

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и Информационных технологий
институт
Информационные системы
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой ИС

_____ Л.С. Троценко
подпись, дата инициалы, фамилия

«13» июня 2018г.

БАКАЛАВАРСКАЯ РАБОТА

09.03.02 Информационные системы и технологии

Сравнительный анализ моделей представления знаний в интеллектуальных
системах

Руководитель

подпись, дата

П.П. Дьячук
инициалы, фамилия

Студент КИ14-13Б 031402261
номер группы, зачетной книжки

подпись, дата

Д.Н. Кулик
инициалы, фамилия

Нормоконтролер

подпись, дата

Ю.В. Шмагрис
инициалы, фамилия

Красноярск 2018

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и Информационных технологий
институт
Информационные системы
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой ИС

_____ С.А. Виденин
подпись, дата инициалы, фамилия

«2» марта 2018г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы

Студенту: Кулику Даниле Николаевичу

Группа: КИ14-13Б Направление 09.03.02 «Информационные системы и технологии»

Тема выпускной квалификационной работы: Сравнительный анализ моделей представления знаний в интеллектуальных системах

Утверждена приказом по университету № 4896/с от 05.04.2018

Руководитель ВКР: П. П. Дьячук, кандидат педагогических наук, доцент кафедры «Информационные системы», ИКИТ СФУ.

Исходные данные для ВКР: Список требований к выполняемой работе, методические указания научного руководителя, учебные пособия

Перечень разделов ВКР: Представление знаний в интеллектуальных системах, сравнительный анализ моделей представления знаний, построение базы знаний на основе продукционно-фреймовой модели

Перечень графического материала: Презентация, выполненная в Microsoft Office PowerPoint 2016.

Руководитель ВКР

подпись, дата

П.П. Дьячук
инициалы, фамилия

Задание принял к исполнению

подпись, дата

Д.Н. Кулик
инициалы, фамилия

«2» марта 2018 г.

РЕФЕРАТ

Выпускная квалификационная работа по теме «Сравнительный анализ моделей представления знаний в интеллектуальных системах» содержит 55 страниц текстового документа, 19 иллюстраций, 17 использованных источника.

МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ, ПРОДУКЦИОННАЯ МОДЕЛЬ, ФРЕЙМОВАЯ МОДЕЛЬ, ОНТОЛОГИЯ, БАЗА ЗНАНИЙ.

Объектом исследования являются неформальные модели представления знаний в интеллектуальных системах.

Целью бакалаврской работы является исследование и проведение сравнительного анализа моделей представления знаний, а также создание базы знаний на основе продукционно – фреймовой модели.

Основные задачи:

- выявить особенности представления и использования знаний в интеллектуальных системах.
- изучить основные характеристики и основы построения моделей представления знаний.
- провести сравнительный анализ моделей представления знаний.
- построить базу знаний на основе продукционно – фреймовой модели.

Основные результаты:

- выявлены особенности представления и использования знаний в интеллектуальных системах
- изучены о основные характеристики и основы построения моделей для представления знаний.
- проведен сравнительный анализ моделей с другими моделями, используемыми для представления знаний
- построена база знаний на основе продукционно – фреймовой модели

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Представление знаний в интеллектуальных системах.....	5
1.1 Интеллектуальные системы	5
1.2 Знания и их представление	9
2 Сравнительный анализ моделей представления знаний	17
2.1 Основные характеристики и основы построения моделей представления знаний	17
2.1.1 Продукционная модель.....	17
2.1.2 Фреймовая модель.....	21
2.1.3 Онтология, как разновидность сетевой модели.....	26
2.2 Сравнение и выявление особенностей моделей представления знаний	33
3 Построение базы знаний на основе продукционно-фреймовой модели	42
Заключение	53
Список использованных источников	54

ВВЕДЕНИЕ

Специалисты в области инженерии знаний ведут активные поиски эффективных сочетаний моделей представления знаний и разработки интегрированных моделей представления знаний, которые смогли бы объединить и взаимно дополнить различные средства и методы представления знаний и в полной мере реализовать свойства знаний.

Поэтому можно сделать вывод о том, что исследования в этой области очень актуальны на сегодняшний день, поскольку необходимо разрабатывать новые подходы к изучению моделей представления знаний в ИИС

В соответствии с поставленной целью в работе решались следующие задачи:

- выявление особенностей представления и использования знаний в интеллектуальных системах.
- комплексное представление методологии использования исследуемых моделей для представления знаний.
- проведение сравнительного анализа неформальных моделей с другими моделями, используемыми для представления знаний и выявление сильных и слабых сторон исследуемых моделей.
- построение базы знаний на основе продукционно – фреймовой модели.

1 Представление знаний в интеллектуальных системах

1.1 Интеллектуальные системы

Прежде чем характеризовать интеллектуальные информационные системы надо определить, что мы понимаем под словом «интеллект». Если проанализировать разнообразные определения, то можно охарактеризовать интеллект как способность мозга (и не только человеческого) планомерно вести организованную и контролируемую познавательную деятельность, приобретать и запоминать знания и на основе приобретенных знаний и существующего опыта эффективно решать разнообразные задачи.

Применительно к ИС понятие «интеллект» означает способность системы самообучаться, создавать программы на основе заложенных знаний и самостоятельно решать задачи в самых разнообразных ПО.

Поэтому для современных ИИС характерен достаточно высокий уровень автоматизации (как процесса обработки информации, так и процесса принятия решений) и применение наиболее наукоемких технологий.

Алгоритм решения задач, ставящихся перед ИИС, строится на сложных рассуждениях, логических обобщениях и выводах, требующих достаточно высокой профессиональной квалификации и изобретательности.

Чтобы эффективно решать поставленные задачи ИИС должна обладать следующими свойствами:

- способностью решать задачи даже в том случае, когда зависимость между основными показателями установлена не полностью или вовсе не установлена (в пределах какого-либо класса);
- способностью работать с данными, изменяющимися в процессе обработки;
- способностью к саморазвитию системы и самостоятельному извлечению знаний из накопленного опыта конкретных ситуаций.

– возможностью использовать информацию, которая не хранится в явном виде, а выводится из имеющихся в БД.

При этом ИИС должна также уметь работать с недостаточно полной или неточной информацией и принимать так называемые «прозрачные решения», т.е. объяснение решений должно быть понятно не только профессионалу, но и пользователю, не обладающему достаточными знаниями в той или иной ПО.

Классифицировать ИИС можно различными способами. Вот несколько классификаций, предложенных А.В Остроух¹.



Рисунок 1 – Классификация ИИС по критериям интеллектуальности

Можно также классифицировать ИИС в зависимости от типа решаемой задачи. В этой классификации выделяются системы управления и справочные системы, системы компьютерной лингвистики, системы распознавания,

¹ Остроух А.В. Интеллектуальные системы / А.В. Остроух. – Красноярск: Научно-инновационный центр, 2015. Стр.9-14

игровые системы и системы создания интеллектуальных информационных систем.



Рисунок 2 – Классификация ИИС в зависимости от решаемой задачи

В зависимости от используемых методов ИИС могут быть разделены на жесткие, мягкие и гибридные.



Рисунок 3 – Классификация ИИС в зависимости от используемых методов

Применение ИИС наиболее эффективно в том случае, когда решаются слабо структурированные задачи, в которых при принятии решений должны учитываться слабо формализуемые факторы, например экономические, политические или социальные².

Из всех понятий, которыми оперирует ИС, ключевым является понятие «знания». Д.А. Поспелов характеризовал саму область ИИ как "науку о знаниях, о том, как их добывать, представлять в искусственных системах, перерабатывать внутри системы и использовать для решения задач"³. Знания в области ИИ – это закономерности предметной области (принципы, связи, законы), полученные в результате профессионального опыта и практической деятельности, дающие возможность специалистам ставить и решать задачи в этой области⁴.

Но что же мы понимаем под «знаниями»? Рассмотрению этого вопроса посвящена следующая часть данной работы.

² Козлов А. Н. Интеллектуальные информационные системы: Учебник /А. Н. Козлов. Пермь: ФГБОУ ВПО Пермская ГСХА, 2013. С. – 11.

³ Поспелов Д. А. Ситуационное управление: теория и практика / Д. А. Поспелов. Москва: Наука, 1986, С. 7.

⁴ Гаврилова Т.А. Базы знаний интеллектуальных систем / Т. А. Гаврилова. СПб.: Питер, 2001, С. 384.

1.2 Знания и их представление

Понятие «знания» не имеет однозначного определения. Если говорить о «человеческих» знаниях, то они приобретаются людьми посредством познания мира, при целенаправленном обучении, путем самообразования, знания – это итог осмысления людьми их практического жизненного опыта. Поэтому можно предложить следующее определение: знания – это устойчивые представления, понятия и теории, сформировавшиеся в мозгу человека в результате познания окружающего мира, проверенные практикой и подтвержденные логикой, позволяющие неоднократно использовать их для решения разнообразных задач.

Понятие «знания» не идентично понятию «данные», хотя они и очень тесно связаны. Данные представляют из себя формализованную информацию, которая обрабатывается, хранится и передается средствами автоматизации профессиональной деятельности. Знания задаются набором конкретных фактов и свойств (которые соответствуют определенному понятию и касаются ПО), либо свойствами, которые соответствуют конкретному понятию и схемой функциональных зависимостей между атрибутами. Данные – это, например, показатели температуры, воздуха, воды, давления, скорости, геометрические параметры - длина, ширина, высота, громкость звука, сила тока, напряжение. Для того, чтобы стать знаниями данные должны приобрести определенные свойства, обусловленные сложной природой знаний и отличающие их от данных.

Первое свойство знаний – это *интерпретируемость*. Это означает, что каждая информационная единица (ИЕ) должна иметь уникальное, свойственное только ей имя, по которому ИС будет вести поиск и отвечать на запросы, упоминающие этот идентификатор. Если у данных, хранящиеся в памяти, не будет уникального имени, то возможности их идентификации системой не будет и сделать это сможет только программа.

Следующим важным свойством знаний является *структурированность*, означающая возможность рекурсивного вложения одних ИЕ в другие, т.е. каждая ИЕ может включаться в состав другой ИЕ (либо несколько ИЕ могут быть объединены в более крупную ИЕ) и, в то же время, из каждой ИЕ могут быть вычленены ее составляющие. Однако, структура ИЕ должна быть достаточно гибкой, т.е. такой, чтобы между отдельными ИЕ можно было установить различные взаимоотношения (например, «часть - целое», «элемент - класс»).

Такое свойство, как *связность* означает, что в каждой информационной базе между ИЕ должны существовать различные взаимосвязи, отражающие отношения между отдельными ИЕ. После того, как в памяти интеллектуальной системы выстраивается структура отношений, фрагменты этой структуры получают возможность самостоятельно определять новые ИЕ. Среди отношений можно выделить отношения *структуризации* (с их помощью которых задаются иерархии ИЕ), *функциональные* отношения (сообщающие информацию, которая позволяет вычислить одни ИЕ через другие), *каузальные* отношения (устанавливают причинно-следственные зависимости), а также *семантические* отношения (устанавливают связи и зависимости отдельных ИЕ в едином семантическом пространстве). Между ИЕ могут существовать также связи, которые определяют порядок выбора ИЕ из памяти, указывают на несовместимость ИЕ в одном описании и т.п.

Еще одним немаловажным свойством знаний является их *активность*. В ЭВМ процессы инициируются командами, а данные используются этими командами лишь в случае необходимости. В ИИС любые действия инициируются знаниями, хранящимися в системе, т.е., выполнение программ в ИС активизируется текущим состоянием информационной базы, а источником активности становится появление в базе новых фактов, описаний событий, или установление связей⁵.

