



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
имени Н.Э. БАУМАНА

Учебное пособие

А.И.Власов

Методические указания по выполнению домашнего задания
по курсу

«Системы искусственного интеллекта»

МГТУ имени Н.Э. Баумана

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
имени Н.Э. БАУМАНА

А.И.Власов

Методические указания по выполнению домашнего задания
по курсу

«Системы искусственного интеллекта»

Москва
МГТУ имени Н.Э. Баумана

2012

УДК 681.3.06(075.8)
ББК 32.973-018
И201

А.И.Власов
Методические указания по выполнению домашнего задания по курсу «Системы искусственного интеллекта» –
М.: МГТУ им. Н.Э. Баумана, 2012. – 33 с.: ил.

В данных методических указаниях рассматриваются вопросы реализации экспертной системы по поддержке конструкторско-технологического проектирования в рамках домашнего задания по курсу «Системы искусственного интеллекта» за 9 семестр по кафедре ИУ4 «Проектирование и технология производства ЭС» МГТУ им.Н.Э.Баумана. Основное внимание уделено моделированию базы знаний, установлению связей и проведению декомпозиции узлов древовидной структуры. Представлена логическая модель системы, модели компонентов и развертывания. Приведены примеры методического обеспечения комплекса: руководства пользователя и инструкций по развертыванию системы.

Ил. 39. Табл. 5. Библиогр. 7 назв.

УДК 681.3.06(075.8)

© МГТУ им. Н.Э. Баумана, 2012

СОДЕРЖАНИЕ

	с.
СПИСОК ОСНОВНЫХ ПОНЯТИЙ И ТЕРМИНОВ	5
СПИСОК УСЛОВНЫХ СОКРАЩЕНИЙ	8
ВВЕДЕНИЕ	9
1 РАЗРАБОТКА И РЕАЛИЗАЦИЯ В СООТВЕТСТВИИ С УТВЕРЖДЕННЫМ ТЕХНИЧЕСКИМ ЗАДАНИЕМ ПРОГРАММНОГО КОМПЛЕКСА ЭКСПЕРТНОЙ СИСТЕМЫ	15
1.1 Структурно-функциональная модель ЭС	15
1.1.1 Контекстная диаграмма работы ЭС	15
1.1.2 Первый уровень декомпозиции «Работа ЭС»	16
1.1.3 Разработка иерархической диаграммы работы ЭС	18
1.1.4 Спецификация ролей пользователей системы	18
1.2 Модели вариантов использования и деятельности экспертной системы	20
1.2.1 Диаграмма вариантов использования ЭС	20
1.2.2 Диаграмма последовательности действий прецедента управления фреймами	22
1.2.3 Диаграмма кооперации для прецедента управления фреймами	23
1.2.4 Диаграмма последовательности действий прецедента синтеза решения	24
1.2.5 Диаграмма состояний ЭС для синтеза решения	24
1.2.6 Диаграмма деятельности для авторизации	25
1.2.7 Диаграмма деятельности для авторизованных пользователей ЭС, имеющие уровень доступа «администратор»	27
1.2.8 Диаграмма деятельности для авторизованных пользователей ЭС, имеющие уровень доступа «эксперт»	28
1.3 Логическая модель ЭС	30
1.3.1 Диаграмма пакетов ЭС	30
1.3.2 Диаграмма классов для БД	31
1.3.3 Диаграмма классов системы ввода-вывода	33
1.3.4 Разработка физической информационной модели (классы сущности)	35
1.3.5 Диаграмма деятельности синтеза решения	36
1.4 Модель реализации ЭС - диаграмма компонентов ЭС	37
1.5 Модель развертывания ЭС	40
1.5.1 Вычислительный кластер «ННС»	40
1.5.1.1 Структура вычислительного кластера «ННС»	40
1.5.1.2 Пользовательский ресурсный хаб	41
1.5.1.3 Информационный портал вычислительного кластера	42
1.5.2 Диаграмма развертывания комплекса	42
Выводы по разделу	44

2 МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ЭКСПЕРТНОЙ СИСТЕМЫ	45
2.1 Реализация инсталляционного комплекта	45
2.2 Развертывание ЭС	45
2.2.1 Постановка задачи	45
2.2.2 Генерация SQL-скриптов	45
2.2.3 Установка файлов-скриптов	45
2.3 Инструкция работы пользователя	46
2.3.1 Модуль «авторизация»	46
2.3.2 Модуль «Регистрация»	47
2.3.3 Модуль «Синтез решения»	47
2.3.4 Модуль «Кабинет»	48
2.4 Инструкция работы эксперта	49
2.4.1 Модуль «Слоты»	49
2.4.2 Модуль «Редактирование слота»	49
2.4.3 Модуль «Прототипы»	50
2.4.4 Модуль «Редактирование прототипа»	51
2.4.5 Модуль «Экземпляры»	51
2.4.6 Модуль «Редактирование экземпляра»	52
2.5 Инструкция работы администратора	52
2.5.1 Модуль «Пользователи»	52
2.5.2 Модуль «Карточка пользователя»	53
Выводы по разделу	54
ЗАКЛЮЧЕНИЕ	56
Список использованных источников	58
ПРИЛОЖЕНИЕ А. ЛИСТИНГИ SQL-СКРИПТОВ	59
ПРИЛОЖЕНИЕ Б. ЛИСТИНГИ PHP-СКРИПТОВ	61

СПИСОК ОСНОВНЫХ ПОНЯТИЙ И ТЕРМИНОВ

понятие	определение
Методическая деятельность по дисциплине	Комплекс мероприятий по организации и методическому обеспечению учебного процесса в виде совокупности следующих процессов: – разработка программы дисциплины для определенного учебного плана и ее регулярная корректировка в установленном порядке; – постановка соответствующего курса лекций; – организация и методическое обеспечение практических (семинарских) занятий; – организация и методическое обеспечение лабораторных работ; – организация и методическое обеспечение самостоятельной работы студентов; – методическое обеспечение текущего контроля успеваемости и промежуточной аттестации студентов по дисциплине; – обеспечение студентов учебниками, учебными пособиями и учебно-методической литературой по дисциплине; – обеспечение студентов электронными учебными материалами по дисциплине; – обеспечение учебного процесса лабораторным оборудованием и компьютерной техникой; – методическое обеспечение применяемых инновационных методов и технологий; – контроль и организация методического обеспечения по дисциплине
Обеспечение качества	Комплекс взаимосвязанных действий в рамках системы менеджмента качества образовательной деятельности университета по эффективной эксплуатации учебно-методических комплексов для разных дисциплин в виде единообразных компонентов информационной модели организации и методического обеспечения учебного процесса с целью получения комплексных оценок оценки качества предоставляемых образовательных услуг по конкретной дисциплине
Учебно-методический комплекс	Совокупность регламентирующих документов, учебных материалов и материально-технической базы по конкретной дисциплине и сведений, характеризующих условия реализации учебного процесса по данной дисциплине, является частью основной образовательной программы высшего учебного заведения, разрабатывается на основе государственного образовательного стандарта, создается с целью оптимизации содержания обучения, повышения качества и эффективности проведения занятий по дисциплине, написания качественных учебников, учебных пособий и учебно-методической литературы и входит в состав контролируемых нормативных документов кафедры. УМК состоит из методического, материально-технического, информационного, методического (обучающего (содержательного) и контролирующего) обеспечений.
Программа дисциплины	Основной методический компонент по дисциплине, содержащий – цели и задачи изучения дисциплины, соотнесенные с общими целями основной образовательной программы, в том числе имеющие междисциплинарный характер или связанные с задачами воспитания; – содержание дисциплины, структурированное по видам учебных занятий с указанием их объемов; – учебно-методическое обеспечение дисциплины, включая перечень основной и дополнительной литературы, методические рекомендации (материалы) преподавателю и методические указания студентам; – требования к уровню освоения программы и формы текущего контроля успеваемости и промежуточной аттестации студентов.
Методические рекомендации	Комплекс материалов представляемых в виде приложения к программе дисциплины и должны указывать на средства и методы обучения, способы учебной деятельности студентов, применение которых для освоения тех или иных тем и разделов наиболее эффективно.
Методические указания для студентов	Комплекс материалов, раскрывающих рекомендуемый режим и характер различных видов учебной работы по данной дисциплине, в том числе запланированных видов самостоятельной работы студентов (СРС)

Регламент контроля компетенций	Материалы, устанавливающие содержание и порядок проведения текущего контроля успеваемости и промежуточной аттестации студентов по данной дисциплине, должны формироваться с учетом действующего «Положения об организации учебного процесса»
Блок дисциплин	Крупная часть учебной программы, имеющая определенную функциональную и логическую завершенность по отдельному направлению области компетенций по отношению к установленным целям и результатам обучения, воспитания.
Модуль дисциплин	Логически завершенная часть учебной программы, состоящая из совокупности связанных дисциплин.
Модуль дисциплины	Часть учебной дисциплины, имеющая определенную логическую завершенность по отношению к установленным целям и результатам обучения, воспитания.
Программа дисциплины	Программа, соответствующая требованиям ГОС ВПО и учитывающая специфику подготовки студентов по избранному направлению или специальности.
Учебник	Печатное или электронное издание, содержащее систематическое изложение учебной дисциплины, соответствующее программе и официально утвержденное в качестве данного вида издания. Основное средство обучения. Учебник является центральной частью учебного комплекса и содержит материал, подлежащий обязательному усвоению
Учебное пособие	Учебное издание, официально утвержденное в качестве данного вида издания, частично заменяющее, или дополняющее учебник. Учебные пособия предназначены для расширения, углубления и улучшения усвоения знаний, предусмотренных программами и изложенных в учебнике. Кроме того, учебные пособия выпускают по вновь создаваемым курсам или курсам по выбору
Курс лекций	Материал, подготовленный с целью организации самостоятельной работы студентов. В нем на основе преподавательского конспекта лекций полностью рассмотрено содержание дисциплины. Курс лекций обычно является авторской работой, в которой возможно рассмотрение конкретных проблем, спорных вопросов с позиций автора.
Конспект лекций	Материал, который освещает содержание дисциплины в наиболее обобщенной, компактной форме, предоставляя студентам лишь самую существенную информацию. Обычно готовится по новой дисциплине или в дополнение к учебнику. Как правило, в нем рассматривают новые проблемы, раскрывают альтернативные решения, характеризуют оригинальные направления развития отрасли, конкретных объектов техники или разделов науки.
СДО	Система дистанционного обучения
LMS	Learning Management System (Системы управления обучением)
LCMS	learning content management system (Системы управления учебным контентом)
IMS QTI	Question & Test Interoperability (Спецификация совместимости вопросов и систем тестирования)
LOM	Learning Objects Metadata (Метаданные об учебных объектах)
ADL Co-Laboratory (ADL Co-Lab) Network	Сеть организаций созданных для совместного исследования, разработки и определения основных инструментов, стандартов, содержания и рекомендаций для продвинутого распределенного обучения.
SCORM Run-Time Environment (RTE) Application Program Interface (API)	Механизм связи для информирования системы управления обучением (LMS) о состоянии учебного объекта (начало прохождения, завершение, условие возникновения ошибки). Интерфейс прикладных программ используется для установки и просмотра данных (балл, временной лимит и т. д.) между системой управления обучением (LMS) и учебным объектом.
Assets	Электронное представление данных (текст, картинки, звук, web-страницы или другие виды данных), которые могут быть переданы средствами web-технологий.
Content Organization (организация построения материалов)	Стандарт, описывающий структуру и пути взаимодействия с материалом

Content Packaging (Размещение материалов)	Стандарт для идентификации и обмена учебными ресурсами между различными системами и инструментами. «Размещение материалов» также может определять «Организацию построения материалов» (Content Organization).
SCORM Run-Time Environment (RTE) Data Model	Стандарт набора элементов, используемых для определения связанной информации, такой как статус учебного ресурса. В самом простом виде модель данных определяет элементы, которые должны быть «знакомы» как системе управления обучением (LMS), так и учебному материалу. Система управления обучением (LMS) должна следить за состоянием запрошенных во время сессии элементов.
Learning Management System (LMS) (Система управления обучением)	Программное обеспечение, которое автоматизирует административные события во время обучения с помощью набора средств, которые запускают учебный объект, следят за прогрессом ученика, упорядочивают учебный объект и создают отчет для структуры, контролирующей деятельность обучающегося
Metadata (Метаданные)	Информация о материале включающая описание характеристик и св язей между категориями.
Asset Metadata	Метаданные могут быть использованы вместе с учебными объектами для сопровождения этих учебных объектов информацией о их связи с обучающими системами. Эти метаданные нужны для облегчения многократного использования и изучения структуры, к примеру, в рамках, хранилищ данных при создании нового материала.
Content Organization Metadata	Метаданные описывают «организацию построения материалов» (Content Organization). Целью использования метаданных описания построения материалов является упрощение понимания структуры, к примеру, хранилища данных и сопровождение информацией об организации внутри материала, которое входит в функции «организацию построения материалов» (Content Organization).
Sharable Content Object (SCO) Metadata	Метаданные могут быть применены к учебным объектам (SCO) для описания информации о материале, который содержит объект. Эти метаданные нужны для облегчения многократного использования объекта и возможности изучения его структуры.
Sharable Content Object (SCO) (Учебный объект)	Коллекция одного или нескольких учебных материалов, представляющая собой единичный ресурс, который можно запустить и который можно связать с системой управления обучением (LMS) с помощью рабочей среды SCORM (SCORM Run-Time Environment (RTE)). Учебный объект (SCO) представляет собой самый низший уровень в иерархии ресурсов который можно связать с системой управления обучением (LMS) с помощью рабочей среды SCORM (SCORM Run-Time Environment (RTE))
The Sharable Content Object Reference Model (SCORM®)	Согласованный набор спецификаций и стандартов который описывает взаимосвязь учебных объектов, модулей данных и правил для доступности учебного объекта внутри системы построенной по такой же модели
SCORM Content Aggregation Model (CAM) (Модель соединения учебного материала)	Метод, который описывает компоненты, используемые в образовательной практике, как использовать эти компоненты для обмена между системами, как описать эти компоненты для удобства поиска и упрощения представления, и как определить правила для этих компонентов
SCORM Run-Time Environment (RTE) (Рабочая среда SCORM)	RTE описывает требования системы управления обучением (LMS) по созданию рабочей среды (к примеру, процесс запуска учебного объекта, стандартизация соединения между учебным объектом и системой управления обучением (LMS), и стандартизация элементов модели данных, используемых для передачи информации в соответствии с опытом учащегося и уровнем сложности материала).
SCORM Sequencing and Navigation (SN) (Навигация и последовательность действий)	Информация и правила которым должна следовать система управления обучением для выстраивания учебного процесса. Эта информация входит в рамки «структуры учебного материала» и определяет организационную часть «Размещения материалов» (Content Packaging).

СПИСОК УСЛОВНЫХ СОКРАЩЕНИЙ

АСМ	– атомно-силовая микроскопия
АСУ	– автоматизированная система управления
АЧХ	– амплитудно-частотная характеристика
ВОПГ	– высокоориентированный пиролитический графит
БД	– база данных
БЗ	– база знаний
МПВ	– метод постоянной высоты
МПТ	– метод постоянного тока
ОС	– операционная система
ПК	– персональный компьютер
ПО	– программное обеспечение
ППС	– пилообразная периодическая структура
ПКЗ	– проектно – конструкторские задачи
РП	– рабочая память
СЗМ	– сканирующая зондовая микроскопия
СТМ	– сканирующая туннельная микроскопия
СУБД	– систему управления базами данных
СЭМ	– сканирующий электронный микроскоп
ТЗ	– техническое задани
ЦАП	– цифро-аналоговый преобразователь
ЭС	– экспертная система
API	– Application Programming Interface
СМАР	– комплект инструментов для моделирования знаний
CSS	– Cascading Style Sheets (каскадные таблицы стилей)
FFT	– Fast Fourier Transform
HTML	– hypertext markup language
J2SE	– Java 2 Standard Edition
JRE	– Java Runtime Environment
MYSQL	– свободная система управления базами данных, принадлежащая Oracle Corporation
OpenGL	– Open Graphics Library
PHP	– Personal Home Page (скриптов язык программирования)
RUP	– Rational Unified Process
SQL	– Structured Query Language (язык структурированных запросов)
UI	– user interface (интерфейс пользователя)
UML	– Unified Modeling Language

ВВЕДЕНИЕ

Актуальность данной работы заключается в том, что в ходе развития микро и нанотехнологий возникла острая потребность в обеспечении единой точки доступа к понятийной информации по технологиям микро- и наносистем, предоставление общего механизма поиска и визуальной навигации по семантическим знаниевым картам понятий данной предметной области [1].

В настоящее время для накопления, систематизации и хранения данных и для оперирования знаниями применяют *базы знаний*. Полноценные базы знаний содержат в себе не только фактическую информацию, но и правила вывода, допускающие автоматический синтез концепций о вновь вводимых фактах и, как следствие, осмысленную обработку информации.

База знаний – важный компонент интеллектуальной системы. Наиболее известный класс таких программ – это экспертные системы. Они предназначены для поиска способов решения проблем из некоторой предметной области, основываясь на записях БЗ и на пользовательском описании ситуации.

Экспертная система (ЭС) – компьютерная программа, способная частично заменить специалиста-эксперта в разрешении проблемной ситуации. Современные ЭС начали разрабатываться исследователями искусственного интеллекта в 1970-х годах, а в 1980-х получили коммерческое подкрепление. Предтечи экспертных систем были предложены в 1832 году С. Н. Корсаковым [2], создавшим механические устройства, так называемые «интеллектуальные машины», позволявшие находить решения по заданным условиям. Например, определять наиболее подходящие лекарства по наблюдаемым у пациента симптомам заболевания.

В информатике экспертные системы рассматриваются совместно с базами знаний как модели поведения экспертов в определенной предметной области с использованием процедур логического вывода и принятия решений, а базы знаний — как совокупность фактов и правил логического вывода в выбранной предметной области деятельности.

Объектом исследования в рамках данной работы являются «знания» в области технологий микро- и наносистем, их методы исследования, области применения, маршруты производства и проектирования. Информация относительно микро и нанотехнологий весьма обширна и известны попытки ее систематизировать и аккумулировать в концентрированном виде и как-то обобщить. Кроме обычных баз данных (информационных, патентных) можно указать на пример построения «дерева нанотехнологий», а также на различные «дорожные карты нанотехнологий». Эти прототипы плохо структурированы и могут решать ограниченное число локальных задач.

Появление сред для построения концептуальных графических схем (СМАР) с возможностью неоднократной детализации вложенных концептов создало предпосылки для построения семантических баз знаний с визуальной интерпретацией семантики понятий. В качестве наиболее значимой работы по концепт-картам и их использованию для систематизации и структурирования знаний и обучения следует указать работы Дж. Новака и А. Канаса [3]. В них впервые была сформулирована методика расстановки концептуальных понятий в вершины ориентированного графа. Ребра графа отражали связи между концептами. Была также предусмотрена методика расшифровки концепта путем вложения подчиненной структуры в соответствующую вершину графа.

Задачами проекта являются: предоставление российскому научному сообществу возможности использования ресурсов экспертной системы для семантической поддержки методов формализации и сравнения научно-технологических решений в области технологий микро- и наносистем, которое должно обеспечить совершенствование научно-технической и инновационной деятельности в сфере nanoиндустрии, информационную поддержку при создании новых технологий и при использовании наноматериалов и нанотехнологий; обеспечение единой точки доступа к информации по технологиям микро- и наносистем, предоставление общего механизма поиска и визуальной навигации по семантической информации, реализуемой на платформе тонкого клиента. Указанные возможности должны ускорить внедрение в практику технологий микро- и наносистем и облегчить их выход на соответствующие рынки.

Цель работы: структуризация, комплексный анализ и представление в виде онтологии комплекса научных, технических и технологических знаний по технологиям микро- и наносистем, сетевая каталогизация и обеспечение доступа к семантическим ресурсам по предметной области, способствующие решению научных, инженерных, производственных и образовательных задач в сфере технологий микро- и наносистем.

Для достижения поставленной цели в работе решен следующий комплекс задач:

- Разработка онтологии предметной области технологий микро- и наносистем, включающей:
 - определение содержания и границ предметной области технологий микро- и наносистем;
 - определение и согласование задач, возлагаемых на разрабатываемую онтологию;
 - формирование иерархической структуры лиц, заинтересованных в использовании онтологии,
 - разработку и реализацию модели запросов указанных лиц;
 - разработку базисной структуры онтологии технологий микро- и наносистем, в том числе:
 - формирование объектной структуры предметной области технологий микро- и наносистем;
 - выделение базисных типов сущностей предметной области технологий микро- и наносистем (материалы, методы и технологии, задачи);
 - построение иерархической объектной модели базисных типов сущностей предметной области технологий микро- и наносистем;
 - определение структуры и содержания запросов к разрабатываемой онтологии;
 - формирование базисной семантики онтологии;
 - согласование структуры основных аксиом и фреймов онтологии;
 - разработку и утверждение базисной структуры концептов онтологии;
 - определение структуры базы знаний (БЗ) для анализа методов и технологий, а также сравнения научно-технических решений в области технологий микро- и наносистем;
 - определение источников и формирование системы каталогизации информационных объектов в области технологий микро- и наносистем;
 - проверку уровня компетентности проектируемой онтологии.
 - разработка структуры базы знаний по технологиям микро- и наносистем в терминах семантического языка описания онтологий для анализа методов и технологий, а также сравнения научно-технических решений в области технологий микро- и наносистем.
- опытная апробация подсистемы генерирующего синтеза бз, включающей не менее 5 примеров решения задач прогнозирования появления новых решений в области технологий микро- и наносистем.
- создание базы знаний по технологиям микро- и наносистем на основе контекстной модели базисных типов сущностей предметной области технологий микро- и наносистем, включающее:
 - определение задач, возлагаемых на разработанную онтологию,
 - разработку структуры и содержания запросов к разработанной онтологии,
 - разработку и реализацию навигации в виде концептуальной карты и посредством генерационного интерактивного синтеза на основе фреймовой семантической сети,
 - разработку и реализацию системы каталогизации информационных источников, ассоциированных с разрабатываемой базой знаний.
- разработка проектов нормативных документов и технических регламентов по организации взаимодействия с базой знаний научных, образовательных, производственных и коммерческо-внедренческих структур, осуществляющих деятельность в области технологий микро- и наносистем.

- разработка и согласование с заказчиком технического задания на разработку программного комплекса экспертной системы по технологиям микро- и наносистем.
- разработка в соответствии с утвержденным техническим заданием программного комплекса экспертной системы по технологиям микро- и наносистем.
- наполнение базы знаний терминологическими записями, в том числе содержащимися в материалах Учебно-методических комплексов.
- разработка методических материалов по использованию базы знаний технологиям микро- и наносистем для решения задач прогнозирования появления новых технологий.
- внедрение программного комплекса экспертной системы по технологиям микро- и наносистем в организации - участнике национальной нанотехнологической сети.
- организация системы консультирования пользователей базы знаний.

Методы исследований: методы генерационного интерактивного синтеза на основе фреймовой семантической сети, формализация знаний осуществлялась построением концептуальных карт с использованием СМАР-технологий, логическая модель, модель реализации и развертывания осуществлялась при помощи языка UML.

Научная и практическая новизна работы заключается в новом подходе к структуризации, комплексному анализу и представлению комплекса научных, технических и технологических знаний по технологиям микро- и наносистем. Используемый подход заключается в создании онтологии области технологий микро- и наносистем, сетевой каталогизации и обеспечении доступа к семантическим ресурсам по предметной области, способствующие решению научных, инженерных, производственных и образовательных задач в сфере технологий микро- и наносистем. Результатом такого подхода является создание базы знаний по технологиям микро- и наносистем для целей анализа методов и технологий, а также сравнения научно-технических решений в указанной области.

Созданная база знаний обладает уникальными когнитивными свойствами. В системе мыследеятельностной методологии такая база содержит два типа схем:

- объектно-онтологические схемы (изображения объектов деятельности и мышления);
- организационно-деятельностные (представления самой деятельности и мышления).

Схема выступает средством экспликации знания. В схеме присутствует только *функциональное место объекта*, знание рассматривается как наполнение этого функционального места. *Использование в схеме различных функциональных мест для объектов и знаний, позволяет схематизировать как объекты деятельности и мышления, так и саму деятельность и мышление.* Таким образом, знание эксплицируется в схематизации по-разному:

- как наполнение функциональных мест, специально вводимых в схеме для изображения знания;
- как наполнение всех функциональных мест и знаков на схеме в отношении значения (если каждый знак - что-то значит/означает, то содержанием этого значения/означения выступает не что иное, как знание).

Различение в схеме и в каждом отдельном элементе (знаке) схемы функционального места и наполнения этого места содержанием (значением) делает схему не только средством экспликации знания, но и средством познания. Одновременно это обеспечивает возможность контроля усвоения знаний, их адекватного встраивания во внутриличностное семантическое поле обучаемого.

