

**НАЦИОНАЛЬНАЯ АКАДЕМИЯ НАУК УКРАИНЫ
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ, МОЛОДЕЖИ И СПОРТА
ИНСТИТУТ КИБЕРНЕТИКИ имени В.М. ГЛУШКОВА**

А.В. Палагин, С.Л. Кривый, Н.Г. Петренко

**ОНТОЛОГИЧЕСКИЕ МЕТОДЫ
И СРЕДСТВА ОБРАБОТКИ
ПРЕДМЕТНЫХ ЗНАНИЙ**

Монография

Луганск 2012

УДК 004.2: 004.3
П 14

Рекомендовано к изданию Ученым советом Института кибернетики имени В.М. Глушкова Национальной академии наук Украины (протокол № 17 от 25 октября 2011г.)

Рецензенты:

Ф.И.Андон, академик НАН Украины
В.А. Широков, академик НАН Украины

Палагин А.В.

П 14 Онтологические методы и средства обработки предметных знаний:
монография / А.В. Палагин, С.Л. Крывый, Н.Г. Петренко. – Луганск: изд-во ВЛУ им. В. Даля, 2012. – 324 с.

ISBN 988-966-590-

Рассматриваются методы и принципы онтологического подхода к построению и проектированию знание-ориентированных информационных систем с онтолого-управляемой архитектурой, реализующих технологию извлечения предметно-ориентированных знаний из множества естественно-языковых текстов, их формально-логического представления и прикладной обработки. Изложены методики проектирования таких систем и компьютерных онтологий произвольных предметных областей, рассмотрены принципы и алгоритмы функционирования инструментального комплекса автоматизированного проектирования компьютерных онтологий. Приведены примеры построения онтологической системы для заданной предметной области и реализации некоторых процедур Text processing & mining. Предложены алгебра и система списочных структур, формализующие процесс обработки текстовой информации.

Для научных работников в области информатики, специалистов по разработке систем обработки знаний и прикладной лингвистике, студентов и аспирантов специальностей по инженерии знаний.

Розглядаються методи і принципи онтологічного підходу до побудови й проектування знання-орієнтованих інформаційних систем з онтолого-керованою архітектурою, що реалізують технологію добування предметно-орієнтованих знань із множини природно-мовних текстів, їх формально-логічного представлення та прикладної обробки. Викладено методики проектування таких систем і комп'ютерних онтологій довільних предметних областей, розглянуто принципи й алгоритми функціонування інструментального комплексу автоматизованого проектування комп'ютерних онтологій. Наведено приклади побудови онтологічної системи для заданої предметної області та реалізації деяких процедур Text processing & mining. Запропоновано алгебру і систему списочних структур, що формалізують процес обробки текстової інформації.

Для наукових фахівців в області інформатики, спеціалістів по розробці систем обробки знань і прикладній лінгвістиці, студентів й аспірантів спеціальностей з інженерії знань.

ISBN 988-966-590-

УДК 004.2: 004.3

© А.В. Палагин, С.Л. Крывый,
Н.Г. Петренко, 2012
© Східноукраїнський національний
університет імені Володимира Даля,
2012

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

АСС	алгебра списочных структур
АССС	алгебраическая система списочных структур
БД	база данных
БД Ф ПрВ	база данных фактов и правил вывода
БЗ	база знаний
ВТ	вычислительная техника
ВУ	вычислительное устройство
ДО	дистанционное обучение
ЕИ	естественный интеллект
ЕЯ	естественный язык
ЕЯВ	естественно-языковое высказывание
ЕЯО	естественно-языковый объект
ЕЯТ	естественно-языковый текст
ЕЯ-	естественно-языковый
ЗОИС	знание-ориентированная информационная система
ЗнПС	знание-ориентированная поисковая система
ИИ	искусственный интеллект
ИИС	интеллектуальная информационная система
ИИТ	интеллектуальная информационная технология
ИКОН	инструментальный комплекс онтологического назначения
ИнС	инструментальные средства
ИП	интерфейс пользователя
ИР	информационный ресурс
ИТ	информационная технология
КГ	концептуальные графы
КЛП	когнитивный лингвистический процессор
КО	компьютерная онтология
КС	компьютерная система
ЛБД	лексикографическая база данных
ЛИП	логико-информационный подход
ЛКТ	лингвистический корпус текстов
ЛО ПдО	лингвистическая онтология предметной области
ЛПТЛ	линейная пропозициональная темпоральная логика
МВ	машина вывода
МНИ	междисциплинарные научные исследования
МПЗ	модели представления знаний
НИ	научные исследования
НО ПдО	начальная онтология предметной области
ОВУ	онтология верхнего уровня
ОГ	онтологический граф, онтограф
ОЗнИС	онтолого-знающая информационная система
ОИШ	общая информационная шина

ОО	объектно-ориентированный
ОП	онто-логический подход
ОнП	онтологический подход
ОнС	онтологическая система
ООП	объектно-ориентированный подход
О ПдО	онтология предметной области
ОУИС	онтолого-управляемая информационная система
ОУИС ПдО	онтолого-управляемая информационная система предметной области
О-модель	онто-логическая модель
ПдД	предметная дисциплина
ПдО	предметная область
ПЛИС	программируемая логическая интегральная схема
ПМЛ	пропозициональная модальная логика
ПП	прикладной процессинг
ППП	подсистема прикладного процессинга
ППФ	правильно построенная формула
ПрП	проблемное пространство
САПР	система автоматизированного проектирования
СИИ	семанτικο-информационный интерпретатор
СС	семантическая сеть
СУБД	система управления базой данных
ТД	текстовый документ
ТЗ	техническое задание
УГО	управляющая графическая оболочка
ФКЗ	формализация качественных знаний
ФЛМ ПЗ	формально-логические методы представления знаний
ФЛП	формально-логическое представление
ЭБ	электронная библиотека
ЭК	электронный курс
ЭК логика	экзистенциально-конъюнктивная логика
ЭлК	электронная коллекция
ЭнС	энциклопедический словарь
ЭС	экспертная система
ЯКМ	языковая картина мира
ЯОИС	языково-онтологическая информационная система
ЯОКМ	языково-онтологическая картина мира
ЯПЗ	язык представления знаний
FOL	First Order Logic
IDEF	Integrated DEFinition
KIF	Knowledge Interchange Format
КР	Knowledge Processing
KR	Knowledge Representation
MDD	Model Driven Development
NLP	Natural Language Processing
ODA	Ontology Driven Architecture
OWL	Ontology Web Language
RDFS	Resource Description Framework Schema

ПРЕДИСЛОВИЕ

В монографии сделана попытка комплексного представления проблем обработки предметно-ориентированных знаний (в научно-технической сфере), построения и проектирования знание-ориентированных информационных систем с онтолого-управляемой архитектурой как нового класса средств вычислительной техники. Такие системы названы *онтолого-управляемыми информационными системами*. Они реализуют интегрированную информационную технологию, включающую компьютерную обработку естественно-языковых объектов, заданных лингвистическим корпусом текстов и описывающих некоторую предметно-проблемную область, извлечение знаний с целью формально-логического представления и автоматизированной обработки предметно-ориентированных знаний.

Сделан акцент на естественно-языковое (текстовое) представление предметных знаний, так как объём текстовой информации, в первую очередь в сети Интернет, увеличивается стремительными темпами, за которыми не поспевают темпы совершенствования потребительских (архитектурных, технологических) характеристик современных персональных компьютеров. Имеющийся в настоящее время разрыв между хорошо проработанными методами и средствами обработки естественно-языковой информации (ЕЯ-информации) для решения прикладных задач в узкоспециализированных предметных областях, с одной стороны, и недостаточностью таковых применительно к решению комплексных задач, связанных с анализом и пониманием ЕЯ-информации, её формально-логическим представлением, извлечением предметных знаний с их последующим использованием в общезначимых предметных областях, в том числе и для проведения сложных междисциплинарных научных исследований, сдерживает развитие и применение систем обработки знаний, реализующих (в том числе) в полном объёме технологии Data & Text Mining. Особо следует отметить дефицит качественных методов и средств знание-ориентированной обработки ЕЯ-информации для украинского сегмента.

Авторы глубоко благодарны за предоставленную возможность использования материалов совместных публикаций сотрудникам Института кибернетики имени В.М. Глушкова НАН Украины д.т.н. Кургаеву А.Ф. (в 7.4), к.т.н. Величко В.Ю. (в 5.3), м.н.с. Михайлюку А.В. (в 3.2), аспирантам Севрук А.О. (в 8.3) и Бибикову Д.С. (в 5.2), а также Луганского национального университета имени Тараса Шевченко – к.т.н. Тихонову Ю.Л. (в 8.2), а также за ряд ценных замечаний и советов при подготовке монографии – сотрудникам ИК НАН Украины д.т.н. Опанасенко В.Н. и к.т.н. Лисовому А.Н.

ВВЕДЕНИЕ

В настоящее время интеллектуальные информационные технологии (ИИТ) всё глубже проникают во все сферы научно-технической деятельности человеческого общества. Центральным их звеном являются технологии инженерии знаний, включая процессы управления знаниями, поэтому успехи данного направления во многом определяются интеллектуальным уровнем и общей эффективностью компьютерных систем (КС). К сожалению, многие идеи этого раздела искусственного интеллекта (ИИ) сегодня так и остаются не реализованными полностью или частично по ряду причин.

Первой причиной является недостаточное понимание исследователями принципов и механизмов восприятия, отображения, осознания и обработки информации/знаний в мозге человека и, как следствие, соответствующие компьютерные методы и алгоритмы не совершенны. Не вызывает сомнения тот факт, что ИИТ и соответствующие КС будут и дальше интенсивно развиваться в направлениях, соответствующих наиболее существенным признакам разумной деятельности, в том числе распознавания, преобразования и понимания знаковых систем (включая естественно-языковые). Результатом указанных комплексных задач является генерация совокупного смысла, заложенного в анализируемых знаковых системах. При этом под смыслом понимается то, что делает знаковую систему текстом, а последний – источником знаний.

Второй причиной является недостаточная эффективность большинства современных КС: последние в процессе управления знаниями должны оперировать не примитивными данными (в традиционном понимании), а знаниями, представленными в подходящей формальной теории. Разработка и использование новых информационных технологий, таких как суперкомпьютерные и GRID-вычисления, многоагентные системы и др., со своей стороны, также требуют знание-ориентированного подхода. Ориентация на аппаратную поддержку (реализацию) основных процедур манипулирования знаниями на базе современных микроэлектронных технологий (например, ПЛИС-технологии, РИМ-технологии, реконфигурируемого компьютеринга) с применением соответствующих САПР поможет в значительной степени повысить эффективность современных КС.

Особенностью интеллектуальных информационных систем (ИИС) являются интеграционные процессы, которые влияют на развитие их архитектуры и функциональных возможностей. Системная интеграция междисциплинарных знаний как таковых, а также технологии их эффективного формирования, представления, обработки и использования являются сильными стимулирующими факторами на этом пути. Не менее важным фактором является естественная интеграция двух областей ИИ, которые раньше развивались почти параллельно и независимо: knowledge-engineering и компьютерная лингвистика. Указанную интеграцию можно рассматривать как закономерное эволюционное развитие разных научных теорий, заложенных ещё с давних времён и известных как логика и онтология. Отсутствие развитой теоретической базы системологии междисциплинарного взаимодействия и квалифицированных кадров (инженер по знаниям должен быть специалистом высокой квалификации в областях онтологического инжиниринга, фор-

мальных логических теорий, прикладной лингвистики, целой совокупности математических теорий, когнитологии, психологии и др.) является, по-видимому, *третьей причиной*.

Сегодня очевидно, что развитие онтологического инжиниринга, а также знание-ориентированных информационных систем (ЗОИС) связано с совершенствованием методов компьютерной обработки ЕЯ-информации и предметных знаний. Нетрудно заметить, что хорошо проработанные методы и средства обработки ЕЯ-информации для решения прикладных задач в узкоспециализированных предметных областях (ПдО) недостаточны для решения комплексных задач, связанных с формально-логическим представлением ЕЯ-информации, извлечением предметных знаний и их последующим использованием в произвольных ПдО.

Отсюда следует актуальность и важность разработки новых научных методов и подходов к автоматическому анализу и глубинно-семантическому пониманию ЕЯ-информации, её формализованному представлению в рамках подходящей теории, автоматическому обнаружению и извлечению новых знаний, технологий и, наконец, соответствующих инструментальных средств автоматизированного построения баз знаний предметных областей.

Решение данной проблемы является основополагающим на пути разработки общей теории понимания, формирования, представления и обработки естественно-языковых знаний методами ИИ и является целью данной работы.

Исследования этой проблемы традиционны для направления инженерии знаний. Большое внимание ей уделял ещё В.М. Глушков, в частности сформулировав и описав концептуальную парадигму работы со знаниями. Известны весомые вклады в её решение Н. Хомским, М. Минским, А. Тарским, Д. Дэвидсоном, Монтегю, Д.А. Пospelовым, Т.А. Гавриловой, Н.Н. Леонтьевой, Э.В. Поповым, В.А. Широковым и другими. Многие из работ по другим направлениям ИИ, математической логике можно также отнести к данной области.

Особенностью данных исследований является их междисциплинарность, сочетание различных теорий, решений и средств их достижения. И вместе с тем, как указывалось выше, единой теории пока нет.

Структурно весь материал монографии представлен в виде восьми логически связанных глав и трёх приложений.

В *первой главе* описан общий подход к исследованию предметно-ориентированных знаний, начиная с онто-гносеологического анализа. Такой анализ рассматривает абстрактную систему <Субъект – Объект – Отношение> и взаимодействие её составных частей между собой. Методы изучения каждого такого взаимодействия рассматриваются в разных научных теориях – гносеологии, онтологии, теории отражения и логике.

Рассмотрена естественная схема взаимодействия человека с окружающим миром: *входная информация* → “*сознание – понимание – знания*”. В ней выделено компонент “сознание”, играющий роль персонифицированного инструмента, который вырабатывает совокупность предметно, ситуативно или причинно-связанных сущностей, которые образуют “сознательную” картину мира. Выработка указанных сущностей, построение и использование на их основе картины мира реализуется как результат восприятия и преобразования входной информации в когнитивном цикле, ядром которого является *механизм понимания*. Сознание человека имеет языковой статус, и это выражено понятием “языковое сознание”, а сознательная картина мира при этом представлена “языковой” картиной мира (ЯКМ), которая является не только системой знаний, а и некоторым буфером, который связывает общие знания с профессиональными. Все знания человечества (или полная картина

мира) представлены в виде “пирамиды” знаний, состоящей из частичных картин мира. При этом предполагается, что каждая картина мира строится по тем же законам системного подхода, что и ЯКМ, а интерпретация процесса развития знаний может быть использована как в рамках одной предметной области, так и для описания полной картины мира.

Выполнен сущностный анализ понятия “знание” как *исходной, преобразующей и конечной* составляющей процесса познания (“знание посредством знаний преобразуется в новое знание”). Предложены частные определения таких взаимосвязанных понятий, как *смысл, знание, данные и информация*, выражающие собой наибольшую проблемность исследований в области построения систем обработки знаний, в том числе и ЗОИС, оперирующих ЕЯ-информацией. Рассмотрены основные свойства знаний – внутренняя интерпретируемость, структурированность, связность, семантическая метрика и активность. На основе выполненного анализа синтезирована обобщённая схема “эволюционирования” знаний.

Используя известные представления *онтологий верхнего уровня*, разработаны методологические основы построения многоуровневых компьютерных онтологий и системного анализа, необходимого для построения строгих математических моделей произвольной предметной области (проблемы, задачи) и их интерпретации для получения конкретных результатов, и в частности *компьютерной онтологии верхнего уровня и системы категорий* как знаний об общей структуре действительности. При этом система категорий выступает по отношению ко всякому новому знанию как некоторый фиксированный *категориальный каркас*, который имеется до познания данного конкретного класса явлений и определяет формы этого познания, так что всякое новое содержание, доставляемое конкретным исследователем, подводится под одну из категорий и упорядочивается по правилам категориального каркаса *Универсального*, рассматриваемого как три (относительно) независимые группы категорий: *Материальное – Абстрактное; Независимое – Относительное – Промежуточное; Продолжительное – Событийное*. Для категорий онтологии верхнего уровня предложена формально описанная система аксиом.

Рассмотрена формальная постановка задачи извлечения знаний из естественно-языковых текстов (ЕЯТ), основным моментом в которой является построение двух фундаментальных отношений, присутствующих практически в каждом ЕЯТ. Это отношения эквивалентности и частичного порядка, которые названы *общезначимыми*. Первое отношение определяет классы синонимических объектов, а второе отношение – иерархию подчинённости классов эквивалентности. Оба эти отношения составляют основу построения онтологий.

Выполнен анализ трёх базовых аспектов разработки онтолого-управляемых информационных систем, а именно – *онтологический, логический и методологический*. Эти аспекты имеют свои, в общем случае фиксированные объекты исследований, соответствующие процессу познания или разработке некоторой ИИС. В связи с ними все объекты могут быть поделены на три группы: система сущностей (или объектов реального мира), система знаний и система обработки сущностей в соответствии с данной системой знаний. Первая группа является предметом онтологического исследования, вторая группа – логического исследования и третья – методологического.

Разработана онтолого-классификационная схема концептуальных понятий, отображающая проблематику (структуру) многочисленных научных исследований в области ИИ, нижние уровни которой представляют объект и предмет исследований при решении научно-технической проблемы построения онтолого-управляе-

мой информационной системы (ОУИС) с обработкой больших объёмов ЕЯ-информации.

Проанализированы известные структурные направления онтологического инжиниринга как перспективного инструмента предметных и междисциплинарных научных исследований и предложены их расширения, основными из которых являются системная интеграция научных знаний (включая персонифицированные знания) и инструментальный комплекс онтологического назначения (ИКОН), “работающий” на протяжении всего процесса реализации интегрированной информационной технологии обработки ЕЯ-информации.

Во *второй главе* рассмотрены логико-математические основы, используемые при разработке и проектировании ОУИС. Основное внимание уделено теориям множеств, отношений и алгоритмов. При этом рассмотрены формальные логические языки, такие как исчисление высказываний и исчисление предикатов первого порядка. Приведены многочисленные примеры, поясняющие использование рассматриваемых математических понятий.

Читателям, знакомым с основами указанных математических теорий, эту главу можно пропустить без ущерба понимания остального материала.

В *третьей главе* рассмотрены различные формы представления знаний, извлечённых из естественно-языковых высказываний (ЕЯВ), для подмножеств логики первого порядка, а также некоторые формализованные аспекты извлечения новых знаний. При этом описание сопровождается многочисленными примерами.

Выполнен сравнительный анализ формальных логических языков первого порядка для представления ЕЯВ, таких как силлогизмы Аристотеля, логика высказываний и алгебра логики, исчисление предикатов, экзистенциально-конъюнктивная логика, продукционная система, дизъюнкты Хорна, фреймовые системы и объектно-ориентированные системы, семантические сети, концептуальные графы (КГ) и язык KIF (Knowledge Interchange Format).

На основе выполненного анализа (по существенным характеристикам) подмножеств FOL (First Order Logic), в том числе с их реализациями в действующих системах и нотациях, сделан вывод, что в них можно представить сложные утвердительные предложения и вопросы к их частям, заменяя некоторые компоненты представления (силлогизма, формулы, графа) неизвестной переменной, ответы на которые могут быть получены или решением уравнения, где значение формул с переменной должно быть истинным, или поиском фрагмента графа-вопроса в общем графе и установлением значений неизвестных узлов или дуг. Результаты анализа представлены в виде таблицы, из которой видно, что: наибольшей выразительностью обладают три системы: исчисление предикатов, концептуальные графы и KIF. Это может служить критерием выбора именно этих систем логики в качестве основных для представления и анализа знаний. Выбор формы представления и анализа знаний в общем случае зависит от конкретной задачи. Некоторые задачи можно эффективно решать навигацией в КГ по алгоритмам работы с графами. Ряд задач не имеет методов и средств их решения на графах – в таком случае следует применять универсальные общие методы логического вывода в системе исчисления предикатов или KIF.

Детально рассмотрено формализованное представление ЕЯВ в форме КГ, даны формальные определения КГ и семантической сети, приведены примеры, поясняющие эти понятия. Рассмотрено также представление ЕЯВ в форме концептуальных графов Дж. Соуы, выявлено ряд недостатков такого представления. Предложены пути устранения этих недостатков, которые основаны на принципах и механизмах построения языково-онтологической информационной системы (ЯОИС)

с задействованием таких компонент как “Языково-онтологическая картина мира (ЯОКМ)”, “Онтология ПдО” и “Грамматический процессор”.

Раскрыты методы извлечения знаний из ЕЯТ на примерах лексико-грамматического анализа, силлогистики Аристотеля и её теоретико-множественной интерпретации.

В *четвёртой главе* рассмотрены принципы организации и методика построения *компьютерных онтологий*. Сформулировано определение компьютерной онтологии и онтологической системы, состоящей из *онтологии объектов, онтологии процессов и онтологии задач*, а также общую методологию проектирования онтологических знаний ПдО.

Особый статус предметных знаний приобретают языковые знания и соответствующие онтологии. Рассмотрены особенности лингвистических онтологий и проанализирована наиболее известная из них – WordNet. Основные характеристики наиболее известных онтологий сведены в таблицу.

Рассмотрена ЯОКМ – лингвистическая онтология, которая ориентируется на украинский и русский языки и состоит из лексикографической базы данных, моделей морфологических, синтаксических, поверхностно-семантических и глубинно-семантических структур естественного языка (ЕЯ), модели интегрированной интерпретации входного ЕЯТ.

Выделен особый вид системной интеграции междисциплинарных знаний – интеграции лингвистических и предметных знаний. Проанализированы известные проекты такой интеграции и предложена формализация её некоторых аспектов и оценок.

Выполнен анализ известных методик IDEF (Integrated DEFinition) – проектирования сложных объектов, в том числе онтологий, и сравнение с предложенным *системно-онтологическим анализом*. Оказывается, что методики IDEF и системно-онтологический подход используют одни и те же множества сущностей ПдО и проблемного пространства (ПрП), акцентируя внимание на различных совокупностях характеристик и атрибутов; конечные цели этих двух подходов отличаются: для IDEF – функционально-блочная модель, а для системно-онтологического подхода – онтолого-содержательная модель.

Описаны методики проектирования онтологии объектов, онтологии процессов, онтологии задач и соответствующий обобщённый алгоритм, при этом предпочтение отдано автоматизированному способу проектирования, так как даже для сравнительно небольших предметных областей списки терминов-понятий, терминов-процессов, терминов-отношений и терминов, описывающих классы задач, могут составлять несколько тысяч записей.

В *пятой главе* рассмотрены оригинальные алгебро-логические и лингвистические модели формализации и обработки ЕЯТ, а также формальная постановка задачи анализа. Выделены особенности анализа ЕЯТ, определяющие направленность на формирование понятийной структуры, т. е. на автоматическое извлечение знаний из текстов и их прагматическую интерпретацию в терминах прикладных задач. При этом текст рассматривается как объект разных уровней анализа: как знаковая система, как грамматическая система, система знаний о мире, источник исходных данных для решения прикладных задач. Каждый уровень имеет свои особенности, свои средства выражения, а, следовательно, предполагает наличие специфических методов обработки.