⁵ Поспелов Д. А. Искусственный интеллект. - В 3-х книгах. Кн.2. Модели и методы: Справочник / Д.А. Поспелова – Москва: Радио и связь, 1990, С.9

Функциональная целостность также одно из важных свойств знаний, она подразумевает возможность выбора желаемого результата, времени и средств получения результата, а также средств анализа достаточности полученного результата.

Проанализировав свойства знаний можно прийти к следующему выводу - главным свойством знаний, отличающих их от данных, является то, что знания дают возможность получать новые знания.

Применительно к сфере ИИС можно определить понятие «знания» следующим образом: знания — это особый вид информации, совокупность структурированных и формализованных теоретических и эмпирических закономерностей, фактов, положений, принципов, законов различных ПО, представленных в различных формах, обладающих определенными свойствами, связанных разнообразными специфическими отношениями, полученных профессионалами в результате обобщения жизненного опыта и позволяющих решать прикладные задачи в конкретной ПО.

Классификация знания проводится по разным признакам:

– *по способу существования*: фактические и эвристические (т.е., те, которые заложены изначально в ИС, но не имеют статуса абсолютной истинности в конкретной ПО, а также те, которые ИС накапливает в процессе функционирования);

– *по способу использования*: факты, т.е., знания типа «X – это Y» («арбуз — это ягода»), правила в виде знаний для принятия решений («ЕСЛИ А, ТО В») и метазнания, указывающие способы использования уже имеющихся знаний;

– *по форме представления* знания разделяют на *декларативные* (занесенные в память ИС структурированные данные, отражающие свойства, факты, количественные и качественные характеристики ПО); *процедурные* (алгоритмы, которые задают закономерности конкретной ПО, и описывают процедуры обработки фактов и последовательность действий для решения

задач); *понятийные* (т.е., определения понятий, их свойств, описание взаимосвязей и отношений между ними);

– *по отношению к классам объектов и отдельным объектам: интенциональные* знания — знания, относящиеся к определенному классу объектов или характеризующие этот класс и *экстенциональные* знания, т.е.. имеющие отношение к определенному объекту из конкретного класса;

– *в зависимости от представляемой ПО* знания можно разделить на *универсальные* - общие знания о мировых законах, которые можно применить к любой ПО (например, о законах времени и пространства реального физического мира, о закономерностях причинно-следственного характера, логического характера) и *специальные*, характерные только для одной конкретной ПО;

В научной литературе выделяют также *«жесткие»* знания (с их помощью можно получить точные рекомендации при заданных начальных условиях) и *«мягкие»* (т.е., знания, которые допускают вариантность рекомендаций и множественность решений); *морфологические и синтаксические* знания (знания о том, какие правила применяются при построении структуры описываемого явления, объекта вне зависимости от их смысла и содержания), *семантические* знания (раскрывают смысл и значение описываемого явления или объекта) и др.

С учетом особенностей использования знаний в ИИС они могут быть классифицированы так, как показано на рисунке 4.



Рисунок 4 – Классификация знаний в интеллектуальных системах.

Под представлением знаний понимают выбор способов использования знаний ИС, которые дают возможность произвести логический вывод, основываясь на имеющихся в базе знаниях т.е. делает возможным создавать знания из знаний.

Фактически, представление знаний в ИИС – это формализация и структурирование знаний эксперта, специализирующегося в конкретной ПО с использованием определенного языка. С помощью формализации и структурирования отражаются главные характерные признаки представляемых знаний, они необходимы для облегчения решения поставленных задач⁶.

⁶ Белоус Е. С. Современные модели представления знаний в обучающих системах / Е. С. Белоус, В. А. Кудинов, М. Э. Желнин // Ученые записки. Электронный научный журнал Курского государственного университета. – 2010. С. 3.

Экспертные знания в различных ПО обычно разделяют на *формализованные*, т.е., образующие формальную систему истинные, универсальные, но достаточно точные знания (суждения), которые могут быть сформулированы в виде формул, моделей, таблиц, законов, алгоритмов, использующихся при разработке программ и *неформализованные*.

Неформализованные знания, как правило представляют из себя описательную, словесную совокупность эмпирических правил и суждений, не достаточно конкретных, они основываются на интуиции эксперта и его опыте работы. Неформализованные знания не могут быть представлены в числовой форме, они требуют символьного представления, алгоритмическое представление неформализованных знаний либо невозможно, либо крайне затруднено из-за ограниченности ресурсов (например, времени или оперативной памяти), цели задачи при представлении неформализованных знаний нельзя выразить в терминах точно определенной целевой функции.

На сегодняшний день существует много моделей представления знаний в ИС, решающих проблему формализации знаний. Все они могут быть условно разделены на *логические*, или формальные модели, основывающиеся на теоретическом подходе, и *эвристические*, или формализованные, базирующиеся на эвристическом подходе.



Рисунок 5 – Модели представления знаний

В основе всех *логических моделей* находится понятие формальной системы (теории), при построении логических моделей факты и утверждения, необходимые для решения задачи, представляются в виде формул в логической системе. Знания представляются совокупностью формул, а получение новых знаний происходит в результате реализации логического вывода⁷.

Формула в логических моделях задается четырьмя символами:

(T, P, A, R),

где T – алфавит теории, счетное множество терминальных элементов, формирующих все выражения формальной системы,

⁷ Головчинер М.Н. Введение в системы знаний: Курс лекций /Под ред. М. Н. головчинер. – Томск, 2011 С.21.

P – синтаксические правила, с помощью которых из T образуют синтаксически правильные совокупности,

A – множество аксиом формальной системы (априорно считаются истинными),

R – конечное множество правил вывода, которые позволяют получать из одних синтаксически правильных выражений другие.

Применение логических моделей позволяет хранить в БЗ только знания, образующие множество A , поскольку все остальные знания могут быть получены посредством реализации ЛВ.

Средства выражения отношений между отдельными единицами знаний достаточно скудны и представлены синтаксическими правилами используемой формальной системы.

Модели, основанные на эвристическом подходе, во многом превосходят формальные модели и по возможностям адекватного представления ПО, и по эффективности используемых правил вывода. Для эвристических моделей присущ разнообразный набор средств, которые передают характерные особенности определенной ПО. Поэтому эвристические модели представляют больший интерес для исследования и анализа.

В следующей главе будут рассмотрены три наиболее часто используемые модели представления знаний, основанные на эвристическом подходе.

2 Сравнительный анализ моделей представления знаний

2.1 Основные характеристики и построение моделей представления знаний

2.1.1 Продукционная модель представления знаний

Продукционные модели на сегодняшний день можно считать наиболее распространенными моделями представления знаний. Понятия «продукция» и «продукционная система» впервые были введены Е. Постом в его работах по теории алгоритмов в 1943г. Однако Пост в качестве продукции рассматривал только ту часть, которую сейчас называют ядром ($A \rightarrow B$, ЕСЛИ A , ТО B)⁸.

Продукционные модели достаточно близки логическим моделям, в них достаточно эффективно реализуется ЛВ, но представление знаний происходит более наглядно, чем в классических логических моделях. Главное отличие от логических моделей состоит в отсутствии жестких ограничений и это позволяет изменять интерпретацию элементов продукции.

Продукционная модель – это модель, основанная на правилах и позволяющая представить знание в виде предложений типа: «ЕСЛИ (условие, или *антецедент*), ТО (действие, или *консенквент*)». При этом условием является определенное предложение-образец, по которому производится поиск в базе знаний, а действием является набор определенных действий, которые выполняются при успешном результате поиска. Новые данные, которые могут быть получены в результате вычислений (или взаимодействия с пользователем) генерируются и также добавляются в базу.

Каждое правило состоит из одной или нескольких пар «атрибут-значение». Истинность пар «атрибут-значение», хранящихся в рабочей памяти систем, основанных на продукционных моделях, устанавливается в процессе

⁸ Поспелов Д. А. Искусственный интеллект: в 3-х кн. Кн.2. Модели и методы: Справочник / Под ред. Д.А. Поспелова – Москва: Радио и связь, 1990. – 304 с.

решения определенной задачи к определенному моменту времени. По мере срабатывания правил в процессе решения содержимое рабочей памяти может изменяться. Правило срабатывает, если факты, содержащиеся в рабочей памяти, при сопоставлении совпадают с антецедентом анализируемого. В таком случае заключение сработавшего правила заносится в рабочую память. Актуализация и выбор продукции происходит с помощью процедур управления продукциями, заданных в системе.

Объем фактов, содержащихся в рабочей памяти, может меняться в итоге реализации ЛВ – как правило, он увеличивается (уменьшится он может в том случае, если в результате действия правила факт удаляется из памяти). В процессе реализации ЛВ каждое правило из базы может сработать только один раз.

Набор продукционных правил, содержащий необходимые для проведения ЛВ знания, образует продукционную БЗ. Как правило, данные представляются в следующем виде «объект – атрибут – значение». Под объектом понимается некий объект ПО, под атрибутом — его характеристика.

В общем виде продукция может быть представлена следующим выражением:

$$(i); Q; P; A \rightarrow B; N,$$

где i – имя (идентификатор) продукции. С его помощью конкретная продукция может быть идентифицирована из всего множества продукций. Идентификатор продукции может быть выражен как в словесной форме, так и в виде порядкового номера продукции.

Q – характеризует область применения продукции. Разделении всего множества знаний на отдельные области экономит время при поиске решения задач.

Р – это условие применимости ядра продукции. Как правило **Р** является логическим выражением. Ядро продукции активизируется только при истинном **Р**. Если **Р** ложно, то ядро продукции не может быть использовано.

Ядро: $A \rightarrow B$ – это основной элемент продукции. Обычное ядро имеет вид предложения: «ЕСЛИ **A**, ТО **B**», однако могут быть и достаточно сложные конструкции, с допущением альтернативного выбора. В этом случае ядро может приобретать вид: «ЕСЛИ **A** ТО **B**₁, ИНАЧЕ **B**₂». Нужно учесть, что логическая причинно-следственная связь **A** и **B** возможна только при истинности **A** (если же **A** не является истинным, то о **B** нельзя ничего сказать, т.к. из ложного условия может следовать все, что угодно, хотя для импликации принято считать, что следует истина). Кроме связи типа «причина – следствие», между **A** и **B** могут существовать связи «действие -действие», «знания – действие» и другие.

N – описывает постусловия продукции, активизирующиеся после реализации ядра. Фактически, постусловия – это либо алгоритм действий, подлежащих выполнению после подтверждения истинности **B**, либо описание изменений, которые должны быть внесены в БЗ после реализации продукции. Выполнение **N** может происходить сразу же после реализации продукции или спустя определенное время.

Логический вывод представляет из себя поиск ответа на вопрос пользователя, заданный ИС. Для поиска ответа ИС анализирует накопленные на текущий момент знания и выбирает наиболее подходящий элемент из списка разрешенных. Предусматриваются и невозможность найти ответ на заданный вопрос. Поэтому, результатом ЛВ может быть как определенное значение из списка разрешенных значений, так и сообщение о невозможности получить ответ.

Логический вывод, реализуемый ИС, происходит следующим образом. У пользователя запрашивается идентификатор (имя объекта), для которого необходимо установить значение, а также известные пользователю знания о ПО, помимо тех, что заложены в БЗ. Используя сведения, полученные от

пользователя, ИС из хранящихся в БЗ знаний выводит новые знания. Этот процесс повторяется несколько раз, до тех пор, пока ИС не сможет найти ответ на поставленный вопрос, т.е. присвоить объекту какое-либо из разрешенных значений или значение «неизвестно». Полный сеанс работы ИС называется консультацией.

Существуют производственные системы с прямым (нисходящим) выводом, управляемым данными, и с обратным (восходящим) выводом, управляемым целями.