Над графическими моделями с-тар можно производить некоторые действия (операции), которые определяют целевую трансформацию базы знаний:

Фокусировка – изменение исходной графической модели с-тар, заключающееся в выделении главных – фокусных – графических операторов и (или) связей между ними.

Расстановка акцентов и приоритетов – изменение исходной графической модели, заключающееся в выделении важных и (или) первоочередных графических операторов и (или) связей.

Достройка – изменение исходной графической модели, заключающееся в добавлении необходимых графических операторов (по типу интерполяции).

«Запуск» – построение операционной графической модели, исходя из имеющейся предметной.

«Остановка» – построение предметной графической модели, исходя из имеющейся операционной.

Смена знака, выворачивание, построение наоборот («диверсионная графика») – построение графической модели, противоположной исходной. (инверсия, метод отрицания иконструирования).

В основу подхода к созданию базы знаний положены мировые достижения в области создания спецификаций на форматы описания информационных ресурсов (Protégé Programming Development Kit (PDK), W3C, RDF(S), OWL, XML Schema, SOA, CMAP).

Разработанный программный комплекс основан на общепринятых открытых международных стандартах, в том числе стандартах Web-технологий и Web-сервисов (XML, HTTP, SMTP, FTP, SOAP, WSDL и др.), а программная реализация осуществлялась с использованием современных языков, технологий программирования и средств разработки информационных систем.

Практическая ценность работы состоит в том, что формализованная модель знаний обеспечила возможность создания экспертной системы, позволяющей хранить и осуществлять поиск знаний по предметной области. Реализации экспертной системы по технологии тонкого клиента не ставит жестких требований к аппаратному обеспечению пользователя или эксперта.

Реализованные технические решения обеспечивают минимизацию затрат эксплуатирующей организации на поддержание программного комплекса в актуальном состоянии, возможность с минимальными затратами проводить интеграцию тематических баз знаний внешних и смежных производителей.

Результаты работы применимы для использования во всех узлах национальной нанотехнологической сети, где циркулирует профильная информация, а также в головной научной организации и головных организациях отраслей по тематическим направлениям деятельности ННС.

Разработка базы знаний осуществлялась на основе современных моделей и методов формализованного представления знаний. При этом обеспечивается функция автосохранения материалов при возникновении программно-аппаратных сбоев.

Количество концептов в базе знаний покрывает 90% понятий предметной области технологий микро- и наносистем. Уровень аксиоматизации базы знаний допускает инверсные запросы по 60% концептам. Степень покрытия понятий области соответствующими концептами и степень аксиоматизации базы знаний определена по методологии, разработанной исполнителем.

Объектами защиты от несанкционированного доступа являются:

- файлы бд.
- файлы индекса подсистемы индексирования.
- данные, хранящиеся в бд комплекса.
- несанкционированный запуск и останов сервисов сбора информации и индексирования.

– исполняемые файлы субд, сервера приложений, операционной системы.
защита информации от несанкционированного доступа должна быть реализована на следующих уровнях:

- аутентификация и авторизация доступа средствами операционной системы.
- разграничение доступа к файлам с помощью средств операционной системы .
- аутентификация и авторизация доступа к данным и настройкам субд с использованием средств субд .

- использование аутентификации при обращении к подсистеме управления комплекса.

Создаваемый программный комплекс имеет встроенные средства разграничения доступа к их частям и должен предусматривать:

- централизованное администрирование;

- возможности ограничения работы пользователя программного комплекса рамками любой из ее частей, а в рамках ее частей должно быть предусмотрено разделение полномочий пользователя на просмотр и изменение данных;

- возможности ведения собственного аудита и настраиваемой регистрации собственных событий, а также событий производимых любыми пользователями программного комплекса;

- технические (программно-технические) средства защиты информации как входящие в состав операционных систем, так и поставляемые дополнительно;

- оборудование локальных вычислительных сетей;

- оборудование высокоскоростных телекоммуникационных сетей.

Для предотвращения некорректной работы программного комплекса необходимо реализовать:

- авторизацию доступа к данным;

- семантический и синтаксический контроль исходных данных;

- вывод сообщений об ошибках;

- возможность повторного ввода данных.

Для знаниевых концептуальных карт выделяют следующие функциональные черты:

- **доступность** – возможность определить и получить доступ к образовательным компонентам из одной удаленной точки и доставить их во многие другие.

- **приспособляемость** – возможность приспособлять обучение под индивидуальные и корпоративные нужды.

- **экономичность** – возможность увеличить эффективность и продуктивность за счет снижения цены и времени, требуемых для обеспечения удаленного обучением.

- **долговечность** – возможность выстоять при технологическом развитии и изменении без стоимостных затрат на перепроектирование, реконфигурацию и изменения исходного кода.

- **совместимость** – возможность взять образовательный компонент, разработанный в одной системе с одним набором средств и платформ, и использовать его в другой системе с другим набором средств и платформ.

- **возможность повторного использования** – гибкость в возможности многократного присоединения образовательного компонента в различных приложениях и контекстах.

Результаты проекта:

- Онтология предметной области технологий микро- и наносистем.

- База знаний по технологиям микро- и наносистем.

- Подсистема генерирующего синтеза БЗ, включающей не менее 5 примеров решения задач прогнозирования появления новых решений в области технологий микро- и наносистем.

- Визуализация базы знаний по технологиям микро- и наносистем на основе контекстной СМАР-модели базисных типов сущностей предметной области технологий микро- и наносистем.

- проекты нормативных документов и технических регламентов организации взаимодействия базы знаний с научными, образовательными, производственными и коммерческо-внедренческими структурами в области технологий микро- и наносистем.

- техническое задание на разработку программного комплекса экспертной системы по технологиям микро- и наносистем.

- программный комплекс экспертной системы по технологиям микро- и наносистем, разработанный в соответствии с утвержденным техническим заданием.

- библиотека семантических карт понятий, представляющих формализацию терминологических записей, в том числе содержащимися в материалах Учебно-методических комплексов, поставленных в рамках Государственного контракта №П761 от 12 ноября 2008 года «Закупка учебно-методического обеспечения для подготовки кадров по программам высшего профессионального образования для тематического направления ННС «Наноинженерия», и увеличение общего количества записей не менее чем на 10 процентов от ранее созданного прототипа.

- методические материалы по использованию базы знаний по технологиям микро- и наносистем для решения задач прогнозирования появления новых технологий.

- система консультирования пользователей базы знаний для решения научных, инженерных, производственных и образовательных задач по технологиям микро- и наносистем.

Понятийный словарь предметной области реализован на основе анализа литературы [4-74].

Реализация результатов: при реализации проекта были обеспечены следующие требования к техническому обеспечению:

- программное обеспечение, используемое во всех компонентах системы, совместимо с ос linux.

- программный комплекс работает на компьютерном оборудовании с intel-совместимой архитектурой и использует инфраструктуру реализованную в рамках государственных контрактов от 18 августа 2008 года №01.647.11.2002 «создание распределенной системы сбора, хранения, обработки и управления потоками научных и технологических данных для nanoиндустрии».

- пропускная способность канала доступа из научно-образовательной сети к серверу разработки составляет 1гб.

- программная реализация осуществлена с использованием современных языков и технологий программирования.

- технические средства для функционирования базы знаний размещены в одном из data-центров, реализованных в рамках государственного контракта от 18 августа 2008 года №01.647.11.2002 «создание распределенной системы сбора, хранения, обработки и управления потоками научных и технологических данных для nanoиндустрии».

- технические средства, которые позволяют обеспечить желаемый уровень производительности базы знаний и время реакции системы на запрос пользователя при одновременном количестве запросов (в пределах 3 сек. при 10 одновременно обрабатываемых запросах):

- управляющий узел: HP DL360R05;

- обрабатывающий узел: 2 x HP DL360R05 (Dual Intel® Xeon® Processor E5430 (12M Cache, 2.66 GHz, 1333 MHz FSB)/16G, 75 G);

- система хранения данных: NetApp FAS2050-R5 в составе:FAS2050, NetApp Select,20x750GB SATA,-C,R5;

- система резервного копирования: Sun SL48 w/ 1HP LTO4 SCSI;

- коммуникационная инфраструктура: Gigabit Ethernet на основе оборудования Cisco.

- для апробации и отладки трудоемких программных модулей обеспечено полное зеркалирование создаваемого программного обеспечения и базы знаний на отдельный выделенный сервер разработки.

- сервер разработки использовался также в качестве горячего резерва основного сервера в целях обеспечения бесперебойной работы всей системы в целом.

- для обеспечения зеркалирования использованы следующие виды программно-технических средств и обеспечения: операционная система семейства Unix (CentOs).

Апробация работы: в рамках проекта проведена опытная **апробация** подсистемы генерирующего синтеза базы знаний, включающий пять примеров решения задач прогнозирования появления новых решений в области технологий конструкторско-технологического проектирования ЭС.

Структура и объем работы: работа состоит из двух разделов.

1 РАЗРАБОТКА И РЕАЛИЗАЦИЯ В СООТВЕТСТВИИ С УТВЕРЖДЕННЫМ ТЕХНИЧЕСКИМ ЗАДАНИЕМ ПРОГРАММНОГО КОМПЛЕКСА ЭКСПЕРТНОЙ СИСТЕМЫ ПО ТЕХНОЛОГИЯМ МИКРО- И НАНОСИСТЕМ

1.1 Структурно-функциональная модель ЭС

1.1.1 Контекстная диаграмма работы ЭС

На верхнем уровне декомпозиции (рисунок 1.1) модель процесса работы ЭС[5] представлена в виде контекстной диаграммы, которая отражает взаимодействие работы «Работа ЭС» с внешней средой.

На входе данной работы имеются знания, полученные от эксперта, и диалог с пользователем.

Работа с системой производится экспертами и пользователями.

Входные воздействия преобразуются системой в готовый технологический процесс сборки пропеллерных часов.



Рисунок 1.1 – Контекстная диаграмма «Работа ЭС»

Спецификация на работу «Работа ЭС» представлена в таблице 1.1.

Таблица 1.1 - Спецификация на работу «Работа ЭС»

Name	Работа ЭС
Definition	Преобразование знаний и диалога в решение с помощью экспертов и пользователей
Input Arrows of " Работа ЭС "	
Name	Definition
Знания	Знания о предметной области технологий микро- и наносистем, полученные от эксперта
Диалог	Диалог с пользователем
Output Arrows of " Работа ЭС "	
Name	Definition
Решение	Синтезированное системой решение
Mechanism Arrows of " Работа ЭС "	
Name	Definition
Эксперт	Эксперт, наполняющий БЗ
Пользователь	Пользователь, работающий с системой

Из приведенной диаграммы видно, что система преобразует знания и диалог в готовое решение, с помощью экспертов и пользователей.

1.1.2 Первый уровень декомпозиции «Работа ЭС»

На первом уровне декомпозиции (рисунок 1.2) модель представляет собой 4 взаимосвязанных работ, представленные в нотации IDEF0: «Работа со слотами», «Работа с фреймами», «Работа с экземплярами» и «Синтез решения»[6].

Работа с системой производится экспертами и пользователями. Входные воздействия преобразуются системой в решение.

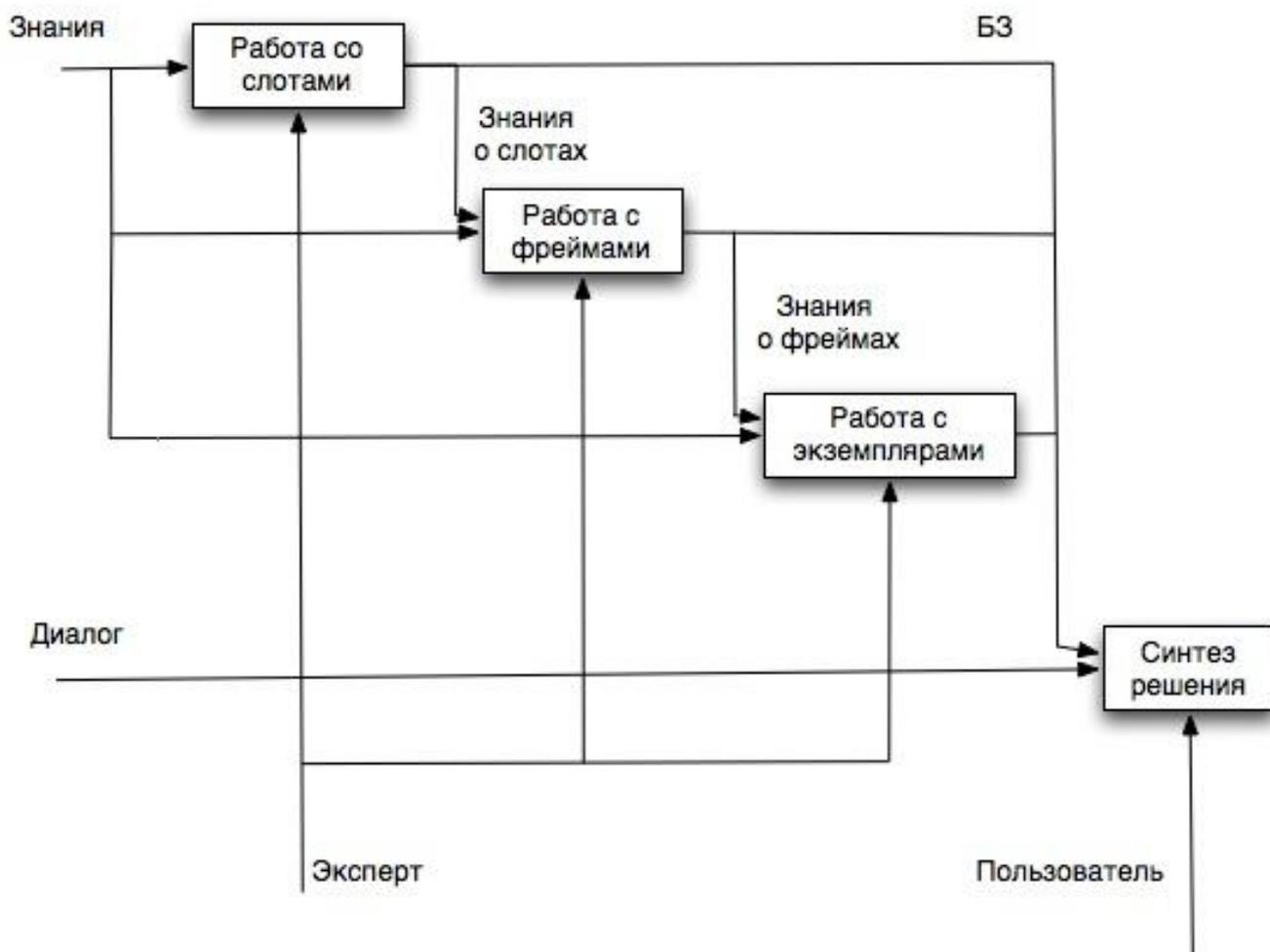


Рисунок 1.2 – Первый уровень декомпозиции «Работа ЭС»

Спецификации на работы приведены в таблицах 1.2 - 1.5.

Таблица 1.2 - Спецификация на работу «Работа со слотами»

Name	Работа со слотами
Definition	На данном этапе эксперт создает, редактирует и удаляет информацию о слотах
Input Arrow of "Работа со слотами"	
Name	Definition
Знания	Знания, полученные от эксперта
Output Arrows of "Работа со слотами"	
Name	Definition
БЗ	База знаний, заполняемая экспертом
Знания о слотах	Знания, введенные экспертом о слотах

Таблица 1.3 - Спецификация на работу «Работа с фреймами»

Name	Работа с фреймами-прототипами
Definition	На данном этапе эксперт создает, редактирует и удаляет информацию о фреймах-прототипах
Input Arrows of "Работа с фреймами"	
Name	Definition
Знания	Знания, полученные от эксперта
Знания о слотах	Знания, введенные экспертом о слотах
Output Arrows of "Работа с фреймами"	
Name	Definition
БЗ	База знаний, заполняемая экспертом
Знания о фреймах	Знания, введенные экспертом о фреймах

Таблица 1.4 - Спецификация на работу «Работа с экземплярами»

Name	Работа с экземплярами
Definition	На данном этапе эксперт редактирует и удаляет значения слотов экземпляров
Input Arrows of "Работа с экземплярами"	
Name	Definition
Знания	Знания, полученные от эксперта
Знания о фреймах	Знания, введенные экспертом о фреймах
Output Arrows of "Работа с экземплярами"	
Name	Definition
БЗ	База знаний, заполняемая экспертом

Таблица 1.5 - Спецификация на работу «Синтез решения»

Name	Синтез решения
Definition	На этом этапе проводится диалог с пользователем, получаются ответы на вопросы и синтезируется решение
Input Arrows of "Синтез ТП"	
Name	Definition
БЗ	База знаний, заполняемая экспертом
Диалог	Диалог с пользователем
Output Arrows of "Работа ЭС "	
Name	Definition
Решение	Синтезированное системой решение

Работы “Работа со слотами”, “Работа с фреймами” и “Работа с экземплярами” служат для наполнения БЗ экспертом: знаний о фреймах и слотах.

Результат этих работ в виде БЗ знаний поступает на вход следующей работы “Синтез решения”, которая совместно с диалогом с пользователем и синтезирует решение.

1.1.3 Разработка иерархической диаграммы работы ЭС

Иерархическая диаграмма дерева узлов (рисунок 1.3) представляет собой полную декомпозиционную модель синтезируемого решения, анализ которой очень важен для понимания бизнес-процесса в целом.



Рисунок 1.3 – Диаграмма дерева узлов

На дереве узлов не представлена последовательность работы системы, однако она включает в себя все реализуемые системой функции. На дереве узлов можно увидеть конечный результат работы.

1.1.4 Спецификация ролей пользователей системы

Система рассчитана на следующие группы пользователей:

Администраторы:

- системный администратор;
- администратор баз данных;
- прикладной администратор.

Инженеры по знаниям:

- системный инженер по знаниям (модератор);
- эксперт предметной области.

Пользователи:

- «гость»;
- авторизованный пользователь.

Все специалисты могут совмещать обслуживание портала с обслуживанием других информационных систем заказчика. Также роли обслуживающего персонала могут быть объединены.

Система должна быть рассчитана на обслуживание специалистами, прошедшими подготовку на авторизованных учебных курсах в соответствии с требованиями настоящего документа.

Применяемые средства не должны требовать от администратора портала применение навыков программирования или управления с использованием командного интерпретатора (командная строка).

Таблица 1.6 - Требования к квалификации обслуживающего персонала группы «Администраторы»

Роль	Функции	Численность	Квалификация
1	2	3	4
Системный администратор	<p>Настройка, диагностирование, оперативные контроль и оптимизация загрузки аппаратных средств Комплекса; Контроль целостности компонент Комплекса. Резервное копирование компонент Комплекса. Восстановление компонент Комплекса. Оперативный контроль и администрирование Комплекса, восстановление функционирования Комплекса при возникновении нештатных ситуаций.</p> <p>Установка необходимых обновлений ОС, базового и антивирусного ПО. Реализация политики информационной безопасности Комплекса. Классификация пользователей Комплекса в соответствии с выработанное политикой информационной безопасности.</p> <p>Формирование индивидуальных профилей пользователей. Комплекса. Периодические контроль соответствия прав доступа пользователей к информационным ресурсам и сервисам Комплекса, установленному регламенту.</p> <p>Анализ журналов регистрации штатных компонент Комплекса, а также средств защиты, регистрация попыток НСД и оперативное реагирование на такие попытки. Расследование случаев НСД к Комплексу. Сопровождение проблем, возникших у пользователей Комплекса, связанных с функционированием аппаратных и базовых программных средств (за исключением СУБД), сетевой инфраструктуры и средств защиты информации.</p>	от 1 до 2	Квалификация администратора: Необходимы навыки с оборудованием и программным обеспечением, используемыми в Комплексе.
Администратор баз данных	<p>Настройка, диагностирование, оперативный контроль и оптимизация загрузки СУБД Комплекса. Контроль целостности системных баз данных Комплекса.</p> <p>Резервное копирование, архивация наборов системных баз данных Комплекса. Восстановление данных в системных базах данных Комплекса.</p> <p>Сопровождение проблем, возникших у пользователей Комплекса, связанных с функционированием СУБД.</p>	от 1 до 2	Опытный пользователь ПК. Квалификация администратора: Необходимы отличное знание и опыт работы с СУБД, используемыми в Комплексе.

Продолжение таблицы 1.6

1	2	3	4
Прикладной администратор	Организация работы обслуживающего персонала. Задание полномочий пользователей по доступу к информационным ресурсам и сервисам Комплекса на уровне прикладного ПО. Сопровождение проблем, возникших у пользователей Комплекса, и связанных с функционированием разработанного прикладного ПО Комплекса. Обобщение потребностей пользователей в части информационных ресурсов и общесистемных функций Комплекса и разработка предложений по их совершенствованию.	Производственная	Опытные пользователь ПК. Стандартные навыки работы с Web-интерфейсами, офисными программными продуктами.

Роли необходимые для функционирования системы «работа ЭС» представлены в таблице 1.7.

Таблица 1.7 – Спецификация ролей пользователей системы

Роль группы	Описание
Администратор (admin)	Описать
Инженер по знаниям (Expert)	Сотрудник, наполняющий БЗ
Пользователь (User)	Пользователь, использующий систему для генерацию технологического процесса

Данные роли предполагается использовать для создания Автоматической системы управления конструкторско-технологического проектирования.

1.2 Модели вариантов использования и деятельности экспертной системы

1.2.1 Диаграмма вариантов использования ЭС

Диаграммы вариантов использования применяются для моделирования представления системы с точки зрения вариантов использования. Большей частью это подразумевает моделирование контекста системы, подсистемы или класса либо моделирование требования к этим элементам.

На рисунке 1.4 представлена диаграмма вариантов использования разрабатываемой ЭС, позволяющая представить ожидаемое поведение системы. На данной диаграмме представлены действующие лица, они же актеры (эксперт, администратор, пользователь), а также все возможные действия, регистрируемые в экспертной системе.

В таблице 1.8 дана спецификация на актеров диаграммы вариантов использования ЭС. В таблице 1.9 дана спецификация на возможные варианты использования подсистемы.

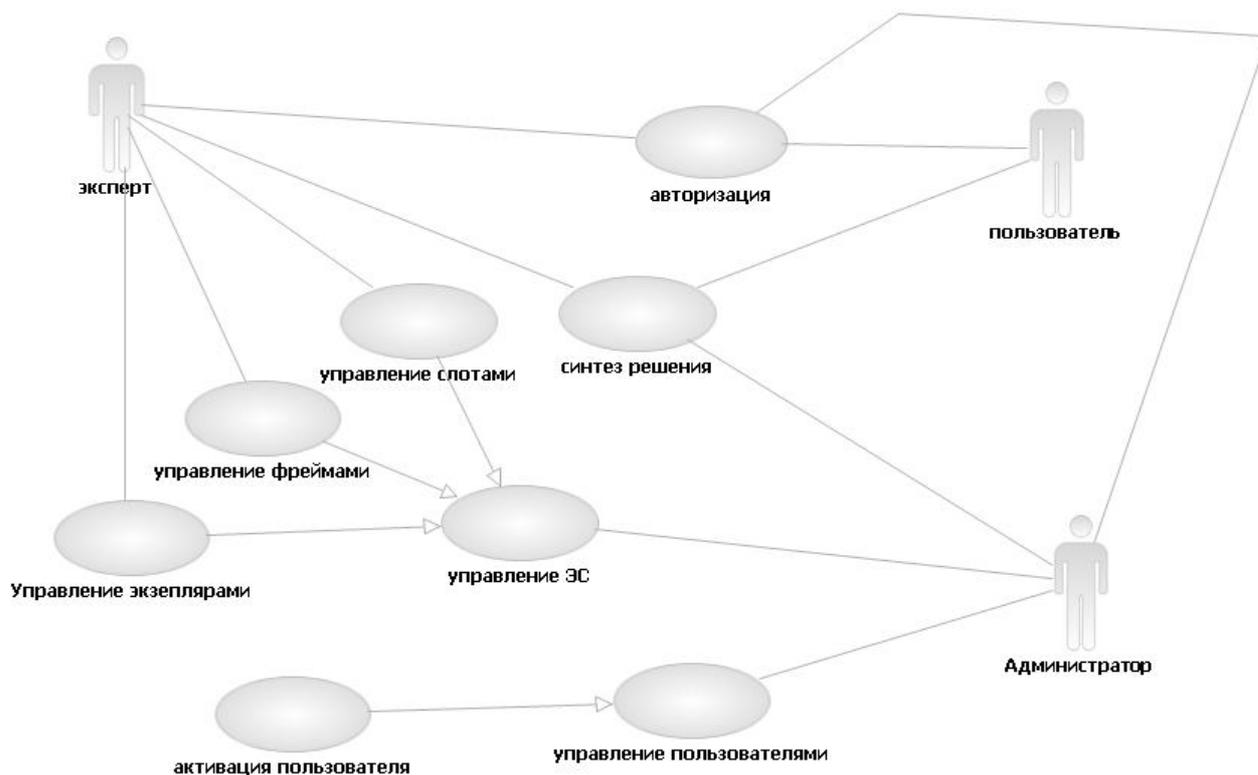


Рисунок 1.4 – Диаграмма вариантов использования ЭС

Таблица 1.8 – Спецификация на актеров диаграммы вариантов использования

Актер	Описание
Эксперт	Эксперт в области технологий микро- и наносистем. Заполняет, проверяет, редактирует базу знаний экспертной системы
Пользователь	Пользователь системы. Производит поиск.
Администратор	Администратор экспертной системы. Управляет пользователями системы, имеет неограниченные права на изменение как структуры база знаний, так и таблиц самой БД.