Предложен алгебро-логический подход к анализу и пониманию ЕЯТ, включающий алгебру и систему списочных структур. Введено понятие *списка элементов* и определены функции и операции над ними, а в качестве элементов списка

рассматриваются слова ЕЯ. При этом алгебраическая система получается путём добавления в алгебру списочных структур *предиката равенства* и *условного оператора*.

Предложена информационная модель этапов лингвистического анализа и рассмотрено их функционирование на содержательном уровне.

В *шестой главе* разработаны методологические основы проектирования ЗОИС с обработкой естественно-языковых объектов (ЕЯО) на основе онтологического подхода. Такие ЗОИС в монографии названы онтолого-управляемыми (или с онтолого-управляемой архитектурой). В онтолого-управляемую информационную систему входят две подсистемы: языково-онтологическая информационная подсистема, обеспечивающая обработку языковых знаний, и онтолого-управляемая информационная подсистема (ОУИС ПдО), обеспечивающая обработку знаний некоторой предметной области.

Рассмотрены основы *онто-логического подхода* проектирования ОУИС. Такой подход учитывает логико-информационную и онтологическую концепции проектирования, а также виртуальную парадигму, при которой архитектура компьютерной системы ориентирована на технологию реконфигурируемого процессинга. Последняя обеспечивает адаптивность системы благодаря наличию в ОУИС архитектурных и технологических возможностей настройки в условиях априорной и текущей неопределённостей на основе обучения и опыта. Важной отличительной особенностью онто-логического подхода по сравнению с логико-информационным (кроме онтологизации) является добавление ещё одного (предметного) уровня управления ОУИС.

Предложена блок-схема процесса синтеза структуры ОУИС, состоящая из блоков синтеза ЯОИС, ОУИС ПдО и отображения языковых знаний в их формальное представление.

Сформулированы характерные черты ЗОИС с онтолого-управляемой архитектурой, из которых основными являются: композиция онтологий разного уровня и назначения, как по вертикали, так и по горизонтали (по вертикали интегрируются онтология верхнего уровня, онтология домена предметных областей и онтология предметной области, а по горизонтали интегрируются онтологии предметных знаний и знаний проблемного пространства). Особо отмечено, что предметные знания взаимодействуют с языковыми знаниями через лингвистическую онтологию предметной области; что эффективно многократное использование онтологии предметной области и онтологии задач для разных наборов типовых задач, а вследствие двойной парадигмы предметных знаний и онтологического управления оказывается возможным включение компонентов архитектуры ОУИС в архитектурно-структурную организацию инструментального онтологического комплекса автоматизированного построения онтологических баз знаний предметных областей.

Рассмотрена архитектура и информационная модель ЯОИС, одной из отличительных особенностей которой в сравнении с известными лингвистическими процессорами является наличие ЯОКМ. Последняя, по сути, представляет собой онтологическую базу данных лексики естественного языка, компьютерно-ориентированную информационную структуру, которая является необходимым компонентом любой знание-ориентированной системы, обрабатывающей ЕЯТ.

Рассмотрена архитектурно-структурная организация ОУИС ПдО, содержащей онтологическую базу знаний (в общем случае конечное множество системно интегрированных онтологических баз знаний), машину вывода, подсистему прикладного процессинга и интерфейс с пользователями и/или внешней средой, а также обобщённый алгоритм проектирования ОУИС в целом.

Седьмая глава содержит описание инструментального комплекса онтологического назначения, который реализует относительно самостоятельные интегральные технологии – *автоматизированного построения онтологии ПдО* на основе обработки соответствующего лингвистического корпуса текстов и *системной интеграции онтологических знаний* в некотором домене предметных областей.

Разработана методология проектирования ИКОН, алгоритмов его функционирования и программно-аппаратных средств. Для этого проанализированы функциональные возможности известных инструментальных средств (ИнС) построения онтологий с учётом таких факторов как поддерживаемые формализмы и формат представления, множество предоставляемых пользователю сценариев работы с онтологическими структурами, архитектура приложения, интерфейс пользователя и доступность. Выявлен ряд недостатков, основные из которых: отсутствие процедур автоматического (автоматизированного) формирования компонент онтологии; англоязычный интерфейс с пользователем, в котором (для большинства ИнС) не предусмотрено присвоение имён компонентам онтологии на русском или украинском языке; структуризация концептов в онтологии выполняется только по одному типу отношений; для большинства общедоступных ИнС не предусмотрена работа с большими объёмами онтологических знаний.

ИКОН состоит из трёх подсистем, интегрирующих информационные ресурсы, программно-аппаратные средства обработки и естественный интеллект инженера по знаниям и эксперта, которые, взаимодействуя между собой, реализуют совокупность алгоритмов автоматизированного итерационного построения понятийных структур предметных знаний, их накопления и/или системной интеграции. Программно-аппаратные средства ИКОН реализованы по модульному принципу. Разработаны обобщённые алгоритмы проектирования.

Представлены методологические основы системной интеграции междисциплинарных научных знаний, разработаны алгоритм и программные средства объединения двух и более онтологий в близких предметных областях.

Восьмая глава посвящена прикладным аспектам функционирования ОУИС и инструментального комплекса онтологического назначения. Рассмотрены примеры автоматизированного построения онтологической базы знаний и онтологии задач ПдО “Проектирование вычислительных устройств и систем на ПЛИС”, применения онтологического подхода к разработке электронных курсов обучения, в частности для предметной дисциплины “Базы данных”, формализованного построения онтологии текстового документа и разработки некоторых процедур Text Processing. Предложена общая архитектура знание-ориентированной поисковой системы, функционирующей в составе ИКОН и ОУИС.

В приложении приведены: описания онтологий верхнего уровня SUMO и Mikrokosmos; методика проектирования простой онтологии ПдО и онтограф домена “Информатика”, а также краткий толковый словарь по онтологическому инжинирингу.

Изложенный в монографии материал относится к *Computer Science*, в частности, разработке знание-ориентированных информационных систем, компьютерных онтологий и формализации процесса анализа и понимания естественно-языковых текстов.

1. ИССЛЕДОВАНИЕ ПРЕДМЕТНО-ОРИЕНТИРОВАННЫХ ЗНАНИЙ

1.1. Онто-гносеологический анализ предметных знаний

Онто-гносеологический анализ знаний вообще, и предметных знаний в частности, предполагает использование в единстве исследовательских методов онтологии, гносеологии и логики. Прежде чем перейти к детальному рассмотрению научно-практических задач анализа знаний, обратимся к его философским аспектам, а точнее к той совокупности общих научных дисциплин, подходы, методы и принципы которых непосредственно влияют на формирование и становление указанного выше анализа. Такими дисциплинами являются *Онтология*¹, *гносеология* (или *теория познания*), *общая логика* и *теория отражения* [1]².

Содержание указанных дисциплин, в той или иной степени, характеризует триаду {*Субъект (S) – Объект (O) – Отношение (R)*}. Под *субъектом* понимается нечто действующее и познающее (в каком-то роде синоним к термину *Наблюдатель*). В более широком смысле под *субъектом* можно понимать воздействующую и отражающую систему любого уровня. Под *объектом* понимается отражаемая и подвергающаяся воздействию система, нечто познаваемое и изменяемое субъектом. С точки зрения системы $\Lambda = \langle O, S, R \rangle$ известны следующие (нестрогие) определения упомянутых выше научных дисциплин [1].

Онтология – это учение о всеобщих свойствах и отношениях объективного мира, теория всеобщего, которая рассматривает объект вне его отношения к субъекту.

Логика (общая) – это учение о всеобщих структурах познания субъекта, рассматриваемых как таковые, вне их отношения к объекту.

Теория отражения изучает отражательное отношение субъекта к объекту. Познание – частный случай и высшая форма отражения, а потому **теория познания** – есть результат спецификации общей теории отражения.

Роль указанных выше научных дисциплин рассмотрим на примере, где в качестве объекта выступает научная теория.

Логика рассматривает структуру этой теории независимо от того, что она отражает, каковы объективные основы этой структуры. С позиций общей *теории отражения* потребуется охарактеризовать то общее отношение *R* между субъектом *S* и объектом *O*, которое выражается в теории. Синтез теории отражения и логики для целей теории познания даёт общую **методологию**. Она “знает” и объективную направленность, и логическую структуру познавательных операций, но абстрагируется от социальных и психологических условий, их генезиса и функционирования.

¹ Этот термин написан с прописной буквы, потому что в данном контексте он применим к онтологии как философской дисциплине, чтобы отличить от компьютерной онтологии.

² Сказанное можно в равной степени отнести и к двум другим аспектам исследования – методологическому и логическому [1].

Общее взаимодействие перечисленных научных дисциплин и системы Λ представлено на рис. 1.1.

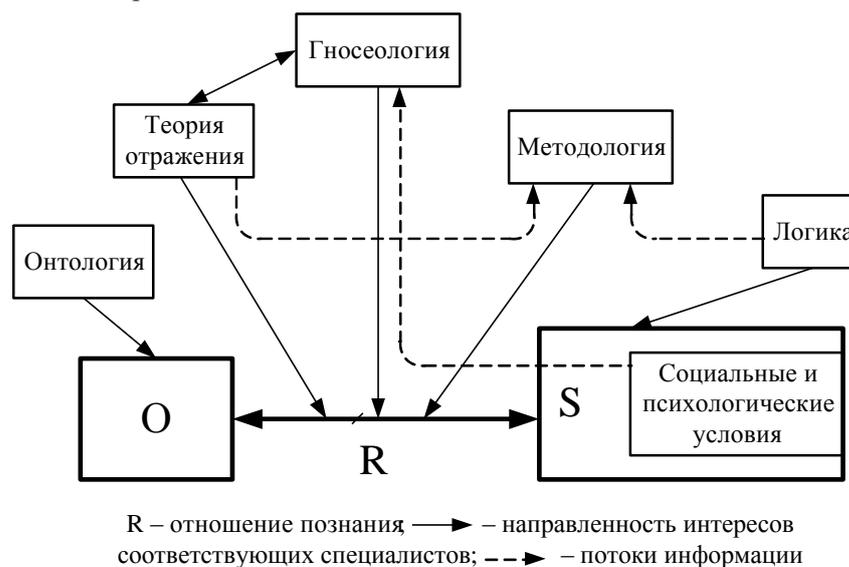


Рис.1.1. Общая структура и взаимосвязи научных теорий, изучающих систему “объект↔субъект”

Таким образом, в системе Λ разные специалисты, занимающие разные позиции, изучают разные предметы.

Конечно, описанная модель взаимодействия научных дисциплин и системы Λ является приближённой. В ней не рассмотрены несколько уточняющих моментов, в частности:

- а) границы общей логики;
- б) процесс воздействия субъекта на объект;
- в) в общем случае субъект и объект могут меняться местами;
- г) как применение математических методов в методологии порождает новую дисциплину – **технологиию** познания, где методология переходит от общих принципов анализа к процессу синтеза (проектирования).

Если система Λ рассматривается на абстрактном уровне, то её анализом должна заниматься онтология как философская дисциплина. Если в качестве объекта выбрана некоторая ПдО (что представляет наибольший практический интерес), то его рассмотрением должна заниматься “компьютерная онтология предметной области” (О ПдО). Аналогичным образом можно перейти к онтологии домена прикладных областей, интегрирующей множество взаимосвязанных предметных областей, и, наконец, – к онтологиям процессов и задач [2–9].

1.1.1. Общесистемный аспект исследования “пирамиды” знаний

Особенность создания современных информационных технологий состоит в стремлении к максимальной интеграции результатов двух областей искусственного интеллекта, которые когда-то развивались параллельно и независимо: knowledge-engineering и компьютерной лингвистики (когнитивной семантики). Можно сказать, что это стремление отображает в общем случае естественную схему взаимо-

действия человека с окружающим миром: *входная информация* → “*сознание – понимание – знания*”. В последней не указаны промежуточные (ощущение, рефлексия и др.) процедуры когнитивного цикла, чтобы выделить “конструктивную” триаду. Сознание здесь играет роль персонифицированного инструмента, который вырабатывает совокупность предметно, ситуативно или причинно-связанных сущностей, которые образуют “сознательную” картину мира. Выработка указанных сущностей, построение и использование на их основе картины мира реализуется как результат восприятия и преобразования входной информации в когнитивном цикле, ядром которого является *механизм понимания*. Сознание человека имеет языковой статус, и это выражено понятием “языковое сознание”, а сознательная картина мира при этом представлена “языковой” картиной мира, которая является не только системой знаний, а и некоторым буфером, который связывает общие знания с профессиональными. Для понимания естественного языка в прагматическом плане важно выделить соответствующий механизм, который характеризуется как “...преобразование представления на естественном языке в логические выражения” [10].

Архитектуру ЯКМ в общем виде можно представить как иерархическую сеть, структура которой, опираясь на базовые категории познания и сферы бытия (пространство, время, движение, качество, количество, отношение, отдельное) или более ёмкие (материя, сознание, практика), содержит три базовых взаимосвязанных уровня:

- категориальный (концептуальный);
- понятийного толкования (слов и словосочетаний);
- грамматического разбора.

Первый ориентирован на формализацию построения и правил вывода на семантическом (концептуальном) языке понятий. Вторым может быть аналогизирован с современными толковыми словарями (в электронном исполнении), но обладающими замкнутой системой связей между лексемами в развитых лексических структурах и сильным взаимодействием с верхним (концептуальным) уровнем, при котором последний обеспечивает понятийное толкование лексем и снятие многозначности. Уровень грамматического разбора обеспечивает информацию о морфологических (род, число, падеж, вид, время и т. д.), синтаксических (атрибутивных, количественных, обстоятельственных, служебных и др.) и семантических (объектных, ситуационных, признаковых, актантно-ролевых, лексико-функциональных) характеристиках лексем и их семантико-синтаксических конструкций. Технологической основой ЯКМ могут служить модели реляционных баз данных, обеспечивающих функции взаимодействия в виде «ЕЯ–SQL» запросов в качестве естественно-языкового интерфейса [11].

Таким образом, ЯКМ можно представить в виде виртуального индексированного информационного поля таблиц лексико-семантических отношений (целое-часть, род-вид, множество-элемент, причина-следствие, объект-функция и пр.) и их компонентов (столбцов и строк) с наборами семантических характеристик лексем и более сложных лексических конструкций. Основой верхнего уровня ЯКМ являются отношения типа: род-вид, целое-часть. Наиболее крупными таксономическими категориями нижних уровней являются семантико-частиречные характеристики отношений.

Знания в конкретной (*i*-й) предметной области (ПдО) представляют собой *i*-е расширение общей картины мира, так что полную картину мира (или пирамиду знаний) можно представить в виде, показанном на рис. 1.2, здесь *i* проходит через позиции философской, биологической, химической, физической и т. д. частичных

картин мира (КМ) или, что то же самое, научных знаний в каждой из поименованных предметных областей. При этом каждая из них ($КМ_i = ПдО_i$) строится по тем же законам системного подхода (целостности, иерархии, классификации и структурирования), что и ЯКМ. Применяя методы системных переходов [11] можно процедуру, реализующую преобразования по формуле Брукса ($K(S) + \Delta I = K(S+\Delta S)$, где $K(S)$ – исходные знания, ΔI – новая порция информации, $K(S+\Delta S)$ – выходные знания), сформулировать в терминах *обобщения* смысла, т. е. ($i \rightarrow j$)–переходов, из чего следует, что новые, неизвестные ранее знания, по сути, представляют собой определённое обобщение (интерпретацию) уже известных знаний в категориях вышестоящих уровней иерархии их представления. На самом деле процесс развития знаний более сложен и глубок. Во-первых, в отличие от процедуры обобщения в структуре ЯКМ, научное обобщение происходит в условиях, когда j -й уровень иерархии не определён. Во-вторых, получение новых знаний об окружающем нас реальном мире часто связано с изменением представлений (отрицанием) определённых старых знаний. Этот процесс можно схематично проиллюстрировать (рис. 1.3). Здесь $j = 1, 2, \dots$ – уровни обобщения знаний: от элементарных (аксиоматических) знаний до полных; стрелки указывают направление развития знаний: \rightarrow – в сторону расширения элементарных знаний, \uparrow – в сторону обобщения существующих знаний; dI_j, dI_i – приращения знаний по соответствующим осям координат. Зона, ограниченная пересечением двух кривых, – это область знаний, инвариантных относительно новых и старых теорий. Зоны S_c и S_n – это области взаимоисключающих разделов теорий. Данная интерпретация процесса развития знаний может быть использована как в рамках одной предметной области, так и для описания полной картины мира. В последнем случае нетривиальными представляются междисциплинарные функции ЯКМ в плане креативного взаимодействия $ПдО_i$ и процессов конвергенции их теорий и технологий (рис. 1.2).

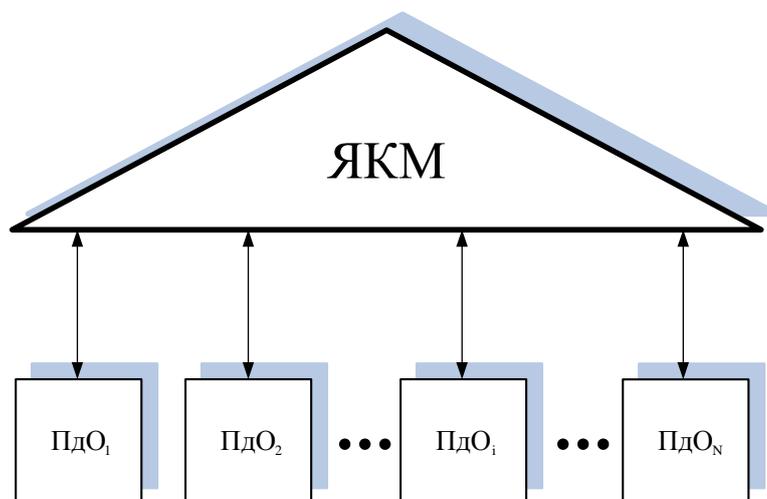


Рис. 1.2. “Пирамида” знаний – полная картина мира

Речь идёт, в частности, о построении формального языка категориального уровня, обеспечивающего эффективное обобщение, перенос и внедрение приёмов, методов и теорий из одной $ПдО$ в другую, что, по-видимому, повысит эффективность процесса междисциплинарного взаимодействия. Как следует из рассмотренного выше анализа взаимодействия общих научных дисциплин, именно теория

познания определяет основные принципы и направления исследовательского процесса в системе Λ . Вместе с тем важно отметить радикальное изменение и самой системы научного познания. Размываются четкие границы между практической и познавательной деятельностью. В системе научного знания происходят интенсивные процессы их дифференциации и интеграции, развиваются комплексные междисциплинарные и трансдисциплинарные исследования, новые способы и методы познания, методологические установки, появляются новые элементы картины мира, выделяются новые, более сложные типы объектов познания, характеризующиеся историзмом, универсальностью, сложностью организации, которые раньше не поддавались теоретическому (математическому) моделированию. Отметим, что проблемы и закономерности указанного взаимодействия исследуются в рамках нового междисциплинарного направления – *синергетики*, которое, вообще говоря, можно рассматривать как современный этап развития кибернетики и системных исследований.

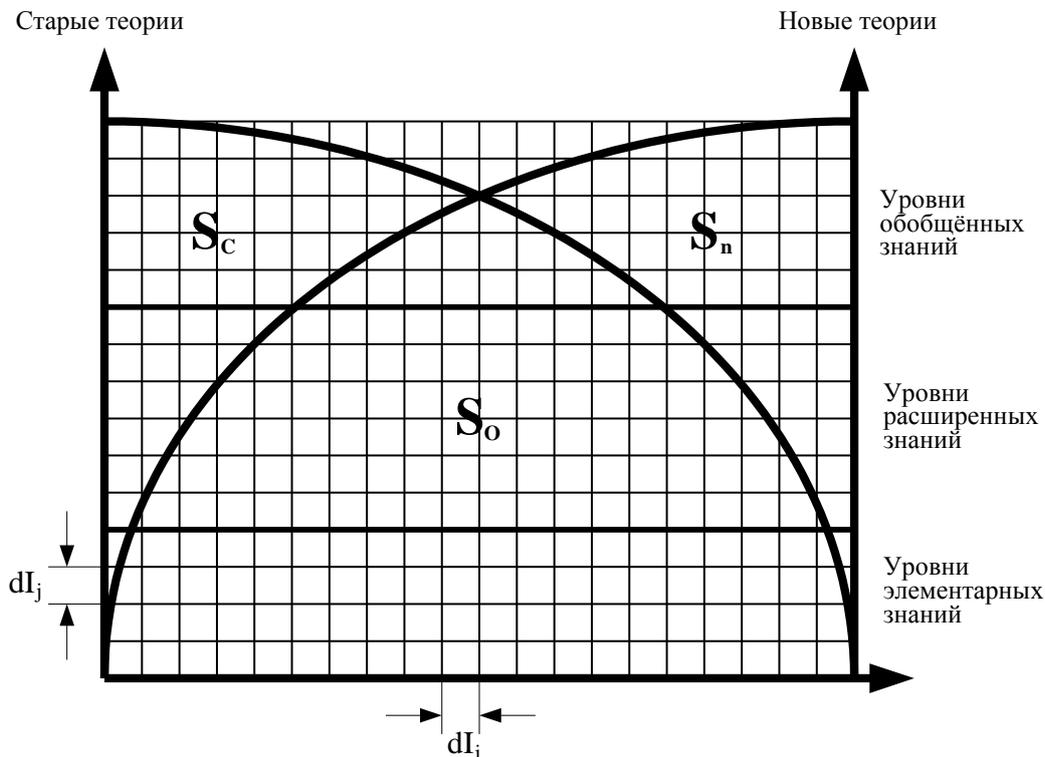


Рис. 1.3. Развитие знаний

Синергетика (от греч. $\sigma\upsilon\nu$ – «совместно» и $\epsilon\rho\upsilon\sigma$ – «действующий») – междисциплинарное направление научных исследований, задачей которого является изучение природных явлений и процессов на основе принципов самоорганизации систем (состоящих из *подсистем*) [12–14]. Синергетика изначально позиционировалась как междисциплинарный подход, как принципы, управляющие процессами самоорганизации, одни и те же безотносительно к природе систем. Нас же интересуют конкретные научно-практические задачи, в их конкретной постановке – разработка развивающихся знание-ориентированных систем, открытых баз знаний

базовых доменов прикладных областей и их пополнение, разработка эффективных механизмов извлечения знаний, их обнаружения и формального представления.

1.1.2. Концептуальная парадигма работы со знаниями

В своё время В.М. Глушков в качестве одного из основных направлений развития кибернетики выделил *поиск новых принципов построения ЭВМ* [15]. В рамках этого направления, в результате “эволюции человеческого знания, напоминающей восхождение по спирали или лестнице с огромными ступенями” была сформирована и развивается *концептуальная парадигма работы со знаниями*. Это понятие является сложным, опирается на два понятия, определяемые терминами: “концептуальная парадигма” и “работа со знаниями”.

“Парадигма [15] часто выглядит как простая, элегантная и правдоподобная концептуализация данных, на основании которой возможно объяснение большей части наблюдаемых фактов. Она выражает некоторое убеждение, критерий ценности, технический приём, разделяемый и применяемый членами определённого общества. Парадигма столь же существенна для науки, как наблюдение и эксперимент.