В системах с прямым выводом поиск ведется от фактов к заключениям. Условия **A**, включающие известные факты, проверяются и активизируются те продукции, для которых **A** истинно. После этого в рабочую память заносятся промежуточные заключения **B'**, которые в дальнейшем выступают как дополнительные факты для **A'** и так до тех пор, пока не будет получено итоговое заключение **B**.

В системах с обратным выводом первоначально выдвигается определенная гипотеза **B**, после чего начинается поиск промежуточных фактов **A'**, которые эту гипотезу подтверждают. Затем в рабочую память заносятся промежуточные факты **A'**, являющиеся промежуточными гипотезами **B'**. Если принятая гипотеза приводит к известным фактам **A**, то она считается итоговым заключением.

Существуют также системы с двунаправленными выводами.

Прямой вывод обычно используется в том случае, когда все (или большинство) исходные данные заданы в постановке задачи, когда при большом количестве целей имеется дефицит способов использования фактов, или в том случае, когда существуют проблемы в формировании цели или гипотезы.

Обратный вывод используется:

– если цель поиска явно присутствует в постановке задачи или может быть легко сформулирована (например, при доказательстве математических теорем);

– если имеется большое количество правил, которые на основе полученных фактов продуцируют постоянно возрастающее число заключений и целей (благодаря своевременному отбору целей обратный вывод позволяет избежать широкого разветвления, что повышает эффективность поиска);

– если в задаче не приводятся исходные данные, однако априори они должны быть известны решателю. Например, выбирается предварительный медицинский диагноз (гипотеза), а потом под него подбираются симптомы (факты).

При выполнении условия применимости одновременно для нескольких продукции возникает проблема выбора продукции или их группы (в случае возможности параллельной обработки), которая должна быть активизирована для эффективного и быстрого достижения поставленной цели. Решение этой проблемы обеспечивает система активизации продукций.

2.1.2 Фреймовая модель представления знаний.

В основе фреймовой модели представления знаний лежит концепция М. Минского, одного из основателей лаборатории ИИ. Фактически фреймовая модель является систематизированной моделью человеческого сознания и памяти. Использовать данный вид модели для представления знаний М. Минский предложил в 1979 г.

Во фреймовых моделях связи между элементами модели выражаются через *бинарные отношения*. Фрейм – это структура, представляющая (описывающая) определенный объект в виде атрибутов и их значений. Вся информация при этом содержится в слотах – отдельных единицах модели. Фреймы могут быть классифицированы, как фреймы-структуры, фреймы-операции, фреймы-ситуации, фреймы-сценарии, фреймы-роли⁹.

Структура фрейма может быть представлена в следующем виде:

⁹ Головчинер М.Н. Введение в системы знаний: Курс лекций /Под ред. М. Н. головчинер. – Томск, 2011 С.21.

(ИМЯ ФРЕЙМА:

(имя 1-го слота: значение 1-го слота),

(имя 2-го слота: значение 2-го слота),

(имя N-го слота: значение N-го слота).

Каждый фрейм имеет уникальный идентификатор (имя) описываемого понятия или явления. Описания основных элементов называются *слотами*. Имена слотов, включенных в один фрейм, также должны быть уникальными. Имя слота должно быть уникальным в рамках одного фрейма. Имя слота обычно выступает лишь в качестве идентификатора слота, хотя в редких случаях иногда оно может иметь особое значение. Слоты задают условия, подлежащие выполнению, если между значениями найдено соответствие.

Например, имя слота DDESENDANTS указывает на дочерний фрейм, FINEDBY на пользователя, определяющего фрейм, DEFINEDON на дату определения фрейма, MODIFIEDON на дату модификации фрейма и т.п.

Есть имена слотов, которые используются для представления структурированных объектов, например HASPART, RELATIONS и другие. Это так называемые «системные» слоты. Они применяются при редактировании БЗ и для управления выводом.

Особое значение имеет слот, называемый «демон» - так называют автоматическую присоединенную процедуру, инициируемую при обращении к данному слоту.

Слот «демон» может быть нескольких видов:

- Демон типа IF-NEEDED, вычисляющий неопределенные значения либо производящий поиск значения в слотах других фреймов;
- Запуск демона типа IF-ADDED происходит при изменении значения слота;
- Для удаления значения слота существует демон типа IF-REMOVED.

Присоединяемая процедура также иногда используется в качестве значения слота. Причем правило, в соответствии с которым происходит запуск

присоединяемой процедуры может быть получено даже из другого фрейма. Поэтому присоединяемая процедура может рассматриваться в качестве процедуры управления выводом. Благодаря этой особенности существует возможность использования фреймов в продукционных моделях.

Слот может содержать как конкретные характеристики объектов, так и процедуры, и алгоритмы вычисления значений этих характеристик с использованием значений других слотов, отсылки к конкретным слотам других фреймов, или ссылки на другой фрейм. Значением слота может стать информация любого вида, используемая в качестве описания объектов ПО или необходимая для определения их характеристик.

Сеть фреймов образуется в том случае, когда имя одного фрейма становится значением слота другого фрейма (при этом слот отображает существующие между фреймами отношения).

Слот фрейма – экземпляра может получить значение одним из следующих способов:

- по умолчанию (от одноименного слота фрейма-прототипа);
- наследуя свойства от фрейма, указанного в слоте АКО, значением которого является имя фрейма – родителя;
- по формуле;
- через присоединенную процедуру;
- при диалоге с пользователем;
- из внешней БД.

Наследование свойств других объектов является одной из главных особенностей фреймовой модели представления знаний. Возможность наследования очень важна в случаях, когда для представления новых состояний объектов используются базовые фреймы.

Наследование свойств происходит при помощи АКО-связей. Слот АКО содержит имя расположенного на более высокой иерархической ступени фрейма, от которого наследуются значения слотов.

В иерархических фреймовых системах существуют так называемые *указатели наследования* (ссылки), применяемые при реализации процедуры наследования одноименными слотами определенного фрейма значений из слотов фрейма верхнего уровня.

Указатели могут иметь разнообразные обозначения:

- Unique – указывает на то, что наследования значения не происходит;
- Same – указывает на наследование значения;
- Range – указывает, что значения слота находятся в рамках значений одноименного слота фрейма, расположенного на верхнем уровне.
- Override – показывает, что значение наследуется при отсутствии значения текущего слота, но если значение определено, то оно будет уникальным.

Фрагмент иерархической БЗ, использующей фреймовую модель представления знаний, показан на рисунке 6. Три из четырех фреймов наследуют определенные свойства «вышестоящих» фреймов, которые представляют модели верхних уровней иерархии. Значения отдельных слотов этих трех фреймов наследуются из других слотов (слоты «АКО»), показаны также слоты, поименованные символами С1, ... , С5. В разных фреймах эти символы означают разные признаки понятий, которые соответствуют разным фреймам. Например, для фрейма «Прозаическое литературное произведение» признаки С1, С2 и С3 будут означать жанры литературного произведения («роман», «повесть», «рассказ»), а для фрейма «Война и мир» - соответственно признаки «автор», «год написания» и «год издания».

Наследуемые значения также различны. Например, фрейм «Роман» наследует от вышестоящего фрейма его название в качестве значения своего признака («Ако»), отвечающего на вопрос: что такое «роман», как литературная категория. Фрейм «Война и мир» наследует от вышестоящего фрейма уже два значения, одно из которых говорит о том, что «Война и мир» – это роман, а другое – содержит характеристику «эпический».

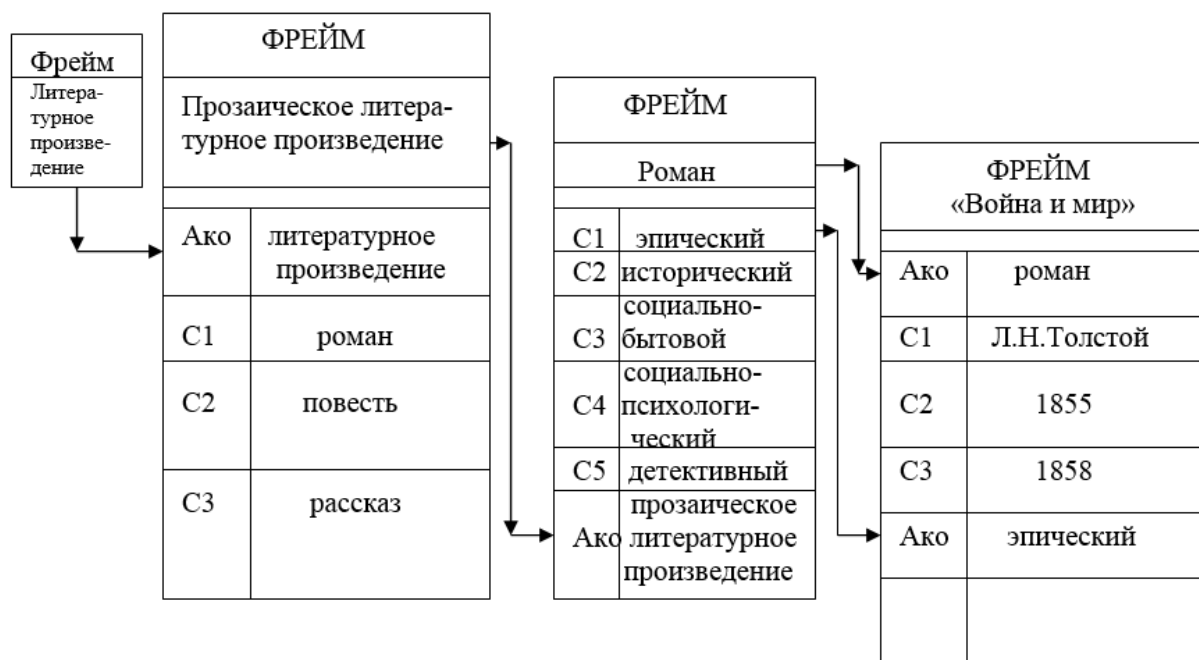


Рисунок 6 – Фрагмент иерархической структуры с фреймовым представлением знаний

Декларативный компонент БЗ, основанной на фреймовой структуре разделен на две части: основные, базовые знания об объектах ПО и их отношениях (находятся во фреймах – прототипах), и текущие знания о реальных объектах представляемой ПО (располагаются во фреймах-экземплярах).

Процедурная компонента БЗ, основанной на фреймовой модели представления знаний, состоит из совокупности демонов, процедуры наследования и присоединенных процедур. При этом демоны и присоединенные процедур выполняют функцию знаний и средств управления ЛВ. Поэтому фреймовая модель позволяет представлять одновременно как декларативные, так и процедурные знания, а тройной механизм управления выводом - процедуры наследования, демоны и присоединенные процедуры обеспечивают любого из известных механизмов ЛВ (хотя при этом возникает проблема – как отделить декларативные знания от процедурных).

2.1.3 Онтология, как разновидность сетевой модели

В информационные науки понятие онтологии пришло из философии, где оно обозначает один из разделов, изучающий принципы, структуру, закономерности такой философской категории, как «бытие». В компьютерных технологиях оно стало применяться относительно недавно, активная разработка онтологий и их применение начались в конце девяностых годов. Связано это было с резким увеличением объемов Интернет-ресурсов, накоплением большого количества знаний, увеличением числа непрофессиональных пользователей. Разработчикам необходимо было решить проблему систематизации и структурного представления знаний о разнообразных ПО, возможности обработки текстов на естественном для пользователей языке, а также облегчить извлечение неявных знаний из хранилищ. Комплексным решением этих проблем и стала разработка онтологий.

В научной литературе можно найти несколько различных толкований понятия онтология».