Таблица 1.9 – Спецификация прецедентов диаграммы вариантов использования

Прецедент	Описание
Авторизация	Представления системе для получения необходимых прав
Управление пользователями	Управление правами пользователей, удаление пользователей
Активация пользователя	Предоставление необходимых прав пользованием системы новым пользователям, зарегистрированных пользователей.
Управление ЭС	Управление экспертной системой. Заполнение базы знаний.
Управление фреймами	Добавление, удаление, редактирование фреймов.
Управления экземплярами	Удаление, редактирование значений слотов для экземпляра.
Синтез решения	Поиск решений, трассировка диалога с пользователем

Диаграмма вариантов использования представленная на рисунке 1.4 содержит субъект – экспертную систему для работы с базой знаний предметной области технологий микро- и наносистем, действующих лиц – эксперта, пользователя и администратора.

1.2.2 Диаграмма последовательности действий прецедента управления фреймами

Главным объектом диаграммы (рисунок 1.5) является пользователь типа «эксперт», который взаимодействует с системой через диалог, осуществляемый «формой для работы с ЭС». Связь с базой данных осуществляется объектом prototype_entry, который генерирует sql-запросы и выдает результаты форме для работы с ЭС.

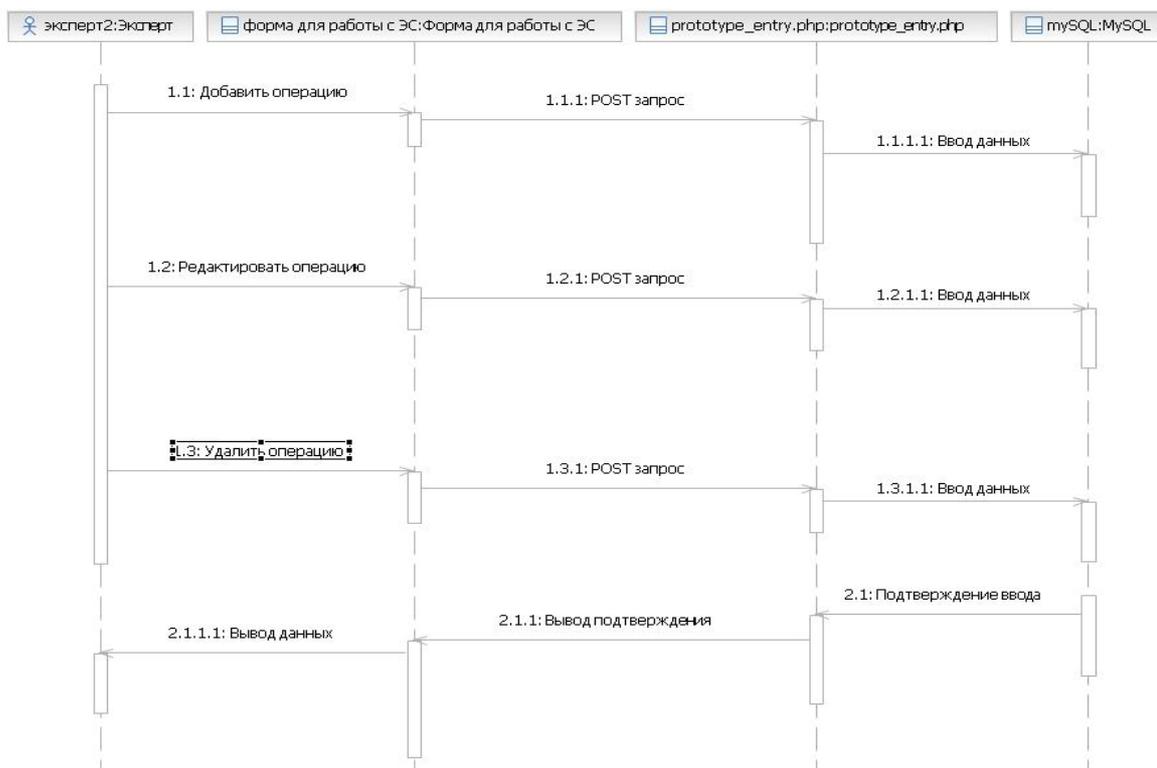


Рисунок 1.5 – Диаграмма последовательности действий прецедента управления фреймами

Спецификация объектов этой диаграммы приведена в таблице 1.10.

Таблица 1.10 – Спецификация объектов диаграммы последовательности

Объекты/методы	Описание
Добавить операцию	Занесение новых данных в БД
Редактировать операцию	Редактировать существующие данные в БД
Удалить операцию	Удаление существующие данные из БД
POST запрос	Структурированная передача данных для обрабатывающему скрипту
Ввод данных	Формирование SQL запроса
Подтверждение вводу	Ответ от СУБД об успешном вводе данных
Вывод подтверждения	Генерация HTML для вывода данных
Вывод данных	Вывод данных в формате HTML

Данная диаграмма последовательности, главным образом, позволяет понять на каких временных данных объекты взаимодействуют друг с другом. Выявлено, что все цепочки последовательностей вызовов методов у объектов порождает объект типа «эксперт».

1.2.3 Диаграмма кооперации для прецедента управления фреймами

На этой диаграмме нумеруются сообщения для обозначения их временной последовательности. На рисунке 1.6 представлена диаграмма кооперации для создания, удаления, редактирования слотов, прототипов и экземпляров.

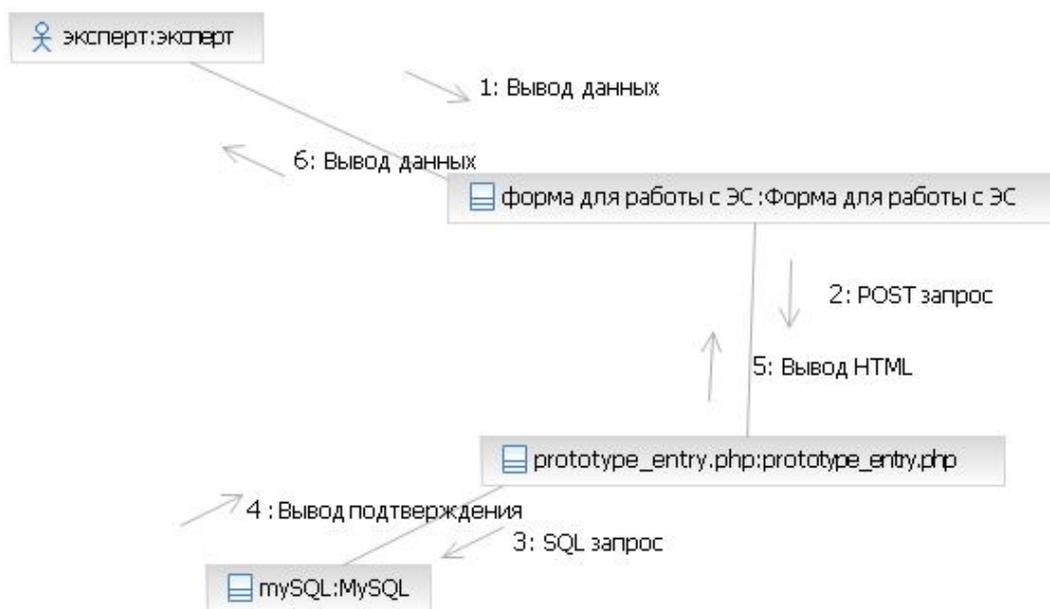


Рисунок 1.6 – Диаграмма кооперации для прецедента управления фреймами

Таким образом, диаграмма взаимодействия можно строить двумя способами – уделяя основное внимание временной упорядоченности сообщений (диаграмма последовательности) или структурным отношениям между взаимодействующими объектами (диаграмма кооперации).

1.2.4 Диаграмма последовательности действий прецедента синтеза решения

Для выявления цепочки последовательности взаимодействия объектов для прецедента синтеза решения также составлена диаграмма последовательности (рисунок 1.7). Здесь, как и для прецедента управления фреймами, порождающим объектом является объект «пользователь».

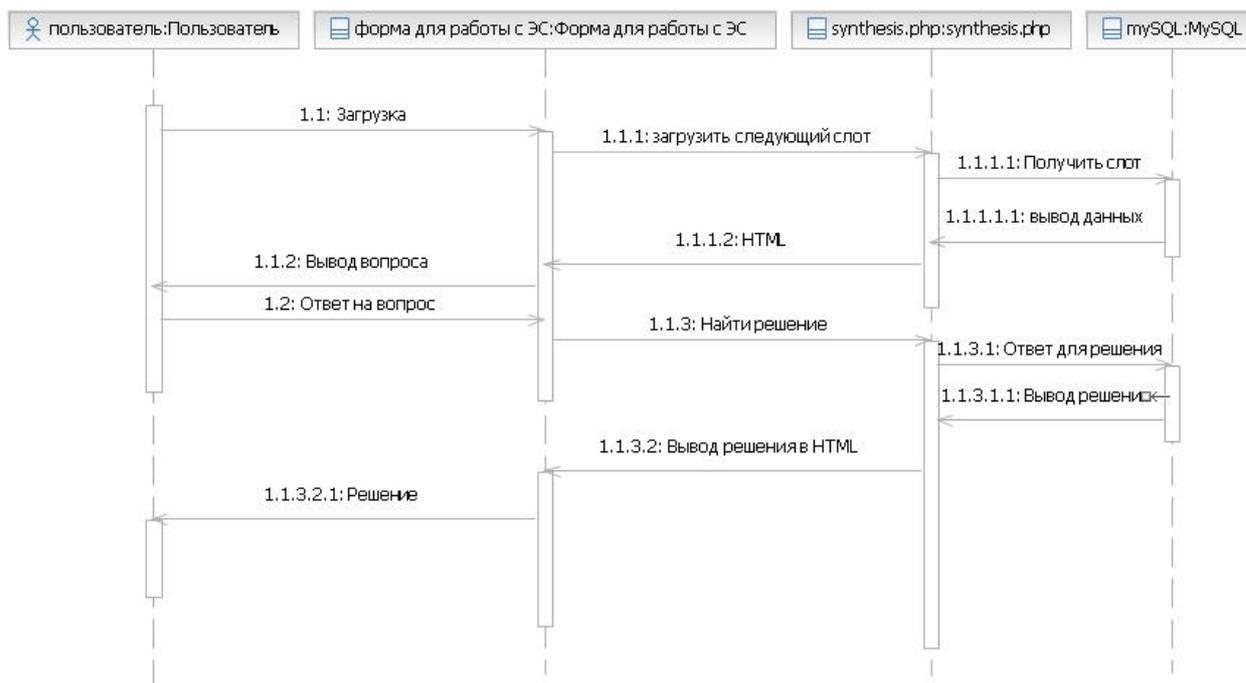


Рисунок 1.7 – Диаграмма последовательности действий прецедента синтеза решения.

Таблица 1.11 – Спецификация объектов диаграммы последовательности

Объекты/методы	Описание
Пользователь	Любой пользователь системы имеет право провести синтез решения
Загрузка	Загрузка исходного состояния для проведения поиска решения
Загрузить следующий слот	Загрузка слота для вывода вопроса пользователю
Получить слот	Составление SQL запроса для получения слота из БД.
Вывод данных	Вывод данных из БД
HTML	Генерация HTML для диалога с пользователем
Вывод вопроса	Форма для ввода ответа
Ответ на вопрос	Заполненная форма ответа
Найти решение	Анализ базы знаний на основе трассировки синтеза
Ответ для решения	SQL запрос ответа
Вывод решения	Получения решения из БД
Вывод решения в HTML	Генерация HTML для отображения ответа пользователю

Диаграмма последовательности действий содержит процесс поиска решения с помощью экспертной системы конечным пользователем. В результате работы ЭС генерируется решение из предметной области технологий микро- и наносистем.

1.2.5 Диаграмма состояний ЭС для синтеза решения

Система последовательно выводит в диалог с пользователем ЭС вопросы, на которые ему необходимо ответить.

Анализируя ответы система подбирает решение. Если решение не найдено, пользователю выводится соответствующее сообщение.

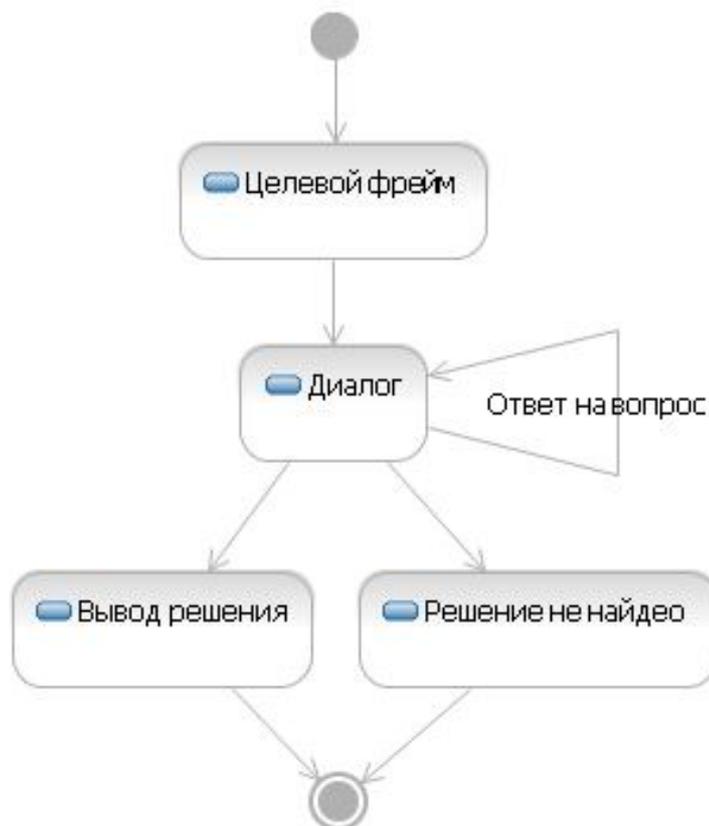


Рисунок 1.8 – Диаграмма состояний ЭС для синтеза решений

Таблица 1.12 – Спецификация для диаграммы состояний

Состояние	Описание
Целевой фрейм	Выбор корневого фрейма
Диалог	Диалог экспертной системы с пользователем, в процессе которого пользователь отвечает на вопросы ЭС
Вывод решения	ЭС выводит решения. В случае отсутствия решения выводит сообщение об отсутствии решения.
Решение не найдено	ЭС выводит сообщение об отсутствии решения

Диаграмма деятельности, представленная на рисунке 1.8, показывает процесс передачи потоков управления при синтезе решения в ЭС.

1.2.6 Диаграмма деятельности для авторизации

Авторизация пользователя проводится по паре логин/пароль. После авторизации пользователя уникальный номер пользователя в базе сохраняется в сессию браузера для запоминания пользователя. Диаграмма изображена на рисунок 1.9, а её спецификация в таблице 1.13.

Таблица 1.13 Спецификация для диаграммы деятельности

Состояние	Описание
Открытие сессии	Вызов функции session_start()
Получение ID пользователя	Вывод ID пользователя, сохраненный в сессии.
Ввод логина пароля	Заполнение форм для авторизации
Обращение к БД	Выполнение SQL запроса
Проверка пароля	Проверка совпадения пары логин/пароль
Вывод сообщения об ошибке	Вывод сообщения пользователю об ошибке авторизации
Сохранение сессии	Запоминание ID пользователя в сессии
Авторизация пользователя	Получение данных о пользователе: имя, роль, email

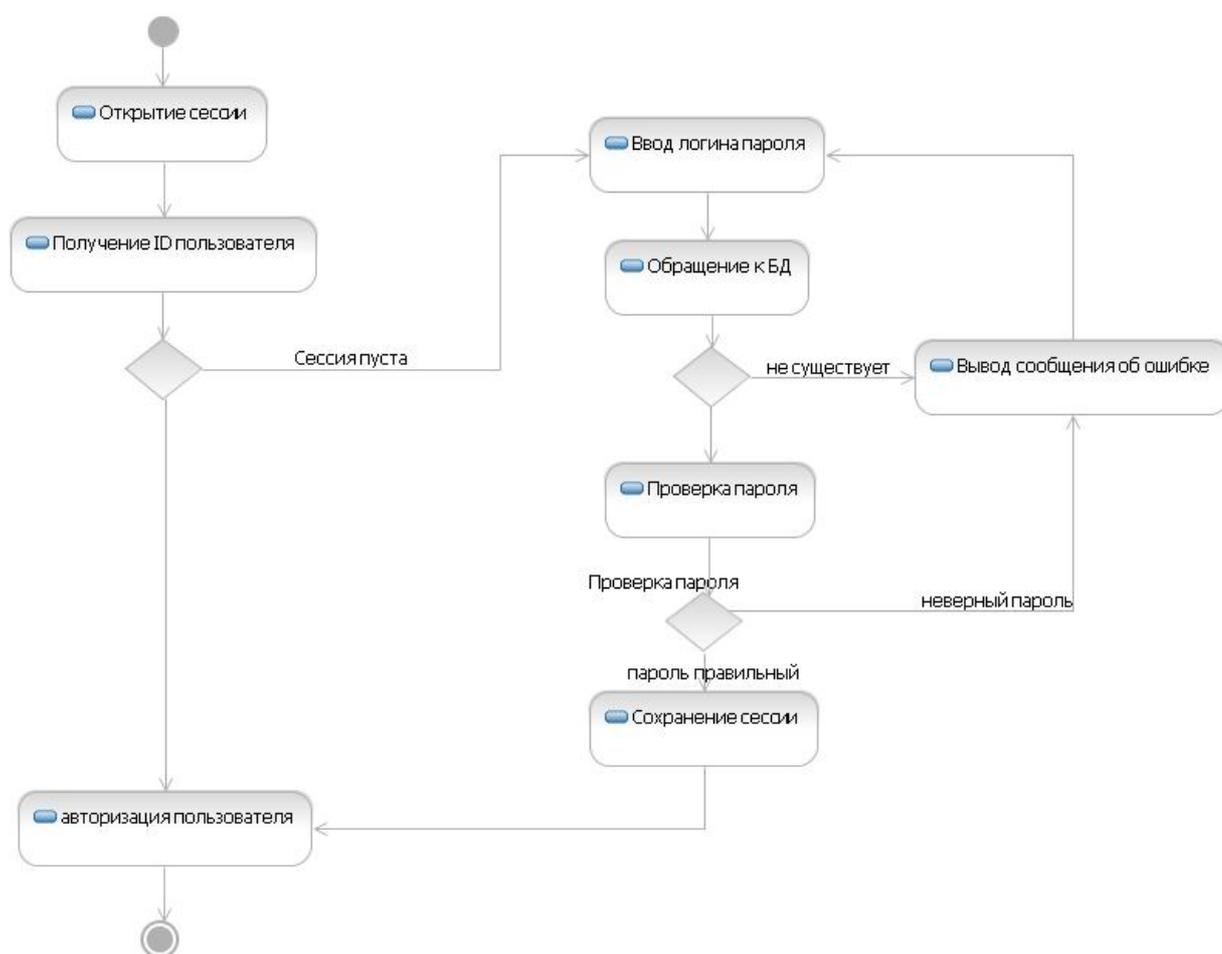


Рисунок 1.9 – Диаграмма деятельности для авторизации

Диаграмма деятельности, представленная на рисунке 1.9, показывает процесс передачи потоков управления при авторизации в ЭС.

1.2.7 Диаграмма деятельности для авторизованных пользователей ЭС, имеющие уровень доступа «администратор»

На рисунок 1.10 приведена диаграмма состояний для экспертной системы в целом имея право доступа «Администратор». Данный уровень доступа дает пользователю безграничный уровень доступа к системе.

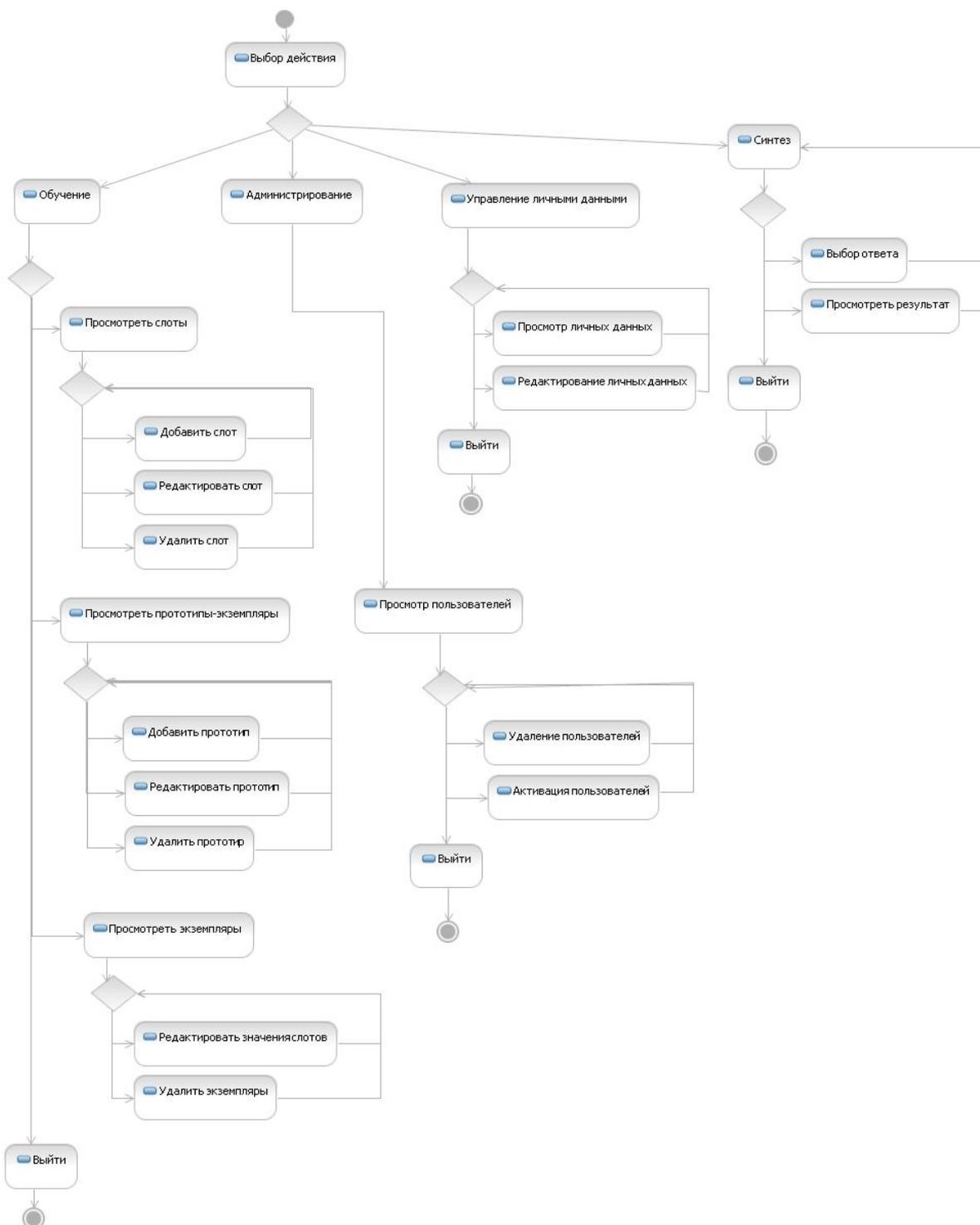


Рисунок 1.10 – Диаграмма деятельности для авторизованных пользователей ЭС, имеющие уровень доступа «администратор»

Таблица 1.14 – Спецификация для Диаграмма деятельности для авторизованных пользователей ЭС, имеющие уровень доступа «администратор»

Состояние	Описание
Выбор действия	Выбор действия для работы с ЭС
Обучение	Процесс работы с базой знаний
Просмотреть слоты	Список слотов в базе знаний
Добавить слот	Добавить новый слот в базу знаний
Редактировать слот	Редактирование существующего слота в базе знаний
Удалить слот	Удалить слот из базы знаний
Просмотреть прототипы-экземпляры	Вывод древовидной структуры иерархии фреймов
Добавить прототип	Добавить новый прототип
Редактировать прототип	Редактировать имя, родителя, слоты прототипы. Включения состояния экземпляра
Удалить прототип	Удалить прототип из базы знаний
Просмотреть экземпляры	Вывод всех экземпляров в базе знаний
Редактировать значения слотов	Ввод значение слотов экземпляра
Удалить экземпляры	Удалить экземпляры из базы знаний
Администрирование	Управление системой
Просмотр пользователей	Вывод пользователей, зарегистрированных в системе
Удалить пользователей	Удалить пользователей из БД
Активация пользователей	Активация пользователей находящиеся в состоянии ожидания после регистрации
Управление личными данными	Кабинет пользователя
Просмотр личных данных	Получения данных об учетной записи пользователя, просмотр статуса активации, уровня доступа
Редактирование личных данных	Изменение личных данных и пароля для доступа к системе
Синтез	Диалог нахождения решения
Выбор ответа	Выбор ответа из предоставляемого списка на ответ
Просмотреть результат	Вывод результата и трассировки решения

Диаграмма деятельности, представленная на рисунке 1.8, показывает процесс передачи потоков управления при работе с ЭС авторизованными пользователями, у которых установлен уровень доступа - администратор.