После того как парадигма сформирована и начинается поступательное развитие науки, большинство известных фактов, как правило, находят объяснение. Однако в процессе исследований и наблюдений неминуемо обнаруживаются такие феномены, которые не находят своего объяснения. Их становится всё больше и больше, что приводит к необходимости смены существующей парадигмы. Так назревает новая революция и подъём на новую ступень развития знаний.

Таким образом, парадигмы несут в себе не только познавательный, но и утверждающий характер и поэтому существенно влияют на процесс развития науки, ускоряя или замедляя его. ... Когда парадигму принимает большая часть научного сообщества, она становится нормативной точкой зрения и инструментом – мощным катализатором научного прогресса”.

Термин “*работа со знаниями*” по принадлежности относится к области инженерии знаний, а по принадлежности к онтологической категории – к понятию-процессу. Последний общепринято определять экстенционально, через перечисление входящих в него подпроцессов. К таким подпроцессам работы со знаниями обычно относят *извлечение и приобретение знаний, представление знаний и манипулирование знаниями*. В область аспектов работы со знаниями, кроме указанных подпроцессов, включают также методы и средства. В онтологическом инжиниринге в эту область включают онтологию процессов, онтологию задач и онтологию объектов [4, 5, 16, 17 и др.].

1. Извлечение знаний из различных источников, в том числе из естественно-языковых объектов, включает два основных раздела: анализ исходной информации, формализацию качественных знаний и интеграцию знаний. Первый раздел связан с созданием методов, позволяющих переходить от знаний, выраженных, в том числе в естественно-языковой форме, к их аналогам, пригодным для ввода в память знание-ориентированной информационной системы. Вторым разделом связан с интеграцией знаний, получаемых от различных источников, в некоторую взаимосвязанную и непротиворечивую систему знаний о ПдО. Без решения этой проблемы вряд ли возможно создать полное представление о ПдО.

2. **Приобретение знаний.** Знаний, содержащихся в источниках информации, отчуждённых от специалиста, как правило, недостаточно. Значительную часть профессионального опыта эти специалисты не могут выразить словесно (профессиональное умение или интуиция). Поэтому для того, чтобы приобрести такие знания, нужны специальные приёмы и методы. Они используются в инструментальных системах по приобретению знаний, создание которых – одна из задач инженерии знаний. Полученные от экспертов знания нужно оценить с точки зрения их соответствия ранее накопленным знаниям и формализовать для ввода в память системы. Кроме того, знания, полученные от различных экспертов, необходимо согласовать между собой, так как нередки случаи, когда эти знания оказывались внешне несовместимыми и даже противоречивыми. Рассматриваемый подпроцесс включает такие разделы как *организация работы с экспертами, оценка и формализация знаний и согласование знаний*.

3. **Представление знаний.** Этот процесс предполагает разработку формальной научной теории, включающей построение *модели знаний, системы представления знаний* и *базы знаний* (БЗ). В инженерии знаний системы *представления знаний* включают совокупность процедур, необходимых для записи знаний, извлечение их из памяти и поддержки хранения знаний в рабочем состоянии. Системы представления знаний оформляются как БЗ, являющиеся естественным развитием баз данных (БД). Представление знаний – это соглашение о том, как и в какой формальной теории описывать реальный мир. В естественных и технических науках принят следующий традиционный способ представления знаний. На естественном языке вводятся основные понятия и отношения между ними. Но при этом используются ранее определённые понятия и отношения, смысл которых уже известен. Далее устанавливается соответствие между характеристиками (чаще всего количественными) понятий и подходящей математической моделью. Основная цель представления знаний – строить математические модели реального мира и его частей.

Системой представления знаний называют совокупность средств, позволяющих:

- описывать знания о предметной области с помощью языка представления знаний;
- организовать хранение знаний в системе (накопление, анализ, структурное обобщение и организация знаний);
- выводить новые знания из имеющихся и объединять их;
- находить требуемые знания;
- обновлять знания;
- осуществлять интерфейс между системой и пользователем.

4. **Манипулирование знаниями.** К этому процессу относятся такие разделы как *пополнение знаний, классификация знаний, обобщение знаний, вывод на знаниях* (резольюционные методы, квазиаксиоматические системы и системы правдоподобного вывода), *рассуждения с помощью знаний, объяснения на знаниях*, решения прикладных задач ПДО.

Новые знания, поступающие в БЗ, должны вместе с теми сведениями, которые уже были ранее записаны в неё, сформировать расширение поступивших знаний. Среди этих процедур особое место занимают *псевдофизические логики* (времени, пространства, действий и т. д.), которые, опираясь на законы внешнего мира, пополняют поступившую в БЗ информацию. Знания ЗОИС образуют упорядоченные структуры, что облегчает поиск нужных знаний и поддержание работоспособности БЗ. Для этого используются различные классифицирующие

процедуры. Типы классификаций могут быть различными: родовидовые, “часть–целое”, ситуативные (когда в одно множество объединяются знания, релевантные некоторой типовой ситуации). Указанные процедуры строятся на основе *теории классификаций*. В процессе классификации часто происходит абстрагирование от отдельных элементов описаний (фрагментов знаний об объектах или явлениях), появляются обобщённые знания, что приводит, в конце концов, к абстрактным знаниям, для которых нет прямого прообраза во внешнем мире.

Вывод на знаниях зависит от модели, которая используется для их представления. Если в качестве представления используются логические системы или продукции, то вывод на знаниях становится близок к стандартному логическому выводу. Это же происходит при представлении знаний в каузальной форме. Во всех этих случаях в ЗОИС используются методы вывода, опирающиеся на идеи метода резолюций.

Возможность появления в памяти ЗОИС новых фактов и сведений приводит к тому, что начинает нарушаться принцип монотонности, лежащий в основе функционирования всех систем, изучаемых традиционной математической логикой. Немонотонность вывода в открытых системах вызывает немалые трудности. В последние годы сторонники логических методов в ИИ делают попытки построить новые логические системы, в рамках которых можно было бы обеспечить немонотонный вывод. По сути системы, с помощью которых представляются знания о предметных областях, не являются строго аксиоматическими, как классические логические исчисления. Поэтому и системы, которые возникают при таких условиях, следует называть *квазиаксиоматическими*. В таких системах вполне возможна смена исходных аксиом в процессе вывода. И, наконец, еще одна особенность вывода на знаниях – неполнота сведений о предметной области и протекающих в ней процессах, неточность входной информации. А это означает, что выводы в ЗОИС носят не абсолютно достоверный характер, как в традиционных логических системах, а приближенный, правдоподобный характер. Такие выводы требуют развитого аппарата вычисления оценок правдоподобия и методов оперирования ими.

Поскольку ЗОИС принимают решения, опираясь на знания, которые могут быть неизвестны пользователю, решающему свою задачу с помощью ЗОИС, то он может усомниться в правильности полученного решения. ЗОИС должна обладать средствами, которые могут сформировать пользователю необходимые объяснения. Объяснения могут касаться процесса получения решений, оснований, которые были для этого использованы, способов отсеечения альтернативных вариантов и т. п. Все это требует развитой *теории объяснений*.

На основе проведенного выше анализа на рис. 1.4 синтезирована обобщённая схема ввода и обработки входных данных, семантической переработки информации и работы со знаниями. В ней также отражены связи между соответствующими процессами, последовательности которых составляют информационные технологии обработки данных, текст-процессинга, работы со знаниями и логико-информационного подхода [18, 19], составляющих основу методологии проектирования знание-ориентированных информационных систем с обработкой ЕЯО и формализованных знаний.

Концептуальная парадигма работы со знаниями утвердилась в связи и по мере развития концепции интеллектуальных информационных систем, согласно которой главной информационной единицей компьютерной обработки стало “знание”. Независимо и в рамках разных областей предметных знаний развиваются

информационные технологии: автоматической обработки естественного языка – в компьютерной лингвистике, представления знаний – в математической логике. Начали появляться научные разработки, предлагающие конструктивные подходы к интеграции указанных технологий.

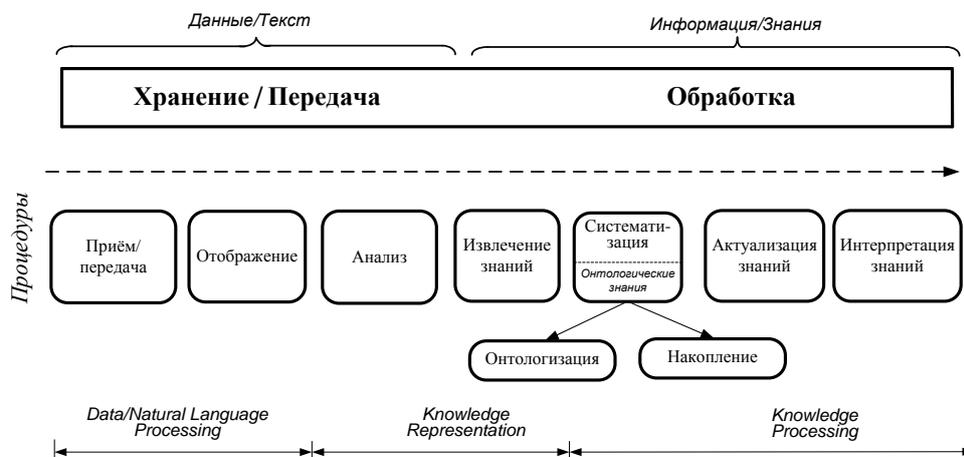


Рис. 1.4. Схема работы со знаниями

1.1.3. Сущностный анализ понятия “знание”

Рассматривая знания как нечто объективное, феноменальное, отражающее исходную, преобразующую и конечную составляющие процесса познания (“знание посредством знаний преобразуется в новое знание”), заметим, что совокупность научных дисциплин, имеющих непосредственное отношение к процессам мышления, понимания, осознания и получения новых знаний, не может обходиться без выработки своих собственных понятий, определённым способом фиксирующих свойства и закономерности её объектов. Из таких основополагающих понятий, как *смысл, знание, (знаковая) система, текст, объект, отношение, предмет, язык, структура, связь*, рассмотрим только первые два понятия, выражающие собой наибольшую проблемность исследований в области построения систем обработки знаний, в том числе и ЗОИС, оперирующих ЕЯО. Но, прежде рассмотрим определение категории “понятие”, являющейся “кирпичиком” для более сложных форм мышления – суждения и умозаключения.

Понятие обычно определяют как одну из основных форм мышления; этим подчеркивается важная роль его в познании. А само мышление может рассматриваться как процесс оперирования понятиями; именно благодаря понятиям мышление приобретает характер обобщённого отражения действительности. Выделение предметов – одна из основных функций понятия в процессе познания. Выделяемые в понятиях классы предметов – это опорные пункты в познании, вокруг которых концентрируются все знания [20].

Понятие есть целостная совокупность суждений, в которых что-либо утверждается об отличительных признаках исследуемой сущности, ядром которой являются суждения (или утверждения) о наиболее общих и в то же время существенных признаках этой сущности.

Каждое понятие характеризуется объёмом и содержанием. Объём и содержание понятия – две взаимосвязанные стороны понятия. **Объём** – класс обобщён-

ных в понятии предметов, *содержание* – совокупность (обычно существенных) признаков, по которым произведено обобщение и выделение предметов в данном понятии. Объём понятия является определяющим при формировании иерархической структуры соответствующего онтографа (ОГ), а содержание – при аксиоматизации его (ОГ) вершин.

Известно, что теории смыслов в завершённом виде не существует, а потому и общепринятого определения понятия “смысл” также нет, как нет однозначного понимания и с точки зрения его формального представления и преобразования в компьютерной системе. В [10, 11, 15, 20–22] этот термин вводится с гносеологических позиций, через его свойства и атрибуты, взаимодействие с другими объектами. Вот наиболее распространённые определения с точки зрения проектирования компьютерных систем.

Смысл – это есть то, что делает знаковые системы текстом [23].

Смысл – некоторый непрерывный, невербальный конструкт, *осмысление* – интерпретация в индивидуальной концептуальной системе [11].

Смысл – это есть сеть значений в определённых позициях и оперативный алгоритм для решения проблем [15].

И далее [23] отмечается, что “...природа смысла может быть раскрыта только через одновременный анализ *семантической триады*: “*смысл–текст–язык*”. Текстовое раскрытие смысла происходит через знаковые системы, которые мы воспринимаем как языки. Таким образом, каждый элемент указанной выше триады раскрывается через два других. Включая в триаду язык, мы вносим представление о том, что сама триада становится возможной, только когда есть *НАБЛЮДАТЕЛЬ* – носитель сознания, воспринимающий тексты и оценивающий смыслы. Триада становится синонимом сознания”, скажем точнее – языкового сознания.

Определение 1.1.

1. **Понимание** – это фиксация смысла естественно-языкового объекта.
2. **Фиксация смысла** естественно-языкового объекта – это проецирование ЕЯО на языковую картину мира.
3. **Смысл** – это индекс проекции ЕЯО на ЯКМ.

Известно также достаточно много определений понятия “знания”, которые по способу представления можно разделить на две основные группы – общего характера и адаптированные применительно к их компьютерной обработке.

Из первой группы можно выделить следующие, наиболее характерные определения.

Знание – целостная и систематизированная совокупность научных понятий о закономерностях природы, общества и мышления, накопленная человечеством в процессе активной преобразующей производственной деятельности и направленная на дальнейшее познание и изменение объективного мира [21].

Знания есть совокупность понятий и представлений об объективной действительности, их внутренне взаимосвязанных систем (суждений, положений, концепций, теорий и т. д.), вырабатываемых обществом в процессе познания и преобразования мира. Зародившееся в эпоху первобытного духовно-физического синкретизма трудовой деятельности человека и продуцируемое в последующем в специализированной (научной) деятельности общества, в первую очередь для достижения практических целей, знание всё шире охватывает объект, всё глубже проникает в его тайны, т. е. развивается как в экстенсивном, так и в интенсивном плане [24].

Эти примеры определений достаточно корректны, дают общее представление о том, что такое знание и познание как когнитивные процессы, но мало полезны

для практических приложений и проектирования соответствующих компьютерных систем.

Из второй группы можно выделить следующие определения, признающиеся многими исследователями.

Знание – это информация, структурированная (формализованная) таким образом, что над ней можно производить логические операции (прежде всего логического вывода) [25].

Знание – это сложно организованная информационная структура, хранимая в памяти интеллектуальных информационных систем и включающая в себя сведения об объектах и отношениях предметной области, процессах взаимодействия объектов во времени и в пространстве, правилах осуществления логического вывода [3].

Определения второй группы уже дают некоторое обобщённое представление о системе, основанной на знаниях, но все же этого недостаточно, чтобы построить указанную систему. Для этого необходимо также, по крайней мере, знать:

- чем знание отличается от подобных ему понятий – данных и информации;
- какими общими свойствами обладают знания;
- какие существуют источники знаний;
- какие существуют способы представления и обработки знаний в КС;
- какие существуют механизмы выявления новых знаний.

Поэтому в задачах информатики можно (и более удобно) определить категорию *знания* косвенно – через его свойства и методы обработки. В частности, применительно к задаче *извлечения знаний* из текста корректным будет

Определение 1.2. Под термином “*знание*” будем понимать совокупность фактов, извлечённых из заданного текста и представленных в виде формул (возможно, некоторого ограниченного фрагмента) языка предикатов первого порядка или другой подходящей логической системы, которые отражают смысл, заложенный в исходном тексте (предполагается существование модели, на которой извлечённое знание можно проверить на истинность или ложность).

Свойства знаний

Известно [3, 4], что знания характеризуются рядом свойств, отличающих их от традиционных моделей данных. К таким свойствам можно отнести.

Внутренняя интерпретируемость. При хранении знаний в памяти ИИС наряду с традиционными элементами данных хранятся информационные структуры, позволяющие интерпретировать содержимое соответствующих ячеек памяти.

Структурированность. Знания состоят из отдельных информационных единиц, между которыми можно установить классифицирующие отношения: род – вид, класс – элемент, тип – подтип, часть – целое и т. п.

Связность. Между информационными единицами предусматриваются связи различного типа: причина – следствие, одновременно, быть рядом и др. Данные связи определяют семантику и прагматику предметной области.

Семантическая метрика. На множестве информационных единиц, хранимых в памяти, вводятся некоторые шкалы, позволяющие оценить их семантическую близость. Это позволяет находить в информационной базе знания, близкие к уже найденным.

Активность. С помощью данного свойства подчеркивается принципиальное отличие знаний от данных. Выполнение тех или иных действий в ИИС инициируется состоянием базы знаний. При этом предполагается, что появление новых фактов и связей может активизировать систему.

Отметим, что это свойство является фундаментальным для понятия “знание”. Исходя из него введено понятие онтолого-управляемой архитектуры КС, так как понятие “*онтологическая база знаний*” является видовым по отношению к понятию “*база знаний*”.

Кроме того, свойство актуальности знаний может порождать процесс актуализации, который имеет определяющее значение в цепочке переходов «*хранилище_данных* → *востребованная_информация* → *знания*». Однако для нас более актуальной (с практической точки зрения) является проблема не первичности компонент цепочки, а их принадлежность. Известны следующие описания принадлежности указанных компонент, признаваемые многими исследователями.

Определение 1.3. *Данные* – это зафиксированные объекты или явления материального или абстрактного мира.

Определение 1.4. *Информация* – это понятие категории верхнего уровня, характеризующее объекты и явления материального и абстрактного мира, представленные в виде символов, которыми оперирует человек или компьютер.

На любом этапе преобразования требование актуализации соответствующей компоненты является обязательным (но не достаточным) условием перехода в более высокую степень абстрактности. И если для первого перехода («*хранилище_данных* → *востребованная_информация*») механизмы актуализации достаточно хорошо изучены и проработаны, то для второго перехода («*востребованная_информация* → *знания*») указанные механизмы слабо изучены и реализованы в разрозненных экспериментальных разработках.

Проблема извлечения знаний из текста представляется не только не тривиальной, но и весьма сложной, несмотря на несомненные достижения Computer science в этой области. Как справедливо замечено в [25], “Текст есть знаковая конструкция и часто содержит знание. Но текст есть не знание, а только его источник. Знания из текста ещё нужно извлечь. Человеку или КС”.

Выполненный анализ позволяют синтезировать некоторую обобщённую схему “эволюционирования” знаний, представленную на рис. 1.5. Заметим при этом, что в языке данной монографии словоформы терминов “знание”, “сознание” и “познание” имеют общий корень, а сами термины также образуют триаду, компоненты которой фиксируют объект, субъект и процесс познания.

Входом является некоторая знаковая система, которую можно интерпретировать как совокупность данных. Далее происходит “осмысление” входных данных, для чего обязательно присутствие *Наблюдателя* (человек, компьютер), как носителя высших форм мышления – языкового сознания. И если у него имеется “внутренний языковой интерпретатор, настроенный на данную знаковую систему”, то данные превращаются в информацию.

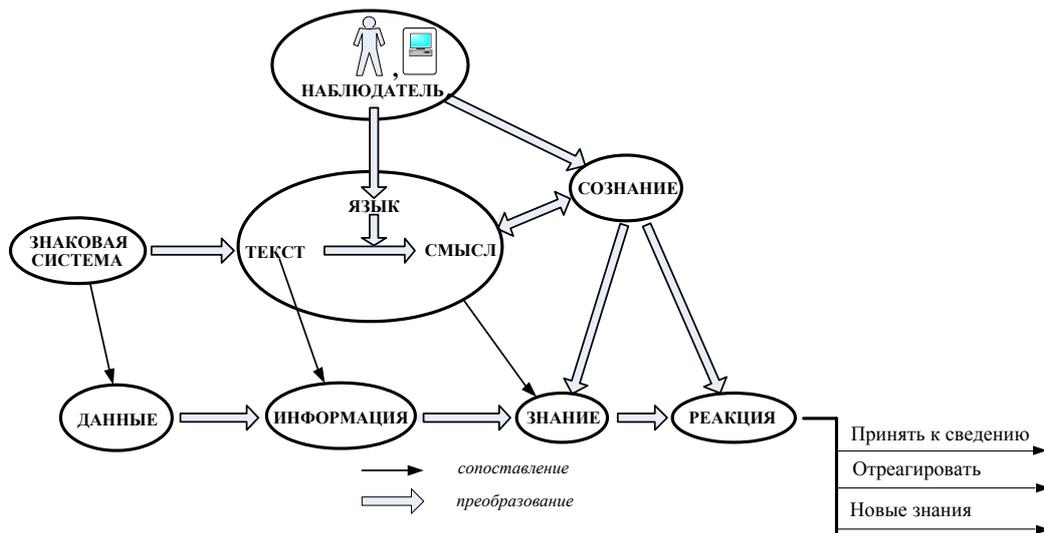


Рис.1.5. Обобщённая схема “эволюционирования” знаний

Далее в работу включается “смысловой интерпретатор” (если он имеется у *Наблюдателя*), который превращает информацию в знания или происходит осознание поступившей информации. На заключительном этапе обработки входного сообщения происходит естественная реакция человека: принять к сведению, отреагировать или пополнить базу знаний новой порцией знаний [10].

1.1.4. Категориальный уровень знаний

Термину “*компьютерная онтология верхнего уровня*” в философской литературе соответствует синоним “*система категорий*”. Изучению последней посвящены многочисленные научные труды философов, логиков, лингвистов [6–9, 26]. Разработка методологических основ построения многоуровневых компьютерных онтологий предопределила необходимость системного анализа, необходимого для построения строгих математических моделей произвольной предметной области (проблемы, задачи) и их интерпретации для получения конкретных результатов, и в частности компьютерной онтологии верхнего уровня (ОВУ) и *системы категорий* как знаний об общей структуре действительности [27–30]. Система категорий определяется как последовательность таких ступенек познания, которые должны быть пройдены в исследовании любого объекта. Осознание того, что категории отражают всеобщие характеристики любого объекта, описывают общую структуру взаимодействия не только субъекта с объектом, но и любых объектов между собой, достигается опосредованно, как вывод, следующий из доказательства того, что деятельность может быть успешной лишь в том случае, если она подчиняется общим известным законам Вселенной [1]. Система категорий выступает по отношению ко всякому новому знанию как некоторый фиксированный *категориальный каркас*, который имеется до познания данного конкретного класса явлений и определяет формы этого познания, так что всякое новое содержание, доставляемое конкретным исследователем, подводится под одну из категорий и упорядочивается по соответствующим правилам [31, 32].

Результатом системного анализа, в частности, являются приведенные ниже утверждения, описывающие *методологические основы исследования и систематизации категорий* [1, 27, 31–37]:

- в системе категорий формируются предельно общие принципы, которые должны лечь в основу систематизации знаний в целом, при этом всеобщность категорий не означает их тавтологичности;
- единство онтологического и гносеологического подходов;
- гносеологическим принципом субординации категорий является движение от абстрактного к конкретному;
- рассмотрение категорий как ступенек познания;
- необходимость строгих определений категорий;
- необходимость общего понятийного каркаса как основы интерпретации частных научных теорий;
- категориальные отношения субъекта к объекту являются проявлением категориальных отношений между любыми объектами;
- возможно построение различных систем категорий (допущение множественности систем), которые будут отражать категориальные каркасы структур знаний, образующихся при решении частно-научных задач. Эти различные системы категорий должны быть субординированы через единый, базовый, *неопределяемый минимум понятий*, составляющий базовую систему, дающую общий язык для построения других систем (онтологию категорий верхнего уровня). Только с помощью базовой системы категорий можно решить следующие общенаучные задачи:
 - а) выделить семантические множители для всеобщих понятий единого языка науки;
 - б) последовательно и строго определить всеобщие категории, исходя из неопределяемого минимума;
 - в) дать основу для построения онтологической атрибутивной модели материи на том уровне, который отражён современной наукой;
 - г) обеспечить разработку системы логических операций с учётом направленности на познание всеобщих сторон любого объекта;
 - д) разработать категориальный каркас для построения общей системной методологии решения задач анализа и синтеза;
 - е) создать “точный” категориальный язык для более детальных категориальных исследований.