Под онтологией одни исследователи понимают неформальную концептуальную систему, служащую основой для БЗ, другие определяют онтологию, как особую семантическую систему, формально представляющую термины некоторой ПО, а также правила их применения. Под онтологией также подразумевают словарь, который используется логической теорией. Но чаще всего в сфере ИИС используется определение, данное Томасом Грубером: «Онтология – это спецификация концептуализации¹⁰».

По мнению Т. Грубера, для полноценного развития систем ИИ необходимо создать определенный механизм, позволяющий БЗ

¹⁰ Митрофанова О. А. Онтологии как системы хранения знаний / О. А. Митрофанова, Н. С. Константинова // Всероссийский конкурсный отбор обзорно – аналитических статей по приоритетному направлению «Информационно – телекоммуникационные системы». 2008. С. 54.

взаимодействовать и обмениваться заложенными в них знаниями друг с другом при решении задач. Для такого взаимодействия необходимо *описание* знаний, понятий, связывающих их отношений, т.е. *спецификация*. При этом такая спецификация должна иметь настолько формальный вид, чтобы быть понятой любой БЗ, независимо от ПО, быть легко понята человеком и должен быть известен язык описания. Под концептуализацией Т. Грубер подразумевал составление такого описания.

Спецификацию Томас Грубер предлагал производить двумя способами: классическим описанием знаний на языке логики предикатов и других формализмов, и в форме онтологии, представляющей иерархию объектов, которые связаны отношениями обобщения¹¹.

Если исходить из определения Т.Грубера, то под онтологией следует понимать иерархическое описание декларативных знаний определенной ПО. При этом одна ИС может представлять несколько спецификаций, которые будут соответствовать разным областям знаний, выступая в роли хранилища библиотек онтологий и обеспечивая обмен между ними. К спецификации, понятной для человека, присоединяется описание в классическом формализованном виде. Таким образом, описание знаний ПО происходит дважды. Развитие современных языков дает возможность объединить эти описания.

Таким образом, можно сформулировать следующее определение: онтология – это базирующееся на концептуализации формальное представление связанных отношением иерархии декларативных знаний определенной ПО.

Такое определение позволяет выделить основные компоненты онтологии: таксономия терминов ПО, описание их смысла и правила их использования и обработки.

¹¹ Скоркин О. А. Онтология в проектировании систем информационных технологий / О. А. Скоркин // Вопросы социальной теории. – т.6. – вып.1-2. – С. 239.

Поэтому в простейшем виде модель онтологии может быть задана тройкой:

$$O = (X, R, F),$$

где X – конечное не пустое множество концептов (понятий) ПО, представленной онтологией,

R – конечное не пустое множество отношений между концептами,

F — конечное не пустое множество функций интерпретации, заданных на концептах и(или) отношениях.

В данной модели для всех компонентов существует естественное ограничение – конечность и не пустота.

Самым простым примером онтологии можно считать классификации в той или иной ПО, например, таблица Д.И.Менделеева.

Рассматривая подробную структуру онтологии, можно выделить следующие ее составные элементы: понятия; отношения; аксиомы; отдельные экземпляры, словарь, свойства(атрибуты) и соответствие между терминами и понятиями.

Ядром онтологии являются понятия с общими свойствами (атрибутами), объединяющиеся в классы. Классы содержат описания понятий и их атрибутов. Классы могут подразделяться на подклассы, наследующие атрибуты родительских классов, а могут состоять и из единичных экземпляров. Классы – это общие категории, которые образуют иерархическую систему (системы) и описывают понятия определенной ПО.

Отношения связывают между собой понятия и классы и описывают их. Основными связями (вертикальными) между классами являются отношения порядка, образующие иерархию (например, таксономические отношения класс-подкласс). Основные связи и понятия образуют скелет онтологии – ациклический граф. Граф онтологии принято считать частным случаем

семантической сети. На рисунке 7 показана схема онтологии ресурсов (электронных документов) и их атрибуты¹².

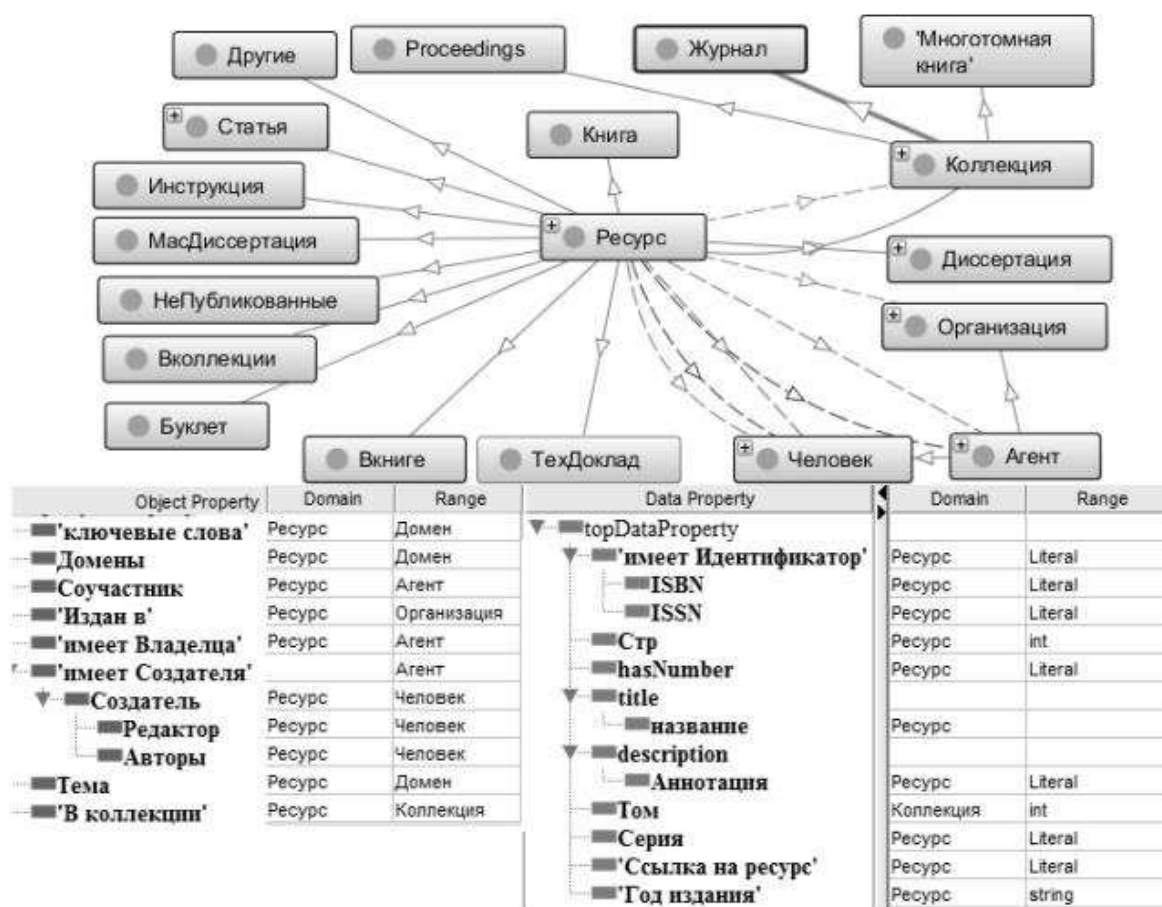


Рисунок 7 – Схема онтологии ресурсов (электронных документов) и их атрибуты

Аксиомы являются очевидными утверждениями, которые связывают и соотносят понятия и отношения и из которых впоследствии выводятся другие умозаключения. Аксиомы содержат информацию, которая не отражается в онтологии при установлении отношений между понятиями, правила автоматического добавления информации, ограничения, накладываемые на определенные отношения и необходимые для выведения умозаключения и т.п.

¹² Хоай Ле, Тузовский А. Ф. Использование онтологий в электронных библиотеках / Ле Хоай, А. Ф. Тузовский // Известия Томского политехнического университета, т.320 №5. 2012 г. С. 36.

Под *экземплярами* понимают конкретные элементы, относящиеся к определенной категории ПО.

Онтологии, в силу больших различий между ними, могут быть классифицированы по различным основаниям.

В зависимости от количества и качества включаемых понятий онтологии разделяют на онтологии *верхней, средней и нижней зоны*.

Онтологии *верхней зоны* состоят из 100-500 концептов и включают достаточно абстрактные и универсальные категории. Для этих онтологий характерен высокий уровень обобщений, поэтому составление аксиом затруднено. Однако эти онтологии имеют преимущество – их можно использовать в разнообразных областях и во многих языках.

Онтологии, содержащие от 500 до 100 000 концептов, относятся к *онтологиям средней зоны* и требуют вывода достаточно большого количества аксиом.

Онтологии нижней зоны принято также называть онтологиями предметной области. Они включают в себя от 200 до 2000 концептов, используются экспертами по предметным областям, поскольку ориентированы на определенную ПО и описывают конкретную ПО и специфику ее отношений. Это ограничивает круг решаемых такой онтологией задач и затрудняет использование онтологии в рамках других ПО. В таких онтологиях, однако ограничение рамками конкретной ПО делает возможным построения в онтологии большого количества аксиом и правил¹³.

Также онтологии подразделяют на *глубинные* (использующие глубинную семантику) и *поверхностные*, строящиеся на поверхностной семантике и определяющие понятия через непосредственное значение слова. Основной проблемой поверхностных онтологий является многозначность слов и выбор нужного значения.

¹³ Бубарева О. А. Модель, алгоритмы и программное обеспечение интеграции данных информационных систем на основе онтологий (на примере ВУЗа): диссертация кандидата технических наук: 05.13.11 / Бубарева О. А. – Новосибирск, 2015. – С. 24-25.

Можно также отметить деление онтологий на *общие онтологии, онтологии области знания и прикладные онтологии*. Эта классификация основывается на предмете концептуализации онтологии.

Выделяют *детализированные и простые* онтологии (проблемой детализированных онтологий является достаточно «громоздкий» язык написания); так называемые *отсылочные онтологии*, которые используются в оффлайн режиме и основаны на достаточно сложных теориях и *онтологии совместно используемые*, работающие в онлайн режиме и имеющие более простую структуру.

Исходя из языка пользователя разделяют онтологии на *одноязычные и многоязычные*. В настоящее время созданы онтологии, которые представляют знания на нескольких языках (EuroWordNet, MikroKosmos).

Независимо от классификации онтология должна соответствовать следующим критериям:

- естественность, т.е. онтология должна быть понятной пользователю и легко передавать подразумеваемый смысл
- объективность – онтология должна объективно отражать суть понятий и отношений между ними;
- логичность и последовательность. Эти критерии подразумевают, что утверждения онтологии не должны противоречить друг другу, отношениям, связывающим понятия и экземпляры и не должны нарушать иерархию понятий;
- онтология должна предусматривать возможность внесения новых элементов без нарушения общей структуры онтологии;
- специализация онтологии должна быть возможно минимальной, так как ориентация онтологии на решение одной конкретной задачи осложняет ее последующее использование, т.е. онтология должна быть максимально универсальной.

Разработка онтологии состоит из последовательных стадий. Вначале определяется цель создания онтологии, область и масштаб онтологии, т.е.

сфера ее применения и круг пользователей, т.е. происходит *спецификация*. Далее происходит построение терминологической базы и после достижения достаточного объема словаря - построение классификационных деревьев концептов, описание отношений между концептами, описание экземпляров, атрибутов классов и отдельных экземпляров, аксиом и формул, правил и ограничений. т.е. *концептуализация*.

Следующими этапами являются формализация и реализация онтологии, т.е. создание вычислительной модели на конкретном языке представления знаний.