1.2.8 Диаграмма деятельности для авторизованных пользователей ЭС, имеющие уровень доступа «эксперт»

На рисунке 1.11 приведена диаграмма состояний для экспертной системы в целом имея право доступа «Эксперт». Данный уровень пользователя предоставляет пользователя добавлять, изменять, удалять данные в базе знаний.

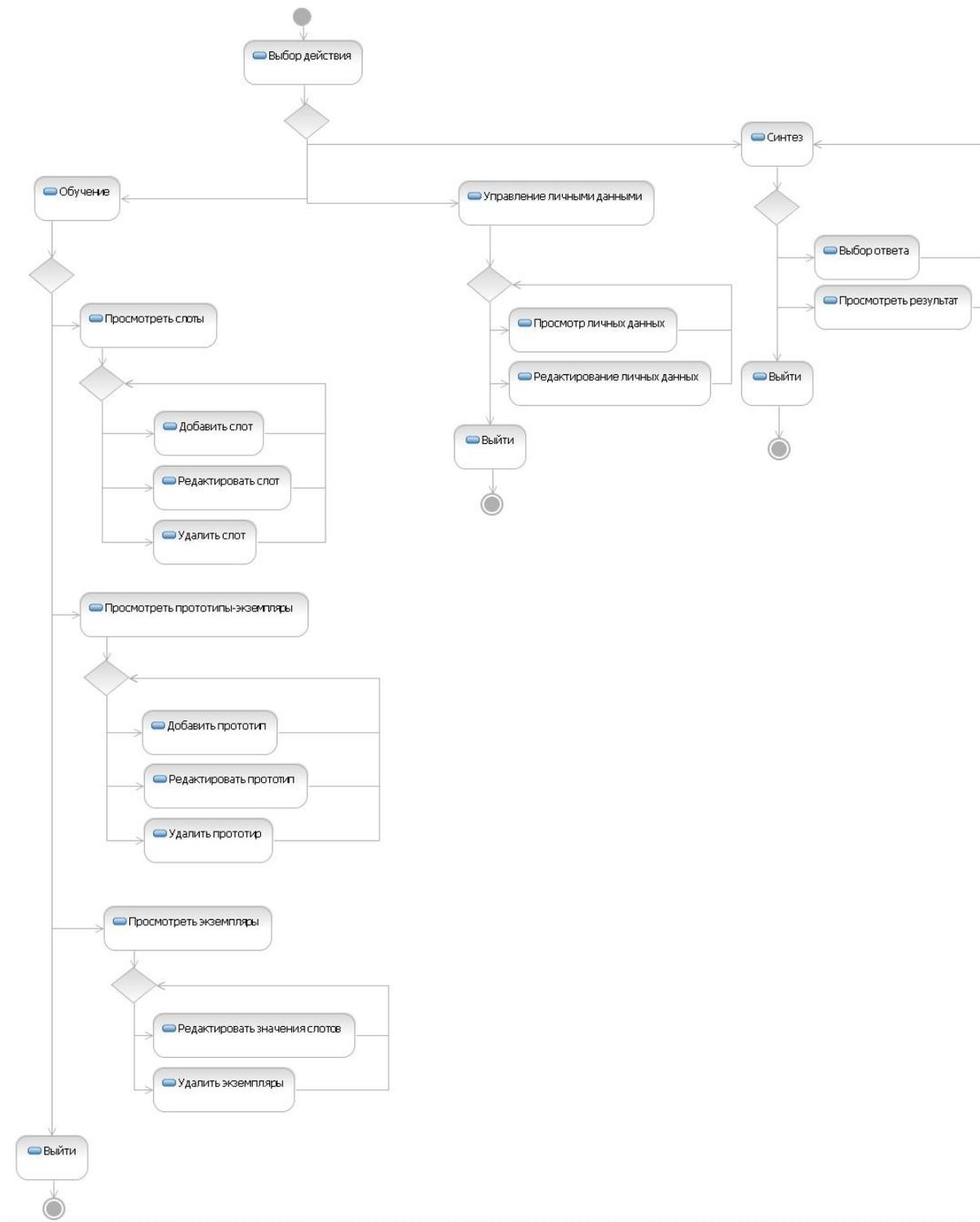


Рисунок 1.11 – Диаграмма деятельности для авторизованных пользователей ЭС, имеющие уровень доступа «эксперт»

Таблица 1.15 – Спецификация для диаграммы деятельности для авторизованных пользователей ЭС, имеющие уровень доступа «эксперт»

Состояние	Описание
1	2
Выбор действия	Выбор действия для работы с ЭС
Обучение	Процесс работы с базой знаний
1	2
Просмотреть слоты	Список слотов в базе знаний
Добавить слот	Добавить новый слот в базу знаний
Редактировать слот	Редактирование существующего слота в базе знаний
Удалить слот	Удалить слот из базы знаний
Просмотреть прототипы-экземпляры	Вывод древовидной структуры иерархии фреймов
Добавить прототип	Добавить новый прототип
Редактировать прототип	Редактировать имя, родителя, слоты прототипы. Включения состояния экземпляра
Удалить прототип	Удалить прототип из базы знаний
Просмотреть экземпляры	Вывод всех экземпляров в базе знаний
Редактировать значения слотов	Ввод значение слотов экземпляра
Удалить экземпляры	Удалить экземпляры из базы знаний
Управление личными данными	Кабинет пользователя
Просмотр личных данных	Получения данных об учетной записи пользователя, просмотр статуса активации, уровня доступа
Редактирование личных данных	Изменение личных данных и пароля для доступа к системе
Синтез	Диалог нахождения решения
Выбор ответа	Выбор ответа из предоставляемого списка на ответ
Просмотреть результат	Вывод результата и трассировки решения

Диаграмма деятельности, представленная на рисунке 1.8, показывает процесс передачи потоков управления при работе с ЭС авторизованными пользователями, у которых установлен уровень доступа - эксперт.

1.3 Логическая модель ЭС

1.3.1 Диаграмма пакетов ЭС

Данная экспертная система состоит из множества различных компонентов. Среди этих компонентов были выявлены те, которые имеют тенденцию изменяться совместно. На основе этого все компоненты были объединены в соответствующие им пакеты: «модуль ui», «модуль core», «модуль expert», «модуль templates», «apache», «MySQL». Взаимодействие между этими пакетами представлено на рисунок 1.12.

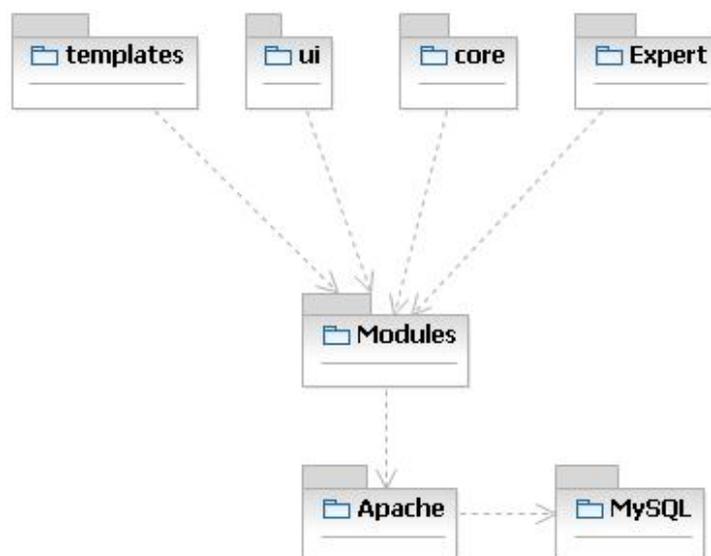


Рисунок 1.12 – Диаграмма пакетов ЭС

В таблице 1.16 представлена спецификация на пакеты, представленные на диаграмме пакетов.

Таблица 1.16 – Спецификация на пакеты, представленные на диаграмме пакетов.

Объекты/методы	Описание
Expert	Модуль для работы с ЭС
Core	Модуль для работы БД, конфигурациями системы и работы с пользователями
Ui	Модуль для генерации визуальных элементов
Templates	Пакет html-шаблонов для генерации html-страниц с соответствующим выводом данных
Apache	Web-сервер, принимающий запросы со стороны клиента, обрабатывающий их и передающий СУБД
MySQL	СУБД + физические файлы базы данных

Пакеты Apache и MySQL – представляют из себя готовые решения и уже установлены перед развертыванием самой системы. Пакеты Expert, Core – является ключевыми пакетами, в которых содержится вся логика. Пакеты Templates и UI – является, второстепенными и служат для создания интерфейсов-диалогов с пользователем.

1.3.2 Диаграмма классов для БД

Диаграмма классов для БД представлена на рисунке 1.13.

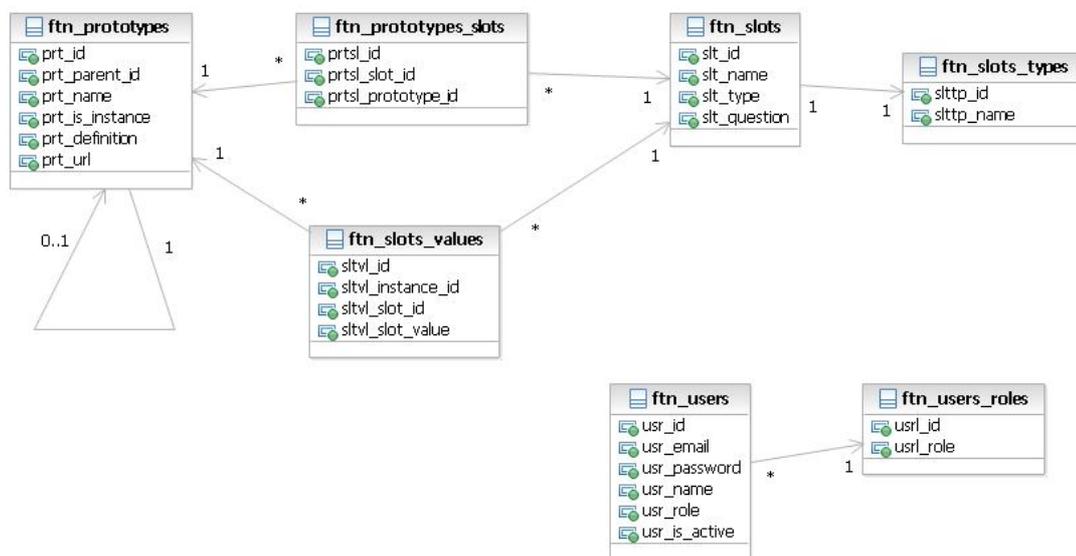


Рисунок 1.13 – Диаграмма классов для БД

Спецификация сущностей и атрибутов БД приведена в таблице 1.17.

Таблица 1.17 – Спецификация сущностей и атрибутов БД

Класс	Атрибут/метод	Описание
1	2	3
ftn_prototypes	prt_id	PRIMARY_KEY
	prt_parent_id	Уникальный номер родителя
	prt_name	Имя прототипа
	prt_is_instance	Прототип является экземпляром
	prt_definition	Определение
	prt_url	Ссылка на описание технологии
ftn_prototypes_slots	prtsl_id	PRIMARY_KEY
	prtsl_slot_id	Уникальный номер слота
	prtsl_prototype_id	Уникальный номер прототипа
ftn_slots	slt_id	PRIMARY_KEY
	slt_name	Имя слота
	slt_type	Тип данных
	slt_question	Вопрос, который выводится в диалог
ftn_slots_types	slttp_id	PRIMARY_KEY
	slttp_name	Имя типа значения слота
ftn_slots_values	sltvl_id	PRIMARY_KEY
	sltvl_instance_id	Уникальный номер прототипа-экземпляра
	sltvl_slot_id	Уникальный номер слота
	sltvl_slot_value	Значение слота
ftn_users	usr_id	PRIMARY_KEY
	usr_email	Е-mail пользователя
	usr_password	Пароль пользователя

Продолжение таблицы 1.17

1	2	3
	usr_name	Имя пользователя
	usr_role	Роль пользователя
	usr_is_active	Статус пользователя
ftn_users_roles	usr_l_id	PRIMARY KEY
	usr_l_name	Имя роли

Все атрибуты содержат префикс вида (prefix)* для соблюдения псевдонимичности. При этом атрибуты одного класса имеют одинаковый префикс.

1.3.3 Диаграмма классов системы ввода-вывода

Диаграмма классов для системы ввода-вывода которая реализует генерацию HTML-страниц представлена на рисунке 1.14.

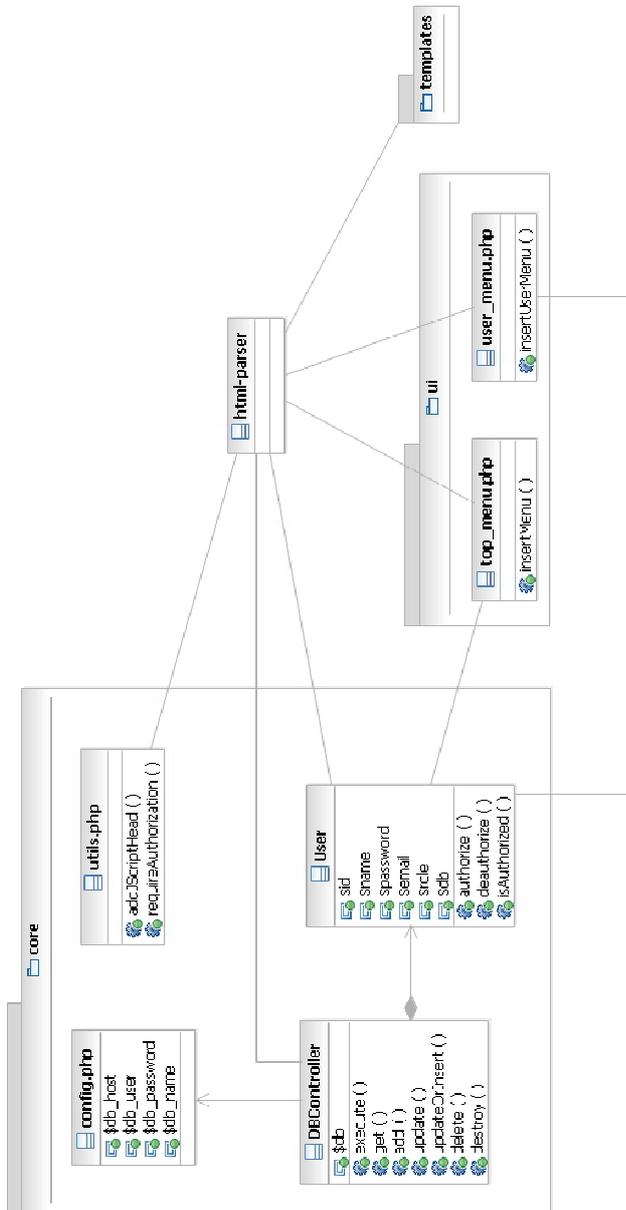


Рисунок 1.14 – Диаграмма классов системы ввода-вывода

Спецификация сущностей и атрибутов представлена в таблице 1.18.

Таблица 1.18 – Спецификация сущностей и атрибутов представлена

Класс	Атрибут/метод	Описание
config.php	db_host	Хост, на котором размещается СУБД
	db_user	Имя пользователя для подключения к СУБД
	db_password	Пароль для подключения к СУБД
	db_name	Имя базы данных
utils.php	addJScripthead()	Подключение внешних JS-скриптов
	requireAuthorization()	Проверка авторизации пользователя для страниц, на которых требуется авторизация
DBController	execute()	Выполнить SQL-запрос
	get()	SQL-запрос для получения данных
	add()	SQL-запрос добавление новых данных в БД
	update()	SQL-запрос для обновления данных в БД
	updateOrInsert()	SQL-запрос добавление, либо обновление, если запись уже существует в БД
	delete()	SQL-запрос удалить данных из БД
	destroy()	SQL-запрос закрыть соединение с БД
User	id	уникальный номер пользователя в системе
	name	имя пользователя в системе
	password	пароль пользователя в системе
	email	e-mail пользователя в системе, по которому производится авторизация
	role	роль пользователя в системе
	db	ссылка на объект DBController
	authorize()	авторизация пользователя
	deauthorize()	Деавторизация пользователя
isAuthorized()	Проверить авторизован ли пользователь, проверка сохраненной сессии	
top_menu.php	insertMenu()	Составление меню в соответствии с ролей пользователя
user_menu.php	insertUserMenu()	Ссылка на кабинет пользователя
html-parser	print()	Генерации HTML-страницы на основе шаблоны и полученных данных

Атрибуты и методы в таблице 1.18 различаются наличием скобок в конце: названием со скобками – это метод, без скобок – атрибут.

1.3.4 Разработка физической информационной модели

На основе логической модели была составлена физическая информационная модель, на этапе создания которой каждому атрибуту сущности был присоединен тип данного поля с возможностями выбранной СУБД MySQL. Результатом работы явилась физическая структура, представленная на рисунке 1.15. Параметры типа данных приведены в таблице 1.19.

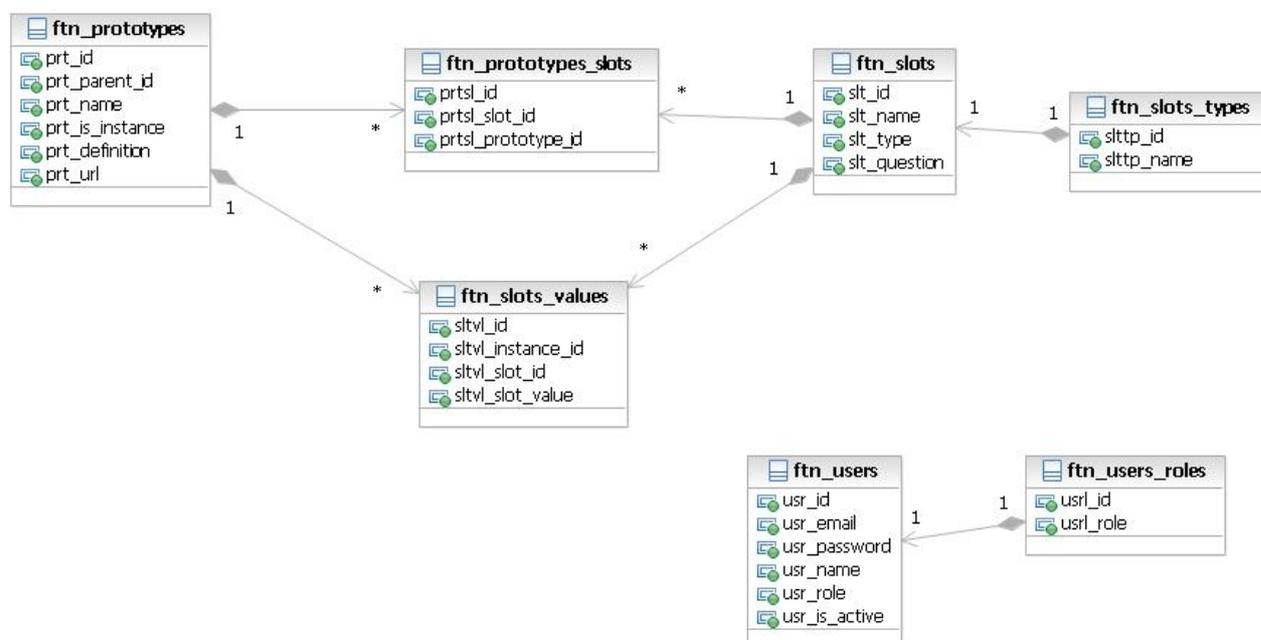


Рисунок 1.15 – Физическая структура данных в БД

Таблица 1.19 – Спецификация структуры данных в БД

Table	Field	Type	Allow Null	Key
1	2	3	4	5
ftn_prototypes	prt_id	int	N	PRI
	prt_parent_id	int	Y	-
	prt_name	char	Y	-
	prt_is_instance	enum	N	-
	prt_definition	text	Y	-
	prt_url	text	Y	-
ftn_prototypes_slots	prtsl_id	int	N	PRI
	prtsl_slot_id	int	Y	-
	prtsl_prototype_id	int	Y	-
ftn_slots	slt_id	int	N	PRI
	slt_name	char	Y	-
	slt_type	int	N	-
	slt_question	text	Y	-
ftn_slots_types	slttp_id	int	N	PRI
	slttp_name	char	Y	-
ftn_slots_values	sltv1_id	int	N	PRI
	sltv1_instance_id	int	N	-
	sltv1_slot_id	int	N	-
	sltv1_slot_value	char	Y	-

Продолжение таблицы 1.19

1	2	3	4	5
ftn_users	usr_id	int	N	PRI
	usr_email	char	N	-
	usr_password	char	N	-
	usr_name	char	Y	-
	usr_role	int	N	-
	usr_is_active	enum	N	-
ftn_users_roles	usr1_id	int	N	PRI
	usr1_role	char	N	-

Таблица 1.19 содержит следующие поля:

- Table
- Field
- Type
- Allow Null
- Key

Table – это имя таблицы в базе данных. *Field* – имя поля в таблицы. *Type* – типа данных хранимых в полях (field) таблицы. *Allow Null* – указывает на то, что значение данного поля может/не может быть пустым (Null). *Key* – свойство, указывающее на то, что данное поля является ключевым в таблице (Primary Key).

1.3.5 Диаграмма деятельности синтеза решения

Поиск решения (проведения синтеза решения) является ключевым элементов и самым сложным в экспертной системы.

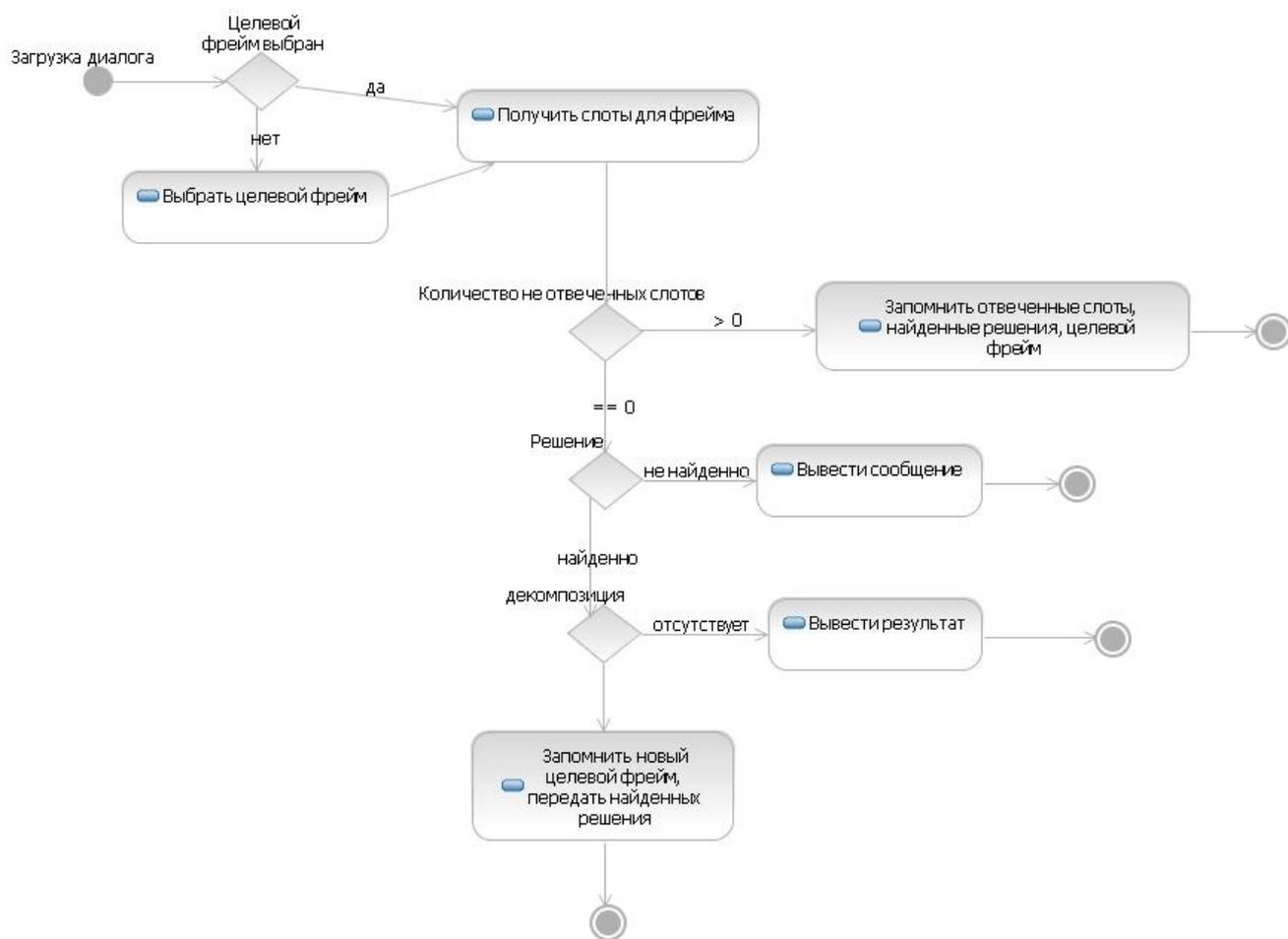


Рисунок 1.16 – Алгоритм поиска решения

Поиск состоит из следующих этапов:

- выбор целевого фрейма (прототипа)
- выборка слотов, принадлежащих целевому прототипу
- диалог с пользователем, в ходе которого последний отвечает на вопросы в соответствии с заданными слотами
- если необходимо провести декомпозицию, то для нового целевого прототипа (фрейма) провести такой же маршрут.