Определения категорий в ОВУ носят функциональный характер, т. е. они отличаются друг от друга специфической функцией в теории познания.

Объединённые из различных источников принципы и механизмы построения системы категорий позволяют синтезировать методологические основы проектирования компьютерной ОВУ.

Анализ принципов генерирования категорий верхнего уровня

В своем первом трактате “Категории” Аристотель представил десять основных категорий для классификации чего-то, что может быть сказано или утверждено о чём-то. Метод Аристотеля образования новых категорий по родовому принципу и его отличительным свойствам является фундаментальным и по своей сути представляет *принцип дихотомии*, т. е. разделение на две противоположные части. Позднее неоплатоник Порфирий привел в порядок категории Аристотеля в виде дерева, а средневековые логики развили его в более детализированную иерархию, которую они называли “дерево Порфирия”.

Фундаментальный принцип формирования категорий предложил Кант, построив свою таблицу, которая состоит из четырех групп, по три категории в каждой. Он рассматривал эту таблицу как принципиальную каркасную структуру, отличную от структуры Аристотеля. Аналогичную триадическую структуру поддерживал и Гегель.

Наиболее полное развитие *триадический принцип* формирования категорий получил в работах Пирса. Он принял за основу структуры триад Канта и Гегеля, и после расширенного анализа осознал, что существует три базовые категории более общего характера. Последние он назвал *Первичность*, *Вторичность* и *Третичность*.

“Первое есть концепция бытия или существование, *независимо от чего-нибудь* еще. Второе есть концепция бытия, *связанного с* чем-либо ещё. Третье есть концепция *посредничества*, с помощью чего первое и второе объединены в отношение” [29].

В различении триад Пирс подчеркивал равный статус всех трёх категорий. Формально, эти три вида категорий характеризуются минимальным числом сущностей, которые должны быть включены в их определение. *Первичность* может быть определена одноместным предикатом $P(x)$, который описывает сущность x её встроенными качествами, независимыми от чего-нибудь внешнего к x . *Вторичность* требует двухместного предиката $Q(x,y)$, который описывает некоторую реакцию между сущностью x и независимой от x сущностью y . *Третичность* требует как минимум трехместного предиката $R(x,y,z)$, который описывает, как сущность x служит связующим звеном между двумя другими сущностями y и z . Пирс утверждал, что нет необходимости выходить дальше за границы *Третичности*, так как все следующие “*ичности*” могут быть определены в терминах триад.

Немного позже, независимо от Пирса, Уайтхед разработал свою метаонтологию, которая включает восемь “категорий существования”, составляющих две триады Пирса, дополненные двумя категориями для генерирования комбинаций. Уайтхед определил категории для *действительных сущностей*, *присоединений* (prehension) и *объединений* (nexus), которые представляют триаду материальных *Первичности*, *Вторичности* и *Третичности*. Материальная действительность существует независимо от каких-либо других сущностей. *Присоединение* есть направленное отношение или реакция между двумя сущностями. *Объединение* – связывание двух или более присоединений.

Описанный триадический принцип формирования категорий или трихотомии называют ещё *принципом Пирса* формирования онтологических категорий.

Кроме известных работ по математике Лейбниц внёс значительный вклад в разработку формальных математических основ формирования онтологических категорий и определения операций для их обработки. Вместо оперирования терминами категорий он назначал категориям определенные простые числа и операцией умножения генерировал новые категории, а операцией деления определял принадлежность “тип-супертип”. Произведения простых чисел формировали своего рода иерархию, названную решёткой. Со временем, такое генерирование категорий было названо *комбинационным методом Лейбница*.

Три описанных выше фундаментальных метода генерирования онтологических категорий, используемых как для метаонтологий верхнего уровня, так и для онтологий прикладных областей равноценны и нет никакого формального способа определения приоритетной значимости одного из них. При разработке онтологии домена приоритет устанавливается самим исследователем на основе его интуиции и профессиональных знаний. Примером являются многочисленные варианты мета-

онтологий, разработанные нашими предшественниками при создании семейств компьютерных прикладных программ (системы Cус, Chat-80, Microcosmos, WordNet и др.).

Следует подчеркнуть, что независимо от того, как названы сгенерированные категории, важными есть только множества (концептуальных) отношений, которые связывают концепты в онтологии, и система аксиом, правил и фактов, выбранных для конкретной прикладной области.

Синтез категорий верхнего уровня

Иерархическая структура на рис. 1.6 является результатом концептуализации понятий согласно рассмотренных выше трёх фундаментальных принципов классификации [26]. Символ $У$ обозначает нейтральное интуитивное понятие универсального типа, или *Универсум*.

Формально, $У$ является типом наивысшего (нулевого) уровня категоризации, не имеет никакой дифференциации, которому удовлетворяют следующие аксиомы [26]:

- существует некоторый x такой, что: $(\exists x)У(x)$;
- всякий x является элементом $У$: $(\forall x)У(x)$;
- каждая категория есть подкатегорией $У$: $(\forall k: \text{Категория}) k \leq У$, где \leq – знак частичного порядка;
- все подкатегории определяются дополнительными отличиями в $У$, чтобы показать, как они отличаются от $У$ и каждая от другой.

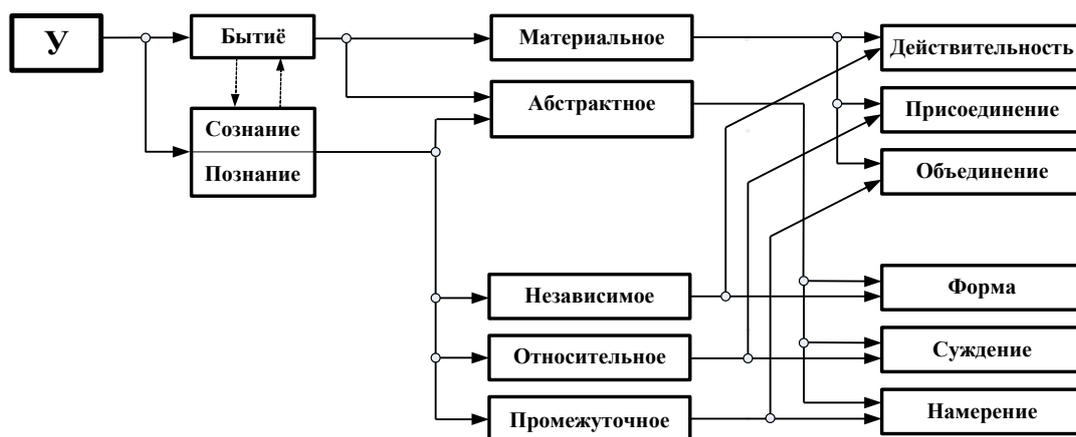


Рис.1.6. Иерархия категорий *Материальное* и *Абстрактное* и *Независимое*, *Относительное* и *Промежуточное*

Далее $У$ разворачивается в две группы категорий второго уровня (первый уровень категорий *Бытие*, *Сознание* и *Познание* будут рассмотрены ниже). Для первой группы: категория *Материальное* – для чего-нибудь, составленного из материи (энергии), и категория *Абстрактное* – для сугубо информационных структур. Вторая группа: *Независимое*, *Относительное* и *Промежуточное*.

Третий уровень делит указанные группы на триады согласно различению Пирса о Первичности, Вторичности и Третичности. Отсюда две группы категорий:

Действительность, Присоединение и Объединение – для *Материального*, а также *Форма, Суждение и Намерение* – для *Абстрактного*.

Итак, шесть категорий третьего уровня сгенерировано пересечением двустороннего и трёхстороннего различий, который отвечает принципу комбинирования примитивов Лейбница, используемого при генерации категорий.

Независимое – категория, включающая материальную *действительную* сущность и абстрактную *форму*, которые характеризуются присущими им качествами или Первичности. В логике независимые сущности могут быть представлены именем сущности или одноместным предикатом $P(x)$, описывающим некоторый аспект сущности x , не принимая во внимание чего-либо внешнего по отношению к x .

Относительное – категория, которая включает категории материального – *Присоединение* и абстрактного – *Суждение*, которые могут быть представлены двухместными предикатами $Q(x, y)$, и отвечает категории Вторичности Пирса. *Присоединение* – это категория материального, которая связывает сущность x , которая присоединяется, с сущностью y , которая присоединяет. *Суждение* – это категория абстрактного, которая связывает сущность x с сущностью y , описанной с помощью x .

Промежуточное – категория, которая отвечает Третичности Пирса и включает категории материального – *Объединение* и абстрактного – *Намерение*. Представляется трёхместным предикатом $R(x, y, z)$. Это трёхместное отношение не может быть выражено конъюнкцией двуместных отношений.

Приведенное описание и рис. 1.6 демонстрируют использование принципа множественного наследования, при котором каждая категория нижнего уровня наследует свойства от двух других категорий предыдущего уровня.

Вытекая из принципа трихотомии в онтологию верхнего уровня включают также **третью группу** категорий третьего уровня, а именно – *Продолжительное* и *Событийное*. Это связано с тем, что “все вещи существуют в движении, но некоторые вещи подвергаются быстрому изменению, в то время как другие остаются сравнительно постоянными” [26]. Постоянный объект имеет постоянные атрибуты-характеристики и называется *Продолжительным*, а процесс (или событие), не имеет постоянных характеристик и называется *Событийным*. Он может быть идентифицирован только своим расположением в некоторой системе координат “пространство-время”. Классификация таких сущностей, как *Продолжительное* или *Событийное*, зависит от масштаба времени и уровня детализации с точки зрения некоторого исследователя.

Таким образом, все производные категории образуются комбинацией трёх базовых групп категорий определения Универсального (рис. 1.7):

Материальное – Абстрактное;

Независимое – Относительное – Промежуточное;

Продолжительное – Событийное.

Каждая из производных категорий представляет комбинацию категорий, из которых она была получена: *Объект*, например, может быть представлен акронимом МНП: Материальное-Независимое-Продолжительное.

Можно заметить, что на рис. 1.7 представлены не все категории, которые могли быть сгенерированы по комбинационному методу Лейбница [26].

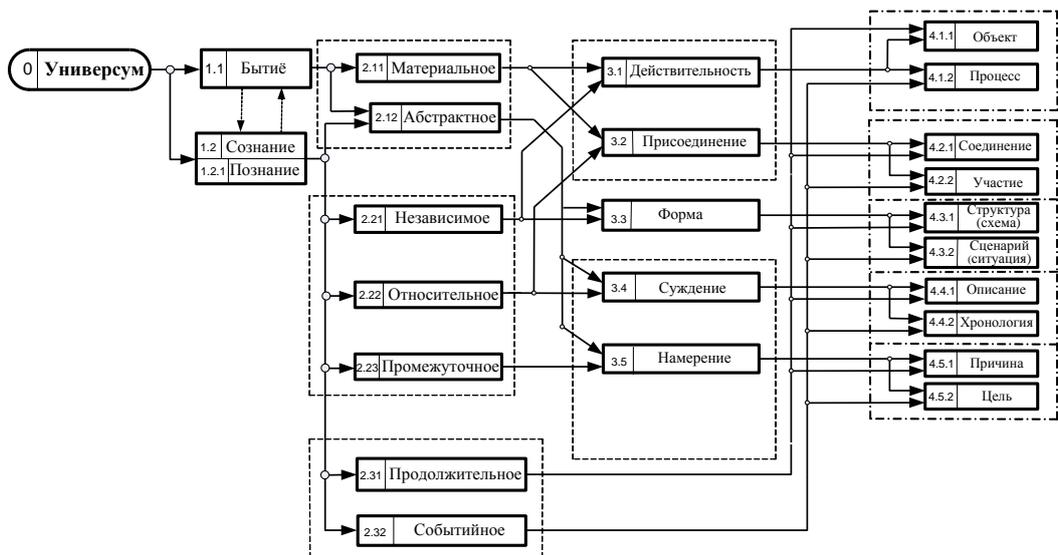


Рис. 1.7. Иерархия категорий верхнего уровня

Рассмотрим более детально построение онтологической структуры для понятий – категорий верхнего уровня [26, 32, 38].

- Самый верхний (нулевой) уровень (порядковый номер уровня указан слева в прямоугольнике) представлен одной вершиной, которая названа ”Универсум“ (У).

- На первом уровне две вершины – ”Бытие“ и ”Сознание/Познание“. Этим подчёркивается разная природа материального и абстрактного. На следующих уровнях учитывается взаимное пересечение дуг, которые выходят из вершин второго уровня категорий.

Опишем включённые в иерархию категории соответственно с их структурными связями, а также приведём аксиомы (где возможно) для соответствующих категорий. Подчеркнём, что каждая категория в иерархии наследует все свойства и аксиомы каждой выше расположенной категории.

Известно, что кванторы темпоральной логики переводятся в логику первого порядка (FOL) следующим образом:

$$\Box P(x) \Leftrightarrow (\forall t: T) P(x, t)$$

$$\Diamond P(x) \Leftrightarrow (\exists t: T) P(x, t),$$

где $t \in T$ ассоциируется со временем и эти кванторы в формулах читаются так:

$\Box P(x)$ – необходимо, чтобы в любой момент времени t в будущем была истинна формула $P(x)$;

$\Diamond P(x)$ – возможно, что в некоторый момент времени t в будущем будет истинна формула $P(x)$.

Темпоральные кванторы связаны между собой отношением двойственности

$$\Diamond P(x) \Leftrightarrow \neg \Box \neg P(x).$$

Второй уровень представлен семью вершинами.

2.11. **Материальное** (М) – это сущность, которая имеет местонахождение в системе координат пространство-время. Для него можно привести следующие аксиомы:

- что-нибудь материальное x существует в некотором месте y :

$$(\forall x:M)(\exists y:Место) loc(x, y), \text{ где } loc(x, y) \text{ – соответствующий предикат}$$

местонахождения сущности x в месте y ;

- что-нибудь материальное x существует (находится) в некоторый момент времени t :

$$(\forall x:M)(\exists t:Время) time(x, t), \text{ где } time(x, t) \text{ – соответствующий предикат}$$

существования сущности x в момент времени t .

2.12. **Абстрактное** (А) – это сущность, для которой выполняются следующие аксиомы:

- нет никакой абстрактной сущности x , размещенной в месте (пространстве) y :

$$\neg(\exists x:A)(\exists y:Место) loc(x, y);$$

- нет никакой абстрактной сущности x , которая существует в момент времени t :

$$\neg(\exists x:A)(\exists t:T) time(x, t).$$

2.21. **Независимое** (Н) – это сущность x , которая характеризуется некоторой естественной первичностью, независимой от других сущностей. Для неё предикаты $Иметься(x, y)$ и $Иметься(y, x)$ не обязательно должны быть всегда истинными, где y – любая другая сущность, т. е.

$$(\forall x:H)(\exists t:T)\neg(\exists y)(Иметься(x, y, t) \vee Иметься(y, x, t)).$$

2.22. **Относительное** (О) – это сущность x во взаимоотношении к некоторой другой сущности y , для которой справедлива следующая аксиома

$$(\forall x:O)(\forall t:T)(\exists y)(Иметься(x, y, t) \vee Иметься(y, x, t)).$$

Для произвольной относительной сущности x должна существовать некоторая сущность y такая, что любой из предикатов $Иметься(x, y)$ или $Иметься(y, x)$ является обязательно истинным в любой момент времени $t \in T$ в будущем.

2.23. **Промежуточное** (Пр) – это сущность x , которая структурирует другие сущности (например, y и z) в их взаимозависимости. Для неё справедлива следующая аксиома

$$\begin{aligned} & (\forall x:Пр)(\forall y, z:Сущность)(\forall t:T) (Иметься(x, y, t) \wedge Иметься(x, z, t)) \\ & \Leftrightarrow (\forall x:Пр)(\forall y, z:Сущность)(\forall t:T) (Иметься(y, z, t) \vee Иметься(z, y, t)). \end{aligned}$$

2.31. **Продолжительное** (П) – это сущность, чьи условия истинности продолжают быть такими, что x распознаётся на некотором расширенном интервале времени t . Для неё справедлива следующая аксиома: продолжительное x имеет только пространственные характеристики и никаких временных. В любое время t , когда x существует, все составные сущности из x существуют в то же самое время t . Условия её распознавания для продолжительного независимы от времени.

Сущностной реализацией материального продолжительного является объект, а абстрактного продолжительного – схема, которая может использоваться для характеристики данного объекта.

2.32. **Событийное** (С) – это сущность, которая не имеет устойчивой идентификации на протяжении некоторого интервала времени. Для неё удовлетворяют следующие аксиомы:

- временные части событийного, которые называют стадиями, существуют в разное время;
- пространственные части событийного, которые называют участниками, могут существовать в то же самое время, но событийное может иметь разных участников на разных стадиях;

- нет никаких условий идентичности, которые могут быть использованы для идентификации двух событий, которые не перекрываются в области координат пространство-время.

Третий и четвертый уровни иерархии категорий сгенерированы пересечением категорий более высоких уровней.

- Для третьего уровня отведены следующие категории.

3.1. **Действительное** (МН) – это материальная сущность (М), чье существование является независимым от любой другой сущности. Как компоненты категория ”Действительное“ включает категории ”Объект“ и ”Процесс“.

3.2. **Присоединение** (МО) – это материальная сущность (М), связанная определенным отношением с некоторой сущностью или сущностями. Для неё выполняется проверка отношения ”*x* присоединить *y*“, выражающая факт включения *y* в сущность *x*. Если это имеет место, то категория ”Присоединение“ может быть выражена предикатом ”*присоединить(x, y)*“.

3.3. **Форма** (АН): абстрактная (А) информация является независимой от любого варианта реализации либо кодирования. Формы могут существовать в том же самом смысле, как и математические объекты типа множеств или отношений.

3.4. **Суждение** (АС): абстракция (А) связывает определенным отношением некоторую сущность или сущности. В логике утверждение (суждение) требует, чтобы данная абстракция отвечала некоторой конфигурации привлеченных сущностей. Сложные суждения представляются конструкциями составного предиката типа математического выражения (или диаграммы).

3.5. **Намерение** (АПр): абстракция (А) рассматривается как посредническая между другими сущностями.

- Четвертый уровень представлен следующими категориями.

4.1.1. **Объект** (МНП): действительность (МН) рассматривается как нечто продолжительное, сохраняющее тождественность больше некоторого промежутка времени. Хотя не существует материальной сущности, которая всегда остается постоянной, объект может быть распознан соответственно с характеристиками, которые остаются устойчивыми в течение их жизненного цикла.

4.1.2. **Процесс** (МНС): действительность (МН) рассматривается как явление событийное. В зависимости от масштаба времени и уровня детализации, одна и та самая действительная сущность может рассматриваться как устойчивый объект или динамический процесс.

4.2.1. **Соединение** (МОП): присоединение (МО) рассматривается как нечто продолжительное (больше некоторого интервала времени). Сущность, которая присоединяется, есть объект, находящийся в устойчивых взаимоотношениях к некоторой продолжительной сущности, которая присоединяет.

4.2.2. **Участие** (МОС): присоединение (МО) рассматривается как явление событийное в течение интервала времени. Сущность, которая присоединяется, есть процесс, находящийся в устойчивых взаимоотношениях с некоторой продолжительной сущностью, которая его присоединяет.

4.3.1. **Структура** (АНП): форма (АН) рассматривается как нечто продолжительное. Схема – это и есть абстрактная форма, чья структура не имеет времязависимых взаимоотношений. (Например, геометрические формы, синтаксические структуры предложений на некотором языке и пр.).

4.3.2. **Сценарий** (АНС): форма (АН) рассматривается как некоторая структурированная времязависимая последовательность событий. Примерами сценария являются компьютерная программа, ноты для исполнения музыки, кинофильм и т. п.

4.4.1. **Описание** (АОП) – это суждение (АО) относительно продолжительного, которое формулирует, как именно некоторая схема характеризует некоторый аспект (или конфигурацию) продолжительного (материального или абстрактного).

4.4.2. **Хронология** (АОС) – это суждение (АО) относительно событийного, которое связывает некоторый сценарий со стадиями некоторого событийного. Компьютерная программа, например, есть сценарий; компьютер, который выполняет программу, обеспечивает процесс, а номер команды в этом потоке команд, есть хронология.

4.5.1. **Причина** (АПрП) – это намерение (АПр) некоторого агента относительно некоторого продолжительного.

4.5.2. **Цель** (АПрС) – это намерение (АПр) некоторого агента, определяющего взаимодействие сущностей в ситуации. Основные аксиомы для Цели унаследованы от “своих” категорий высших уровней: Абстрактного, Промежуточного и Событийного. Они связывают цели с действиями и агентами.

- **Случайность.** Если агент x выполняет действие y , чья цель – местонахождение z , описанное суждением p , то возможно, что z не мог бы состояться или что p не мог бы быть истинным для z .
- **Успех или неудача.** Если агент x выполняет действие y , чья цель – ситуация z , описанная суждением p , то говорят, что x является успешным, если z происходит и p истинно для z ; иначе говорят, что x потерпел неудачу.

Построение полной иерархии концептов на основе таксономических и мереологических концептуальных отношений представляет сложную научную задачу. Ниже рассмотрено одну из ветвей иерархии концептов, которая следует из категории *Действительность* [26]. Эта ветвь выбрана не произвольно, а по глубокому убеждению, что на её нижних уровнях концентрируется значительная часть понятий реального мира.

Фрагмент онтологии категории *Действительность* (а точнее для её “продолжительной” составляющей) представлен на рис. 1.8.

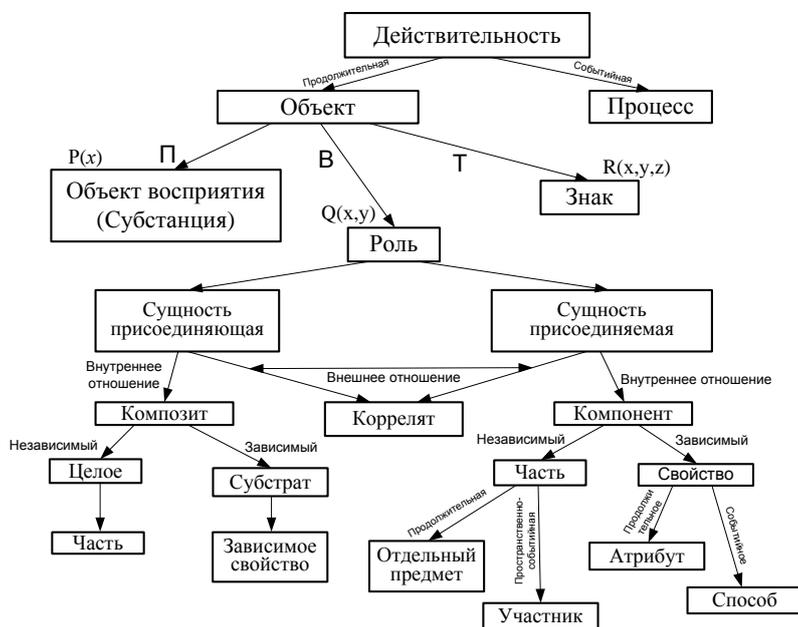


Рис. 1.8. Онтологическое представление категории “Действительность” в ОВУ с использованием классификации Уайтхеда

Категория *Объект восприятия* включает все осязаемые сущности и близка к категории *Субстанция* Аристотеля. С целью разработки метаонтологии для этой ветви был проведен анализ соответствующих фрагментов как идеографических словарей Роже, Кесареса, так и известных современных онтологий (СУС, WordNet, смыслового каркаса Ж.П. Соколовской, естественной классификации Соловьевой Е.А. и др.), что дало возможность сделать некоторые выводы, а именно:

- фрагмент составляет всего несколько метаонтологических уровней, ниже которых лежат дисциплинарные и междисциплинарные онтологические уровни (например, вариант онтологии прикладной области “Вычислительная техника”, разработано в [3]);
- фрагмент является простым, так как строится (как правило) на основе одного таксономического отношения (род-вид);
- “смысловой каркас” фрагмента в современной интерпретации близок к известному “Дереву Порфирия”; ниже приведен именно такой фрагмент метаонтологии для категории *Объект восприятия* (рис. 1.9).