Для реализации онтологии важен выбор языка спецификации, максимально приближающего машинную семантику систем к реальному миру. Языки спецификации онтологий делятся на простые, на языки, основанные на дескриптивных логиках и на фреймах (OKBC, OCML, Flogic), а также на Web-стандартах (XOL, RDF(S), DAML, OIL, OWL, SHOE, UPML). Применение различных форматов описания онтологий при импорте и экспорте данных показано на рисунке 8.

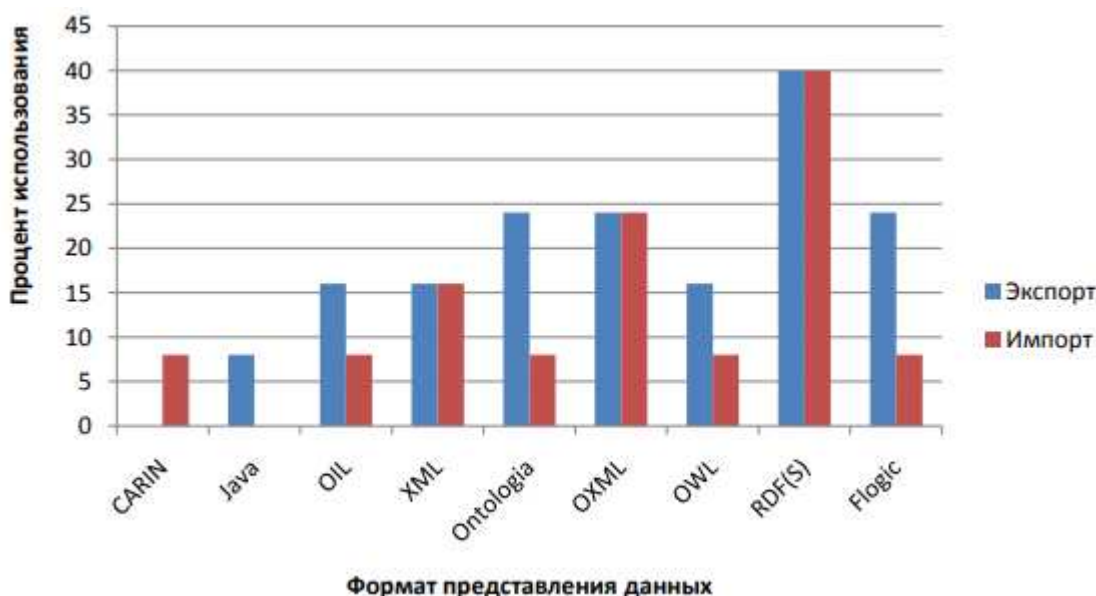


Рисунок 8 – Диаграмма применения различных форматов описания онтологий при импорте и экспорте данных

В числе инструментальных средств обработки онтологий необходимо назвать Protégé (предназначен для проектирования онтологий предметных областей), PROMPT (представляющий собой расширение системы Protégé), Chimaera (программа, предназначенная для объединения онтологий), OntoMerge, OBSERVER, ONION¹⁴.

Если говорить о применении онтологий, то их применение наиболее эффективно там, где необходима обработка данных с учетом их семантики. Это вопросно-ответные системы, системы извлечения знаний, системы ведения диалога «человек – компьютер», актуально применение онтологий при информационном поиске, при автоматическом реферировании и рубрикации текста.

Структура онтологий легко поддается анализу и обработке, онтологии содержат информацию о семантике заданных терминов, об иерархическом строении ПО, о взаимосвязи и отношениях концептов. Все это облегчает программам построение умозаключений и манипулирование онтологиями.

В области ИИ онтологии применяются для формальной спецификации понятий и отношений, характеризующих конкретную ПО.

2.2 Сравнение и выявление особенностей моделей представления знаний

Прежде всего нужно выявить положительные стороны рассматриваемых моделей представления знаний.

В силу того, что продукции не связаны между собой явно, они могут быть модифицированы по модульной технологии. Поэтому *продукционная модель* обладает достаточно высокой *модульностью*. *Наглядность* представления знаний также является сильной стороной продукционной

¹⁴ Бубарева, О. А. Модель, алгоритмы и программное обеспечение интеграции данных информационных систем на основе онтологий (на примере ВУЗа): диссертация кандидата технических наук: 05.13.11 / Бубарева О. А. – Новосибирск, 2015. – С. 33.

модели. Продукционные эвристики близки образу мышления человека-эксперта, можно провести аналогии между продукционной моделью и долговременной памятью человека, поэтому можно утверждать, что этому типу модели представления знаний *естественность*. Родовидовая иерархия понятий в продукционной модели поддерживается как связь между правилами, т.е., изменение в правилах меняет иерархию, поэтому можно говорить о иерархической *гибкости* продукционных моделей. *Простота создания и понимания* отдельных правил; *легкость внесения дополнений* и модификаций и *простота реализации логического вывода* также являются важными критериями при выборе этой модели для представления знаний.

Основные компоненты продукционной модели применяются при построении ИС, ориентированных на самые разнообразные проблемы, это позволяет сделать вывод об *универсальности* модели.

Продукционные модели достаточно удобны для представления логических взаимосвязей между фактами, поскольку они достаточно *формализованы и строги*.

Если говорить об *абстрактности элементов представляемых знаний*, то продукционная модель работает с простейшими составляющими знаний — фактами и правилами.

Основные достоинства систем, основанных на продукционных моделях - это простота представления знаний и организации логического вывода. Такие системы состоят из совокупности трех компонентов: *базы правил* (продукций), *базы фактов*, которая содержит декларативные знания о конкретной ПО (они используются как аргументы в условиях применимости продукций), и *интерпретатора продукций*, который анализирует условия применимости продукций, определяет правила выполнения продукций и порядок управления ЛВ в продукционной системе. Реализация продукционных моделей знаний базируется на языках типа ПРОЛОГ'а.

ИС, основанные на продукционной модели, достаточно *конкретизированы*, т.к. охватывают небольшую, хорошо определенную ПО.

Решения, получаемые такой ИС обладают достаточной эффективностью и удобством в силу определенных ограничений, которые продукционная модель накладывает на решаемую задачу, объем знаний и другие параметры.

Данная модель широко применяется в экспертных системах, в том числе промышленного (коммерческого) уровня¹⁵.

Отличительной чертой *фреймовой модели* является упомянутая выше возможность *комбинировать* декларативные и процедурные знания благодаря процедурным вложениям. Поэтому в этой модели может быть отражена как непосредственная информация о представляемом объекте, так и его функциональные и поведенческие особенности. Использование различных типов фрейм-структур (фрейм-роли, фрейм-сценарии, фрейм-ситуации) для описания самых разнообразных понятий и объектов ПО позволяет подробно и максимально истинно отображать все многообразие знаний. Поэтому данная модель, так же, как и продукционная, является *универсальной*.

Фреймовые модели имеют четко выраженную *иерархическую структуру*, что позволяет строго систематизировать и таксономизировать представляемые знания. При этом фреймовая структура дает возможность представить сложные объекты в виде единого целого, из которого при необходимости могут быть вычленены отдельные, более мелкие составляющие. Это позволяет сделать вывод о том, что эта модель представления знаний обладает *гибкостью и внутренней интерпретируемостью*, а для представляемых знаний характерны *связность и активность*.

Фреймовой модели также, как и продукционной, присуща *естественность*, поскольку принципы ее построения вполне соответствуют логической памяти человека, направленной на комплексное запоминание смысловой составляющей представленных знаний.

¹⁵ Козлов А. Н. Интеллектуальные информационные системы: Учебник /А. Н. Козлов. Пермь: ФГБОУ ВПО Пермская ГСХА. 2013. С. – 62.

Описание ПО в виде фреймов позволяет описывать довольно абстрактные понятия, поэтому этой модели в большей степени, чем продукционной, присуща *абстрактность*.

В отличие от продукционной модели, оперирующей простыми элементами знаний, использование фреймовой модели позволяет также *описывать метазнания*, т.е., знания о процессах, происходящих в самой системе и правилах получения и формирования новых знаний.

Отличительной особенностью фреймовой модели является также *принцип наследования*, позволяющий рационально использовать память и анализировать слабо детализированные ситуации, а также облегчающий построение БЗ, в которой родовая иерархия понятий хранится в явной форме¹⁶.

Благодаря механизму присоединенных процедур в системе, основанной на фреймовом представлении знаний, может быть реализован любой механизм управления выводом на основе обмена сообщениями – объектно-ориентированного метода управления выводом.

В настоящее время активно развиваются методы объектно-ориентированного программирования, поэтому фреймовые модели находят широкое применение. Многие промышленные экспертные системы строятся с использованием специальных языков представления знаний, основанные на фреймовой модели: FRL (Frame Representation Language), KRL (Knowledge Representation Language), фреймовая «оболочка» Карра. Экспертная система MOLGEN, занимающаяся генерированием знаний и исследований в области молекулярной генетики, построена на фреймовой основе¹⁷.

Наиболее эффективно положительные стороны применения исследуемой модели проявляются при структурном описании сложных понятий ПО, а также при решении задач, где необходимы различные способы реализации ЛВ.

¹⁶ Нечаев Н. П. Реализация механизма поддержки принятия решений с использованием фреймовой модели на примере задачи Эйнштейна / Н. П. Нечаев, О. В. Рогозин // Журнал Новые информационные технологии в автоматизированных системах. 2007. С – 87.

¹⁷ Муромцев Д. И. Разработка экспертных систем в Dtools Guvnog // Д. И. Муромцев, М. А. Колчин. – Санкт – Петербург: НИУ ИТМО, 2013. – С.8.

Следует также отметить, что фреймовая система без механизма присоединенных процедур может быть использована как БД системы продукции.

Онтологии, так же, как и предыдущие рассмотренные модели представления знаний, обладают достаточно высокой *универсальностью*, особенно онтологии высшего уровня - метаонтологии. Рассуждения людей основываются не только на знаниях в конкретной ПО, но и на знаниях высокой степени общности (например, описания свойств пространства, времени, личности и т.п.). Онтологии высшего уровня помогают компьютеру учесть эту особенность человеческого мышления и конкретизировать предметные ситуации.

Онтологии основаны на общей терминологии, поэтому они находят применение практически во всех областях, обрабатывающих данные на естественном языке. Кроме того, применение онтологий делает процесс обработки информации намного проще, поскольку помогает приложениям распознавать семантические различия и оттенки понятий ПО, помогает связывать воедино знания, подлежащие компьютерной обработке и представлять их в достаточно доступном пользователю виде. Поэтому можно характеризовать онтологическую модель, как *естественную*.

Онтологии обладают достаточно высокой *активностью и модульностью, и открытостью*: при определении новой онтологии могут использоваться уже ранее построенные онтологии, а появляющиеся в определенной ПО новые языковые конструкции могут быть легко включены в используемые онтологии.

Практически все онтологические системы описывают понятия и элементы ПО в четких и единообразных определённых образцах - терминах, моделях и т.п., причем количество базовых терминов, служащих основой для конструирования всех остальных понятий, достаточно ограничено поэтому онтологическая модель является, как и производственная, *строгой и формализованной*.

Нужно также отметить такие качества данной модели, как *внутренняя полнота и логическая непротиворечивость*. Последнее свойство отличает онтологическую модель, в частности, от продукционной, где при накоплении большого числа продукций они могут начать противоречить друг другу.

Применение онтологий в ИИ позволяет эффективно создавать и использовать БЗ - онтологии эффективно интегрируют декларативные знания, язык онтологий естественен и понятен человеку-эксперту, универсальность языка введенных данных позволяет использовать существующую БЗ неоднократно, поскольку онтология охватывает различные этапы жизненного цикла технической системы и служит основой для построения и трансформации информационных моделей, БД и документов, обеспечивая их *связность* и адекватное понимание.