На основе данного маршрута был составлен алгоритм, представленный на диаграмме (рисунок 1.16). Основной сложностью в реализации данного алгоритма является специфика тонкого клиента на РНР, из-за чего диалог в каждой итерации загружается заново.

1.4 Модель реализации ЭС - диаграмма компонентов ЭС

В разделе 1.3.1 была представлена диаграмма пакетов (см. рисунок 1.12), она отображает структуру разрабатываемой экспертной системы, однако является не полной с точки зрения полноты описания всей системы. Для описания полной структуры экспертной системы составлена диаграмма компонентов.

На рисунок 1.17 представлена диаграмма компонентов. Спецификация компонентов системы представлена в таблице 1.20.

Таблица 1.20 – Спецификация компонентов системы

Компонент	Описание
top_menu.php	Генератор главного меню в соответствии с правами доступа пользователя
user_menu.php	Генерация приветственных сообщений
config.php	Настройки для подключения к СУБД
DBController.php	Класс для работы с запросами к СУБД
User.php	Класс для работы с данными пользователями и их авторизации
utils.php	Вспомогательные функции
cabinet.php	Редактирование личных данных
enter.php	Авторизация пользователя
exit.php	Деавторизация пользователя
registration.php	Регистрация пользователя в ЭС
user_entry.php	Административный компонент для работы с данными пользователей ЭС
users.php	Административный компонент для работы с пользователями с ЭС
instance_entry.php	Редактирование экземпляров
instances.php	Вывод экземпляров базы знаний
prototype_entry.php	Редактирование прототипов
prototypes.php	Вывод прототипов базы знаний
slot.php	Редактирование слотов базы знаний
slots.php	Вывод слотов базы знаний
synthesis.php	Компонент нахождения решений

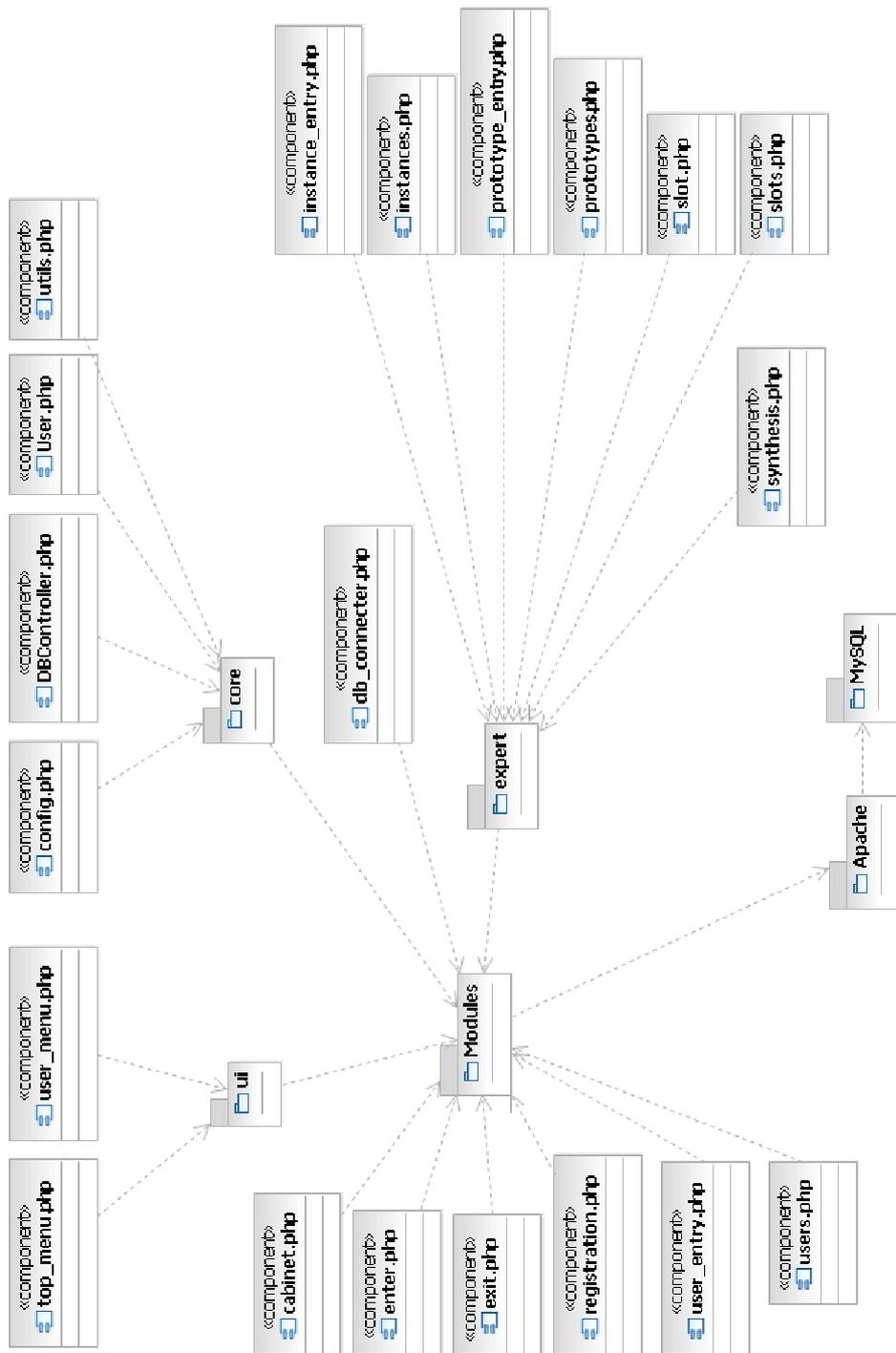


Рисунок 1.17 – Диаграмма компонентов

Таблица 1.20 содержит полную спецификацию по приведенной диаграмме компонентов. Она содержит имена компонентов (имя файлов с кодом на PHP) и их описание.

1.5 Модель развертывания ЭС

1.5.1 Вычислительный кластер «ННС»

1.5.1.1 Структура вычислительного кластера «ННС»

ЭС размещается на «Вычислительном кластере ННС» (рисунок 1.18). Кластер был развернут для решения задач проектирования микро- и наносистем, требующих высокопроизводительных вычислительных ресурсов. Для текущей работы доступна система конфигурации описанной в таблицах 1.21 и 1.22.

Таблица 1.21 – Архитектура вычислительного кластера.

Сервера HP	2 x HP DL360R05 (Dual Intel® Xeon® Processor E5430 (12M Cache, 2.66 GHz, 1333 MHz FSB)/16G, 75 G)
RISC сервера IBM	2 x IBM OpenPower P5 RiskServer 710
Система хранения данных	NetApp FAS2050-R5 в составе:FAS2050,NetApp Select,20x750GB SATA,-C,R5
Система резервного копирования	Sun SL48 w/ 1HP LTO4 SCSI
Коммуникационная инфраструктура	Gigabit Ethernet на основе оборудования Cisco

Таблица 1.22 – Общесистемное программное обеспечение вычислительного кластера

Операционная система	CentOS
Кластерное ПО	MOSIX
Интерфейс передачи сообщений между процессами	MPI2
Система мониторинга	Ganglia
Веб-сервер	Apache 2.2.3
СУБД	MySQL 5.0.77

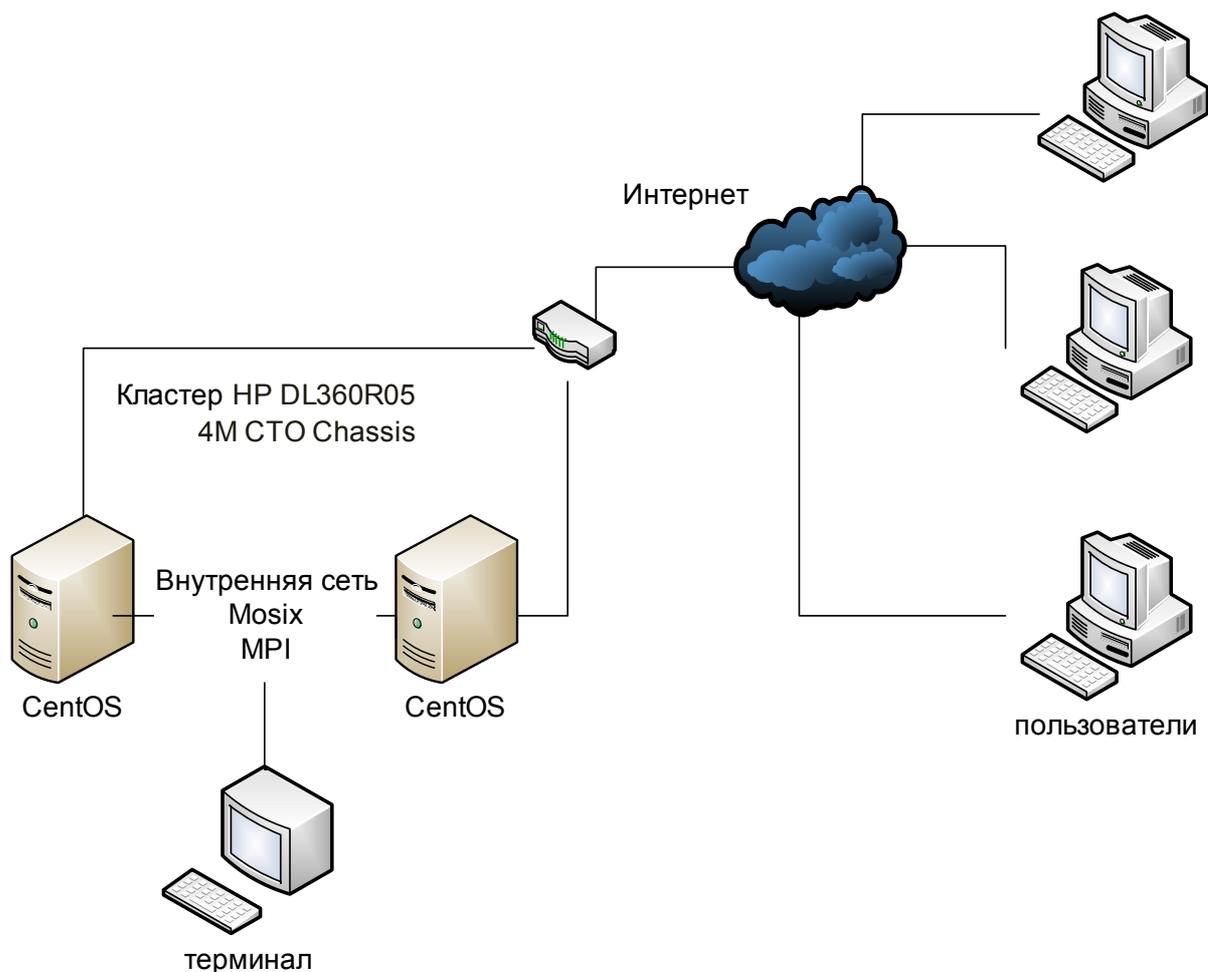


Рисунок 1.18 – Структура вычислительного кластера «ННС»

Вычислительный кластер «ННС» состоит из двух лезвий HP DL360R05, обмен сообщениями между которыми осуществляется по Gigabit Ethernet на основе оборудования Cisco средствами MPI2.

1.5.1.2 Пользовательский ресурсный хаб

Пользовательский ресурсный хаб host.iu4.bmstu.ru, предоставляет студентам возможность размещения своих информационных и расчетных проектов, доступ к хранилищам данных, программным библиотекам, сервисам (FTP, WWW, PHP, MySQL, Oracle и т.п.).

Для каждого пользователя создается домен <http://host.iu4.bmstu.ru/~fio>. Каждый студент, зарегистрированный в системе, имеет возможность хранить свои материалы на своем хостинге, учебные материалы по курсам «Основы конструкторско-технологической информатики», «Системное программирование», «САПР наносистем», «Конструкторско-технологические базы данных».

Сервер host.iu4.bmstu.ru является управляющим для систем хранения и кафедральным архивом.

Программное решение такой задачи - программный комплекс Webmin.

Webmin - это программный комплекс, позволяющий администрировать операционную систему через веб-интерфейс, в большинстве случаев, позволяя обойтись без использования командной строки и запоминания системных команд и их параметров. Используя любой браузер, администратор сервера может создавать новые учётные записи пользователей, почтовые ящики, изменять настройки служб и сервисов, например: веб-сервера Apache, DNS. Однако, в некоторых случаях необходимо знание операционной системы и редактирование конфигурационных файлов

вручную. Кроме того, не все возможности операционной системы и не все программы можно конфигурировать через интерфейс Webmin, например nginx пока не входит в базовый набор.

Webmin состоит из простого веб-сервера и большого количества скриптов (>500), которые собственно и осуществляют связь между командами администратора через веб-интерфейс и их исполнением на уровне операционной системы и прикладных программ. Webmin написан полностью на языке Perl и не использует никаких дополнительных нестандартных модулей. Простота, лёгкость и быстрота выполнения команд - одно из самых больших преимуществ данной панели управления.

Другое важное преимущество - возможность исправлять конфигурационные файлы вручную, так как Webmin не "портит" конфигурационные файлы, в отличие от некоторых других панелей управления, и следует, как правило, политикам дистрибутивов по конфигурированию программ.

Данная панель управления бесплатно распространяется для коммерческого и некоммерческого использования. Авторы этой программы позволяют всем желающим не только бесплатно использовать программу, но и изменять её по своему усмотрению.

Работать с Webmin достаточно просто - нужно запустить браузер, набирать <https://host.iu4.bmstu.ru:10000>, ввести свою авторизационную информацию (логин и пароль) и попасть на страницу администрирования.

При использовании системы Webmin необходимо следовать подробной инструкции пользователя.

1.5.1.3 Информационный портал вычислительного кластера

Информационный портал вычислительного кластера доступен по адресу <http://cluster.iu4.bmstu.ru>. Ресурс посвящен освещению работы вычислительного кластера по направлению «Наноинженерия». На данном портале размещена информация по учебным курсам для школьников, бакалавров, магистров и аспирантов по направлению "Проектирование и технология производства ЭС" (кафедра ИУ4 МГТУ им. Н.Э. Баумана).

Программа для школьников предназначена для учащихся 10-11 классов (возможно и более младших, отбор по собеседованию) общеобразовательных средних школ естественно - научного, физико-математического и подобных профилей. Учащимся профильных физико-математических школ предоставляется возможность познакомиться с основами высокопроизводительных вычислений в области нанотехнологий.

В рамках подготовки специалистов и бакалавров/магистров реализуются образовательные программы подготовки специалистов в области высокопроизводительных кластерных вычислений в наноинженерии. Программа включает в себя курс "Основы конструкторско-технологической информатики".

Магистерские программы ориентированы на изучение системного программирования, проведение исследований разнообразных электромагнитных волн, химических процессов в наносистемах, моделирования и проектирование различных инженерных задач в оптике, например проектирование фотонных кристаллов, волноводов и резонаторных систем.

В распоряжение аспирантов предоставляется доступ к вычислительным ресурсам кластера, у них появляется возможность моделировать и проводить расчеты для своих проектов с помощью установленного на кластере программного обеспечения, а так же участвовать в разработке прикладного ПО.

1.5.2 Диаграмма развертывания комплекса

Расширенное ТЗ для данной курсовой работы содержит четкое требование к минимальному аппаратному и программному обеспечению для ПК пользователей системы (пользователя, эксперта, администратора), а так же и к аппаратному и программному обеспечению сервера, на котором развертывается экспертная система.

Так как данная экспертная система имеет возможность внедрения в другие различные области знаний, необходимо было составить диаграмму развертывания (рисунок 1.19). К каждому узлу системы предъявлены требования для нормального функционирования последней.

Спецификация к диаграмме развертывания приведена в таблице 1.23.

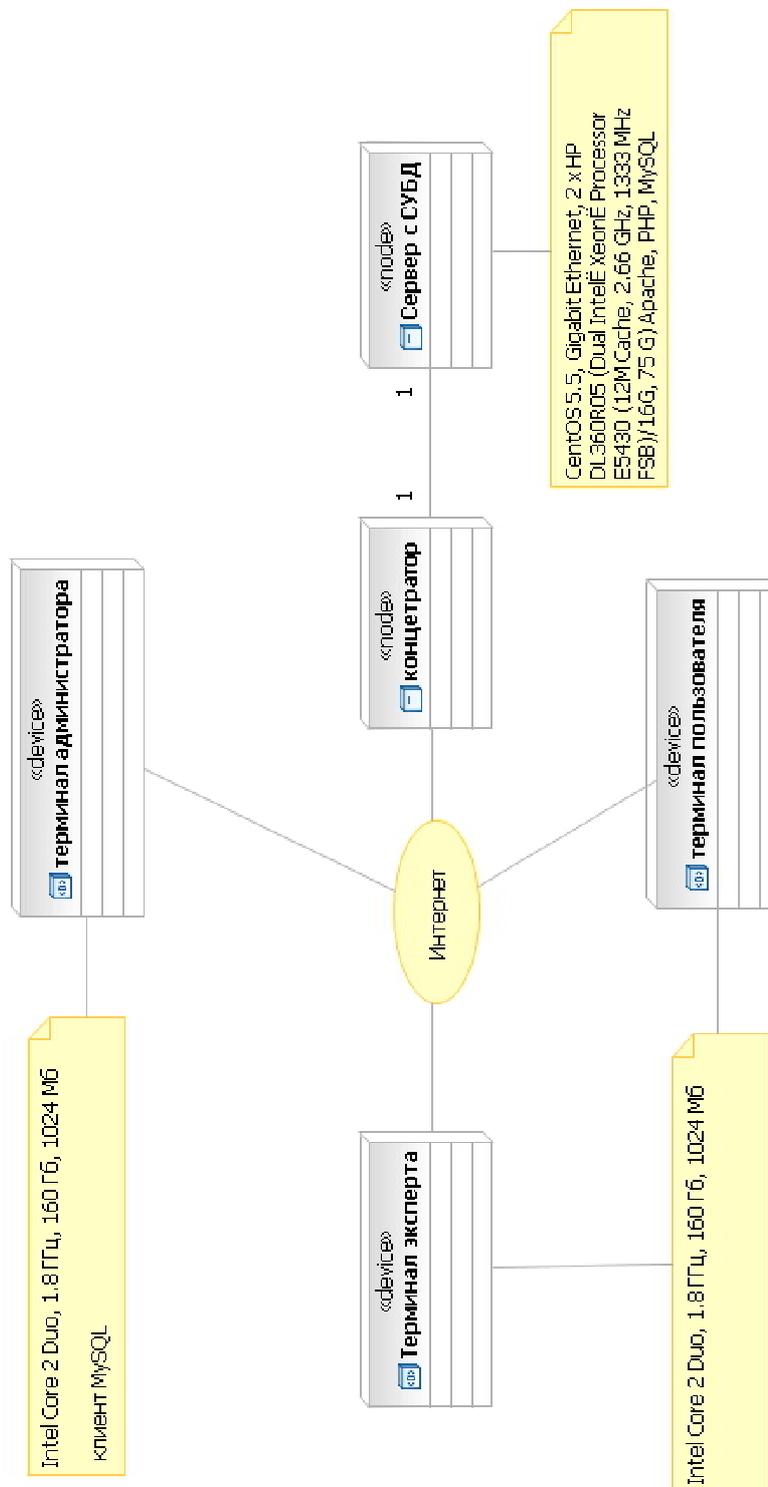


Рисунок 1.19 – Диаграмма развертывания ЭС

Таблица 1.23 – Спецификация к диаграмме развертывания

Процессор/устройство	Описание
терминал администратора, эксперта, пользователя	Intel Core 2 Duo, 1.8 GGz, 160 Gb, 1024 Mb
концентратор	CISCO
Сервер	Вычислительный кластер ННС

На терминале (ПК) администратора необходим быть установленным клиент MySQL. Кроме того, на всех терминалах могут быть установлены любые ОС Windows, Linux или Mac OS X последних поколений, и любой из браузеров IE, Opera, Firefox, Safari, Chrome последних поколений, в которых должны быть включены Cookies, и разрешены выполнения сценариев JavaScript.

ВЫВОДЫ ПО РАЗДЕЛУ

Разработанная модель процессов полностью отражает аспекты поведения системы разрабатываемой ЭС, определяет области её применения и состоит из диаграмм взаимодействия и кооперации, описывающих взаимодействия, состоящие из множества объектов и отношений между ними, включая сообщения, которыми они обмениваются.

Разработка логической модели является начальным этапом, перед, непосредственно, самим написанием кода ЭС:

- Диаграмма пакетов отображает структурный вид системы, отображающий её компоненты.
- Диаграмма системы ввода-вывода отображает отношение классов при построении этой системы. Видно, что для создания HTML-страниц необходим парсер, который на основе полученных данных из БД и необходимых шаблонов генерирует конечный вид страницы, которая является диалогом с пользователем.
- Диаграмма классов для БД позволила определить параметры для создания, непосредственно, самих таблиц в базе данных. Этими параметрами являются название таблицы и полей.

В ходе разработки ЭС была разделена на ряд компонентов: управления базой знаний, модулем работы с СУБД, модуль работы с пользователями и модуль для генерации HTML-страниц.

Была осуществлена детализация структуры ЭС до уровня файлов скриптов и файлов шаблонов. Выявление типа данных для создания модели базы данных было осуществлено при помощи физической модели данных.

Разрабатываемая ЭС основана на технологии тонкого клиента (клиент-серверной архитектуре). Данная технология позволяет пользователям системы использовать средние по мощности ПК, тогда как вся бизнес-логика осуществляется на сервере. ПК, с установленным на нём браузером, в данной случае служит пользователю как терминал доступа к ЭС.

2 МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ЭКСПЕРТНОЙ СИСТЕМЫ

2.1 Реализация инсталляционного комплекта

Данное методическое обеспечение поставляется вместе с экспертной системой и предназначено для следующих пользователей:

- Эксперты – эксперты в предметной области «Технологии микро- и наносистем».
- Администратор – обеспечивает установку ЭС и проведение регламентных работ для обеспечения необходимых условий непрерывной работы ЭС. Управление пользователями ЭС.

Данное методическое обеспечение предназначено для обучения пользованию ЭС и её администрированию.

2.2 Развертывание ЭС на кластере

2.2.1 Постановка задачи

В данном разделе описана последовательность развертывания ЭС на «Вычислительном кластере ННС». Данная работа должна быть выполнена администратором кластера, или разработчиком системы.

Развертывание состоит из следующих этапов:

- а) Установка сторонних программных пакетов – установка СУБД MySQL, веб-сервера Apache, FTP-сервер, интерпритатора PHP и настройка этих компонентов.
- б) Создание базы данных – создание базы данных на сервере и запуск её экземпляра.
- в) Генерация SQL-скриптов – генерация поставляемых SQL-скриптов, реализующих объекты базы данных.
- г) Закачка программных модулей на FTP-сервер и пробный запуск.

