методов и средств концептуального моделирования. Показано, что на сегодня не существует средств создания веб-приложений, позволяющих использовать возможности статической типизации без потери вычислимости отношений на концептах, предоставляемой при их описании в терминах дескрипционных логик. Сделан вывод, что для моделирования и реализации программных систем, поддерживающих возможность настройки на онтологию предметной области, требуется осуществление аппликативного моделирования концептуальных зависимостей.
2. Осуществлено построение аппликативной модели концептуальных зависимостей в форме производящих функций, подразумевающей систематическое применение метаоперации аппликации к абстрактным структурам зависимостей при формировании конкретной онтологии, состоящей из концептов, связанных бинарным отношением частичного порядка.
3. Предложена модель шаблонированных конструкций, связанных с конкретными дескрипциями — модель перегруженных функций. Настройка шаблонированных конструкций на фактические параметры, т. е. диспетчеризация, проводится в процессе аппликации с использованием концептуальных зависимостей. Разработанная модель формализует использование специального полиморфизма по отношению к концептуальным моделям предметных областей.
4. Разработана архитектура и интерфейсы, проведена реализация программного обеспечения, оснащающего разработанные модели. Реализована программная библиотека, позволяющая применять концепты для классификации объектов среды Java, а также определять функции, перегруженные на

основе концептов. Использование данной библиотеки позволяет симметричным образом применять в качестве индивидов как объекты Java, так и объекты, получаемые из RDF-документов; предоставляет унифицированные методы для работы с концептами веб-онтологий и классами объектно-ориентированного языка программирования.

5. Разработан язык шаблонного преобразования RDF-данных на основе концептов веб-онтологий. Предложены алгоритмы верификации и валидации шаблонных систем, применение которых позволяет снизить число ошибок при обработке данных с использованием разработанного инструментария.
6. Создан набор программных инструментов, реализующих разработанный метод. Предложена структура компонентов, показаны способы управления системами на основе метаданных. Полученная архитектура дала свойство адаптивности, сделав программные компоненты инвариантными к предметной области — как по составу объектов, так и по сложности их описаний.
7. Работоспособность предложенных методов и средств подтверждена при внедрении программных систем, управляемых веб-онтологией предметной области, во Всероссийском институте научной и технической информации Российской академии наук (ВИНИТИ РАН) и в Российском государственном архиве научно-технической документации (РГАНТД), что подтверждается соответствующими актами. Результаты внедрения и эксплуатации показали, что использование разработанных средств снижает объем трудозатрат, требуемых для настройки систем на предметную область.

Результаты работы показывают, что поставленную цель построения аппликативной модели концептуальных зависимостей, поддерживающей настройку вычислений на фактические параметры с использованием частично упорядоченных структур концептов, а также разработки на ее основе программного обеспечения, предназначенного для создания веб-ориентированных приложений, настраиваемых на предметную область при помощи ее веб-онтологии, можно считать достигнутой. Практическое внедрение разработанных программных инструментальных средств подтвердило теоретические разработки, предложенные в представленной работе, и показало их применимость к решению задачи построения программных систем на основе концептуальных моделей предметных областей.

Основное содержание диссертационной работы изложено в следующих публикациях:

1. Шапкин П. А. Применение технологий ASP.NET и Семантического Интернета для обеспечения доступа к системе классификационных схем // Научно-техническая информация, серия 2. — 2006. — № 5. — С. 20–26.
2. Shapkin P. A. Explicit Substitutions Calculi // Proceedings of CSIT'2005. Vol. 1. — Ufa, 2005. — P. 57–62.
3. Shapkin P., Shapkin A. Software tools for navigation in document databases. Developing of Information Navigation Service Based on Classification Schemes // Proceedings of the Third International Conference on Web Information Systems and Technologies (WebIST2007). Vol. “Web Interfaces and Applications” (WIA). — INSTICC Press, 2007. — P. 455–458.
4. Shapkin P. A. Web Service for Accessing Classification Scheme Mappings // Proceedings of CSIT'2007. Vol. 1. — Ufa, 2007. — P. 105–109.
5. Залаев Г. З., Меркулов В. Н., Шапкин П. А. Подход к построению АИПС с цифровыми копиями фотодокументов // Сб. трудов конференции «Научно-техническая информация 2007». — М.: ВИНТИ, 2007. — С. 248–249.
6. Шапкин П. А. Поиск документов в разнородных источниках на основе системы классификационных схем // Науч. сессия МИФИ-2008. Сб. науч. трудов. Т. 3 — М.: МИФИ, 2008. — С. 120–121.
7. Шапкин П. А. Описание Web-приложений с использованием аппликативного подхода // Труды конференции по аппликативным вычислительным системам (АВС'2008) — Москва, 2008. — С. 20–21.
8. Shapkin P. A. Potential of Using Ontologies as Models for Web Application Development // Proceedings of the Workshop on Computer Science and Information Technologies (CSIT'2008). Vol. 2. — Ufa, 2008. — P. 249–252.
9. Шапкин П. А. Разработка службы доступа к системе классификационных схем на основе формальной модели связанных понятий. — М., 2009. — 60 с., ил. — Деп. в ВИНТИ РАН 28.09.09, № 575-B2009.

10. Шапкин П. А. Использование онтологий при разработке веб-приложений, настраиваемых на предметную область. // Информационные технологии и вычислительные системы. — 2009. — № 2. — С. 44–50.
11. Shapkin P. A. Developing Web Information Systems On the Basis of Domain Ontology // Proceedings of the Workshop on Computer Science and Information Technologies (CSIT'2009). Vol. 1. — Crete, Greece, 2009. — Pp. 120–123.
12. Шапкин П. А. Разработка веб-приложений на основе онтологий и семантических шаблонов // XIII Международная телекоммуникационная конференция студентов и молодых ученых «Молодежь и наука». Тезисы докладов. Ч. 2. — М.: НИЯУ МИФИ, 2010 — С. 28–29.
13. Шапкин П.А. Модели и методы разработки Веб-приложений на основе онтологии предметной области // Информационные технологии. — М.: Новые технологии, 2010. — № 2. — С. 13–18.
14. Шумский Л. Д., Шапкин П. А. Преобразование данных в формате RDF на основе веб-онтологий // Аппликативные вычислительные системы: Труды 2-й международной конференции по аппликативным вычислительным системам (АВС'2010) / Под ред. В. Э. Вольфенгагена. — М.: НОУ Институт Актуального образования «ЮрИнфоР-МГУ», 2010. — С. 195–204.
15. Шапкин П. А. Вычисления с концептами в аппликативном языке программирования. // Аппликативные вычислительные системы: Труды 2-й международной конференции по аппликативным вычислительным системам (АВС'2010) / Под ред. В. Э. Вольфенгагена. — М.: НОУ Институт Актуального образования «ЮрИнфоР-МГУ», 2010. — С. 205–213.