Классификация категории *Знак* отвечает классификации, разработанной в семиотике Пирса [26]. Она приведена отдельно на рис. 1.10. Пирс разделил семиотику на три ветви соответственно со своим принципом трихотомии: Первичность (П) – синтаксис, Вторичность (В) – семантика и Третичность (Т) – прагматика. Знаки в синтаксисе, или в *Материальном*, имеют значение, исходя из самого знака непосредственно. Знаки в семантике, или *Связном*, имеют значение в зависимости от того, как знак связан с его объектом. Знаки в прагматике, или в *Формальном*, определяются некоторым формальным правилом, которое ассоциируется со знаком и объектом.

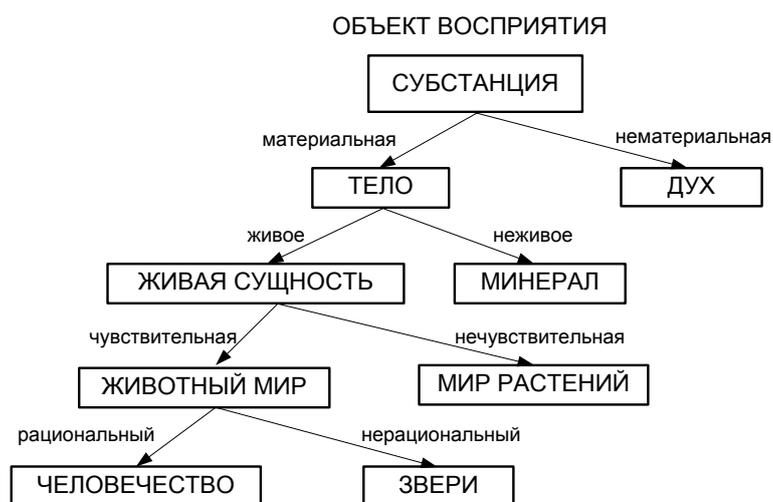


Рис. 1.9. Дерево Порфирия классификации категорий Аристотеля

Категория *Роль* далее подразделяется согласно различению взаимоотношений между присоединяемыми и присоединяющими сущностями [26]. Различение между внутренними и внешними взаимоотношениями определяют виды присоединений. Если сущность, которая поглощается в присоединении, не нарушает форму или существование другой сущности, отношение между ними является внешним. Если поглощение одной сущности в присоединении изменяет вид или делает не-

возможным существование другой, отношение между ними является внутренним. Такое различие генерирует три дальнейших категории.

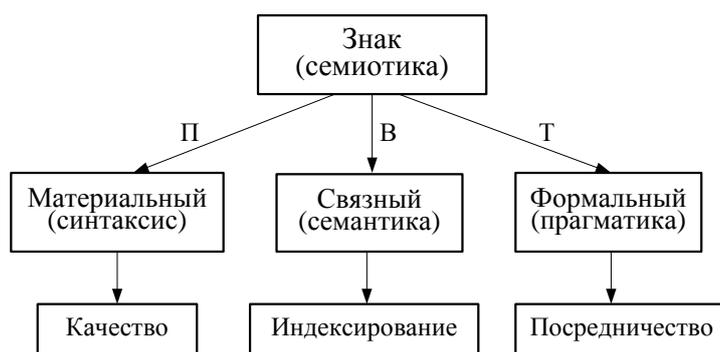


Рис. 1.10. Онтологическое представление категории *Знак* согласно семиотики Пирса

- *Композит*. Внутренняя присоединяющая сущность, названная композитом, опирается на взаимоотношения с каждым компонентом внутри самой себя. Её подтипы различаются видом присоединения: целое, составленное из своих частей, и субстрат – материал, который лежит в основе и поддерживает зависимые свойства (например, размер, вес, форма или цвет).
- *Коррелят*. Внешняя присоединяющая или присоединяемая сущность, названная коррелятом, опирается на взаимоотношения с чем-нибудь вне себя. Примерами могут служить пары сущностей: мать и ребёнок, предприниматель и служащий и др. Коррелят можно считать присоединяемой сущностью в одном присоединении или присоединяющей сущностью обратного присоединения. Отсюда вытекает причина невозможности использования “has”-теста.
- *Компонент*. Внутренняя присоединяемая сущность, названная компонентом, опирается на взаимоотношения к композиту, в который она входит. Его подтипы включают части, существование которых является независимым от целого и свойства, которые не могут существовать без некоторого субстрата.

Две категории *Композит* и *Компонент* на рис. 1.8 далее подразделяются согласно зависимым или независимым различиям. При этом *Композит* подразделяется на *Целое* с независимыми частями, а *Субстрат* – с зависимыми свойствами. Две категории ниже *Компонента* далее классифицированы различиями *Продолжительность* и *Событийность*. С описанными различиями категории, лежащие ниже *Компонента*, подразделены на четыре подтипа.

- *Отдельный предмет*. Части продолжительного называют отдельными предметами (двери и стены дома, области и районы страны и др.).
- *Участник*. Части пространственно-событийного называют участниками (агент, пациент или реципиент действия и др.).
- *Атрибут*. Свойства продолжительного, которые обычно описываются прилагательными, называют атрибутами. Они включают различие сущностей (цвет, вес, форма, размер и др.).
- *Способ*. Свойства событийного, которые обычно описываются наречиями, называют способами (например, такие сущности, как скорость, стиль, интенсивность и др.).

Категории, которые определены простыми семантическими отличиями, имеют сильную корреляцию с синтаксическими категориями естественного языка.

Категории *продолжительного* обычно выражены существительными, а *событийного* – глаголами. *Атрибуты* выражены прилагательными, а *способы* – наречиями. *Участники* выражены обстоятельственными отношениями или *тематическими ролями*, ассоциированными с глаголами.

1.1.5. Источники знаний и проблема формирования новых знаний

Выделяют [3–5, 11, 16, 22, 27–30, 37–42] два основных *источника знаний* – это эксперты, специалисты в своей предметной области и лингвистический корпус текстов (или множество ЕЯО). Если для первого источника методы приобретения знаний достаточно хорошо изучены и проработаны, а также известны соответствующие промышленные экспертные системы, то для второго – разработаны только отдельные методы, не связанные в единую интегрированную технологию, а соответствующие информационные системы носят экспериментальный характер и не совершенны. Актуально усовершенствование известных и разработка новых, более эффективных формальных подходов и моделей обработки, начиная от поиска релевантной информации, её анализа, устранения разного рода неоднозначностей, формально-логического, онтолого-семантического, информационно-кодowego представлений, перехода к формализованному описанию знаний, разработке процедур работы со знаниями и соответствующих им алгоритмов заканчивая получением конкретных результатов пользователя. В более общем виде приведенная выше последовательность этапов обработки знаний, содержащихся в естественно-языковых объектах³, представляется цепочкой технологий Natural Language Processing (NLP) → Knowledge Representation (KR) → Knowledge Processing (KP).

Общая схема компьютерной обработки знаний, содержащихся в ЕЯО, представлена на рис. 1.11 [43]. Здесь в общем виде отражена информационная модель обработки знаний, начиная с поиска на всём информационном пространстве подходящих текстовых данных, соответствующих запросу пользователя (возможно на ЕЯ) и последующего их преобразования (в общем случае) в простые, ситуационные и новые (формальные) знания.

Под *простыми знаниями* понимается поступившая информация о некоторой сущности (сущностях) реального мира, которой соответствует реакция человека – “*принять к сведению*”.

Под *ситуационными знаниями* понимается описание некоторой ситуации, которую необходимо распознать и соответствующим образом “отреагировать”.

Под *новыми знаниями* понимается распознавание приращения ΔS (см. 1.1.1) и пополнение ими БЗ заданной ПДО в некотором формализованном виде.

По сути, схема на рис. 1.11 представляет собой концептуальную модель средств, методов и технологий обработки знаний, содержащихся в ЕЯО.

Извлечение и обработка знаний из ЕЯО является одним из разделов Data mining (Knowledge acquisition, Knowledge extraction, Knowledge processing) и признано перспективным междисциплинарным направлением исследований.

³ Мы не рассматриваем гипертекстовую информационную технологию. Ей посвящена многочисленная литература, например [4, 5 и др.]

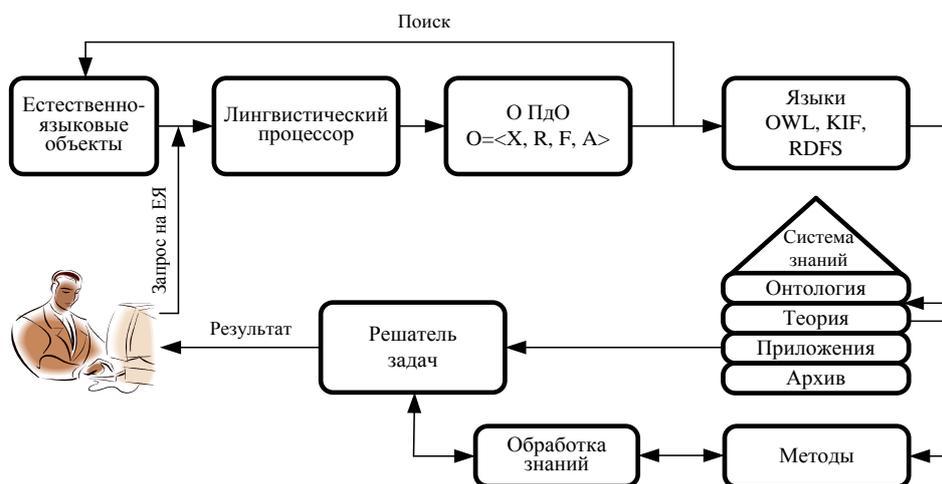


Рис. 1.11 Компьютерная обработка знаний, содержащихся в ЕЯО

Аксиоматизация научных знаний определяется следующими структурными элементами: аксиомы или схемы аксиом (как не доказываемые в данной системе положения); исходные (первоначальные) понятия (явно не определяемые в системе); правила вывода и построения системы знаний; определения как правила введения в теорию новых терминов, абстрактных объектов; теоремы как выводимые высказывания; задачи, следствия, положения, выводимые из теории, леммы, логико-методологические принципы аксиоматизирующей теории. Все вышеназванные аспекты составляют базовую основу формального научного исследования. Наряду с ними научный подход необходимо предполагает ещё одну важную компоненту – интерпретацию. Понятно, что научный смысл имеют лишь те абстракции, которые заведомо могут быть приложены к чему-либо и которые можно сопоставить истине, особенно в тех случаях, когда это практически важно и возможно. Механизмом такого сопоставления истине, поиска логических возможностей её установления и выступает *интерпретация* как логический приём по установлению значения терминологических выражений теорий, знаковой (формальной) системы [41].

Каждому предложению, логически правильно построенному, в интерпретации придаётся некоторое значение, например предложение может быть истинным или ложным. Вопрос же о том, какая из этих возможностей реализована, не решается интерпретацией. Поэтому одна и та же формальная система предполагает множество интерпретаций и только одну истину. Интерпретация гарантирует непротиворечивость теории, но не обеспечивает содержательной истинности данной теории.

Интерпретация имеет свои гносеологические предпосылки. Суть их заключается в том, что в мысленном процессе объективная действительность не просто пассивно отражается, но определённым образом реконструируется, перестраивается. В познании происходит непрерывный процесс образования абстракций и их исключение, перевод одного уровня абстракций на другой. Интерпретация как раз и выступает *логическим средством перехода* от одного уровня знаний к другому, средством установления логических связей между этими уровнями [41, 44].

Известно достаточно много *моделей представления знаний* и соответствующих им методов обработки в КС [3, 5, 17, 25, 42, 45, 46 и др.]. Их выбор для конкретного приложения существенно зависит от характера знаний ПдО, наличия

средств автоматизации построения БЗ, объёма последней, а главное – набора реализуемых соответствующей системой функций.

Следует отметить, что существующие механизмы извлечения новых знаний оперируют только знаниями силлогистического типа. Переход к формированию новых знаний более высокого уровня абстракции методами ИИ, если принять изложенную в [23] вероятностную модель человеческого мышления, соответствует уровню предсознания *Наблюдателя*. На этом уровне знания организованы в сложную *онтологическую структуру* с полным набором связей между концептами. Из сказанного следует, что в настоящее время автоматическое построение БЗ для широких предметных областей представляет сложную *научную проблему*. При этом извне (по отношению к КС) фиксируются уровни расположения базовых понятий ПдО и базовые отношения между ними. Такую структуру, созданную вручную инженером по знаниям на основе тезауруса ПдО, можно представить как *начальную онтологию ПдО*.

Формальная постановка задачи извлечения знаний из ЕЯТ

Прежде чем перейти к рассмотрению системы обработки и извлечения знаний, содержащихся в естественно-языковых текстах, определим понятие “извлечение знаний из ЕЯТ”. С этой целью воспользуемся понятиями, которые используются в constraint programming [47] как одной из форм декларативного программирования.

Пусть дано множество D , на котором определена некоторая конечная совокупность R n -арных отношений на D , то есть $R_i \subseteq D^n$, где $R_i \in R$, $i=1, \dots, k$. Языком ограничений L на D называется некоторое непустое множество $L \subseteq R$. Проблема выполнимости ограничений формулируется таким образом.

Для произвольного множества D и произвольного языка ограничений L на D проблемой выполнимости ограничений $CSP(L)$ есть решение такой комбинаторной проблемы.

Дано: тройка $P = (V, D, C)$, где

– V – конечное множество переменных;

– C – некоторое множество ограничений $\{C_1, \dots, C_q\}$;

– каждое ограничение $C_i \in C$ – это пара (s, R_i) , где s – n -ка элементов из V , которая называется областью ограничения;

– $R_i \in L$ – n_i -арное отношение на D , которое называется отношением ограничения.

Найти: функцию $\varphi: V \rightarrow D$ такую, что $\forall (s, R_i) \in C$, где $s = (v_1, v_2, \dots, v_n)$, n -ка $(\varphi(v_1), \varphi(v_2), \dots, \varphi(v_n)) \in R_i$ или убедиться в том, что такой функции не существует.

Множество D в этом случае называется областью проблемы, а множество всех решений CSP вида $P=(V,D,C)$ обозначается $Sol(P)$.

В случае анализа ЕЯТ с целью извлечения знаний множество D , как область проблемы, интерпретируется как множество объектов, извлечённых из входного текста T , которое факторизовано по некоторым отношениям эквивалентности R (это отношение названо отношением синонимии) и в котором «закодированы» отношения $R_i, i=1, 2, \dots, k$. Переменные из множества переменных $V = \{v_1, v_2, \dots, v_m\}$ принимают свои значения в этом факторизованном множестве

объектов, которые фигурируют в тексте T (это могут быть лексико-грамматические разряды, конкретные объекты (люди, даты, предметы и т. д.)).

Проблемой извлечения знаний из ЕЯТ называется проблема поиска интерпретации $\varphi: V \rightarrow D$ с явным построением отношений R_i из множества $L \subseteq R$. При этом отношения $R_i \in L, i = 1, 2, \dots, k$, извлечённые из текста T , будем называть **элементарными знаниями**.

При анализе ЕЯТ первоочередной задачей есть построение двух фундаментальных отношений, присутствующих практически в каждом ЕЯТ. Это отношение эквивалентности и частичного порядка, которые названы **общезначимыми**. Первое отношение определяет классы синонимичных объектов, а второе отношение – иерархию подчинённости классов эквивалентности. Оба эти отношения составляют основу построения онтологий. Знания, полученные на этом этапе, будем называть первичными. В отношении частичного порядка может вкладываться разный семантический смысл: это может быть отношение таксономии («принадлежать» множеству, классу, группе и т. д.), отношение партономии («состоять из»), отношение генеалогии («отец-сын»), причинно-следственные отношения («если-то»), атрибутивное отношение и т. д.

Как отмечено в [3], наиболее перспективными представляются ЗОИС, функционирование которых опирается на автоматизацию процесса извлечения и формализации содержания естественно-языковых текстов с последующей обработкой формализованного представления этого содержания логико-семантическими методами с ориентацией на конкретную ПДО.

1.2. Область исследования предметно-ориентированных знаний

Анализ проблемы

Теория и практика создания и использования систем, основанных на знаниях (или знание-ориентированных информационных систем), – наиболее актуальное и интенсивно развивающееся направление Computer Science, использование результатов которого повысит эффективность создания инструментальных средств, прикладных систем и применение компьютеров. К сожалению, в настоящее время даже на мировом уровне не получено ощутимых и общепризнанных результатов в широких прикладных областях знаний, хотя можно отметить имеющиеся удачные решения в этом направлении для узкоспециализированных приложений. Сложность указанной проблемы определяется, в частности, сложностью построения, организации и использования больших баз формализованных знаний, а также привлечения целого ряда научных теорий (логики, компьютерной и психологической лингвистики, нейрокибернетики, теории семантических сетей и др.), которые, вполне очевидно, должны способствовать решению проблемы извлечения, формального представления, обработки и системной интеграции знаний и составить концептуально-методологическую основу *теории междисциплинарных научных исследований*.

Как следует из сказанного, исходной информацией для большинства научных исследований являются знаковые системы, представленные в виде ЕЯО. В общем случае такими ЕЯО могут быть большие базы неструктурированных данных, хранящихся в корпоративной памяти, разного рода электронные библиотеки, коллекции документов, текстовая составляющая простого и семантического Web, монографии, научные статьи, научно-технические и деловые документы и пр. Та-

кое уточнение сужает круг научных дисциплин и подходов, методы которых привлекаются к анализу и пониманию ЕЯО, извлечению и формализации из последних знаний, их структурированию и обработке. Среди прочих можно выделить *лингвистику*, которая исследует, в том числе лексический и структурно-грамматический аспекты ЕЯ, и *логику*, которая рассматривает язык в одном ограниченном аспекте, лишь в той мере, в какой он является средством *фиксации, переработки и передачи знаний*. Рассматривая языковые описания, логика в первую очередь интересуется отношением его элементов к обозначаемым объектам и тем, как при помощи определённых связей из этих элементов образуются сложные знаковые системы, выражающие истинные знания об объективном мире. Отправляясь от анализа ЕЯ, логика рассматривает в качестве приоритетных особые языки – искусственные языки науки, возникающие на базе естественных, но отличающиеся рядом важных черт. Для исследования языков (естественных или искусственных) логика использует инструментарий, реализуемый в виде формальных знаковых систем, позволяющий обнаружить законы построения и функционирования, т. е. образования и преобразования систем знаний. Вопрос о выделении объективных средств фиксации знаний и выведении одних “единиц” знаний из других по определённым, объективно значимым правилам, является одним из приоритетных в современных исследованиях по логике [10].

Другим важным направлением в рамках выбранной области исследования является создание ЗОИС, функционирование которых основано на автоматизации процесса формализации содержания ЕЯТ с последующей обработкой формализованного представления этого содержания логико-семантическими методами в интересах решения конкретных задач информационной или информационно-аналитической деятельности специалистов некоторой предметной области. Основной компонентой таких ИС является *подсистема лингвистической обработки ЕЯТ*, задачей которой является распознавание и извлечение знаний, содержащихся в ЕЯТ, и их формализация – с одной стороны, и синтез по формализованному представлению этих знаний их описания на естественном языке – с другой [47].

Таким образом, особенностью лингвистической обработки в таких системах является подчинённость всех её этапов формированию элементов формализованного представления знаний. Однако её реализация невозможна без моделирования в полном объёме *процесса понимания текстовой информации*. Сложность создания таких систем обусловлена в значительной степени отсутствием единой теории понимания ЕЯТ человеком и, как следствие, невозможностью формализации этого процесса применительно к компьютерным системам. Поэтому при разработке ЕЯ-систем часто налагаются различные “спонтанные” ограничения, последствия которых не до конца осознаются разработчиками. Эти ограничения приводят к тому, что многие человеко-машинные системы, на разработку которых тратятся огромные средства, при промышленном использовании не удовлетворяют требованиям решения конечных прикладных задач, связанных с анализом ЕЯ-информации [16].

Одной из важных ветвей современного развития ИИС являются онтолого-управляемые информационные системы. Построение последних тесно связано с разработкой теоретических основ и методологии проектирования, включающих формальный подход, фундаментальные принципы и механизмы, обобщённую архитектуру и структуру системы, формальную модель и методологию проектирования онтологии предметной области, формальную модель представления знаний, обобщённые алгоритмы процедур обработки знаний и др. В свою очередь, каждая из перечисленных составляющих общей методологии проектирования представляет собой сложную информационно-алгоритмическую структуру. Например, разра-

ботка онтологии ПдО тесно связана с концептуализацией онтологических категорий, разработкой и усовершенствованием иерархических структур сущностей на всех уровнях, построением формальной системы аксиом и ограничений. Комплексное решение указанных задач проектирования должно повысить *роль онтологических (концептуальных) знаний* при решении конкретных задач в прикладных областях.

При проектировании знание-ориентированных информационных систем (и в частности ОУИС) существенным является выбор как формально-логического представления знаний, так и источников приобретения и пополнения знаний. В настоящее время признанным “де-факто”, наиболее обширным и общедоступным источником знаний является Интернет-пространство, с его естественным способом представления информации. Указанный способ предопределил появление многочисленных информационных технологий обработки знаний, содержащихся в ЕЯО, в том числе и NLP-технологии.

При создании ОУИС следует выделить три аспекта исследований – онтологический, логический и методологический. Эти аспекты имеют свои, в общем случае фиксированные объекты исследований, соответствующие процессу познания или разработке некоторой ИИС. В связи с ними все объекты могут быть поделены на *три группы*: система сущностей (или объектов реального мира), система знаний и система обработки сущностей в соответствии с данной системой знаний. Первая группа является предметом онтологического исследования, вторая группа – логического исследования и третья – методологического. При этом под *методологией* будем понимать совокупность приёмов, методов и механизмов их взаимодействия, применяемых в процессе исследований.

Большая часть исследований относится к обработке ЕЯО или лингвистического корпуса текстов (ЛКТ) из заданной предметной области. Общепринятого определения для понятия “ЕЯО” неизвестно, а для понятия “Лингвистический корпус текстов” содержательный обзор общепринятых определений выполнен в [48]. Лингвистическим корпусом текстов является любая коллекция текстов по некоторой тематике (из некоторой предметно-проблемной области), которые являются доступными в электронной форме, а также результаты их совместного анализа. Там же приведена методика формирования произвольного ЛКТ.