2.2.2 Генерация SQL-скриптов

После создания базы данных необходимо выполнить скрипты из комплекта поставки для создания требуемых объектов БД:

- create_tables.sql – создание таблиц;
- slot_types.sql – занесение типов значений слотов;
- user_roles.sql – занесение ролей пользователей ЭС;
- admin_user.sql – занесение пользователя Администратор.

После прогона всех скриптов все объекты базы данных будут созданы и готовы к заполнению.

2.2.3 Установка файлов-скриптов

Перед запуском скриптов на сервер необходимо внести конфигурационные данные в файл core/config.php, имеющий содержание показанное в таблице 2.1.

Таблица 2.1 – Конфигурация системы перед инсталляцией на сервер

```
<?php
$db_host = "";
$db_user = "";
$db_password = "";
$db_name = "";
?>
```

Спецификация конфигурационного файла показана в таблице 2.2.

Таблица 2.2 – Спецификация для конфигурационного файла

Параметр	Пояснение
\$db_host	Хост, на котором размещается СУБД. Если СУБД размещается на том же сервера, что и веб-сервер Apache, то здесь нужно указать localhost, или 127.0.0.1
\$db_user	Имя пользователя в СУБД, имеющий доступ к БД
\$db_password	Пароль пользователя в СУБД
\$db_name	Название базы данных созданной для данной ЭС

После конфигурирования файла core/config.php все файлы закидываются на выделенное место на сервер через FTP-сервер.

После установки в системе будет доступен один пользователь с правами доступа «Администратор» со следующими данными для авторизации:

Email – admin@bmstu.ru;

Password – password.

Рекомендуется изменить пароль после первого входа в систему. Эти действия проводятся через модуль «кабинет». Email также можно изменить через модель «кабинет».

Занесение новых пользователей осуществляется через компонент «регистрация», который будет описан ниже.

2.3 Инструкция работы пользователя

2.3.1 Модуль «Авторизация»

Данный модуль (рисунок 2.1) позволяет пользователя представиться системе путем авторизации, для этого необходимо ввести пару email/пароль, с которыми была зарегистрирована учетная запись.

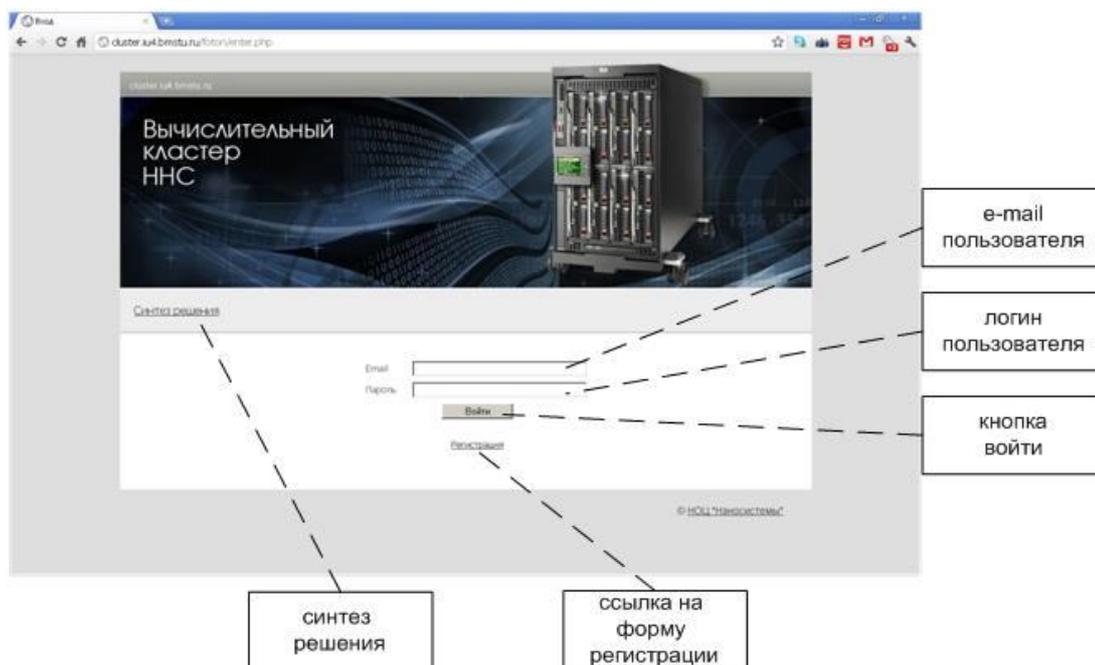


Рисунок 2.1 – Скриншот модуля «авторизация»

Если пара email/пароль не совпадает, или пользователя с таким email не существует в системе будет выведена ошибка.

2.3.2 Модуль «Регистрация»

Данный модуль (рисунок 2.2) служит для регистрации новых пользователей в системе, для этого необходимо ввести имя, email и пароль. Имя может быть написано как кириллицей, так и латиницей.

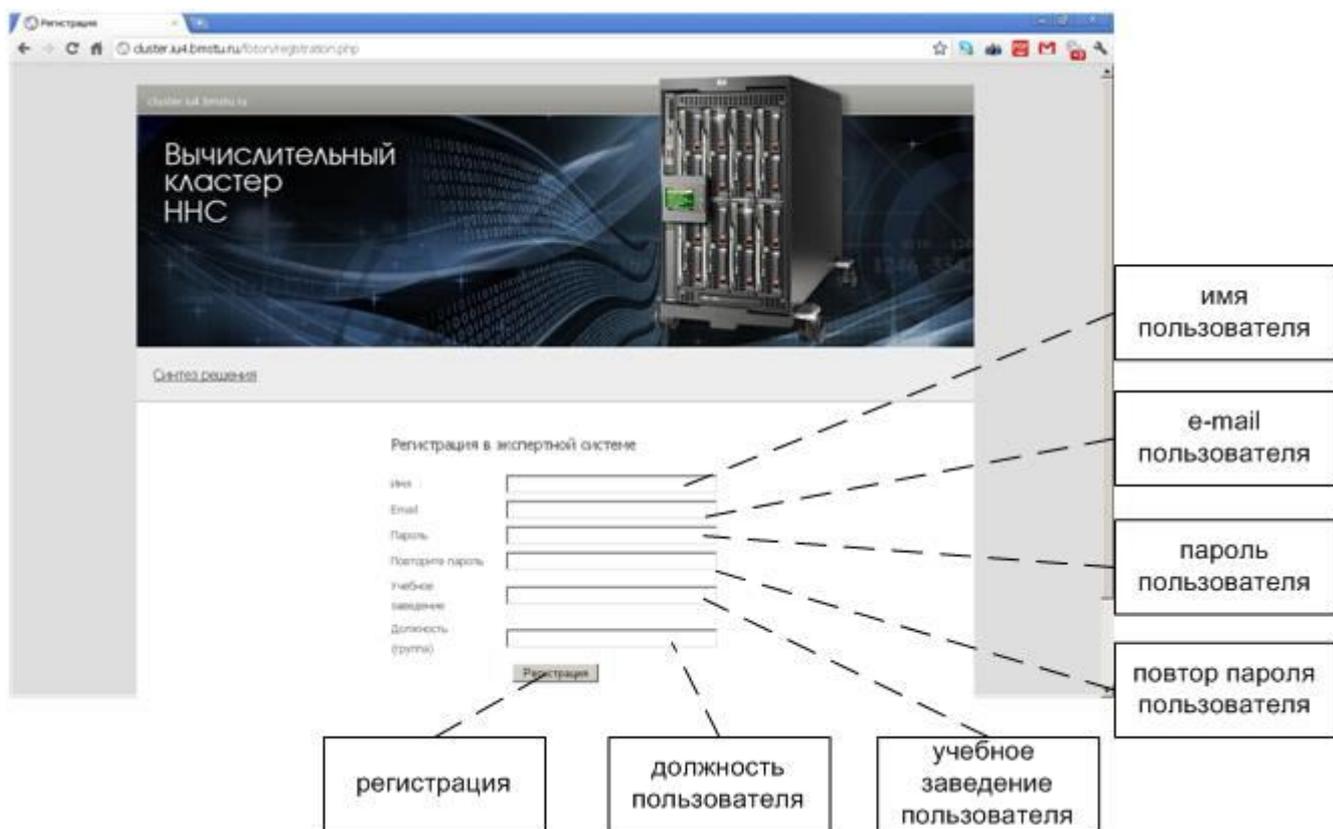


Рисунок 2.2 – Скриншот модуля «регистрация»

После регистрации администратор системы будет осведомлен об этом и рассмотрит заявку. После активации учетной записи администратором пользователь сможет воспользоваться парой email / пароль для авторизации в системе.

2.3.3 Модуль «Синтез решения»

Данный модуль (рисунок 2.3) является ключевым в системе. Этот модуль может быть доступен любому, даже не зарегистрированному пользователю.

Модуль представляет из себя диалог с пользователем вида ответ-вопрос. На ответы пользователя производится синтез для нахождения решения. По мере работы диалога пользователю также выводится трассировка его ответов на вопрос системы

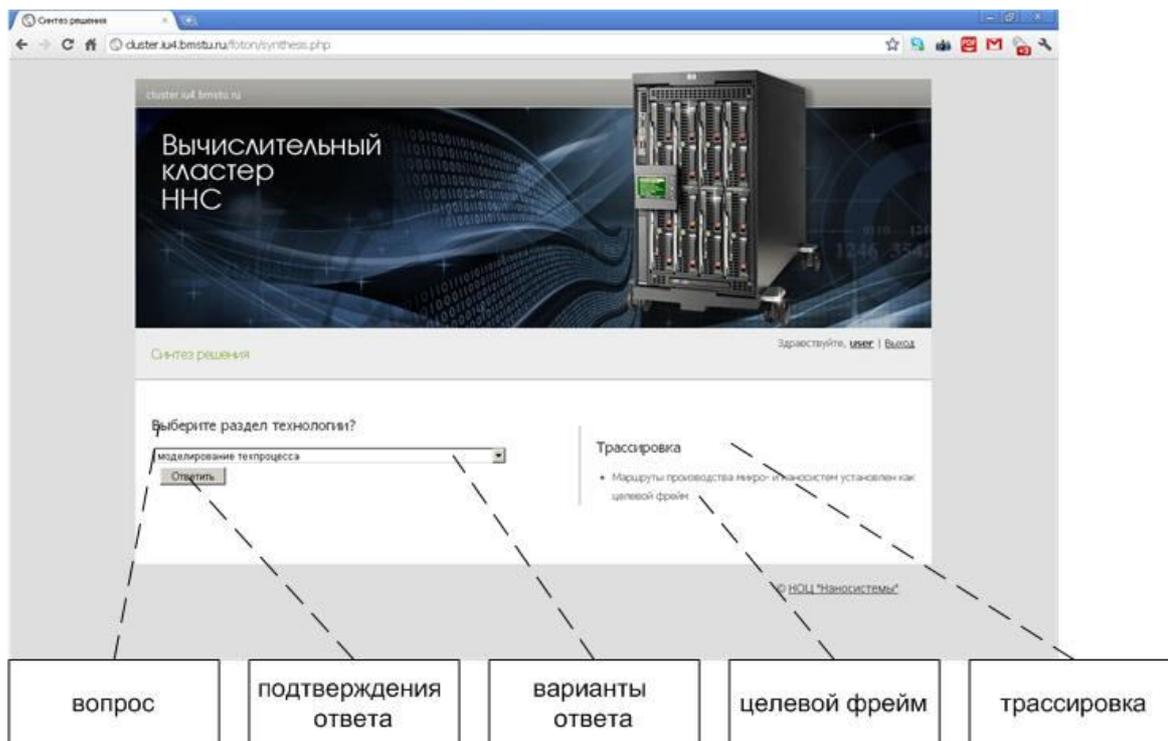


Рисунок 2.3 – Скриншот модуля «синтез решения». Состояние поиска решения в режиме диалога

Если решение не найдено, то пользователь ответственен за занесение нового знаниевого объекта базу знаний. Отправленные данные для нового решения будут отправлены эксперту на рассмотрение. Далее эксперт принимает решение о целесообразности включения нового решения.

2.3.4 Модуль «Кабинет»

Данный модуль (рисунок 2.4) служит для изменения данных пользователя, таких как: имя, email, пароль.

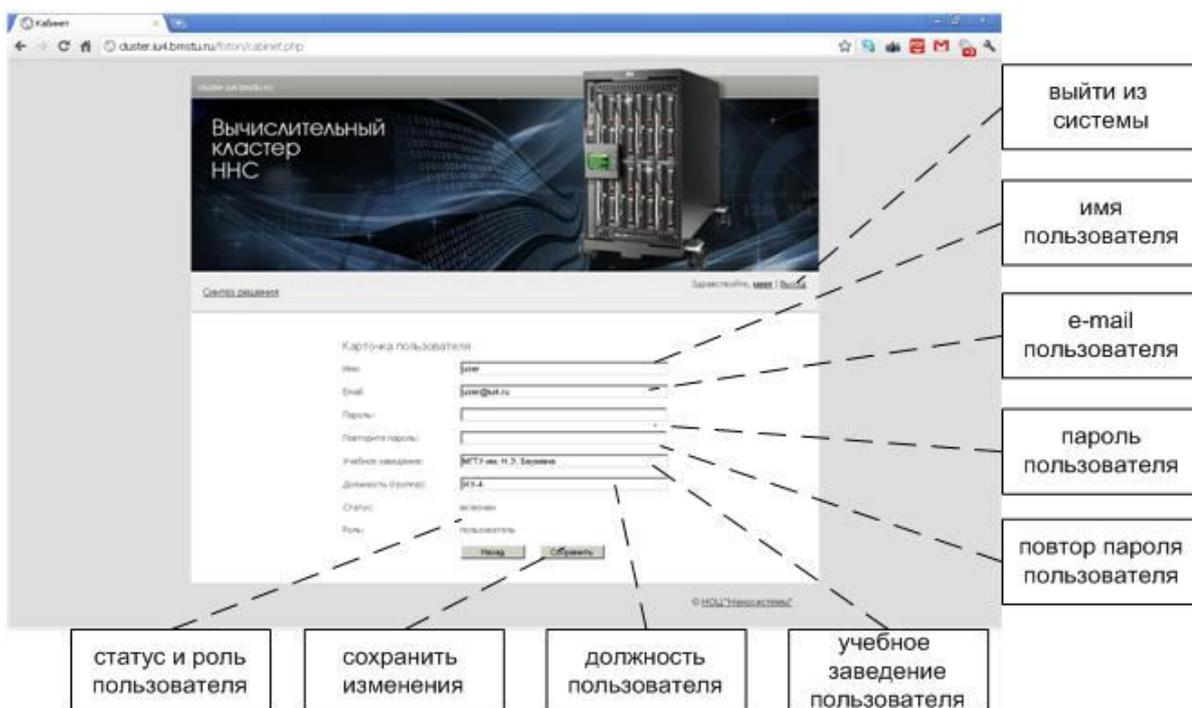


Рисунок 2.4 – Скриншот модуля «Кабинет»

Данный модуль доступен каждому пользователю системы. Помимо редактирования персональных данных пользователь осведомляется с текущим состоянием учетной записи в системе: статус и роль. Статус имеет следующие значения:

- включен;
- выключен.

Роль пользователя служит для предоставления нужного уровня доступа к системе, т.е. возможность эксперт должен иметь доступ к редактированию базы знаний, а администратор имеет полный уровень доступа к всей системе. Данное поле имеет следующие значения:

- пользователь;
- эксперт;
- администратор.

2.4 Инструкция работы эксперта

2.4.1 Модуль «Слоты»

Модуль «слоты» (рисунок 2.5) выводит все занесенные слоты в базе знаний. Система отображает название слота, а также вопрос, который выводится пользователю при диалоге с системой ввода-вывода.

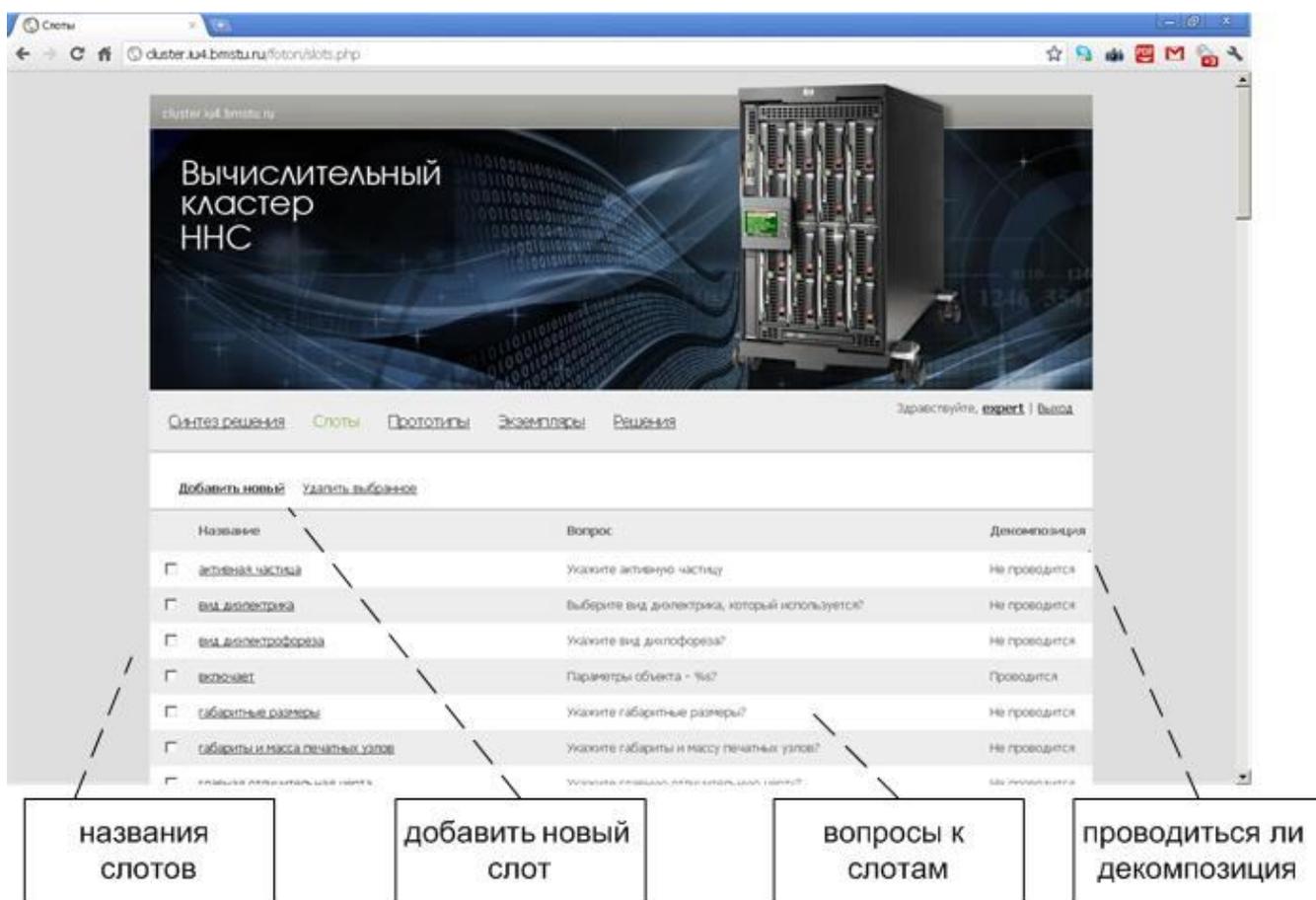


Рисунок 2.5 – Скриншот модуля «слоты»

Доступ к данному модулю имеют пользователи с ролью «эксперт» и «администратор»

2.4.2 Модуль «редактирование слота»

Модуль «редактирование слота» (рисунок 2.6) вызывается при создании нового слота, либо при редактировании уже существующего.

Все заполненные поля – является необходимым условием

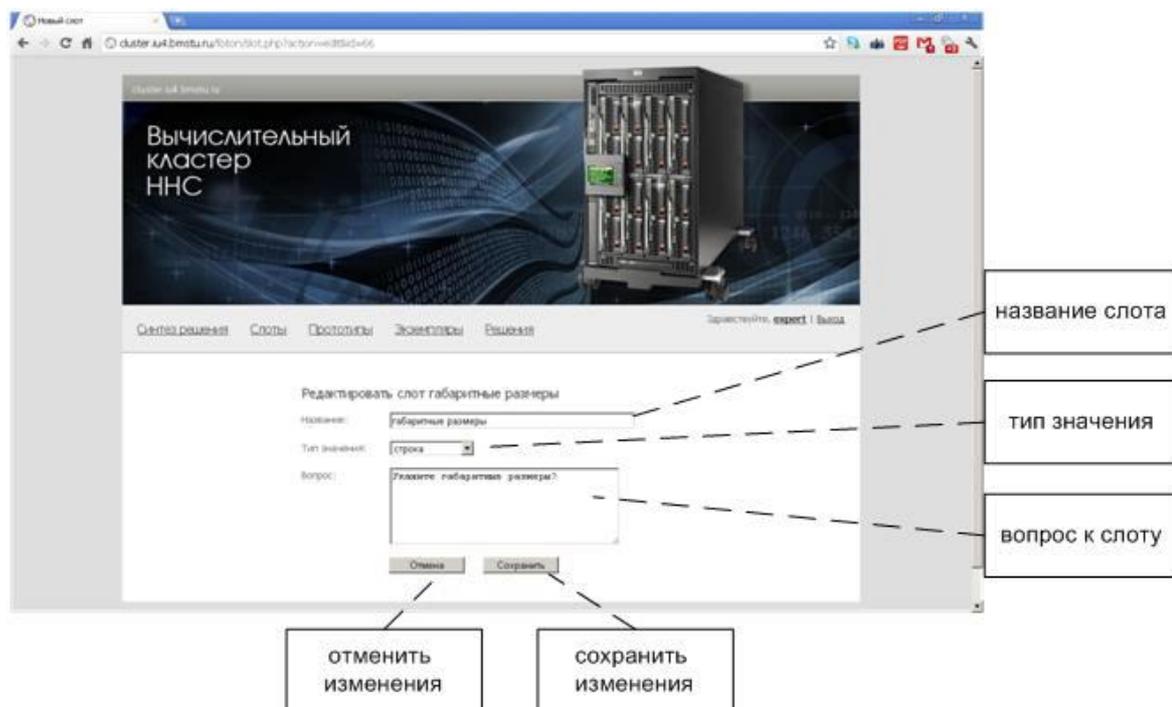


Рисунок 2.6 – Скриншот модуля «редактирования слота»

Доступ к данному модулю имеют пользователи с ролью «эксперт» и «администратор»

2.4.3 Модуль «Прототипы»

Модуль «прототипы» (рисунок 2.7) выводит эксперту все имеющиеся в базе знаний фреймы-экземпляры в соответствии их иерархии. Экземпляры отличаются на фреймов наличием соответствующей иконки.

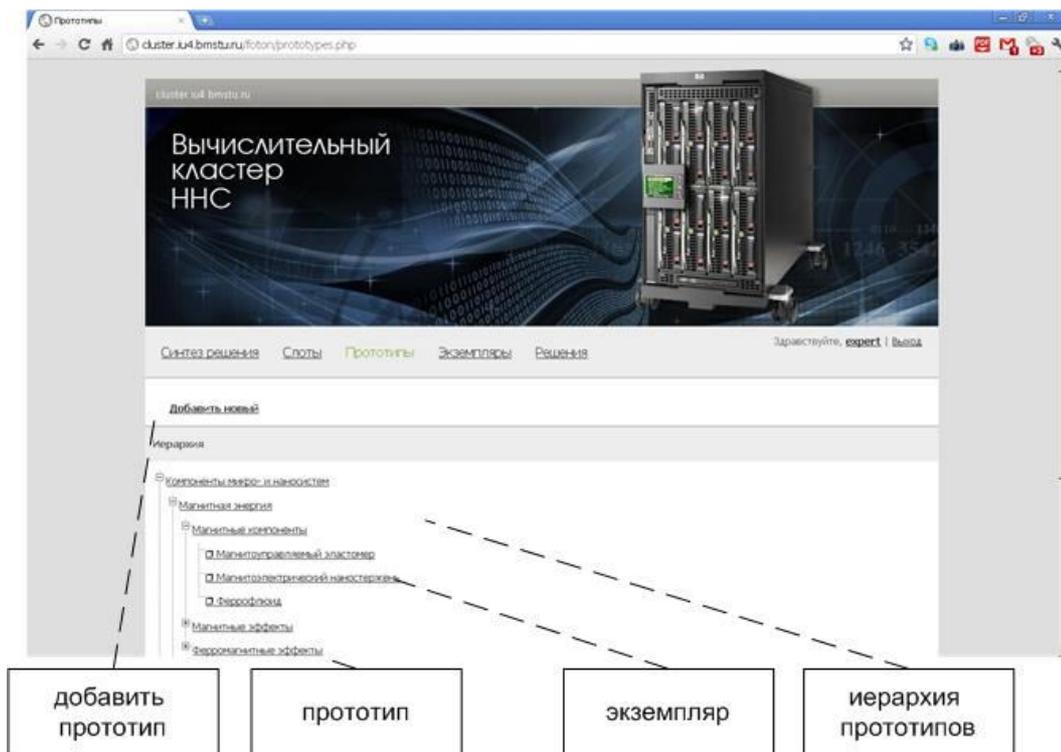


Рисунок 2.7 – Скриншот модуля «прототипы»

Доступ к данному модулю имеют пользователи с ролью «эксперт» и «администратор»

2.4.4 Модуль «Редактирование прототипа»

Модуль «редактирование прототипа» (рисунок 2.8) вызывается при создании нового фрейма (экземпляра). Здесь необходимо ввести имя прототипа, отметить его положение в иерархии. Система сразу же отобразит унаследованные слоты.