Применение онтологий делает эффективным организацию поиска – смысл понятий и терминов в запросах интерпретируется достаточно точно, терминологическая связность обеспечивает максимальное расширение запроса.

В ИИС применение онтологий облегчает принятие решений, поскольку способствует пониманию системой семантики конкретных терминов и составлению высказываний.

В настоящее время онтологическая модель находит широкое применение при построении емких БД и БЗ. Онтологии применяют в разработке приложений для программного обеспечения и управления знаниями, при создании ИИС, веб-приложений, САПР, медицинской информатики, семантических веб-технологий (технология Semantic Web получила развитие именно благодаря особенностям применяемых онтологий).

Среди языков описания онтологий можно назвать OWL — Web Ontology Language, разработанный как расширение RDF и RDFS; KIF (Knowledge Interchange Format — формат обмена знаниями), Common Logic (CL), CycL (онтологический язык, основанный на исчислении

предикатов с некоторыми расширениями более высокого порядка),
DAML+OIL

Теперь рассмотрим недостатки каждой из рассмотренных моделей.

Говоря о продукционных моделях, прежде всего необходимо отметить, что представление знаний посредством данной модели не вполне соответствует структуре знаний, выстраиваемой человеческим мозгом. Поэтому продукционная модель обладает меньшей естественностью, чем фреймовая и онтологическая модели.

Гибкость иерархии понятий, отмеченная выше, как положительная сторона модели, является одновременно и недостатком этой модели, поскольку ее следствием становится недостаточная ясность взаимных отношений правил.

Поскольку в продукционных системах затруднено установление иерархии понятий, оценить цельный образ представляемых знаний также довольно трудно, поэтому представление знаний с помощью продукций иногда называют «плоским»¹⁸.

Необходимость детальной структуризации базы знаний, обязательная проверка полноты и непротиворечивости знаний, правильности заполнения, проверка применимости правил, на которую затрачивается много времени снижают как эффективность обработки знаний, так и эффективность логического вывода, который также не обладает достаточной гибкостью. Кроме того, при работе с продукционными базами данных часто возникают синтаксические и семантические проблемы.

Продукционная модель предназначена для решения довольно простых задач, что делает неэффективным ее применение при решении

¹⁸ Старикова А. В. Создание подсистемы принятия решений в медицинских информационных системах / А. В. Старикова, О. Г. Берестнева, Г. Е. Шевелев, К. А. Шаропин. Томск: Известия Томского политехнического университета. – т.317. – №5. – 2010. – С. 196

многосоставных разнородных задач, а невозможность эффективного описания правил с исключениями является довольно существенным недостатком.

Ядро модели – конструкция ЕСЛИ, - ТО, требует небольшого объема памяти в силу своей простоты, однако объем БЗ, основанной на продукционной модели, растет линейно, по мере включения в нее новых знаний, поэтому БЗ может насчитывать несколько сотен, а то и тысяч правил, поэтому чтобы найти нужное правило необходимо просмотреть всю БЗ. Это негативно сказывается на скорости работы продукционных ИС. К тому же накопление большого количества правил зачастую приводит к тому, что они начинают противоречить друг другу.

Наиболее эффективно применение продукционной модели при разработке небольших систем, содержащих всего несколько десятков правил, т.к. при большом объеме представляемых знаний сильнее проявляются слабые стороны продукционной модели.

Говоря о *фреймовых моделях*, нужно отметить, что обучение систем, основанных на фреймах, затруднено. Эффективное приобретение и формирование новых знаний можно осуществить только в системах со сложной структурой фреймов, которые в свою очередь требуют больших затрат средств и времени. При этом устранение проблемы формирования противоречивых знаний должно осуществляться самой системой.

Фреймовые модели, в силу своей сложной структуры, требуют достаточно большого объема памяти. Поэтому одним из главных недостатков фреймовых моделей можно назвать их высокую сложность, которая снижает скорость работы всей системы и усложняет процесс внесения изменений в родовидовую иерархию.

Несмотря на иерархическую структуру знаний во фреймовых моделях, отдельные части представляемой информации сложно выстроить в последовательные высказывания.

Еще одним минусом ИС, основанных на фреймовой модели, является трудности при обработке исключений.

Усложненная реализация механизма вывода с помощью присоединенных процедур также может быть отнесена к недостаткам модели.

Наиболее эффективно применение фреймовых систем в тех случаях, когда описываемая ПО насчитывает немного исключений, а изменения родовидовых связей происходит достаточно редко.

Говоря о слабых сторонах *онтологий*, необходимо отметить следующее.

Каждая система онтологий понятна и имеет смысл для определенной группы специалистов конкретной ПО, владеющих терминологией, понятийным аппаратом и осведомленных о свойствах понятий представляемой ПО. Поэтому главная задача при разработке онтологии – обеспечить ее доступность для различных групп специалистов. А в связи с постоянным изменением научных знаний, компьютерные системы онтологий требуют разработки средств поддержки целостности и «версионности» онтологий на случай их изменений и дополнений

Представление знаний в онтологических системах производят узко профилированные специалисты. Поэтому языки онтологий довольно специфичны. Поэтому основная задача разработчиков – сделать язык представления онтологий открытым для пользователей, но при этом внутреннее представление онтологий для компьютерного использования и межмашинного обмена должно оставаться достаточно стандартизованным.

Также можно отметить, что для ведения больших онтологий (и большие библиотеки онтологий) необходима разработка специальных средств. Формирование сложных систем онтологий требует соответствующих средств опробования и отладки онтологий. В сложных онтологиях сложно полностью разделить декларативные и процедурные знания, что значительно снижает эффективность их применения.

Интеграция онтологий, представленных на разных языках в разных логиках и моделях, также представляется довольно затруднительной.

3 Построение базы знаний на основе продукционно – фреймовой модели

Задание: построить базу знаний по устранению неисправностей с персональным компьютером на основе продукционно – фреймовой модели.

Выполнение задания включает следующие этапы:

1. Анализ предметной области
2. Реализация базы знаний в программной среде.

Анализ предметной области

Для создания базы знаний необходимо выделить множество всех неисправностей компьютера, которые будет различать экспертная система и выделить совокупность информационных параметров, значения которых позволяют поставить диагноз.

В результате выявлены следующие типы возможных проблем с компьютером:

– Проблемы с запуском. К ним относятся следующие проблемы: Компьютер не запускается, Компьютер запускается, но сразу выключается, при запуске компьютера системный блок сильно шумит.

– Проблемы с аппаратурой. К ним относятся: Отсутствие изображения на экране, искажения на экране, неработоспособность клавиатуры.

– Проблемы с производительностью. К ним относятся: Компьютер тормозит, появляется синий экран.

Далее необходимо составить правила, на основе которых будут выводиться рекомендации. В результате созданы следующие правила:

– Если проблема – компьютер не запускается и системный блок не подключен к сети, то необходимо подключить его к сети и попробовать заново запустить компьютер.

– Если проблема – компьютер не запускается и системный блок подключен к сети, но нарушена целостность кабеля то необходимо заменить кабель и попробовать заново запустить компьютер.

– Если проблема – компьютер не запускается и системный блок подключен к сети, целостность кабеля не нарушена, но блок питания не подключен к материнской плате, то необходимо подключить блок питания к материнской плате и попытаться заново запустить компьютер.

– Если проблема – компьютер не запускается и системный блок подключен к сети, целостность кабеля не нарушена, блок питания подключен к материнской плате и блок питания работает на другом компьютере, то необходим ремонт материнской платы.

– Если проблема – компьютер не запускается и системный блок подключен к сети, целостность кабеля не нарушена, блок питания подключен к материнской плате и блок питания не работает на другом компьютере, то необходим ремонт блока питания.

– Если проблема – компьютер включается, но через какое то время выключается и блок питания исправен, то необходим ремонт материнской платы.

– Если проблема – Компьютер включается, но через какое то время выключается и блок питания не исправен, то необходим ремонт блока питания.

– Если проблема – Системный блок сильно шумит, то необходимо провести чистку кулера процессора.

– Если проблема – Системный блок сильно шумит, и чистка кулера процессора не помогла, то необходимо провести чистку кулера блока питания.

– Если проблема – Системный блок сильно шумит, чистка кулера процессора и кулера блока питания не помогли, то необходима замена жесткого диска.

– Если проблема – Нет изображения на экране и монитор не подключен к сети, то необходимо подключить монитор к сети и попытаться заново запустить компьютер.

– Если проблема – Нет изображения на экране, монитор подключен к сети и монитор работает на другом компьютере то необходим сброс настроек Bios.

– Если проблема – Нет изображения на экране, монитор подключен к сети и монитор не работает на другом компьютере то необходим ремонт монитора.

– Если проблема – искажения на экране и есть повреждения на hdmi кабеле, то необходимо заменить кабель.

– Если проблема – искажения на экране, повреждений на hdmi кабеле нет, и видеокарта не работает на другом компьютере то требуется замена видеокарты.

– Если проблема – искажения на экране, повреждений на hdmi кабеле нет, видеокарта работает на другом компьютере и имеются повреждения материнской платы то требуется ремонт материнской платы.

– Если проблема – искажения на экране, повреждений на hdmi кабеле нет, видеокарта работает на другом компьютере и повреждений материнской платы нет, то требуется ремонт монитора.

– Если проблема – не работает клавиатура и кнопка питания не включена, то необходимо включить кнопку питания.

– Если проблема – не работает клавиатура, кнопка питания включена, клавиатура работает на другом компьютере то необходима переустановка драйвера клавиатуры.

– Если проблема – Компьютер тормозит и есть перегрев комплектующих компьютера, то необходимо заменить термопасту или заменить кулер.

– Если проблема – Компьютер тормозит, перегрева комплектующих компьютера нет, то необходимо провести чистку реестра.

– Если проблема – Компьютер тормозит, перегрева комплектующих компьютера нет, и чистка реестра не помогла, то необходимо удалить программы из автозапуска.

– Если проблема – Компьютер тормозит, перегрева комплектующих компьютера нет, чистка реестра и удаление программ из автозапуска не помогли, то необходимо проверить компьютер на наличие вирусов.

– Если проблема – Появляется синий экран, то необходимо выполнить восстановление системы.

– Если проблема – Появляется синий экран и восстановление системы не помогло, то необходимо выполнить обновление программного обеспечения.

– Если проблема – Появляется синий экран, восстановление системы и выполнение обновления программного обеспечения не помогли, то необходимо обновить драйвера устройств.

Для наглядности все проблемы и методы их устранения можно представить в виде графа, показанного на рисунке 9.