Следует уточнить смыслы употребляемых в монографии некоторых терминов, не всегда совпадающих с их употреблением в других публикациях. Эти термины относятся к входной информации для ЗОИС и сгруппированы в категорию “Естественно-языковая знаковая система”. Ниже её в онтологической иерархии находятся два термина-синонима – множество естественно-языковых объектов и лингвистический корпус текстов. Ниже этих терминов также находятся два термина-синонима – естественно-языковой текст и естественно-языковая информация. И на самом нижнем уровне находятся два термина-синонима – естественно-языковое высказывание и естественно-языковое предложение. Последние термины являются синонимами, так как в монографии рассматриваются только стили текстов “Научно-технический” и “Деловая проза”. Из этого следует, что все предложения таких текстов являются повествовательными и могут рассматриваться как высказывания.

1.3. Общая постановка проблемы

Актуальность исследования

Рассмотренный выше анализ состояния и тенденций развития знание-ориентированных информационных систем и связанных с ними методов компьютерной обработки естественно-языковой информации и предметных знаний свидетельствует о существовании важной *научно-технической проблемы*, заключающейся в разрыве между хорошо проработанными методами и средствами для отдельных этапов обработки ЕЯ-информации для решения прикладных задач в узкоспециализированных предметных областях, с одной стороны, и недостаточностью таковых для решения комплексных задач, связанных с анализом и пониманием ЕЯ-информации, её формально-логическом представлении, извлечением предметных знаний с их последующим использованием в различных предметных областях. Особый интерес представляет разработка методов и средств знание-ориентированной обработки ЕЯ-информации для украинского сегмента.

Сущность сформулированной научно-технической проблемы, решаемой в рамках настоящего исследования, заключается в следующем.

1. Большой объём накопленной ЕЯ-информации и её ежегодный стремительный прирост предъявляют всё более высокие требования к эффективности её переработки.

2. Существующие методы и средства автоматизированной обработки ЕЯ-информации ориентированы на поверхностно-семантический анализ текстовых документов (для узкоспециализированных предметных областей в лучшем случае), с применением простых онтологий, разработанных вручную. Такие методы и средства непригодны для автоматизированного построения представительных баз знаний (с количеством концептов в несколько тысяч и более) в широких предметных областях, и, как следствие, не могут обеспечить удовлетворительных для пользователей решений, получаемых на основе работы с базами знаний предметных областей.

3. Указанные факты свидетельствуют о наличии противоречия между существующими методами и средствами обработки ЕЯ-информации и постоянно повышающимися требованиями к эффективности обработки.

4. Указанное противоречие разрешается путём выработки новых научных знаний, включающих методы и подходы к автоматическому анализу и глубинно-семантическому пониманию ЕЯ-информации, её формализованному представлению в рамках подходящей теории, которая обеспечивает автоматическое обнаружение и извлечение новых знаний, технологию и соответствующие инструментальные средства автоматизированного построения баз знаний предметных областей.

Целью исследований является разработка общетеоретических и методологических основ, а также формализованных методик и алгоритмов построения интегрированной технологии “*Обработка ЕЯО → Представление знаний → Обработка предметных знаний*”, программно-аппаратных средств и инструментария её реализации.

В соответствии с поставленной целью представлены:

- онто-логический подход к проектированию знание-ориентированных информационных систем с обработкой ЕЯО, который является композицией методов и механизмов, присущих онтологическому и логико-информационному подходам,

а также критерии эффективности информационных систем с онтолого-управляемой архитектурой;

- формальная модель анализа и понимания естественно-языковых текстов;
- методика формализованного представления естественно-языковых высказываний с использованием концептуальных графов;
- архитектурно-структурная организация, информационная и программная модели языково-онтологической информационной системы, реализующей технологию обработки ЕЯО;
- информационная модель и архитектурно-структурная организация онтолого-управляемой информационной системы обработки знаний предметной области;
- методика формализованного проектирования онтологии предметной области, опирающаяся на категориальный уровень знаний. В качестве источника знаний используется множество естественно-языковых объектов, представляющих заданную предметную область;
- информационная модель, архитектурно-структурная организация и принципы функционирования инструментального комплекса для автоматизированного построения онтологий предметных областей и системной интеграции междисциплинарных научных знаний.

1.4. Классификация методов и средств обработки ЕЯО

На рис. 1.12 представлена онтолого-классификационная схема концептуальных понятий, которая отображает проблематику (структуру) многочисленных научных исследований в области ИИ [4, 22 и др.]. Её нижние уровни представляют объект и предмет исследований при решении научно-технической проблемы построения ОУИС [4, 28–30, 49 и др.].

На рис. 1.12 приняты следующие обозначения: Мд – материальная действительность, Ад – абстрактная действительность, Т – технология, Ис – искусственный, С – система, Ин – интеллект, И – информация, Зн – знания, ИТ – информационная технология, ИИ – искусственный интеллект, ИС – информационная система, ПдО – предметная область, ИИС – интеллектуальная информационная система, КЛ – компьютерная лингвистика, ЭБ – электронная библиотека, КП – корпоративная память, БЗ ПдО – база знаний предметной области, ОУИС ПдО – онтолого-управляемая информационная система предметной области, ЯОИС – языково-онтологическая информационная система, ЕЯ – естественный язык, ПП – прикладной процессинг, СП – семантический процессор, ЛП – лингвистический процессор, СИИ – семантико-информационный интерпретатор, ЯОКМ – языково-онтологическая картина мира, О ПдО – онтология предметной области, БД ФПрВ – база данных фактов и правил вывода; типы семантических отношений: 1 – род-вид, 2 – целое-часть, 3 – по преобразованию.

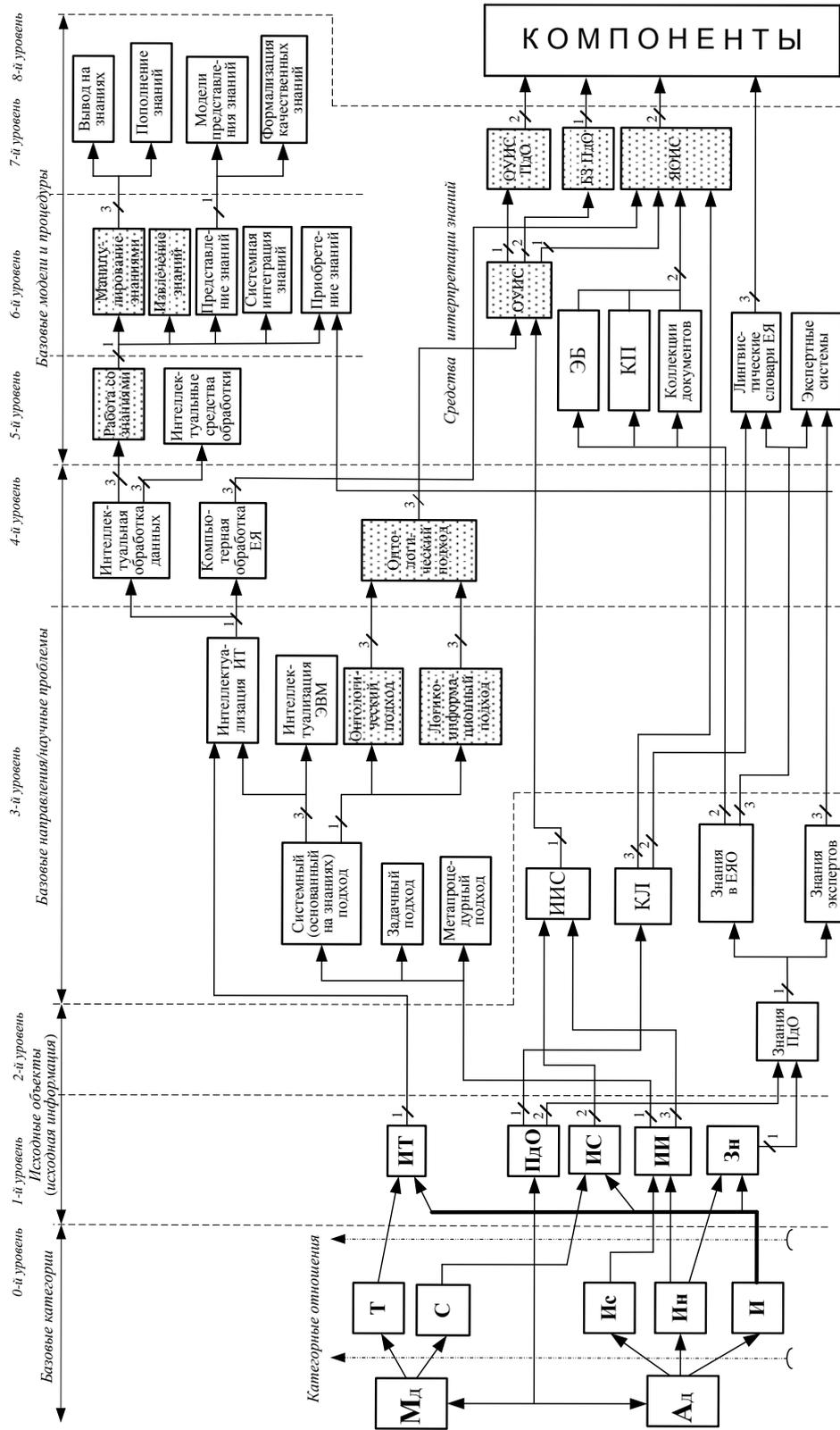


Рис.1.12. Онтолого-классификационная схема методов и средств работы со знаниями

По сути, онтолого-классификационная схема представляет собой иерархию онтологий:

- нулевой уровень на схеме – *уровень категорий*;
- первый и второй уровни – *исходные объекты (исходная информация)* в домене “Информатика”;
- третий и четвёртый уровни – *представляют базовые научные направления, проблемы и подходы* ПдО “Искусственный интеллект”;
- пятый, шестой и седьмой уровни – *базовые методы, модели и процедуры* – представляют верхний и средний уровни онтологии ПдО “Знание-ориентированная обработка предметных знаний”). На этих уровнях сосредоточены основные подсистемы ЗОИС с обработкой ЕЯО;
- восьмой уровень – *модели и средства интерпретации языковых и предметных знаний* – представляет обобщённые понятия нижнего уровня указанной выше онтологии ПдО.
- девятый уровень – *уровень компонент ЗОИС с обработкой ЕЯО*.

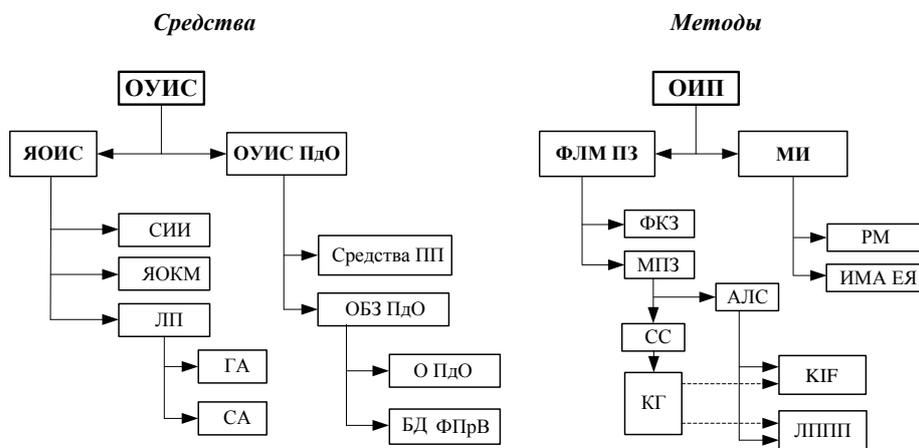


Рис. 1.13. Средства и методы обработки ЕЯ-объектов

На рис. 1.13 представлен детализированный фрагмент онтолого-классификационной схемы (рис. 1.12) (или нижний уровень онтологии предметной области) с признаками деления понятий (и соответствующей группировкой) по объекту и предмету исследований, в котором отражена совокупность научно-технических задач, входящих в круг решения научно-технической проблемы (раздел 1.3). На рисунке приняты следующие обозначения: ГА – грамматический анализатор, СА – семантический анализатор, О БЗ ПдО – онтологическая БЗ ПдО, ФЛМ ПЗ – формально-логические методы представления знаний, ФКЗ – формализация качественных знаний, МПЗ – модели представления знаний, АЛС – алгебро-логические системы, СС – семантические сети, КГ – концептуальные графы, KIF – Knowledge Interchange Format, ЛППП – логика предикатов первого порядка, МИ – методы интерпретации, РМ – резолюционный метод, ИМА ЕЯ – интерпретационный метод анализа естественного языка.

На рис. 1.13 показаны только концепты и связи между ними, существенные с точки зрения разработки методологии проектирования ОУИС.

1.5. Онтологический инжиниринг – перспективное направление предметных и междисциплинарных научных исследований

Онтологический инжиниринг обычно рассматривают как новое направление в методологии разработки интеллектуальных систем. При этом ведущей парадигмой структурирования информационного контента остаются онтологии или иерархические концептуальные структуры, которые формируются аналитиком на основе изучения и структурирования протоколов извлечённых знаний и документации. С методической точки зрения это один из “систематических” и наглядных способов. Онтологический инжиниринг развивает основные положения инженерии знаний – науки о моделях и методах извлечения, структурирования и формализации знаний. Инженерия знаний – это ветвь искусственного интеллекта, в то время как онтологический инжиниринг охватывает более широкий круг приложений от систем манипулирования знаниями, создания электронных курсов и дистанционного обучения до корпоративных систем управления знаниями и принятия решений, интегрированных сетей знаний [50]. Сегодня сфера онтологического инжиниринга уточняется и расширяется в связи с актуализацией *парадигмы компьютерных онтологий*, стремлением максимизировать степень автоматизации построения баз знаний для достаточно широких предметных областей. С этой точки зрения суть онтологического инжиниринга раскрывается при описании *онтологического подхода* к проектированию онтологических баз знаний ПдО и средств их построения, обработки и применения.

Предлагаемые расширения структуры онтологического инжиниринга представлены на рис. 1.14 (новые блоки указанной структуры имеют затемнённую заливку).

Развитие онтолого-управляемых информационных систем тесно связано с концептуализацией онтологических категорий, совершенствованием иерархических структур сущностей на всех уровнях, построением формальной системы аксиом и ограничений и, наконец, решением конкретных задач в прикладных областях. При этом онтологический инструментарий выступает в роли объединительного механизма для разного рода научных знаний, ориентированных на решение одной из главных проблем искусственного интеллекта – компьютерного анализа, синтеза и понимания ЕЯ. Диапазон применений и практическая значимость ОУИС, назначением которых является решение указанной проблемы, постоянно расширяется.

Актуальной для направления онтологического инжиниринга является проблема разработки методологии и инструментальных средств автоматизированного проектирования *формальной (компьютерной) онтологии ПдО*. До сих пор не существует общепринятой методологии. Остаётся открытым вопрос появления конструктивной теории разработки формальных онтологий. *Инструментальная среда* автоматизированного проектирования формальной онтологии ПдО представляет собой сложный программно-аппаратный комплекс, состоящий из оригинальных и известных инструментальных средств, объединённых между собой управляющей программной оболочкой, которая обеспечивает разработчику онтологии автоматизированный и интерактивный процесс проектирования. Она обеспечивает выполнение условий целостности, непротиворечивости, интегрированности, операбельности и др. при создании и применении онтологий.

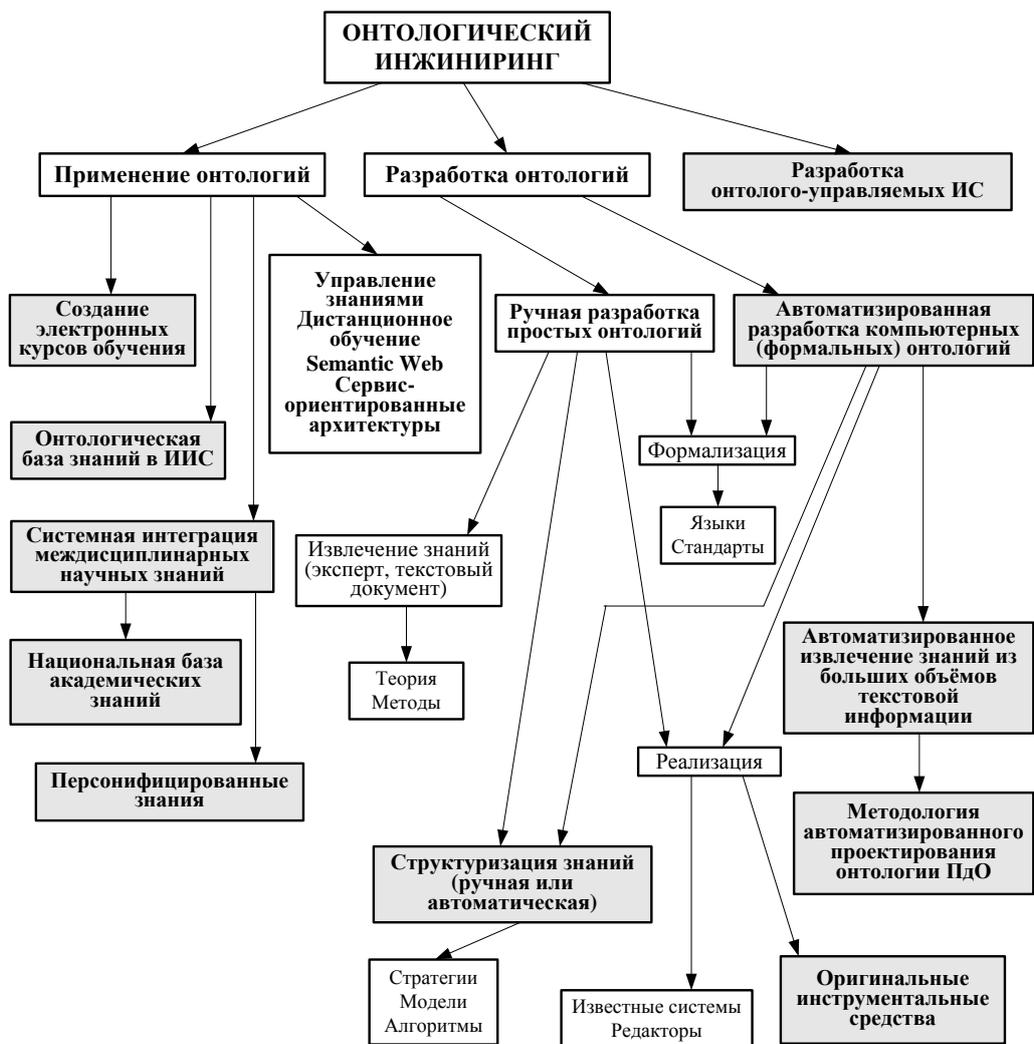


Рис. 1.14. Структура онтологического инжиниринга

2. ЛОГИКО-МАТЕМАТИЧЕСКИЕ ОСНОВЫ ОНТОЛОГО-УПРАВЛЯЕМЫХ СИСТЕМ

2.1. Множества и отношения

2.1.1. Интуитивное понятие множества

Понятие множества относится к аксиоматическим понятиям математики, и точное его определение дать невозможно. Часто принимается формулировка интуитивного понятия множества Г. Кантора, основоположника этой теории:

“Произвольная совокупность определённых предметов нашей интуиции или интеллекта, которые можно отличить один от другого и которые представляются как единое целое, называется *множеством*. Предметы, которые входят в состав множества, называются его *элементами*”.

Существенным пунктом канторовского понятия множества является то, что совокупность предметов рассматривается как один предмет (“представляется как единое целое”). Основное внимание тут переносится с отдельных предметов на совокупности предметов, которые, в свою очередь, можно рассматривать как предметы.

Что касается “предметов нашей интуиции или интеллекта”, то эта формулировка даёт значительную свободу, прежде всего тем, что никак не ограничивает природу предметов, составляющих множество. Множество может состоять, например, из людей, точек плоскости, простых чисел, планет Вселенной. Заметим также, что канторовская формулировка множества даёт возможность рассматривать множества, элементы которых по определенной причине точно задать невозможно. В связи с этим вспомним, что элементы любого бесконечного множества, даже теоретически, нельзя собрать в законченную совокупность (с этой проблемой связана философская абстракция так называемой актуальной бесконечности). Известны также и конечные множества, которые имеют такую же меру неопределенности, как и любое бесконечное множество.

Выясним, наконец, смысл выражений: “которые можно отличить один от другого” и “определённые предметы”. В первом случае для любых двух предметов, которые рассматриваются как элементы данного множества, должна существовать возможность выяснить, различные это предметы, или одинаковые. В другом случае, если задано некоторое множество и какой-нибудь предмет, то можно определить, является ли этот предмет элементом данного множества. Отсюда следует, что всякое множество полностью определяется своими элементами. Это канторовское требование формулируется как

Интуитивный принцип объёмности или аксиома экстенциональности.

Два множества равны тогда и только тогда, когда они состоят из одних и тех же элементов. Равенство двух множеств A и B обозначают $A = B$.

Таким образом, два множества равны, если каждый элемент одного из них является элементом другого и наоборот.

Множество называется **конечным**, если оно состоит из конечного числа элементов. Запись $a \in A$ ($a \notin A$) означает, что a является (не является) элементом множества A . Однозначно определенное множество, элементами которого являются a_1, a_2, \dots, a_n , обозначается $\{a_1, a_2, \dots, a_n\}$.

В качестве примеров рассмотрим следующие.

1. Множество $\{a\}$ – так называемое **одноэлементное множество** – это множество, единственным элементом которого является элемент a .

2. Множества $A = \{2, 4, 6\}$, $B = \{2, 6, 4\}$, $C = \{2, 2, 6, 4, 4\}$, $C' = \{2, 2, 4, 4, 6, 6\}$, ... равны, поскольку состоят из одних и тех же элементов. Первое и второе множества отличаются одно от другого порядком своих элементов, а третье, четвертое и т. д. от первых двух тем, что в них элементы 2, 4 и 6 присутствуют в двух экземплярах каждый.

3. Множества $\{\{a, b\}, \{b, c\}\}$ и $\{a, b, c\}$ не равны, поскольку первое состоит из элементов $\{a, b\}$ и $\{b, c\}$, а второе – из элементов a, b, c .

Последний пример показывает, что необходимо различать предмет и множество, единственным элементом которого является этот предмет.

Рассмотрим более детально ситуацию, которая имеет место в примере 2. Из принципа объёмности следует, что все множества типа множеств C, C' и т. д., которые были приведены в этом примере, равны между собой и равны множеству A .

Множество, которое состоит из элементов некоторого множества A так, что эти элементы могут входить в состав этого множества в каком угодно количестве экземпляров, будем называть **мультимножеством** множества A и обозначать его $M(A)$. С точки зрения теории множеств, множество и его мультимножество – это один и тот же объект, и они могут между собой не различаться. Однако часто, особенно когда речь заходит о представлении множества в памяти вычислительной машины, возникает потребность отличать мультимножество от множества.

Задание множества с помощью фигурных скобок с явным перечислением его элементов целесообразно только тогда, когда множество имеет небольшое количество элементов. Если же множество имеет хотя и конечное, однако большое количество элементов, такое задание множества достаточно громоздко, а в случае бесконечного множества его применение вообще невозможно. Возникает вопрос: как задавать множества, которые имеют большое или бесконечное число элементов?