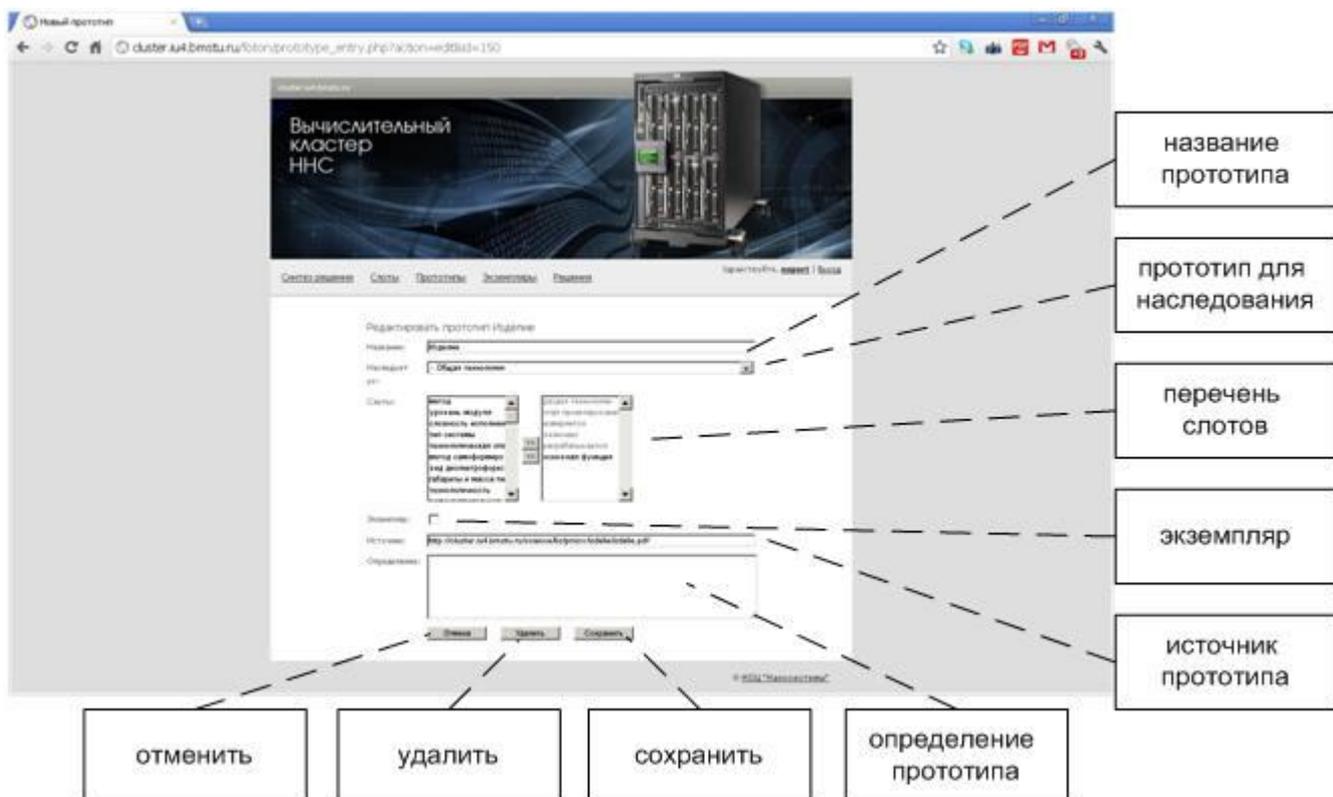


Рисунок 2.8 – Скриншот модуля «редактирования прототипа»

Доступ к данному модулю имеют пользователи с ролью «эксперт» и «администратор»

2.4.5 Модуль «экземпляры»

Модуль «экземпляры» (рисунок 2.9) выводит эксперту список всех экземпляров в базе знаний. Модуль выводит название экземпляров, а также название слотов, которые присвоены данному экземпляру.

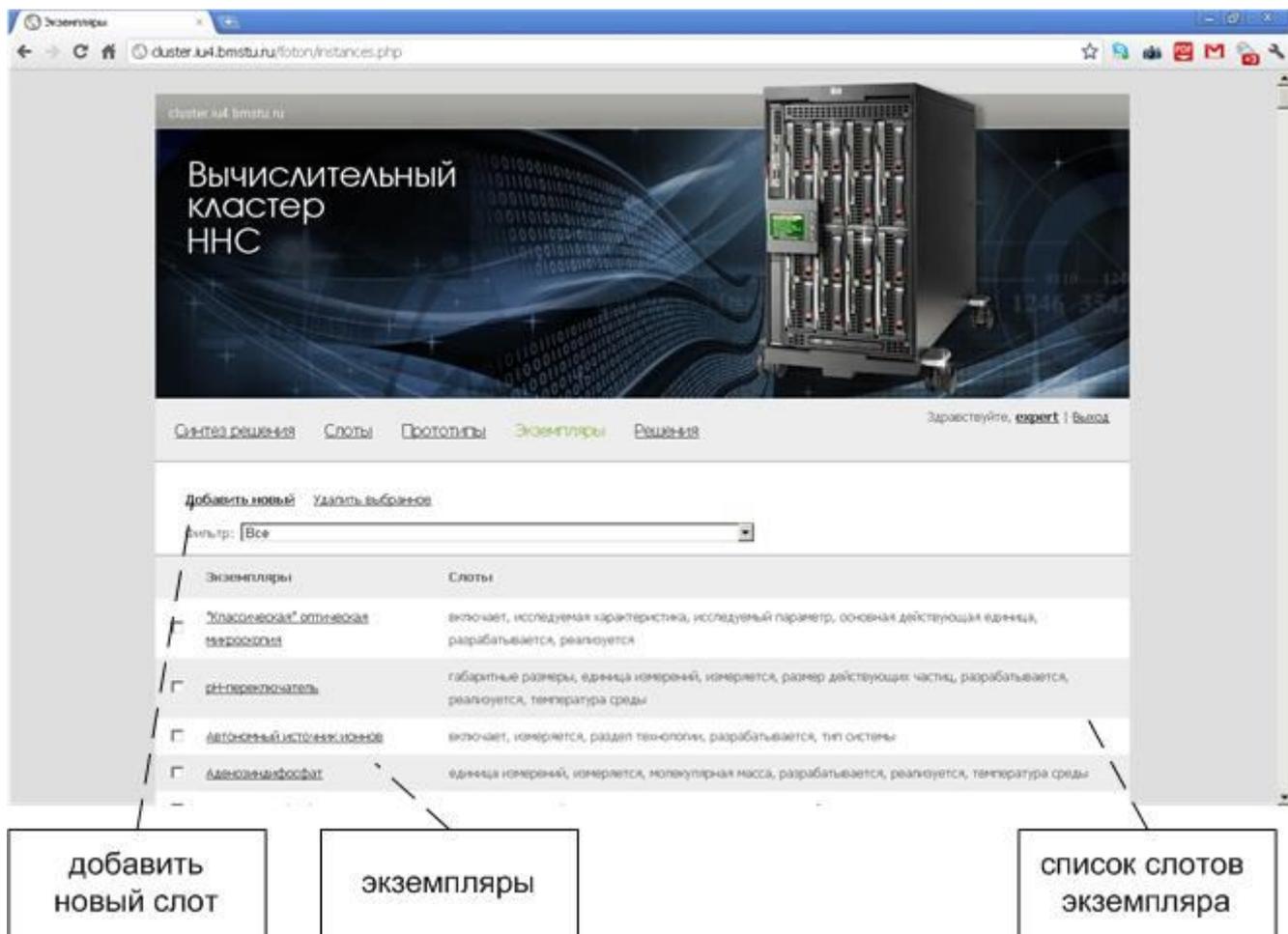


Рисунок 2.9 – Скриншот модуля «экземпляры»

Доступ к данному модулю имеют пользователи с ролью «эксперт» и «администратор»

2.4.6 Модуль «Редактирование экземпляра»

Данный модуль использует тот же диалог, что и модуль «редактирование прототипа» (рисунок 2.8) служит для ввода значений слотов, для данного экземпляра экспертом. Модули «редактирование экземпляра» и «редактирование прототипа» являются одним и тем же компонентом, который автоматически подстраивается под тот или иной тип данных для редактирования.

Визуальное отличие модулей «редактирование экземпляра» и «редактирование прототипа» в компоненте `prototype_entry.php` отличается тем, что форма «Экземпляр», типа `checkbox`, помечена.

2.5 Инструкция работы администратора

2.5.1 Модуль «пользователи»

Модуль «пользователи» (рисунок 2.10) служит для управлением пользователями, зарегистрированными в системе. Важным аспектом данного модуля является отображения статусов пользователей. Администратор обязан рассмотреть заявку новых зарегистрированных пользователей на активации, для последующей их авторизации в системе.

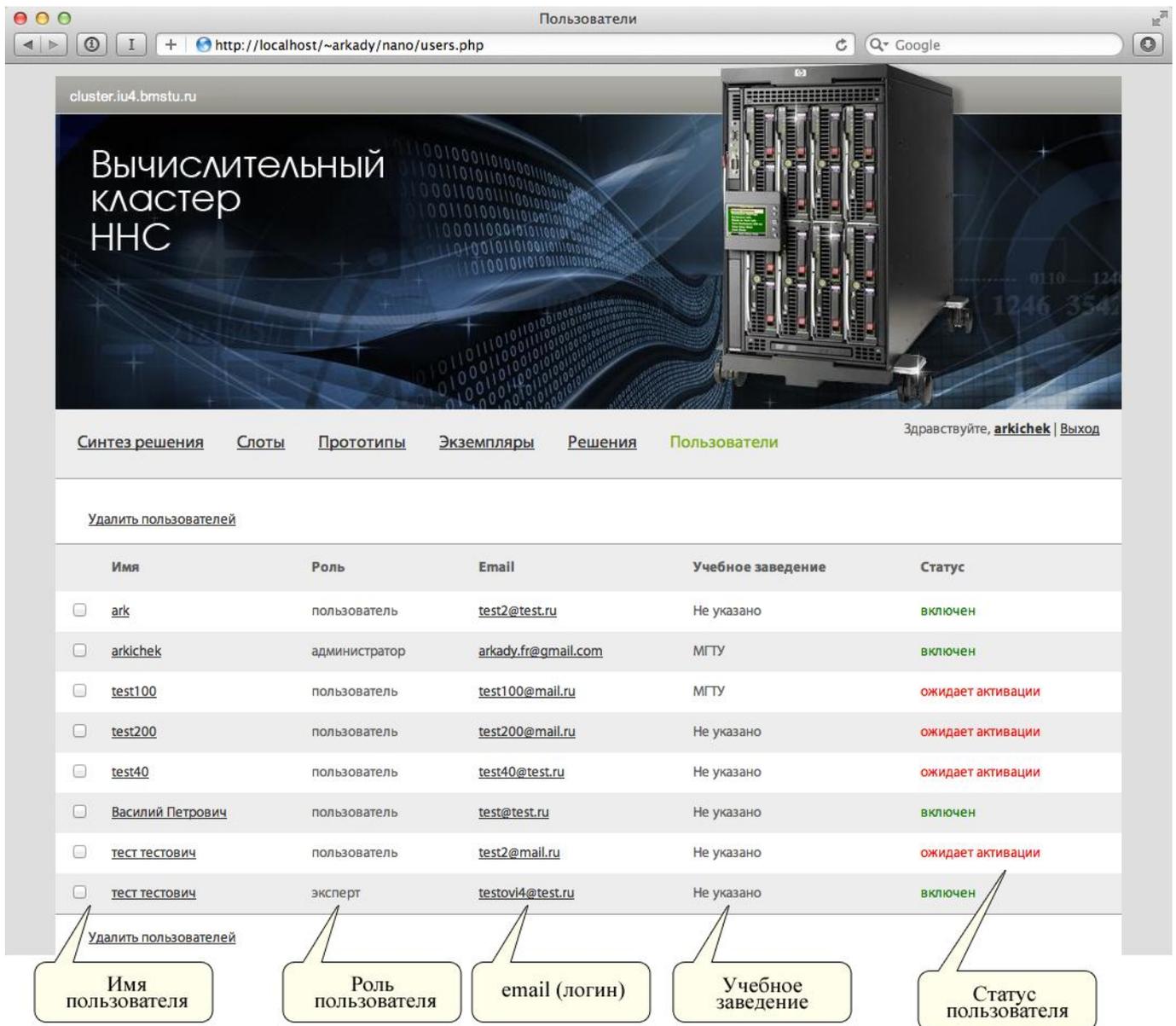


Рисунок 2.10 – Скриншот модуля «пользователи»

Также, данный модуль позволяет удалить пользователей из системы, что лишает их доступа к редактированию базы знаний (для пользователей, имеющих уровень доступа эксперт).

2.5.2 Модуль «Карточка пользователя»

Модуль «карточка пользователя» (рисунок 2.11) позволяет изменить уровни доступа пользователя к системе (роль), активировать его, если пользователь находится в состоянии ожидания авторизации, а также, удалить пользователя из системы.

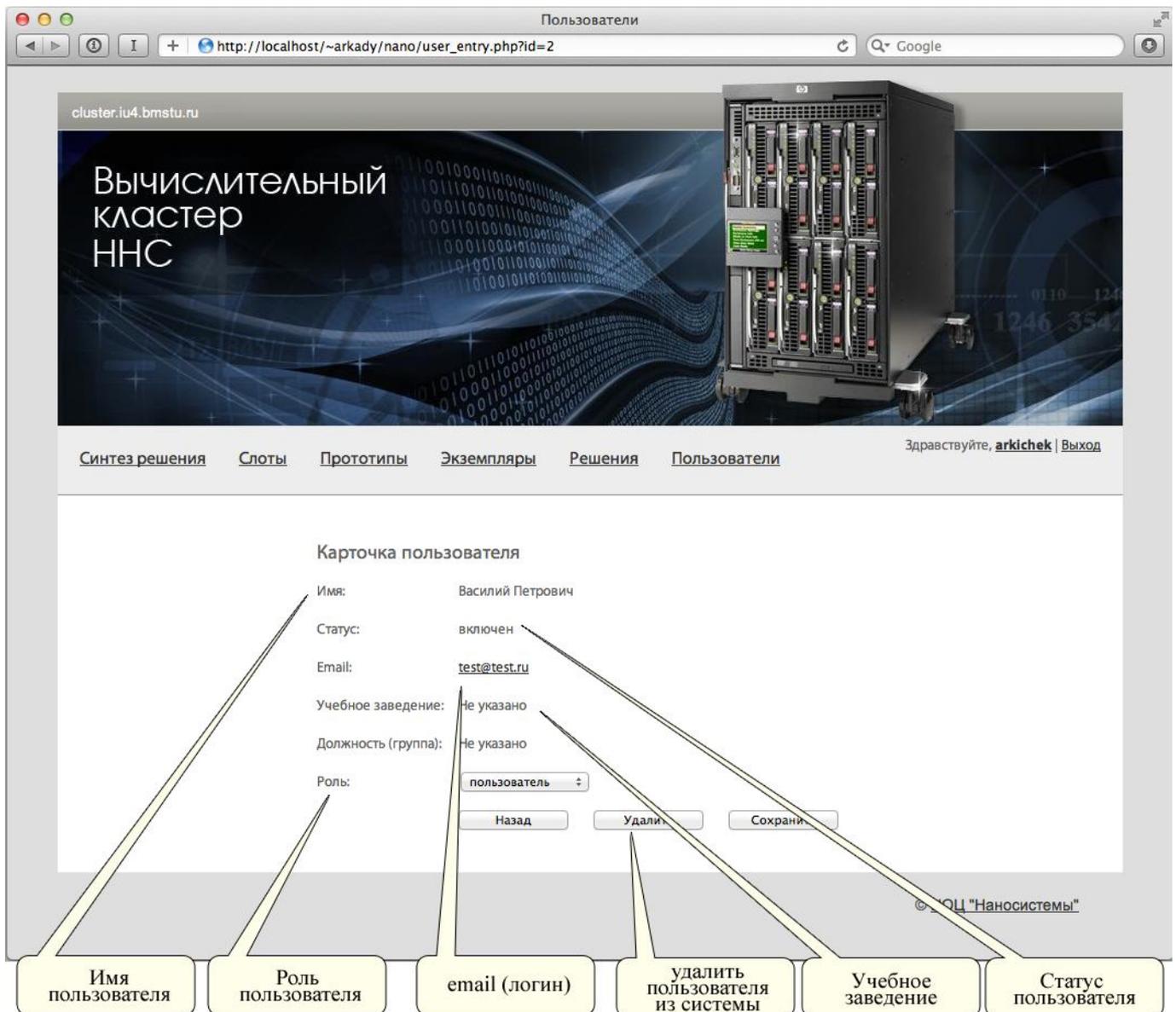


Рисунок 2.11 – Скриншот модуля «карточка пользователя»

В отличие от модуля «кабинет» данный модуль не позволяет изменять личные данные пользователя такие как: имя, email, пароль.

ВЫВОДЫ ПО РАЗДЕЛУ

Разработанная система имеет возможность внедрения и в других областях знаний, в связи с этим помимо руководств пользователей составлены инсталляционные инструкции по развертыванию системы на сервере, удовлетворяющий требованиям к аппаратному и программному обеспечению сервера.

Важным моментом перед размещением файлов скриптов является конфигурирование файла `config.php`, который отвечает за подключение к СУБД.

Руководство пользователя включает в себя подробное описание интерфейсов системы и её возможности. Каждый пользователь, имеющий доступ к системе, имеет один из трёх уровней доступа:

- пользователь;
- эксперт;

- администратор.

Различия заключаются в ограниченном доступе к модулям системы, пользователь имеет доступ к модулям:

- кабинет;
- синтез решения.

Эксперт имеет доступ к следующим модулям:

- кабинет;
- синтез решения;
- слоты;
- прототипы;
- экземпляры.

Администратор имеет доступ ко всем модулям, а именно:

- кабинет;
- синтез решения;
- слоты;
- прототипы;
- экземпляры;
- пользователи.

ЗАКЛЮЧЕНИЕ

Результатом проведенной работы является созданная и развернутая на «Вычислительном кластере ННС» экспертная система по базе знаний предметной области для целей анализа методов и технологий, а также сравнения научно-технических решений в указанной области.

Разработанный программный комплекс экспертной системы по предметной области технологий микро- и наносистем обеспечивает механизм интеллектуального синтеза решений и визуальной навигации по семантической информации в указанной области, реализованной на платформе тонкого клиента.

Экспертная система осуществляет интеллектуальную обработку данных по технологиям микро- и наносистем, реализует структуризацию, комплексный анализ и представление в виде онтологии комплекса исследовательских, технических и технологических знаний по технологиям микро- и наносистем, реализующую сетевую каталогизацию и обеспечение доступа к семантическим ресурсам по предметной области, что направлено на решение научных, инженерных, производственных и образовательных задач в сфере технологий микро- и наносистем.

В состав комплекса входят модули для редактирования базы знаний и синтеза решений (экспертной системы):

- Модуль «Слоты».
- Модуль «Фреймы».
- Модуль «Экземпляры».
- Модуль «Синтеза решений».
- Модуль «Визуализации найденных решений».

Также в состав входят вспомогательные модули для второстепенных задач работы комплекса:

- Модуль регистрации пользователей.
 - Подсистема консультирования пользователей базы знаний.
- Система обслуживается группой следующих специалистов:
- системный администратор;
 - администратор баз данных;
 - прикладной администратор.

От несанкционированного доступа защищены следующие объекты:

- файлы БД;
- файлы индекса подсистемы индексирования;
- данные, хранящиеся в БД Комплекса;
- несанкционированный запуск и останов сервисов сбора информации и индексирования;
- исполняемые файлы СУБД, сервера приложений, операционной системы.

Защита информации от несанкционированного доступа реализована на следующих уровнях:

- аутентификация и авторизация доступа средствами операционной системы;
- разграничение доступа к файлам с помощью средств операционной системы;
- аутентификация и авторизация доступа к данным и настройкам СУБД с использованием средств СУБД;
- использование аутентификации при обращении к подсистеме управления Комплекса.

Созданный программный комплекс имеет встроенные средства разграничения доступа к их частям и должны предусматривать:

- централизованное администрирование;
- возможности ограничения работы пользователя программного комплекса рамками любой из ее частей, а в рамках ее частей должно быть предусмотрено разделение полномочий пользователя на просмотр и изменение данных;

- возможности ведения собственного аудита и настраиваемой регистрации собственных событий, а также событий производимых любыми пользователями программного комплекса;
- технические (программно-технические) средства защиты информации как входящие в состав операционных систем, так и поставляемые дополнительно;
- оборудование локальных вычислительных сетей;
- оборудование высокоскоростных телекоммуникационных сетей.

Для предотвращения некорректной работы программного комплекса реализовано:

- авторизацию доступа к данным;
- семантический и синтаксический контроль исходных данных;
- вывод сообщений об ошибках;
- возможность повторного ввода данных.

В ходе работы над проектом были разработаны следующие алгоритмы[9]:

- Алгоритм сбора информации об источниках гипертекстовых данных.
- Алгоритм выявления новых источников данных .
- Методика взаимодействия подсистемы управления и подсистемы сбора информации.
- Методика взаимодействия подсистемы управления с системой управления базами данных.
- Методика взаимодействия подсистемы сбора данных и подсистемы индексирования данных.

Указанные алгоритмы обеспечивают высокую скорость и надежность при обработке данных.

База знаний по технологиям микро- и наносистем была спроектирована при помощи СМАР-технологий, при помощи которых четко просматривается фреймовая древовидная структура знаний. На основании СМАР-схем была создана логическая модель ЭС при помощи языка UML.

Ключевым момент в ходе создания ЭС явился модуль «Синтез решения», алгоритм которого был рассмотрен в 1 главе.

Сама экспертная система была разработана на основе клиент-серверной технологии. Тонкий клиент реализован при помощи скриптового языка программирования PHP, с применением HTML и CSS для создания интерфейсов. В качестве СУБД используется Oracle.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Поздняев А.С., Власов А.И. Развитие информационно-телекоммуникационного сектора экономики при гармонизации мировой финансовой системы // Вестник МГТУ им. Н.Э. Баумана. серия: Приборостроение. 2010. № 1. - С. 121-125.
2. Корсаков С.Н. Начертание нового способа исследования при помощи машин, сравнивающих идеи / Пер. с франц. под ред. А.С. Михайлова. – М.: МИФИ, 2009, 44 с.
3. Novak J. D., Canas A.J. The Theory Underlying Concept Maps and How to Construct Them. //Technical Report IHMC SmartTools 2006-01 Rev 01-2008, Florida Institute for Human and Machine Cognition, 2008.
4. Буч Г., Рамбо Дж., Джекобсон А. Язык UML. Руководство пользователя.-М.:ДМК ПРЕСС; СПб.:Питер,2004.-429 с.
5. Герман О.В. Введение в теорию экспертных систем и обработку знаний. Учебное пособие. Мн.: ДизайнПро, 1995. - 456 с.
6. Сотник С. Л. Конспект лекций по курсу «Основы проектирования систем искусственного интеллекта», 1997-1998.
7. Зандстра М., PHP: объекты, шаблоны и методики программирования, 3-е издание = PHP Objects, Patterns and Practice, Third Edition — М.: «Вильямс», 2010. — С. 560. — ISBN 978-5-8459-1689-1.
8. Суэринг С., Конверс Т., Джойс П. PHP и MySQL. Библия программиста, 2-е издание = PHP 6 and MySQL 6 Bible — М.: «Диалектика», 2010. — 912 с. — ISBN 978-5-8459-1640-2.
9. Кормен Т., Лейзерсон И. Ч., Ривест Р. Л., Штайн К. Алгоритмы: построение и анализ = INTRODUCTION TO ALGORITHMS — 2-е изд. — М.: «Вильямс», 2006. — С. 1296. — ISBN 0-07-013151-1.
10. Кнут Д. Искусство программирования, том 1. Основные алгоритмы = The Art of Computer Programming, vol.1. Fundamental Algorithms — 3-е изд. — М.: «Вильямс», 2006. — С. 720. — ISBN 0-201-89683-4
11. Нильсен Я., Перниче К. Веб-дизайн: анализ удобства использования веб-сайтов по движению глаз = Eyetracking Web Usability — М.: «Вильямс», 2010. — С. 480. — ISBN 978-5-8459-1652-5.
12. Титтел Э., Ноубл Дж. HTML, XHTML и CSS для чайников, 7-е издание = HTML, XHTML & CSS For Dummies, 7th Edition — М.: «Диалектика», 2011. — 400 с. — ISBN 978-5-8459-1752-2.
13. Zakas N., McPeak J., Fawcett J. Professional Ajax — 2nd ed. — Wrox, 2007. — 624 p. — (Programmer to Programmer). — ISBN 0470109491.
14. Lindley C. jQuery Cookbook. Solutions & Examples for jQuery Developers. O'Reilly Media, 2009. – 478 с.