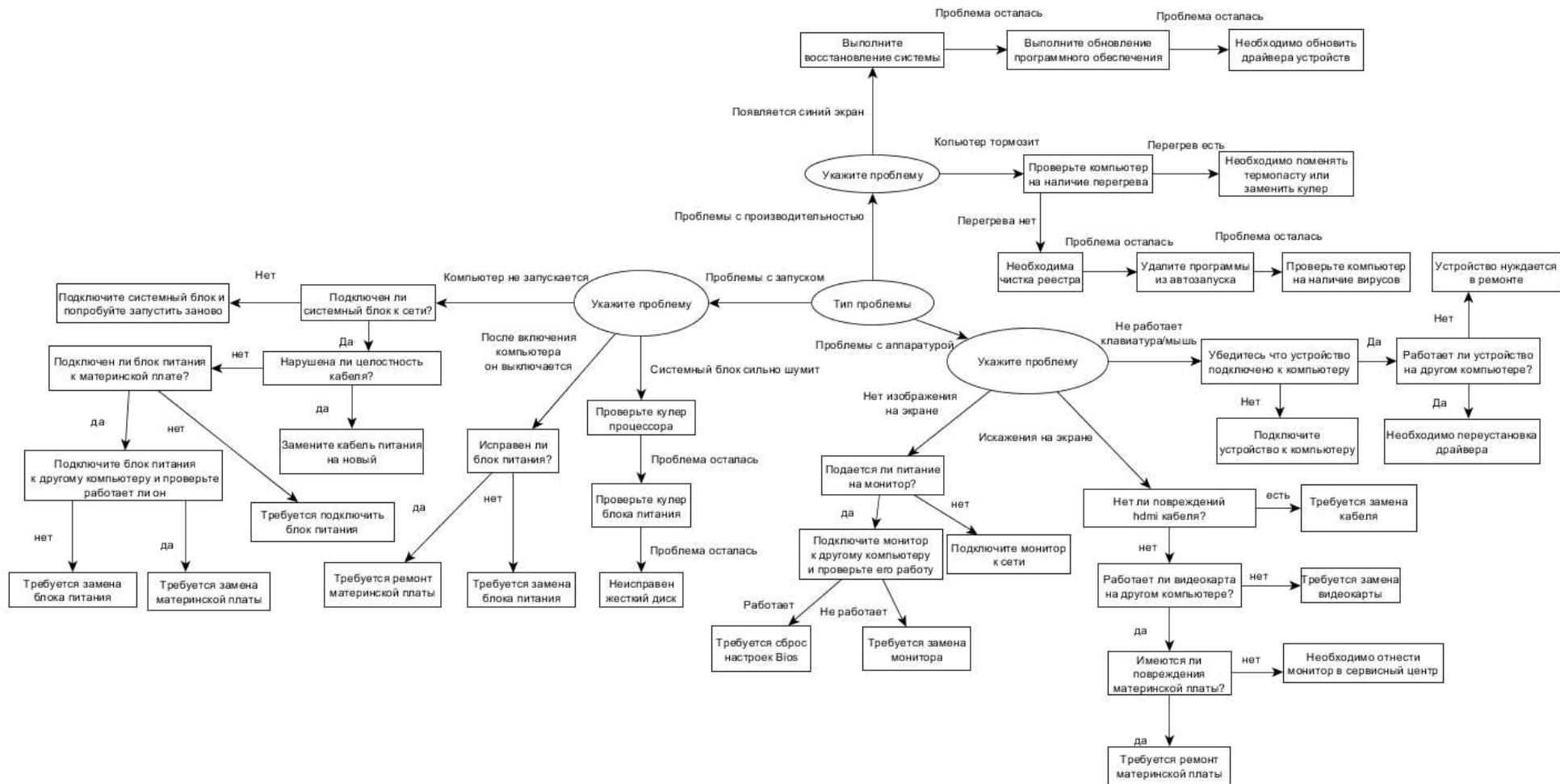


Рисунок 9 – Неисправности персонального компьютера и методы их устранения

Реализация

Созданная база знаний была применена в программной среде Eswin 2.1.

ESWin – программная оболочка для работы с производственно-фреймовыми экспертными системами с возможностью использования лингвистических переменных. Описываемая программная оболочка предназначена для решения задач логического вывода на основе интерпретации правил – продукций с использованием фреймов как структур данных, включающих в себя, в частности, лингвистические переменные.

База знаний состоит и набора фреймов и правил – продукций.

Для описания объектов, понятий, ситуаций в базе знаний используются фреймы. Формат внешнего представления фреймов выглядит так, как показано на рисунке 10.

```
FRAME (<тип фрейма>) = <имя фрейма>
PARENT: <имя фрейма-родителя>
OWNER: <имя фрейма-владельца>
<имя слота 1> (<тип слота>) [<вопрос слота>?]: (<значение 1>;
                                     <значение 2>; ...;
                                     <значение k>)
<имя слота 2> (<тип слота>) [<вопрос слота>?]: (<значение 1>;
                                     <значение 2>; ...;
                                     <значение l>)
.
.
.
<имя слота n> (<тип слота>) [<вопрос слота>?]: (<значение 1>;
                                     <значение 2>; ...;
                                     <значение m>)
ENDF
```

Рисунок 10 – Представление фрейма в программной оболочке Eswin

Пример фрейма для рассматриваемой предметной области показан на рисунке 11.

```
Frame=Проблемы с аппаратурой|
тип(symbol)[Выберите тип]: (На экране появляются горизонтальные полосы, искажения; )
Провод(symbol)[Проверьте целостность самого видео кабеля]: (Исправен; нет;)
EndF
```

Рисунок 11 – Пример фрейма

Правила используются в базе знаний для описания отношений между объектами, событиями, ситуациями и прочими понятиями. На основе отношений, задаваемых в правилах, выполняется логический вывод. В условиях и заключениях правил присутствуют ссылки на фреймы и их слоты.

Формат внешнего представления правил показан на рисунке 12.

```
RULE <номер правила>
<условие 1>
<условие 2>
.
.
.
<условие m>
DO
<заключение 1>
<заключение 2>
.
.
.
<заключение n>
ENDR
```

Рисунок 12 – Представление правила в программной оболочке Eswin

Пример созданного для предметной области правила приведен на рисунке 13.

```

Rule 1
= (Неисправность.Запуск ; Проблемы с запуском)100
= (Проблемы с запуском.Тип ; После запуска компьютера индикатор загрузки виндовс не исчезает)100
= (Проблемы с запуском.проблемы ; Проблема осталась)100
Do
= (Цель.Устранить неисправность в пк; Выполните восстановление операционной системы с загрузочного диска)100
EndR

```

Рисунок 13 – Пример правила

Также правила, содержат коэффициент доверенности. Коэффициент достоверности – число от 0 до 100. Коэффициент достоверности в заключении используется при формировании значения слота фрейма-экземпляра при срабатывании правила.

Рассмотрим пример логического вывода. Например, у пользователя нет изображения на экране, монитор подключен к сети и работает на другом компьютере.

Для начала пользователю необходимо запустить поиск решения. Для этого необходимо выбрать пункт Решение и нажать выбрать Поиск решения.

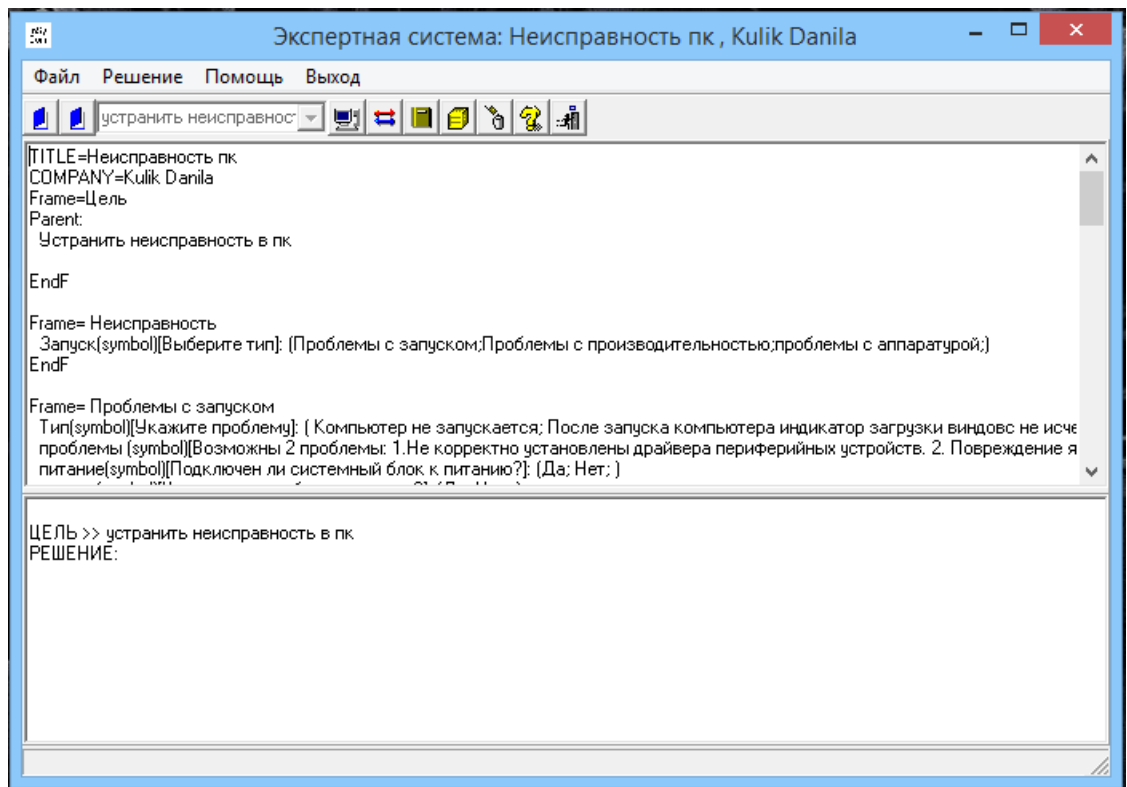


Рисунок 14 – Общий вид окна экспертной системы

Далее, пользователю необходимо ответить на вопросы системы, выбирая нужные параметры. Процесс выбора нужных параметров показан на рисунках 15 – 19.