Ответ связан с понятием свойства. Пока что ограничимся интуитивным понятием свойства.

Под **свойством** предмета x будем понимать такое повествовательное предложение, в котором нечто утверждается относительно предмета x и которое можно характеризовать только как истинное или как ложное по отношению к x .

В качестве примеров рассмотрим следующие:

а) свойствами являются такие записи: 1) 3 делит x ; 2) $x < x$; 3) $x^2 = 2$; 4) $x^2 + 1 > 0$;

б) выражения “индукция очень важна” и “существует такой x , что $2x < 0$ ” не являются свойствами, потому что их нельзя характеризовать как истинные или ложные относительно x .

Пусть $P(x)$ означает некоторое свойство, тогда $P(a)$ будет означать то же самое свойство, но с заменой x на a . Задание множества в терминах свойств достигается с помощью такого принципа.

Интуитивный принцип абстракции или аксиома свертки. Всякое свойство $P(x)$ определяет некоторое множество A с помощью условия: элементами множества A являются те и только те предметы a , которые имеют свойство P .

В силу принципа абстракции всякое свойство $P(x)$ определяет единственное множество, которое обозначают $\{a \mid P(a)\}$ и читают так: “множество всех тех предметов a , что $P(a)$ ”. Заметим, что свойство P может представлять собой способ построения элементов множества $\{a \mid P(a)\}$.

Пусть A – некоторое множество, а $P(x)$ имеет вид $x \neq x$, тогда множество $\{a \mid P(a)\} = \{a \mid a \neq a\}$, очевидно, не имеет элементов. Из принципа объёмности следует, что может существовать только одно множество, которое не имеет элементов. Это множество называется **пустым множеством** и обозначается \emptyset .

В качестве примеров рассмотрим следующие.

1. **Множество натуральных чисел** $\{0, 1, 2, \dots, n, \dots\}$, которое обозначают буквой \mathbf{N} . Это множество можно задать с помощью принципа свертки.

Пусть $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ – множество из десяти элементов, которое будем называть **алфавитом**, а его элементы – **символами**. Всякий набор символов из этого алфавита, записанных один за другим, назовем **словом** в алфавите A . Например, $p = 0020034985700$ – слово в алфавите A , а $p = 00a26543c$ не является словом в алфавите A , поскольку в его запись входят символы a и c , не принадлежащие множеству A . То есть p будет словом в алфавите A тогда и только тогда, когда каждый символ этого слова принадлежит алфавиту A .

Тогда множество $\mathbf{N} = \{p = xy\dots z \mid x, y, \dots, z \in A\}$ составляет множество натуральных чисел. Заметим, что слово $p = 00\dots 0$ и $p' = 0$ – одно и то же число 0, так же, как и числа 00057 и 57. Следовательно, запись одного и того же элемента множества \mathbf{N} может быть различной. Для однозначности записи чисел необходимо потребовать, чтобы первый символ в слове p , которое состоит более чем из одного символа, был отличен от нуля, то есть

$\mathbf{N} = \{p = xy\dots z \mid x, y, \dots, z \in A \text{ и } x \neq 0 \text{ или } 0, \text{ если } p = 0\}$.

2. **Множество натуральных положительных чисел** $\mathbf{N}^+ = \{1, 2, \dots, n, \dots\}$. То есть множество натуральных положительных чисел – это такое множество слов в алфавите A , среди которых нет слова 0.

3. **Множество целых чисел** $\mathbf{Z} = \{\dots, -n, \dots, -2, -1, 0, 1, 2, \dots, n, \dots\}$ – это множество слов в алфавите $B = \{-, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ таких, что $\mathbf{Z} = \{p = xy\dots z \mid x, y, \dots, z \in B \text{ и } x \neq 0 \text{ и } y, \dots, z \neq - \text{ или } 0, \text{ если } p = 0\}$. Слово, первым символом которого является символ “-”, называется отрицательным числом, а слово, первым символом которого является символ, отличный от символа “-”, называется положительным числом.

4. **Множество рациональных чисел** \mathbf{RC} . Состоит из множества всех целых чисел и всех несократимых рациональных дробей вида m/n , где $m, n \in \mathbf{Z}$, $n \neq 0$, то есть $\mathbf{RC} = \{m/n \mid m, n \in \mathbf{Z} \text{ и } n \neq 0\}$.

Подводя итоги, прежде всего, отметим, что множества задаются или с помощью явного перечисления своих элементов (случай конечного множества с небольшим количеством элементов), или с помощью свойств, которым удовлетворяют их элементы (свойство также может выражать способ построения элементов множества). Задание множества будем называть **неизбыточным**, если каждый его элемент входит в данное множество в единственном экземпляре, и **избыточным**, если хотя бы один элемент этого множества входит в его состав более чем в одном экземпляре (случай мультимножества).

Более детальное изложение основ теории множеств и её аксиоматику можно найти [53–55].

2.1.2. Операции над множествами и их свойства

Введем символы \Leftrightarrow , $\exists x$, $\forall x$, \Rightarrow , которые далее будут служить для сокращения выражений “тогда и только тогда, когда”, “существует x такой, что”, “для всякого x ” и “следует” или “вытекает” соответственно.

Множество A называется **подмножеством** множества B ($A \subseteq B$), если все его элементы являются также элементами множества B ($(A \subseteq B) \Leftrightarrow (a \in A \Rightarrow a \in B)$). При этом множество B называется **надмножеством** множества A .

Теперь принцип объёмности можно записать так: $A = B \Leftrightarrow (A \subseteq B \text{ и } B \subseteq A)$. $A \subset B$ означает, что $A \subseteq B$ и A не равно B ($A \neq B$). Если $A \subset B$, то множество A называется **собственным подмножеством** множества B , а множество B – собственным надмножеством множества A .

Покажем, что пустое множество является подмножеством любого множества A . Допустим, что утверждение “ $\emptyset \subseteq A$ ” ложно, то есть, существует хотя бы один элемент x , который принадлежит множеству \emptyset , который не является элементом множества A . Но множество \emptyset не имеет элементов. Значит, утверждение “ $\emptyset \subseteq A$ ” истинно.

Пусть U – некоторое множество, тогда $\mathbf{B}(U)$ – множество всех подмножеств множества U . В этом случае множество U называют **универсальным**, а множество $\mathbf{B}(U)$ – **множеством-степенью** или **булеаном** множества U . Например, если $U = \{1, 3, 5\}$, то $\mathbf{B}(U) = \{\emptyset, \{1\}, \{3\}, \{5\}, \{1, 3\}, \{1, 5\}, \{3, 5\}, \{1, 3, 5\}\}$.

Объединением множеств A и B называется множество, в состав которого входят те и только те элементы, которые входят в состав хотя бы одного из этих множеств. Полученное множество обозначается $A \cup B$, то есть $A \cup B = \{a \mid a \in A \text{ или } a \in B\}$.

В качестве примеров рассмотрим следующие.

1. Пусть $A = \{1, 2, 3\}$, $B = \{1, 3, 4, 6\}$, тогда $A \cup B = \{1, 2, 3, 4, 6\}$.
2. Пусть \mathbb{C} – множество всех чётных натуральных чисел, а \mathbb{H} – множество всех нечётных натуральных чисел, тогда $\mathbb{C} \cup \mathbb{H} = \mathbb{N}$, где \mathbb{N} – множество всех натуральных чисел.

Пересечением множеств A и B называется множество, которое состоит из элементов, входящих в состав как множества A , так и множества B . Полученное множество обозначается $A \cap B$, то есть $A \cap B = \{a \mid a \in A \text{ и } a \in B\}$. Если $A \cap B = \emptyset$, то множества A и B называются **непересекающимися**.

В качестве примеров рассмотрим следующие.

1. Пусть $A, B, \mathbb{C}, \mathbb{H}$ означают множества из предыдущего примера, тогда:
а) $A \cap B = \{1, 3\}$; б) $\mathbb{C} \cap \mathbb{H} = \emptyset$; в) $\mathbb{N} \cap \mathbb{H} = \mathbb{H}$; г) $\mathbb{N} \cap \mathbb{C} = \mathbb{C}$.
2. Пусть A – множество прямых, которые проходят через точку a некоторой плоскости, а B – множество прямых, которые проходят через точку c этой же плоскости. Тогда $A \cap B = \{l\}$, где l – прямая, которая проходит через точки a и c .

Если множество A представляет собой объединение подмножеств $A_1, A_2, \dots, A_n, \dots$, то совокупность подмножеств $\{A_1, A_2, \dots, A_n, \dots\}$ называется **покрытием** множества A . Если же совокупность подмножеств покрытия множества A такова,

что $A_i \cap A_j = \emptyset$ при $i \neq j$, то совокупность $\{A_1, \dots, A_n, \dots\}$ называется **разбиением** множества A , а подмножества A_i – **классами** этого разбиения, $i = 1, 2, \dots, n, \dots$.

В качестве примера покрытия и разбиения множества рассмотрим следующие.

1. Пусть A – множество всех студентов некоторого вуза X , которые его закончили, а A_i – подмножество тех студентов вуза X , которые закончили i -й факультет этого вуза.

Поскольку не исключена возможность, что некто из множества студентов A закончил несколько факультетов данного вуза, и такой человек попадает в несколько соответствующих подмножеств совокупности, то ясно, что совокупность подмножеств A_1, A_2, \dots, A_k является покрытием множества A .

Если же взять совокупность всех студентов вуза X , которые учатся в данное время, то совокупность студентов A_1, A_2, \dots, A_k является, очевидно, разбиением множества всех студентов данного вуза, которые учатся в данное время.

Разностью множеств A и B называется множество $B \setminus A = \{a \mid a \in B \text{ и } a \notin A\}$. Очевидно, что $B \setminus A = B \setminus (A \cap B)$. Если $A \subseteq B$, то $B \setminus A$ называется **дополнением** множества A в множестве B и обозначается A'_B или просто A' , когда B можно определить по контексту.

Симметричной разностью множеств A и B называется множество

$$A \div B = (A \setminus B) \cup (B \setminus A).$$

Введенные операции называют **теоретико-множественными операциями**.

Из определения операции $A \div B$ очевидным образом вытекает такое равенство: $A \div B = (A \cup B) \setminus (A \cap B)$.

Введенные операции объединения, пересечения и разности удовлетворяют определенным законам. Имеет место такая теорема.

Теорема 2.1. Для произвольных подмножеств A, B, C некоторого универсального множества U имеют место такие тождества:

$$M1) A \cup B = B \cup A, A \cap B = B \cap A; \quad (\text{коммутативность})$$

$$M2) A \cup (B \cap C) = (A \cup B) \cap C, \\ A \cap (B \cup C) = (A \cap B) \cup C; \quad (\text{ассоциативность})$$

$$M3) A \cup (B \cap C) = (A \cup B) \cap (A \cup C), \\ A \cap (B \cup C) = (A \cap B) \cup (A \cap C); \quad (\text{дистрибутивность})$$

$$M4) A \cup A' = U, A \cap A' = \emptyset;$$

$$M5) A \cup \emptyset = A, A \cap U = A;$$

Доказательство. Покажем справедливость второго соотношения из M2: $A \cap (B \cap C) = (A \cap B) \cap C$.

Пусть $a \in A \cap (B \cap C) \Rightarrow a \in A, a \in B, a \in C \Rightarrow a \in (A \cap B)$ и $a \in C \Rightarrow a \in (A \cap B) \cap C$ и $A \cap (B \cap C) \subseteq (A \cap B) \cap C$.

Получение обратного включения выполняется аналогично.

Покажем справедливость первого соотношения из M3. С одной стороны, поскольку $(B \cap C) \subseteq B$, то $A \cup (B \cap C) \subseteq A \cup B$. Аналогично $(B \cap C) \subseteq C$ и $A \cup (B \cap C) \subseteq A \cup C$. Следовательно, $A \cup (B \cap C) \subseteq (A \cup B) \cap (A \cup C)$.

С другой стороны, если $a \in (A \cup B) \cap (A \cup C)$, то $a \in A \cup B$ и $a \in A \cup C$. Если $a \in A$, то $a \in A \cup (B \cap C)$. А если $a \notin A$, то $a \in B$ и $a \in C$ и тогда $a \in B \cap C$. Следовательно, $(A \cup B) \cap (A \cup C) \subseteq A \cup (B \cap C)$. Вместе с полученным ранее включением имеем необходимое равенство.

Равенства доказаны.

Остальные соотношения доказываются аналогично. Их доказательство предлагается читателю в качестве упражнения.

Используя законы ассоциативности для операций объединения и пересечения множеств, можно ввести такие сокращения.

Выражения

$$A_1 \cup A_2 \cup \dots \cup A_n = \bigcup_{i=1}^n A_i$$

означают объединение множеств A_1, A_2, \dots, A_n , то есть совокупность тех элементов, которые являются элементами хотя бы одного из множеств A_1, A_2, \dots, A_n , а выражения

$$A_1 \cap A_2 \cap \dots \cap A_n = \bigcap_{i=1}^n A_i$$

означают пересечение множеств, то есть совокупность тех предметов, которые являются элементами каждого из множеств A_1, A_2, \dots, A_n .

Кроме тождеств, которые были приведены в теореме 2.1, существуют и другие полезные тождества для операций над множествами.

Теорема 2.2. Для любых подмножеств A и B некоторого универсального множества U имеют место такие тождества:

M6) $A \cup A = A$,	$A \cap A = A$;
M7) $A \cup (A \cap B) = A$,	$A \cap (A \cup B) = A$;
M8) $A \cup U = U$,	$A \cap \emptyset = \emptyset$;
M9) $(\emptyset)' = U, U' = \emptyset$,	$(A')' = A$;
M10) $(A \cup B)' = A' \cap B'$;	$(A \cap B)' = A' \cup B'$.

Доказательство. Покажем, например, справедливость тождеств M6) и M8).

В силу законов M1) – M5) можем записать:

$$A = A \cup \emptyset = A \cup (A \cap A') = (A \cup A) \cap (A \cup A') = (A \cup A) \cap U = A \cup A,$$

$$A = A \cap U = A \cap (A \cup A') = (A \cap A) \cup (A \cap A') = (A \cap A) \cup \emptyset = A \cap A.$$

Далее, пользуясь только что установленными тождествами, имеем:

$$A \cup U = A \cup (A \cup A') = (A \cup A) \cup A' = A \cup A' = U,$$

$$A \cap \emptyset = A \cap (A \cap A') = (A \cap A) \cap A' = A \cap A' = \emptyset.$$

Доказательство остальных тождеств предлагается читателю в качестве упражнения.

Пользуясь тождествами, приведенными в теоремах 2.1 и 2.2, можно редуцировать сложные выражения, содержащие множества, аналогично тому, как проводится редуцирование выражений в элементарной алгебре.

2.1.3. Декартово произведение множеств. Отношения

Пусть A и B – два множества. Рассмотрим множество $C = \{(a, b) \mid a \in A, b \in B\}$. Это множество называется **декартовым произведением** множеств A и B и обозначается $A \times B$. Если множества A и B конечные и состоят соответственно из m и n элементов, то очевидно, что C состоит из mn элементов.

Самостоятельный интерес вызывает случай, когда множества A и B равны: $A = B$. Для его рассмотрения введем понятие упорядоченной пары.

Упорядоченной парой элементов множества A будем называть объект (a, a') , состоящий из двух, не обязательно разных, элементов a и a' множества A , для ко-

торых указано, какой необходимо считать первым, а какой – вторым. Так если $A = \{1, 2, 3, 4, 5\}$, то упорядоченные пары $(3, 4)$ и $(4, 3)$ следует считать разными, поскольку в первой паре первым элементом является 3, вторым – 4, а во второй паре – наоборот. Упорядоченными парами являются также пары $(1, 1), (2, 2), (3, 3), (4, 4), (5, 5)$.

Множество $C = \{(a, a') \mid a, a' \in A\}$ всех упорядоченных пар (a, a') элементов из множества A называется **декартовым квадратом** множества A и обозначается A^2 .

Понятие упорядоченной пары можно расширить на упорядоченные тройки элементов (a_1, a_2, a_3) , упорядоченные четверки (a_1, a_2, a_3, a_4) и т. д. Вообще, упорядоченная n -ка элементов из множества A – это n не обязательно различных элементов из A , заданных в определенной последовательности. Если

$$(a_1, a_2, \dots, a_n) = (a_1', a_2', \dots, a_n'), \text{ то } a_1 = a_1', a_2 = a_2', \dots, a_n = a_n'.$$

Для того, чтобы отличать упорядоченные пары, тройки и т. д. от неупорядоченных, введем такое обозначение: если A – некоторое множество, то $A^{(2)}, A^{(3)}, \dots$ будут соответственно означать множество неупорядоченных пар элементов, троек элементов и т. д. из множества A , то есть $A^{(2)}$ представляет собой множество всех двухэлементных подмножеств множества A , $A^{(3)}$ – множество всех трехэлементных подмножеств множества A и т. д.

Приведенные выше определения декартового произведения двух множеств и декартового квадрата множества можно обычным способом обобщить и на случай произвольной конечной совокупности множеств.

Декартовым произведением $A_1 \times A_2 \times \dots \times A_n$ множеств A_1, A_2, \dots, A_n называется совокупность последовательностей (то есть совокупность упорядоченных n -ок элементов) вида (a_1, a_2, \dots, a_n) , где $a_i \in A_i, 1 \leq i \leq n$.

Элементы декартового произведения называют еще **кортежами**. Произвольное подмножество R множества $A_1 \times A_2 \times \dots \times A_n$ называется **отношением**, заданным или определенным на множествах A_1, A_2, \dots, A_n . Если $A_1 = A_2 = \dots = A_n = A$, то декартово произведение $A_1 \times A_2 \times \dots \times A_n$ называется **декартовым произведением n -й степени множества A (A^n)**, а отношение R , заданное на множествах A_1, A_2, \dots, A_n , – **n -арным отношением на множестве A** .

Когда $(a_1, a_2, \dots, a_n) \in R$, то говорят, что элементы a_i ($i = 1, 2, \dots, n$) находятся между собой в отношении R , или отношение R истинно для a_1, a_2, \dots, a_n . Если $(a_1, a_2, \dots, a_n) \notin R$, то считают, что R ложно для a_1, a_2, \dots, a_n . При $n = 1$ отношение называется **унарным**, при $n = 2$ – **бинарным**, при $n = 3$ – **тернарным** и т. д.

Поскольку отношения, заданные на A_1, A_2, \dots, A_n , – подмножества множества $A_1 \times A_2 \times \dots \times A_n$, то для них определены операции объединения, пересечения, разности и дополнения:

$$(a_1, a_2, \dots, a_n) \in R \cup R_1 \Leftrightarrow (a_1, a_2, \dots, a_n) \in R \text{ или } (a_1, a_2, \dots, a_n) \in R_1;$$

$$(a_1, a_2, \dots, a_n) \in R \cap R_1 \Leftrightarrow (a_1, a_2, \dots, a_n) \in R \text{ и } (a_1, a_2, \dots, a_n) \in R_1;$$

$$(a_1, a_2, \dots, a_n) \in R \setminus R_1 \Leftrightarrow (a_1, a_2, \dots, a_n) \in R \text{ и } (a_1, a_2, \dots, a_n) \notin R_1;$$

$$(a_1, a_2, \dots, a_n) \in R' \text{ в } A_1 \times A_2 \times \dots \times A_n \Leftrightarrow (a_1, a_2, \dots, a_n) \notin R.$$

Часто R' обозначают $\neg R$ и называют **отрицанием** отношения R .

Довольно часто как в теории, так и на практике используются бинарные отношения. Если R – бинарное отношение, то запись aRb означает, что $(a, b) \in R$, то есть что R истинно для (a, b) .

Пусть $R \subseteq A \times B, R_1 \subseteq B \times C$. Отношение R^{-1} , заданное на множестве $B \times A$, называют **обратным** к R , если $R^{-1} = \{(b, a) \mid aRb\}$, а отношение $R * R_1$, заданное на

множестве $A \times C$, – *произведением* или *суперпозицией* отношений R и R_1 , если $R * R_1 = \{(a, c) \mid a \in A, c \in C \text{ и } (\exists b \in B) (aRb \text{ и } bR_1c)\}$.

Следует заметить, что операция умножения отношений может быть и неопределенной, если в множестве B для данных элементов a из A и c из C не существует соответствующего элемента b . Но, если $A = B = C$, то эта операция всегда определена.

Рассмотрим некоторые полезные свойства бинарных отношений.

Теорема 2.3. Если R, R_1, R_2 – бинарные отношения, заданные на множестве A , то:

- 1) $(R_1 \cup R_2) * R = R_1 * R \cup R_2 * R; R_1 \subseteq R_2 \Rightarrow R_1 * R \subseteq R_2 * R;$
- 2) $(R^{-1})^{-1} = R; R \subseteq R_1 \Rightarrow R^{-1} \subseteq R_1^{-1};$
- 3) $(R * R_1)^{-1} = (R_1^{-1}) * (R^{-1});$
- 4) $(R \cap R_1)^{-1} = (R^{-1}) \cap (R_1^{-1});$
- 5) $(R * R_1) * R_2 = R * (R_1 * R_2).$

Доказательство. 1. Если $(a, b) \in (R_1 \cup R_2) * R$, то существует элемент $c \in A$ такой, что $(a, c) \in R_1 \cup R_2$ и $(c, b) \in R$. Следовательно $(a, c) \in R_1$ или $(a, c) \in R_2$ и $(c, b) \in R$. Отсюда имеем, что $(a, b) \in R_1 * R$ или $(a, b) \in R_2 * R$, то есть $(a, b) \in R_1 * R \cup R_2 * R$. Обратное включение доказывается аналогично.

Вторая часть утверждения следует из того, что, если $R_1 \subseteq R_2$, то $R_1 \cup R_2 = R_2$, отсюда имеем (в силу выше доказанного), что $(R_1 \cup R_2) * R = R_1 * R \cup R_2 * R = R_2 * R$, то есть $R_1 * R \subseteq R_2 * R$.

2. $(a, b) \in R^{-1} \Leftrightarrow (b, a) \in R \Leftrightarrow (b, a) \in (R^{-1})^{-1}$. Откуда следует, что $R = ((R^{-1})^{-1})$.

Для доказательства второй части заметим, что $(a, b) \in R \Leftrightarrow (b, a) \in R^{-1} \Rightarrow (a, b) \in R \Rightarrow (a, b) \in R_1 \Rightarrow (b, a) \in R^{-1} \Rightarrow (b, a) \in R_1^{-1}$, то есть $R^{-1} \subseteq R_1^{-1}$.

3. $(a, b) \in (R * R_1)^{-1} \Leftrightarrow (b, a) \in (R * R_1) \Rightarrow (\exists c \in A) (b, c) \in R \text{ и } (c, a) \in R_1$. Но тогда $(c, b) \in (R^{-1})$ и $(a, c) \in (R_1^{-1}) \Rightarrow (a, b) \in (R_1^{-1}) * (R^{-1})$, то есть $(R * R_1)^{-1} \subseteq (R_1^{-1}) * (R^{-1})$. Обратное включение доказывается аналогично.