ЛИСТИНГИ SQL-СКРИПТОВ

Таблица П.1 Листинг sql-скрипта создания таблиц

```

CREATE TABLE `ftn_prototypes` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `parent_id` int(11) DEFAULT NULL,
  `name` char(255) DEFAULT NULL,
  `is_instance` enum('Y','N') NOT NULL DEFAULT 'N',
  `definition` text,
  `url` text,
  `decomposition` int(11) DEFAULT NULL,
  `is_primary` enum('Y','N') NOT NULL DEFAULT 'N',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8;

CREATE TABLE `ftn_prototypes_slots` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `slot_id` int(11) DEFAULT NULL,
  `prototype_id` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;

CREATE TABLE `ftn_slots` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` char(200) DEFAULT NULL,
  `type` int(11) DEFAULT NULL,
  `question` text,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;

CREATE TABLE `ftn_slots_types` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` char(255) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;

CREATE TABLE `ftn_slots_values` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `instance_id` int(11) DEFAULT NULL,
  `slot_id` int(11) DEFAULT NULL,
  `slot_value` char(255) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=19 DEFAULT CHARSET=utf8;

CREATE TABLE `ftn_users` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `email` char(255) DEFAULT NULL,
  `password` char(255) DEFAULT NULL,
  `name` char(255) DEFAULT NULL,
  `role` int(11) DEFAULT '1',
  `is_active` enum('Y','N') NOT NULL DEFAULT 'N',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8;

CREATE TABLE `ftn_users_roles` (
  `id` int(11) NOT NULL,
  `role` char(255) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Таблица П1.2 – Листинг sql-скрипта для первичного заполнения БД

```
INSERT INTO `ftn_slots_types` (`id`,`name`)
VALUES
  (1, 'число'),
  (2, 'булеевый'),
  (3, 'строка');

INSERT INTO `ftn_users_roles` (`id`,`role`)
VALUES
  (1, 'пользователь'),
  (2, 'эксперт'),
  (3, 'администратор');

INSERT INTO `ftn_users` (`id`,`email`,`password`,`name`,`role`,`is_active`)
VALUES
  (1, 'admin@bmstu.ru', 'adminpassword', 'admin', 3, 'Y');
```

ЛИСТИНГИ PHP-СКРИПТОВ

Таблица П2.1 – Листинг php-скрипта core/DBController.php

```
<?php

/*
 * User.php
 *
 * Версия 1.0
 *
 * Дата 01.01.2011
 *
 * Фамилия Имя Отчество
 */

require_once "config.php";

class DBController
{
    public $db;

    public function __construct()
    {
        global $db_host, $db_user, $db_password, $db_name;

        $this->db = mysql_connect($db_host, $db_user, $db_password) or die
(mysql_error());
        mysql_select_db($db_name, $this->db) or die (mysql_error());
        mysql_query("SET NAMES utf8");
    }

    public function execute($query)
    {
        $result = mysql_query($query) or die (mysql_error());
        return $result;
    }

    public function executeAndGetArray($query)
    {
        $result = $this->execute($query);
        $arrayToReturn = array();
        while ($row = mysql_fetch_array($result, MYSQL_ASSOC))
        {
            $arrayToReturn[] = $row;
        }
        return $arrayToReturn;
    }

    public function get($tableName, $filter = NULL, $sorting = NULL,
$isDistinct = false)
    {
        $query = "SELECT * FROM $tableName";
    }
}
```

```
if ($isDistinct)
    $query = "SELECT DISTINCT * FROM $tableName";

    if ($filter)
        $query .= " WHERE $filter";

    if ($sorting)
        $query .= " ORDER BY $sorting";

    $result = $this->execute($query);
    $arrayToReturn = array();
    while ($row = mysql_fetch_array($result, MYSQL_ASSOC))
    {
        $arrayToReturn[] = $row;
    }
    return $arrayToReturn;
}

public function add($tableName, $params)
{
    $query = "INSERT INTO $tableName SET ";
    $counter = 0;
    foreach ($params as $key => $value)
    {
        if (is_numeric($value))
            $query .= ($counter++ == 0) ? "$key=$value" : ", $key=$value";
        else if (is_null($value))
            $query .= ($counter++ == 0) ? "$key=NULL" : ",
$key=NULL";
        else
            $query .= ($counter++ == 0) ? "$key='$value'" : ",
$key='$value'";
    }
    $result = $this->execute($query);
    if ($result)
        return mysql_insert_id();
    return -1;
}

public function update($tableName, $filter, $params)
{
    $query = "UPDATE $tableName SET ";
    $counter = 0;
    foreach ($params as $key => $value)
    {
        if (is_numeric($value))
            $query .= ($counter++ == 0) ? "$key=$value" : ", $key=$value";
        else if (is_null($value))
            $query .= ($counter++ == 0) ? "$key=NULL" : ",
$key=NULL";
        else
            $query .= ($counter++ == 0) ? "$key='$value'" : ",
$key='$value'";
    }
    $query .= " WHERE $filter";
    return $this->execute($query);
}
```

```
public function updateOrInsert($tableName, $filter, $params)
{
    $rows = $this->get($tableName, $filter);
    $result = false;
    if (count($rows) > 0)
        $result = $this->update($tableName, "id=".$rows[0]['id'],
$params);
    else
        $result = $this->add($tableName, $params);
    return $result;
}

public function delete($tableName, $filter)
{
    if ($filter)
        return $this->execute("DELETE FROM $tableName WHERE $filter");
}

public function destroy()
{
    if ($this->db)
        mysql_close($this->db);
}
}
?>
```

Таблица П2.2 – Листинг PHP-скрипта core/User.php

```
<?php
/*
 * User.php
 *
 * Версия 1.0
 *
 * Дата 2011
 *
 * Аркадий Фраерман arkady.fr@gmail.com
 */

// import
require_once "DBController.php";

class User
{
    // user id in db (usr_id)
    public $id = "";
    // user name in db (usr_name)
    public $name = "";
    // user password in db (usr_password)
    public $password = "";
    // user email in db (usr_email)
    public $email = "";
    // user role in db (usr_role)
    public $role = 1;
    // DBController object
```

```
private $db;

public function __construct()
{
    $this->db = new DBController();
    session_start();
}

/**
 * check pair user name and user password, if ok, then start session
 */
public function authorize()
{
    $filter = "email='".$this->email.'" AND password='".$this->password.'"';
    $result = $this->db->get("ftn_users", $filter);

    if (!$result)
        return 1;

    if ($result[0]['is_active'] == 'N')
        return 2;

    $this->id = $result[0]['id'];
    $this->name = $result[0]['name'];
    $this->password = $result[0]['password'];
    $this->email = $result[0]['email'];
    $this->role = $result[0]['role'];

    $_SESSION['userId'] = $this->id;

    return 0;
}

/**
 * destroy session
 */
public function deauthorize()
{
    $this->id = "";
    $this->name = "";
    $this->password = "";
    $this->email = "";
    $this->role = 1;

    unset($_SESSION['userId']);
    session_destroy();
}

/**
 * check if session in started, if ok, then fill up object properties
 */
public function isAuthorized()
{
    if(isset($_SESSION["userId"]) && $_SESSION["userId"] > 0)
    {
        $filter = "id=".$_SESSION["userId"];
        $result = $this->db->get("ftn_users", $filter);
    }
}
```

```
        if (!$result)
        {
            unset($_SESSION['userId']);
            session_destroy();
            return false;
        }

        $this->id = $result[0]['id'];
        $this->name = $result[0]['name'];
        $this->password = $result[0]['password'];
        $this->email = $result[0]['email'];
        $this->role = $result[0]['role'];

        return true;
    }
    else
        return false;
}
?>
```

Таблица П2.3 – Листинг PHP-скрипта core/utils.php

```
<?php

/*
 * utils.php
 *
 * Версия 1.0
 *
 * Дата 2011
 *
 */

function addJScriptHead($source, $template)
{
    $scriptSource = "<script type=\"text/javascript\"
src=\"\$source\"></script></head>";
    return str_replace("</head>", $scriptSource, $template);
}

function requireAuthorization($user)
{
    $currentUrl = $_SERVER['REQUEST_URI'];
    preg_match("/\/\([A-z]+\.\php)/", $currentUrl, $currentPage);
    $currentPage = $currentPage[1];
    if (!$user->isAuthorized())
    {
        header("Location: enter.php?return_page=$currentPage");
        exit();
    }
}
?>
```

Таблица П2.4 – Листинг PHP-скрипта synthesis.php

```
<?php
```

```
/*
 * synthesis.php
 *
 * Версия 1.0
 *
 * Дата 2011
 *
 */

header("Content-Type: text/html;charset=utf-8");

//-----
// includes
//-----
require_once "ui/top_menu.php";
require_once "ui/user_menu.php";
require_once "core/DBController.php";
require_once "core/User.php";
require_once "core/utils.php";

// DB instance
$db = new DBController();

// user instance
$user = new User();

//-----
// check authorization
//-----
//requireAuthorization(&$user);
$user->isAuthorized();

//-----
// interface
//-----

$template = file_get_contents("templates/main.html");
$template = addJScriptHead("js/jquery.js", $template);
$template = addJScriptHead("js/synthesis.js", $template);
$template = str_replace("::{pageTitle::}", "Синтез решения", $template);

insertMenu(&$template, $user);
insertUserMenu(&$template);
$synthesisTemplate = file_get_contents("templates/synthesis.html");
$template = str_replace("::{content::}", $synthesisTemplate, $template);

print $template;

?>
```

Таблица П2.5 – Листинг JS-скрипта js/synthesis.js

```
/*
 * synthesis.js
 * Логика поиска решений
```

```
*
* Версия 1.0
*
* Дата 2011
*
*/

var targetPrototype = null;
var currentPrototype = null;
var currentPrototypes = null;
var slotsToAnswer = null;
var answeredSlots = null;
var currentPrototypesStack = null;
var decompositionStack = null;
var currentDecompositionQuestion = '';
var decompositionMode = false; // не добавлять в ДЕКОМПОЗИЦИОННЫЙ СТЕК НОВЫЕ
СЛОТЫ, ЧТОБЫ ПРЕДОТВРАТИТЬ ЗАЦИКЛИВАНИЕ

/**
 * init
 */
$(function ()
{
    if (decompositionStack == null)
        decompositionStack = new Array();

    if (targetPrototype == null)
        drawTargetPrototypeChooser();

    $('body').bind('didSelectTargetPrototype', function(e) {
        currentPrototype = targetPrototype;
        decompositionSlots = null;
        addToTrace(currentPrototype['name'] + ' установлен как целевой
фрейм');
        readNode(targetPrototype);
    });

    $('body').bind('didAnswerSlot', function(e, slot, value) {
        answeredSlots.push({'slot':slot, 'value':value});
        addToTrace(slot['name'] + ': ' + value);

        var pulledSlot = slotsToAnswer.pull(0); // извлекаем один слот из
стека
        if (pulledSlot == null)
            findSolution();
        else
        {
            var values = new Array();
            for (var i = 0; i < currentPrototypes.length; i++)
                values.push(currentPrototypes[i]['values'][pulledSlot['id']]);
            drawSlot(pulledSlot, values);
        }
    });
});
```

```

    // После того, как находится конечное решение, проверяем необходимо ли
    // проводить декомпозицию
    $('body').bind('didFoundSolution', function(e, prototype){
        currentDecompositionQuestion = '';
        if (decompositionStack.length > 0)
        {
            var pulledSlot =
            decompositionStack.pull(decompositionStack.length - 1);
            decompositionMode = true;

            $.ajax({
                url:'api/synthesis.source.php',
                method:'GET',
                data:'action=getSlotValues&prototypeId='
            prototype['id'] + '&slotId=' + pulledSlot['id'],
                success: function(data) {
                    var values = $.parseJSON(data);
                    $.ajax({
                        url:'api/synthesis.source.php',
                        method:'GET',
                        data:'action=getPrototype&prototypeId='
            values['decomposition_id'],
                            success: function(data) {
                                currentPrototype = $.parseJSON(data);

                                addToTrace(pulledSlot['question'].replace('%s', prototype['name']));
                                currentDecompositionQuestion =
            pulledSlot['question'].replace('%s', prototype['name']);
                                readNode(currentPrototype);
                            }
                        });
                    }
                });
            }
        });

    $('body').bind('didFoundNewSolution', function(e, parentPrototype){
    });
});

/**
 * Выбор целевого перед началом работы с синтезом
 */
function drawTargetPrototypeChooser()
{
    $('<h3></h3>', {
        text: 'Выберите целевой фрейм:'
    }).appendTo('#synthesis');

    $.ajax({
        url:'api/synthesis.source.php',
        method:'GET',
        data:'action=getTargetPrototypes',
        success: function(data) {
            var prototypes = eval(data);

            if (prototypes.length == 0)
            {
                $('<h3 />', {

```

```

        text:'В базе знаний пуста'
    }).appendTo('#synthesis');
    return;
}

$('<select />', {
    'id':'targetPrototypesSelect'
}).appendTo('#synthesis');

for (var i = 0; i < prototypes.length; i++)
{
    $('<option/>', {
        val:i,
        text:prototypes[i]['name']
    }).appendTo('#targetPrototypesSelect');
}

$('<input/>', {
    'type':'button',
    'value':'Выбрать',
    'style':'width:80px; margin-left:10px',
    click:function(){
        targetPrototype
prototypes[$('#targetPrototypesSelect').val()];
        $('#synthesis').empty();
        $('#body').trigger('didSelectTargetPrototype');
    }
}).appendTo('#synthesis');
});
}

/**
 * Добавить ответный слот в список отображения трассировки
 */
function addToTrace(txt)
{
    $('#synthesisTrace').css('visibility', 'visible');
    $('<li/>', {
        text:txt
    }).appendTo('#traceConatainer');
}

/**
 * Добавить найденное решение в список отображения решений
 */
function addToAnswers(prototype)
{
    $('#solutions').css('visibility', 'visible');
    $('<li/>', {
        html:'<a href="' + prototype['url'] + '">' + prototype['name'] +
'</a>'
    }).appendTo('#solutionsContainer');
}

/**
 * диалог ответов на вопросы для заданного слота

```

```
*/
function drawSlot(slot, values)
{
    $('#synthesis').empty();

    if (currentDecompositionQuestion != '')
    {
        $('<h3></h3>', {
            text:currentDecompositionQuestion
        }).appendTo('#synthesis');
    }

    $('<h3></h3>', {
        text:slot['question']
    }).appendTo('#synthesis');

    switch (slot['type'])
    {
        case '3':
            $('<select>', {
                'id':'slotValuesSelect'
            }).appendTo('#synthesis');

            var valuesOptions = new Array();
            for (var i = 0; i < values.length; i++)
            {
                if (values[i] == null)
                    continue;

                var exists = false;
                for (var j = 0; j < valuesOptions.length; j++)
                {
                    if (values[i]['slot_value'] == valuesOptions[j])
                        exists = true;
                }

                if (!exists)
                {
                    valuesOptions.push(values[i]['slot_value']);
                    $('<option>', {
                        val:i,
                        text:values[i]['slot_value']
                    }).appendTo('#slotValuesSelect');
                }
            }

            // custom answer
            $('<option>', {
                val:-1,
                text:'свой вариант'
            }).appendTo('#slotValuesSelect');
            // listen for adding custom field answer
            $('#slotValuesSelect').change(function(){
                // addCustomField
                if ($(this).val() == -1 && $('#customField').size() ==
0)
                {
                    $('<input>', {
                        'id':'customField',
                        'type':'text',
                        css:'width:150px'
                    }).insertAfter('#slotValuesSelect');
                }
            });
        }
    }
}
```

```

    else if ($(this).val() != -1 && $('#customField').size()
> 0)
    {
        $('#customField').remove();
    }
});

$('<input/>', {
    'type':'button',
    'value':'Ответить',
    'style':'width:80px; margin-left:10px',
    click:function(){
        var answer = $('#slotValuesSelect
:selected').text();
        if ($('#slotValuesSelect').val() == -1 &&
$('#customField').size() != 0)
            answer = $('#customField').val();
        $('#body').trigger('didAnswerSlot', [slot,
answer]);
    }
}).appendTo('#synthesis');

break;
}
}

/**
 * диалог да/нет для конкретизации выбора фрейма
 */
function drawChooser()
{
    var pulledPrototype = currentPrototypesStack.pull(0);
    if (pulledPrototype == null)
    {
        createNewSolution();
        return;
    }

    $('#synthesis').empty();
    $('<h3/>', {
        text:'Это ' + pulledPrototype['name'] + '?'
    }).appendTo('#synthesis');

    $('<input/>', {
        'type':'button',
        'value':'Да',
        'style':'width:80px',
        click:function(){
            $('#synthesis').empty();
            addToAnswers(pulledPrototype);
            addToTrace('Это ' + pulledPrototype['name']);
            readNode(pulledPrototype);
        }
    }).appendTo('#synthesis');

    $('<input/>', {
        'type':'button',
        'value':'Нет',
        'style':'width:80px; margin-left:10px',
        click:function(){
            $('#synthesis').empty();

```

```

        drawChooser();
    }
    }).appendTo('#synthesis');
}

/**
 * Разбор очередного узла в дереве
 */
function readNode(prototype)
{
    var adjacentVertices;
    currentPrototype = prototype;

    // Если конечный элемент является экземпляром - то это ответ
    if (prototype['is_instance'] == 'Y')
    {
        addToTrace('Найдено решение - ' + prototype['name']);
        addToAnswers(prototype);
        $('body').trigger('didFoundSolution', prototype);
        return;
    }

    $.ajax({
        url:'api/synthesis.source.php',
        method:'GET',
        data:'action=getSlots&prototypeId='+prototype['id'],
        success:function(data){
            var slots = eval(data);
            var query = '';
            for (var i = 0; i < slots.length; i++)
                query += '&slotId[]=' + slots[i]['id'];

            $.ajax({
                url:'api/synthesis.source.php',
                method:'GET',

                data:'action=getAdjacentVerticesWithValues&prototypeId='+prototype['id'] +
                ((query == '') ? '' : query),
                success:function(data){
                    adjacentVertices = eval(data);
                    currentPrototypes = adjacentVertices;
                    answeredSlots = new Array();
                    slotsToAnswer = new Array();

                    if (slots.length > 0)
                    {
                        var i;
                        for (i = 0; i < slots.length; i++)
                        {
                            if (slots[i]['type'] == '4')
                            {
                                if (!decompositionMode)

                                decompositionStack.push(slots[i]);
                            }
                            else
                                slotsToAnswer.push(slots[i]);
                        }

                        var values = new Array();
                        for (i = 0; i < adjacentVertices.length;

```

```

i++)
    {
        if (adjacentVertices[i]['values'])
            values.push(adjacentVertices[i]['values'][slotsToAnswer[0]['id']]);
    }
    // Если есть слоты, на которые можно ответить, то
ВЫВОДИМ ДИАЛОГ
    if (slotsToAnswer.length > 0)
    {
        var pulledSlot = slotsToAnswer.pull(0); //
        ИЗВЛЕКАЕМ ОДИН СЛОТ ИЗ СТЕКА
        drawSlot(pulledSlot, values);
    }
    else
    {
        // слотов нет, выведем диалог отсева решений
        slotsToAnswer = null;
        currentPrototypesStack = currentPrototypes;
        drawChooser();
    }
    }
    });
}

/**
 * Поиск решения по результатам ответов на слоты
 */
function findSolution()
{
    $('#synthesis').empty();

    currentPrototypesStack = new Array();

    for (var i = 0; i < currentPrototypes.length; i++)
    {
        var matchingCounter = 0;
        for (var j = 0; j < answeredSlots.length; j++)
        {
            switch (answeredSlots[j]['slot']['type'])
            {
                case '3':
                    if (currentPrototypes[i]['values'] != null &&
                        currentPrototypes[i]['values'][answeredSlots[j]['slot']['id']] != null &&
                        answeredSlots[j]['value'] ==
currentPrototypes[i]['values'][answeredSlots[j]['slot']['id']]['slot_value'])
                        matchingCounter++;
                    break;
            }
        }

        if (matchingCounter == answeredSlots.length)
            currentPrototypesStack.push(currentPrototypes[i]);
    }

    if (currentPrototypesStack.length == 1)

```

```
{
    // Проверим найдено ли одно решение, если нет
    readNode(currentPrototypesStack[0]);
}
else if (currentPrototypesStack.length == 0)
{
    // ничего не найдено - новое решение
    createNewSolution();
}
else
{
    // найдено несколько решений - надо уточнить, чтобы пойти дальше
    drawChooser();
}
}

function createNewSolution()
{
    $('#synthesis').empty();
    $('<h3/>', {
        text:'Найдено новое решение'
    }).appendTo('#synthesis');

    $('<span/>', {
        text:'Введите название нового решения:'
    }).appendTo('#synthesis');

    $('<input/>', {
        'id':'solutionName',
        'type':'text',
        css:'width:100px'
    }).appendTo('#synthesis');

    $('<input/>', {
        'type':'button',
        'value':'Далее',
        click:function(){
            if ($('#solutionName').val() == '')
                alert('Не введено название нового решения');
            else
                submitForm();
        }
    }).appendTo('#synthesis');
}

function submitForm()
{
    $('<form/>', {
        'id':'solutionForm',
        'method':'POST',
        'action':'adder.php'
    }).appendTo('body');

    $('<input/>', {
        'type':'hidden',
        'name':'action',
        'value':'fill'
    }).appendTo('#solutionForm');
```

```

    $('<input/>', {
      'type':'hidden',
      'name':'solutionName',
      'value':$('#solutionName').val()
    }).appendTo('#solutionForm');

    $('<input/>', {
      'type':'hidden',
      'name':'parentId',
      'value':currentPrototype['id']
    }).appendTo('#solutionForm');

    $('<input/>', {
      'type':'hidden',
      'name':'parentId',
      'value':currentPrototype['id']
    }).appendTo('#solutionForm');

    for (var i = 0; i < answeredSlots.length; i++)
    {
      $('<input/>', {
        'type':'hidden',
        'name':'value_for_slot_'+answeredSlots[i]['slot']['id'],
        'value':answeredSlots[i]['slot']['value']
      }).appendTo('#solutionForm');
    }
    $('#solutionForm').submit();
  }

function getUrlVars()
{
  var vars = {};
  var parts = window.location.href.replace(/[?&]+([^\&]+)=([^\&]*)/gi,
function(m,key,value) {
    vars[key] = value;
  });
  return vars;
}

Array.prototype.pull = function (arg) {
  // pulls an element out of an array, returns the single element.
  // array.length becomes one less.
  // may pass the index or value of element to remove
  var oEl, tmp;

  if (typeof arg == "number") {
    oEl = this[arg];
    tmp = this.slice(0,arg).concat(this.slice(arg + 1));
  }
  else if (typeof arg == "string") {
    for (var x = 0; x < this.length; x++) {
      if (this[x] == arg) {
        oEl = this[x];
        tmp = this.slice(0,x).concat(this.slice(x + 1));
        break;
      }
    }
  }
  if (typeof oEl != "undefined") {

```

```
        this.length = 0;
        for (var x = 0; x < tmp.length; x++)
            this[x] = tmp[x];
        return oEl;
    }
    else return null;
}

// Compute the intersection of n arrays
Array.prototype.intersect =
function() {
    if (!arguments.length)
        return [];
    var a1 = this;
    var a = a2 = null;
    var n = 0;
    while(n < arguments.length) {
        a = [];
        a2 = arguments[n];
        var l = a1.length;
        var l2 = a2.length;
        for(var i=0; i<l; i++) {
            for(var j=0; j<l2; j++) {
                if (a1[i] === a2[j])
                    a.push(a1[i]);
            }
        }
        a1 = a;
        n++;
    }
    return a.unique();
};
```