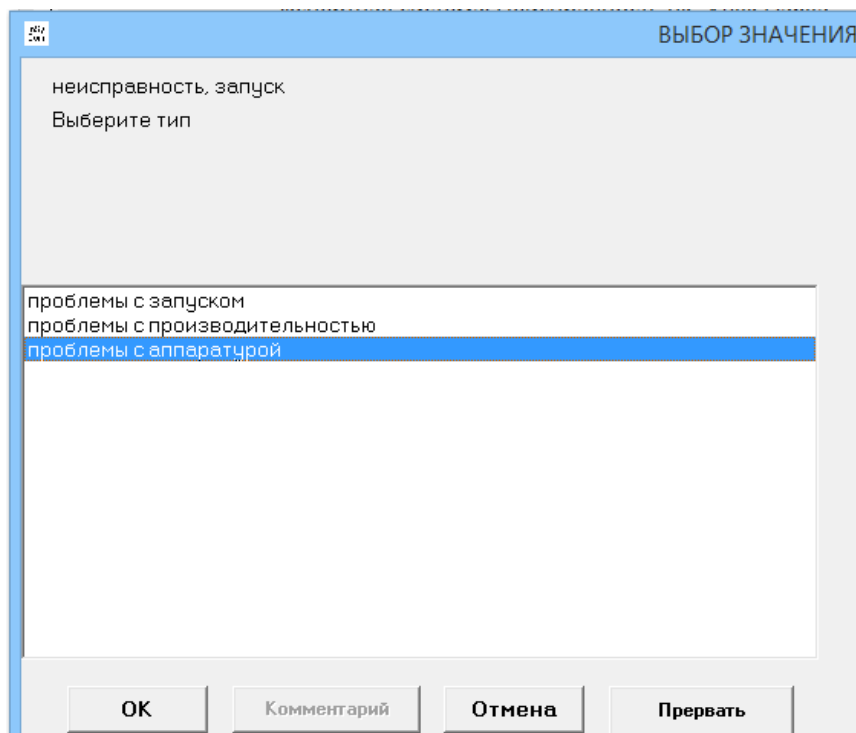


Рисунок 15 – Выбор типа проблемы

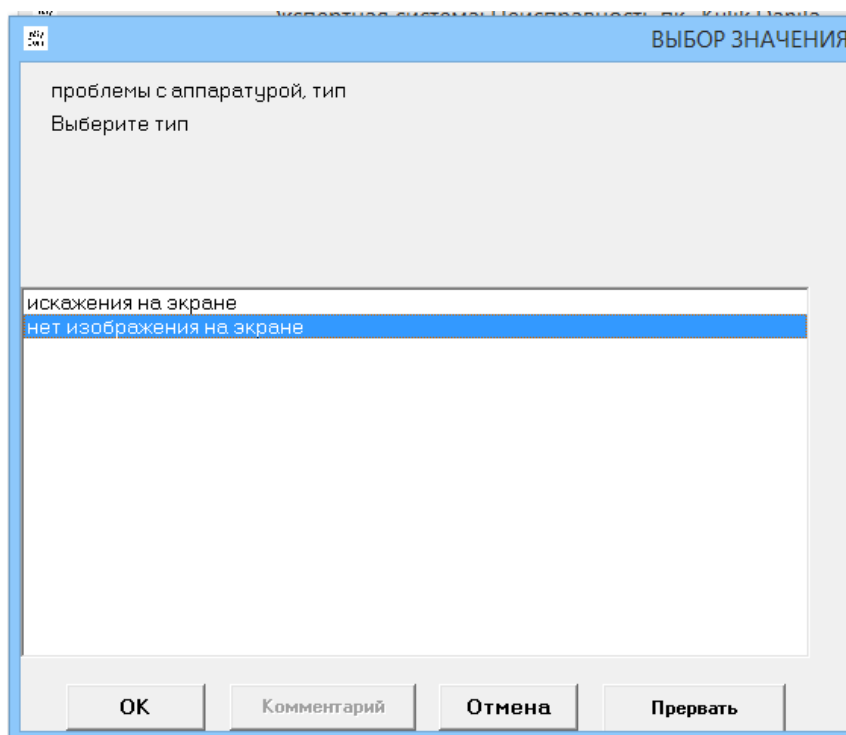


Рисунок 16 – Выбор конкретной проблемы

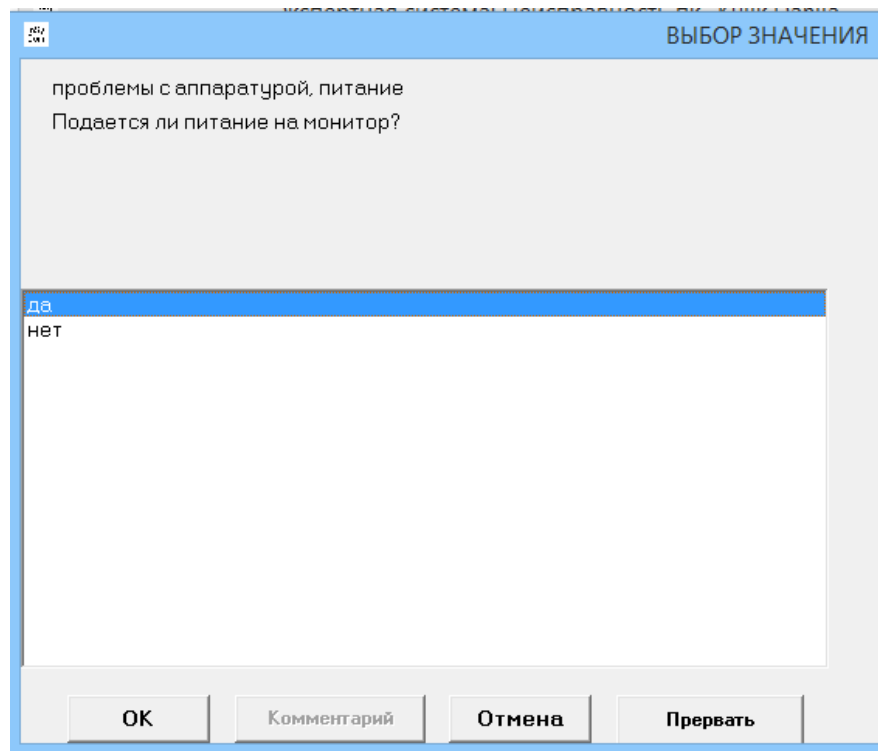


Рисунок 17 – Ответ на вопрос о подаче питания на монитор

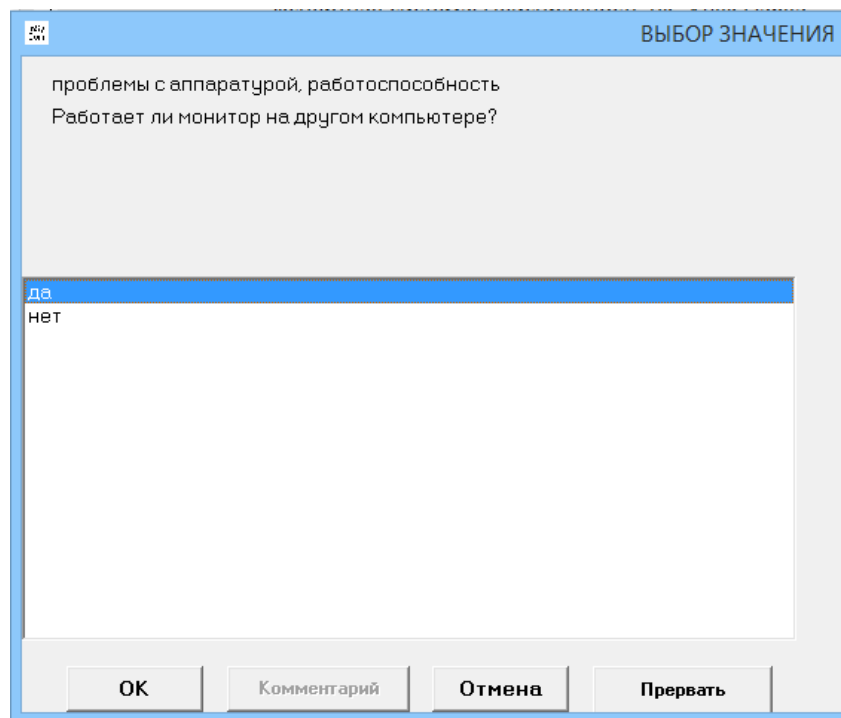


Рисунок 18 – Ответ на вопросе работе монитора на другом компьютере

В результате работы пользователь получает рекомендацию по устранению проблемы.

ЦЕЛЬ >> устранить неисправность в ПК
РЕШЕНИЕ:
цель. устранить неисправность в ПК = необходим сброс настроек bios с уверенностью 100 %
(Правило 8)

Рисунок 19 – Решение

ЗАКЛЮЧЕНИЕ

Подводя итоги анализа исследованных моделей, нужно отметить, следующее.

Для того, чтобы эффективно и полноценно представлять эвристические знания модель представления должна обладать следующими характеристиками: универсальность; наглядность представления знаний; естественность; простота построения и применения; способность при представлении знаний реализовать такое свойство знаний, как активность; открытость БЗ; возможность отражения иерархической структуры представляемых объектов ПО; уметь оперировать не явными знаниями; использовать многоуровневых представлений; проводить быстрый и эффективный ЛВ.

Всем трем рассмотренным моделям в большей или меньшей степени присущи такие свойства, как универсальность, естественность, активность, и наглядность. Модели обладают широкими возможностями адекватного представления ПО, используют достаточно разнообразный набор средств, передающих характерные особенности определенной ПО, в должной мере эффективно организуют ЛВ.

Однако отдельно ни одна из рассмотренных моделей знаний не соответствует всем требованиям. Можно выделить такие свойственные этим моделям недостатки, как сложность получения новых знаний, вероятность формирования противоречивых знаний, сложность наращивания модели по мере накопления знаний, а также ее значительный размер.

Поэтому на практике специалисты применяют комбинированные модели представления знаний для того, чтобы в полной мере реализовать свойства знаний.

Также в ходе выполнения выпускной квалификационной работы была создана база знаний по устранению неисправностей персонального компьютера на основе продукционно – фреймовой модели.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Андрейчиков, А.В. Интеллектуальные информационные системы: Учебник / А. В. Андрейчиков, О. Н. Андрейчикова. – Москва: Финансы и статистика, 2004. – 424 с.
2. Бубарева, О. А. Модель, алгоритмы и программное обеспечение интеграции данных информационных систем на основе онтологий (на примере ВУЗа): диссертация кандидата технических наук: 05.13.11 / Бубарева О. А. – Новосибирск, 2015. – 137 с.
3. Гаврилова, Т.А. Базы знаний интеллектуальных систем: Учебник по проектированию и программированию систем искусственного интеллекта / Т. А. Гаврилова, В. Ф. Хорошевский. – Санкт-Петербург: Питер, 2001. – 384 с.
4. Гаскаров, Д.В. Интеллектуальные информационные системы: Учебник для вузов. – Москва.: Высшая школа, 2005. – 432 с.
5. Головчинер, М. Н. Введение в системы знаний. Курс лекций / М. Н. Головчинер. – Томск. 2011. – 168 с.
6. Гурина, Р.В. Методика конструирования фреймовых схем / Р.В. Гурина // Образовательные технологии. – 2013. – №1. – с. 31 – 31.
7. Козлов, А. Н. Интеллектуальные информационные системы: Учебник / А. Н. Козлов. Пермь: ФГБОУ ВПО Пермская ГСХА. 2013. – 307 с.
8. Коробова, И. Л. Методы представления знаний / И. Л. Коробова. – Тамбов: Издательство ТГТУ, 2003. – 512 с.
9. Митрофанова, О. А. Онтологии как системы хранения знаний / О. А. Митрофанова, Н. С. Константинова // Всероссийский конкурсный отбор обзорно – аналитических статей по приоритетному направлению «Информационно – телекоммуникационные системы». – 2008. – 54 с.
10. Муромцев, Д. И. Разработка экспертных систем в Drools Guvnor / Д. И. Муромцев, М. А. Колчин. – Санкт-Петербург: НИУ ИТМО, 2013. – 108 с.

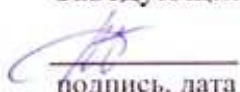
11. Нечаев, Н. П. Реализация механизма поддержки принятия решений с использованием фреймовой модели на примере задачи Эйнштейна / Н. П. Нечаев, О. В. Рогозин // Журнал Новые информационные технологии в автоматизированных системах. 2007. – 96 с.
12. Остроух, А.В. Интеллектуальные системы / А.В. Остроух. – Красноярск: научно – инновационный центр, 2015. – 110 с.
13. Поспелов, Д. А. Искусственный интеллект: в 3-х кн. Кн.2. Модели и методы: Справочник / Под ред. Д.А. Поспелова – Москва.: Радио и связь, 1990. – 340 с.
14. Старикова, А. В. Создание подсистемы принятия решений в медицинских информационных системах / А. В. Старикова, О. Г. Берестнева, Г. Е. Шевелев, К. А. Шаропин. Томск: Известия Томского политехнического университета, т.317, №5. 2010. – 197 с.
15. СТО 4.2-07-2014 Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности. – Введ. 9.01.2014. – Красноярск: ИПК СФУ, 2014. – 60 с.
16. Тоискин, В. С. Интеллектуальные информационные системы / В.С. Тоискин. – Ставрополь: Изд-во СГПИ, 2010. – 188 с.
17. Хоай, Ле. Использование онтологий в электронных библиотеках / Ле Хоай, А. Ф. Тузовский // Известия Томского политехнического университета. – т.320. – №5. – 2012. – 42 с.

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и Информационных технологий
институт
Информационные системы
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой ИС


подпись, дата

Л.С. Троценко
инициалы, фамилия

«13» июня 2018г.

БАКАЛАВАРСКАЯ РАБОТА

09.03.02 Информационные системы и технологии

Сравнительный анализ моделей представления знаний в интеллектуальных
системах

Руководитель


подпись, дата


П.П. Дьячук
инициалы, фамилия

Студент КИ14-13Б 031402261
номер группы, зачетной книжки

Куши 13.06.18
подпись, дата

Д.Н. Кулик
инициалы, фамилия

Нормоконтролер


подпись, дата

Ю.В. Шмагрис
инициалы, фамилия

Красноярск 2018