4. $(a, b) \in (R \cap R_1)^{-1} \Leftrightarrow (b, a) \in R \cap R_1 \Leftrightarrow (b, a) \in R \text{ и } (b, a) \in R_1 \Leftrightarrow (a, b) \in (R^{-1})$ и $(a, b) \in (R_1^{-1})$, то есть $(R \cap R_1)^{-1} = (R^{-1}) \cap (R_1^{-1})$.

5. Пусть $(a, d) \in (R * R_1) * R_2$, тогда существует $c \in A$ такой, что $(a, c) \in R * R_1$ и $(c, d) \in R_2$. Следовательно, существует b такой, что $(a, b) \in R$, $(b, c) \in R_1$ и $(c, d) \in R_2$, а это означает, что $(b, d) \in R_1 * R_2$ и $(a, d) \in R * (R_1 * R_2)$, то есть $(R * R_1) * R_2 \subseteq R * (R_1 * R_2)$. Обратное включение доказывается аналогично.

2.1.4. Примеры отношений

Рассмотрим примеры наиболее важных бинарных отношений.

1. Отношение тождества. Бинарное отношение *тождества*, заданное на множестве A , состоит из всех пар вида (a, a) , где $a \in A$, и обозначается i_A или просто i , если A фиксировано. Пары вида (a, a) называют *диагональными*, а отношение i_A – *диагональю*. Очевидно, что для любого бинарного отношения R , определенного на множестве A , имеет место равенство $i_A * R = R * i_A = R$.

2. Рефлексивное отношение. Бинарное отношение R , заданное на множестве A , называется *рефлексивным*, если $i_A \subseteq R$, то есть, когда оно включает диагональ.

Примером рефлексивных отношений могут служить такие бинарные отношения:

- прямая x параллельна прямой y в плоскости z ;
- студент x – ровесник студента y .

Действительно, в первом случае из элементарной геометрии известно, что две прямые, которые лежат в одной плоскости, параллельны, если они либо совпадают, либо не имеют ни одной общей точки, сколько бы их не продолжали. Поскольку прямая x совпадает сама с собой, то пара (x, x) принадлежит данному отношению. Во втором случае очевидно, что каждый студент – сам себе ровесник.

3. Иррефлексивные отношения. Бинарное отношение R называется *иррефлексивным*, если aRa не имеет смысла. Например, отношение строгого порядка $x < y$ на множестве действительных или рациональных чисел не имеет смысла, потому что оно всегда ложно.

4. Симметричные отношения. Бинарное отношение R , заданное на множестве A , называется *симметричным*, если $aRb \Rightarrow bRa$ ($R \subseteq R^{-1}$).

Примером симметричных отношений могут служить такие бинарные отношения:

- прямая x перпендикулярна прямой y в плоскости z ;
- студент x является соседом по парте студента y .

Действительно, в первом случае из элементарной геометрии известно: если прямая x перпендикулярна прямой y , то и прямая y перпендикулярна прямой x .

Во втором случае каждый студент может убедиться, что когда студент y приходится ему соседом, то студенту x приходится соседом он сам. Заметим, что приведенные отношения не являются рефлексивными.

5. Транзитивные отношения. Бинарное отношение R , заданное на множестве A , называется *транзитивным*, если из aRb и bRc следует aRc ($R^2 \subseteq R$).

Примером транзитивных отношений могут служить такие бинарные отношения:

- город x связан с городом y шоссейной дорогой;
- студент x ровесник студента y ;
- треугольник x подобен треугольнику y ;
- действительное число x больше действительного числа y .

В первом случае, если между городами x и y есть шоссейная дорога и между городами y , z также есть шоссейная дорога, то ясно, что между городами x и z тоже есть шоссейная дорога, которая проходит, например, через город y .

В последних трёх случаях транзитивность очевидна.

6. Антисимметричные отношения. Бинарное отношение R , заданное на множестве A , называется *антисимметричным*, если из aRb и bRa следует $a = b$ ($R \cap R^{-1} \subseteq i_A$).

Примером антисимметричного отношения может служить бинарное отношение включения для множеств, то есть отношение “множество A является подмножеством множества B ”.

В самом деле, если $A \subseteq B$ и $B \subseteq A$, то из аксиомы объёмности следует, что множества A и B состоят из одних и тех же элементов, то есть $A = B$.

7. Отношение эквивалентности. Бинарное отношение R , заданное на множестве A , называется *отношением эквивалентности* или просто *эквивалентностью* на A , если для любых элементов a, b, c из A справедливы такие свойства:

- а) aRa ($i_A \subseteq R$) (рефлексивность)
б) $aRb \Rightarrow bRa$ ($R \subseteq R^{-1}$) (симметричность);

в) из aRb и bRc следует aRc ($R^2 \subseteq R$) (транзитивность),

где i_A – отношение тождества, а $R^2 = R * R$.

Несложно показать, что условия “а”, “б” и “в” эквивалентны таким $i_A \subseteq R$, $R = (R^{-1})$, $R^2 = R$.

Отношение эквивалентности, заданное на множестве A , тесно связано с разбиением множества A на классы. Эта связь выражается такими утверждениями.

Лемма 2.1. *Всякое разбиение множества A на классы задаёт на множестве A отношение эквивалентности.*

Д о к а з а т е л ь с т в о. Пусть $a, b \in A$, положим $aRb \Leftrightarrow a$ и b лежат в одном классе разбиения. Покажем, что полученное бинарное отношение является отношением эквивалентности. Для этого необходимо доказать, что оно рефлексивно, симметрично и транзитивно. Действительно, поскольку a лежит в некотором классе разбиения, то aRa , то есть оно рефлексивно.

Пусть K – некоторый класс разбиения и $a, b \in K$ тогда и $b, a \in K$ то есть $aRb \Rightarrow bRa$. Симметричность доказана.

Из aRb и bRc следует $a, b, c \in K$. Следовательно, aRc , что и следовало доказать.

Лемма 2.2. *Всякое отношение эквивалентности R , определенное на множестве A , задаёт разбиение множества A на классы.*

Д о к а з а т е л ь с т в о. Назовём классом элемента a множество $K(a) = \{x \in A \mid aRx\}$. Из рефлексивности отношения R следует, что $a \in K(a)$, то есть система классов $K(a)$ ($a \in A$) покрывает все множество A . Далее, симметричность отношения R показывает, что если $b \in K(a)$, то $a \in K(b)$, а транзитивность отношения R приводит к тому, что если $b \in K(a)$, то из того, что $c \in K(b)$, следует, что $c \in K(a)$, то есть $K(b) \subseteq K(a)$. Но если $a \in K(b)$, то $K(a) \subseteq K(b)$, значит $K(a) = K(b)$. Отсюда следует, что каждый класс определяется любым своим элементом. Если $K(a) \cap K(b) \neq \emptyset$, то существует такой элемент c из этого пересечения, что cRa и cRb и тогда $K(a) = K(b) = K(c)$, то есть классы $K(a)$ и $K(b)$ совпадают. Таким образом доказано, что система всех различных классов вида $K(a)$ является разбиением множества A . Что и требовалось доказать.

Очевидно, что переход от разбиения S множества A , определяемого отношением R , к отношению эквивалентности R , а после этого переход от отношения R к разбиению множества A , снова приводит к разбиению S . Таким образом, справедлива следующая теорема.

Теорема 2.4. *Между разбиениями множества на классы и отношениями эквивалентности, заданными на этом множестве, существует взаимно однозначное соответствие [56].*

Если R – эквивалентность на A , то классы разбиения, определенные отношением R , называют **классами эквивалентности** отношения R , а множество классов – **фактор-множеством** множества A и обозначают A/R . Число классов эквивалентности отношения эквивалентности R называется **индексом** множества A . Когда число классов эквивалентности конечное, то множество A называется **множеством конечного индекса**.

Выясним, какие из операций теории множеств сохраняют свойство отношения быть отношением эквивалентности, а какие нет.

Теорема 2.5. *Если R, R_1 – отношения эквивалентности, заданные на множестве A , то:*

1. R^{-1} – отношение эквивалентности на A ;
2. $R * R_1$ – отношение эквивалентности на A тогда и только тогда, когда $R * R_1 = R_1 * R$, то есть когда отношения R и R_1 можно поменять местами;

3. $R \cap R_1$ – отношение эквивалентности на A ;

4. R' не является отношением эквивалентности на A .

Доказательство. 1. Поскольку R – отношение эквивалентности, то $R^{-1} = R$ и, следовательно, R^{-1} – тоже отношение эквивалентности.

2. Если $R * R_1$ – отношение эквивалентности, то по доказанному выше $(R * R_1)^{-1}$ – тоже отношение эквивалентности и по теореме 2.3 $(R * R_1)^{-1} = (R_1^{-1}) * (R^{-1}) = R_1 * R$.

Наоборот, если $R * R_1 = R_1 * R$, то из того, что xRx и xR_1x , следует $xR * R_1x$, следовательно, $R * R_1$ рефлексивно. Далее, из того, что $(R * R_1)^{-1} = (R_1^{-1}) * (R^{-1}) = R_1 * R = R * R_1$, следует симметричность $R * R_1$. И, наконец, $(R * R_1) * (R * R_1) = R * R * R_1 * R_1 = R * R_1$ поскольку их можно менять местами и из ассоциативности умножения отношений (см. теорему 2.3). Следовательно, $R * R_1$ транзитивно.

3. Поскольку R и R_1 – эквивалентности на A , то $i_A \subseteq R$ и $i_A \subseteq R_1$, откуда следует, что $i_A \subseteq R \cap R_1$.

По теореме 2.3 и из того, что R и R_1 – эквивалентности, имеем $(R \cap R_1)^{-1} = (R^{-1}) \cap (R_1^{-1}) = R \cap R_1$. Значит отношение $R \cap R_1$ рефлексивно и симметрично. Покажем транзитивность.

Пусть $(a, b) \in R \cap R_1$ и $(b, c) \in R \cap R_1$, тогда $(a, b) \in R$, $(a, b) \in R_1$ и $(b, c) \in R$, $(b, c) \in R_1$. Из $(a, b) \in R$, $(b, c) \in R$ следует, что $(a, c) \in R$, а из $(a, b) \in R_1$, $(b, c) \in R_1$, – что $(a, c) \in R_1$. Отсюда заключаем, что $(a, c) \in (R \cap R_1)$, то есть $R \cap R_1$ транзитивно.

4. Из того, что R – эквивалентность, следует, что $i_A \subseteq R$, но тогда i_A не может быть подмножеством множества R' , то есть R' не рефлексивно и, следовательно, не является отношением эквивалентности.

Объединение отношений эквивалентности в общем случае не является отношением эквивалентности, как показывает следующая теорема.

Теорема 2.6. *Объединение $R \cup R_1$ отношений эквивалентности R и R_1 является эквивалентностью тогда и только тогда, когда пересечение любого класса эквивалентности по R с любым классом эквивалентности по R_1 либо совпадает с одним из них, либо пусто. Если $R \cup R_1$ – эквивалентность, то $R \cup R_1 = R * R_1$ [55].*

8. Замыкание отношений. Пусть R – некоторое бинарное отношение на множестве A :

а) **рефлексивным замыканием** R_1 отношения R называется отношение $R \cup i$, где i – отношение тождества на A (диагональ);

б) **симметричным замыканием** R_s отношения R называется отношение $R \cup R^{-1}$, то есть, если $(a, b) \in R$, то $(a, b) \in R_s$ и $(b, a) \in R_s$;

в) **транзитивным замыканием** R_t отношения R называется отношение $R_t = R \cup R^2 \cup R^3 \cup \dots \cup R^n \cup \dots$, то есть $(a, b) \in R_t$ тогда и только тогда, когда существуют элементы $a_1 = a, a_2, \dots, a_n = b \in A$ такие, что $(a_1Ra_2, a_2Ra_3, \dots, a_{n-1}Ra_n)$.

Если некоторое отношение включает свое симметричное, рефлексивное и транзитивное замыкания, то оно является отношением эквивалентности и наоборот.

В качестве примеров рассмотрим следующие.

Пусть $A = \{a_1, a_2, a_3, a_4\}$ и $R = \{(a_1, a_3), (a_3, a_4), (a_4, a_2), (a_3, a_3)\}$. Тогда

$R_i = \{(a_1, a_3), (a_3, a_4), (a_4, a_2), (a_3, a_3), (a_1, a_1), (a_2, a_2), (a_4, a_4)\}$,

$R_s = \{(a_1, a_3), (a_3, a_1), (a_3, a_4), (a_4, a_3), (a_4, a_2), (a_2, a_4), (a_3, a_3)\}$,

$R_t = \{(a_1, a_3), (a_3, a_4), (a_4, a_2), (a_3, a_3), (a_1, a_4), (a_1, a_2), (a_3, a_2)\}$.

9. Отображения и операции. Отношение F , заданное на множествах A_1, A_2, \dots, A_n, B , называется **функциональным**, если для любого элемента (a_1, a_2, \dots, a_n) из

$A_1 \times A_2 \times \dots \times A_n$ существует не более одного элемента b из B такого, что $(a_1, a_2, \dots, a_n, b) \in F$. Если такой элемент b из B существует для некоторого (a_1, a_2, \dots, a_n) , то он обозначается $F(a_1, a_2, \dots, a_n)$ и записывается так: $b = F(a_1, a_2, \dots, a_n)$.

Пусть $F^{-1}(b) = \{(a_1, a_2, \dots, a_n) \in A_1 \times A_2 \times \dots \times A_n \mid F(a_1, a_2, \dots, a_n) = b\}$ и $\text{Dom}(F) = \bigcup_{b \in B} F^{-1}(b)$. Очевидно, что для всякого функционального отношения F ,

заданного на A_1, A_2, \dots, A_n, B , имеет место включение $\text{Dom}(F) \subseteq A_1 \times A_2 \times \dots \times A_n$.

В случае, когда $\text{Dom}(F) = A_1 \times A_2 \times \dots \times A_n$, F называется **полностью определенным**, когда $\text{Dom}(F) \subset A_1 \times A_2 \times \dots \times A_n$, – **частично определенным**, или просто **частичным**.

Отношение F , заданное на множествах A_1, A_2, \dots, A_n, B , называется **отображением**, или **функцией** из $A_1 \times \dots \times A_n$ в B ($F: A_1 \times A_2 \times \dots \times A_n \rightarrow B$), если F функциональное и полностью определенное. Отношение F называется **частичным отображением**, или **частичной функцией**, если F функциональное и частичное. Число n называется **арностью** функции F .

Если $F: A_1 \times A_2 \times \dots \times A_n \rightarrow B$ и существует b из B такой, что $F(a_1, a_2, \dots, a_n) = b$, то элемент b называют **образом** элемента (a_1, a_2, \dots, a_n) при отображении F , а элемент (a_1, a_2, \dots, a_n) – **прообразом** элемента b . Множество

$$F^{-1}(b) = \{(a_1, a_2, \dots, a_n) \mid F(a_1, a_2, \dots, a_n) = b\},$$

введенное ранее, называют **полным прообразом** элемента b в множестве $A_1 \times A_2 \times \dots \times A_n$.

Отношение $F: A_1 \times A_2 \times \dots \times A_n \rightarrow B$ называют **отображением на B** тогда и только тогда, когда $(\forall b \in B) (F^{-1}(b) \neq \emptyset)$.

Отображение F множества $A_1 \times A_2 \times \dots \times A_n$ на множество B называется **взаимно однозначным отображением**, или **взаимно однозначным соответствием** тогда и только тогда, когда обратное к отношению F отношение F^{-1} является отображением B на $A_1 \times A_2 \times \dots \times A_n$.

Выше была введена операция произведения отношений. Поскольку отображение – это отношение специального вида, то выясним, что собой представляет произведение отображений.

Пусть $F_1: A \rightarrow B$, а $F: B \rightarrow C$ – некоторые отображения. По определению произведения отношений имеем: $(a, c) \in F_1 * F$ тогда и только тогда, когда существует элемент $b \in B$ такой, что $(a, b) \in F_1$ и $(b, c) \in F$, то есть $F_1(a) = b$ и $F(b) = c$, или $F(F_1(a)) = c$ согласно с принятыми выше обозначениями. Таким образом, произведение отношений представляет собой хорошо известную операцию суперпозиции функций. Из последнего пункта теоремы 2.3 следует, что произведение отображений удовлетворяет закону ассоциативности.

Примером **отображения на** и взаимно однозначного отображения может служить отображение, которое связывает множество A с его фактор-множеством A/R , где R – некоторое отношение эквивалентности на A . Отображение $F: A \rightarrow A/R$, которое задаёт для любого элемента a из A тот класс разбиения, которому принадлежит a , называется **натуральным отображением A на A/R** .

Между отношениями эквивалентности, заданными на некотором множестве, и отображениями этого множества на другие множества существует тесная связь. Действительно, если $F_1: A \rightarrow B$ – **отображение на**, то ему соответствует полностью определенное отношение эквивалентности R на A : если $a, b \in A$, то $aRb \Leftrightarrow F_1(a) = F_1(b)$. Сопоставляя каждому элементу x из B его полный прообраз в

A , получаем отображение $F_2: B \rightarrow A/R$, основное свойство которого доказывает следующая теорема.

Теорема 2.7. *Отображение $F_2: B \rightarrow A/R$ является взаимно однозначным отображением, причем $F_1 * F_2 = F$, где F – натуральное отображение A на A/R .*

Доказательство. Определим $F_2: B \rightarrow A/R$ так, чтобы $F_2(b) = K(b) = \{a \in A \mid F_1(a) = b\}$. Ясно, что когда $b \neq b'$, то $K(b) \neq K(b')$. Следовательно, отображение F_2 взаимно однозначное. Далее, пусть $F_1(a) = b$, а $F_2(b) = K(b)$, тогда $a \in K(b)$. Таким образом, произведение $F_1 * F_2$ совпадает с натуральным отображением F . Теорема доказана.

Если $F: A^n \rightarrow B$, то F называют **n -арной функцией** из A в B , а если при этом $B = \{0, 1\}$, то F называется **n -арным предикатом** на множестве A , а элементы $0, 1$ – соответственно **ложью** и **истиной**. Если предикат F имеет то свойство, что для всех $(a_1, a_2, \dots, a_n) \in A$

$$F(a_1, a_2, \dots, a_n) = 1 \quad (F(a_1, a_2, \dots, a_n) = 0),$$

то предикат F называется **тождественно истинным (ложным)** на A .

Между отношениями и предикатами, заданными на одном и том же множестве A , существует взаимно однозначное соответствие. Действительно, пусть F – n -арный предикат на A . Совокупность тех последовательностей из A^n , для которых $F(a_1, a_2, \dots, a_n) = 1$, является отношением на A , соответствующим предикату F . Наоборот, пусть задано произвольное n -арное отношение $R \subseteq A^n$ на A . Полагая

$$F(a_1, a_2, \dots, a_n) = \begin{cases} 1, & \text{если } (a_1, a_2, \dots, a_n) \in R, \\ 0, & \text{если } (a_1, a_2, \dots, a_n) \notin R, \end{cases}$$

получаем n -арный предикат, который соответствует отношению R . Таким образом, n -арные отношения и n -арные предикаты на произвольном множестве находятся во взаимно однозначном соответствии.

Если F – n -арная функция из A^n в A , то F еще называют **n -арной операцией** на A . При $n = 0$ F называется **нульарной операцией**, значением которой является фиксированный элемент множества A .

Операция F называется **частичной**, если F – частичная функция. Пусть $F: A_1 \times A_2 \times \dots \times A_n \Rightarrow B$ и $F^{-1}(b)$ – полный прообраз элемента b в $A_1 \times A_2 \times \dots \times A_n$ при отображении F . Введенное выше множество $\text{Dom}(F) = \bigcup_{b \in B} F^{-1}(b)$ называется

областью определения отображения F , а $\text{Im}(F) = \{b \mid F^{-1}(b) \neq \emptyset\}$ – **областью значений**. Если $F: A_1 \times A_2 \times \dots \times A_n \Rightarrow B$ и $F_1: A_1 \times A_2 \times \dots \times A_n \Rightarrow B$, то $F = F_1$ тогда и только тогда, когда $\text{Dom}(F) = \text{Dom}(F_1)$, $\text{Im}(F) = \text{Im}(F_1)$ и $F(a_1, a_2, \dots, a_n) = F_1(a_1, a_2, \dots, a_n)$ для любого $(a_1, a_2, \dots, a_n) \in A_1 \times A_2 \times \dots \times A_n$.

Далее будем обозначать предикаты, функции, операции строчными латинскими буквами, и если Ω – некоторое множество предикатов, функций или операций, то функция $ar: \Omega \rightarrow \mathbf{N}$, где \mathbf{N} – множество натуральных чисел, называется **функцией арности n** , то есть $ar(\omega) = n$, если $\omega \in \Omega$ и имеет арность n .

10. Отношение частичного порядка. Бинарное отношение O , заданное на множестве A , называется **частичным порядком** на A , если для любых a, b, c из A выполняются свойства:

- | | | |
|-------------------------------------|---------------------------------|-----------------------|
| a1) aOa | $(i_A \subseteq O)$ | (рефлексивность); |
| a2) aOb и $bOc \Rightarrow aOc$ | $(O^2 \subseteq O)$ | (транзитивность); |
| a3) aOb и $bOa \Rightarrow a = b$ | $(O \cap O^{-1} \subseteq i_A)$ | (антисимметричность). |

Частичный порядок на множестве A , как правило, обозначают символом \leq . Если $a \leq b$ для некоторых $a, b \in A$, то говорят, что a меньше или равно b , а также,

что a включается в b либо равно b . Если $a \leq b$ и $a \neq b$, то говорят, что a строго меньше b ($a < b$).

Определение 2.1. Транзитивное и иррефлексивное отношение называется отношением строгого порядка. Это отношение обозначают $<$.

Определение 2.2. Транзитивное и рефлексивное отношение называется отношением квазипорядка или предпорядка. Это отношение обозначают \preceq .

С каждым отношением частичного порядка \leq связано отношение строгого порядка $<$ (этот порядок называют строгой частью отношения \leq):

$$(a < b) \Leftrightarrow a \leq b \text{ и } a \neq b.$$

Наоборот, с каждым отношением строгого порядка $<$ связано отношение частичного порядка \leq :

$$(a \leq b) \Leftrightarrow a < b \text{ или } a = b.$$

С каждым отношением квазипорядка \preceq связаны отношение строгого порядка $<$ и эквивалентности \sim :

$$(a < b) \Leftrightarrow a \preceq b \text{ и } \neg(b \preceq a);$$

$$(a \sim b) \Leftrightarrow a \preceq b \text{ и } b \preceq a.$$

Любое отношение квазипорядка \preceq , заданное на множестве A , индуцирует отношение частичного порядка \leq на фактор-множестве A/\sim :

$$([a]_{\sim} \leq [b]_{\sim}) \Leftrightarrow a \preceq b.$$

Действительно, поскольку отношение \preceq транзитивно и рефлексивно, то и отношение \leq тоже будет транзитивным и рефлексивным. Покажем антисимметричность. $[a]_{\sim} \leq [b]_{\sim}$ и $[b]_{\sim} \leq [a]_{\sim}$ означают, что $a \preceq b$ и $b \preceq a$. Однако отсюда следует, что $a \sim b$, то есть что $[a]_{\sim} \sim [b]_{\sim}$.

Простые свойства частично упорядоченных множеств.

Теорема 2.8 (принцип двойственности). *Отношение, обратное к отношению частичного порядка, также будет отношением частичного порядка.*

До к а з а т е л ь с т в о. Пусть O^{-1} – отношение, обратное к отношению O :

б1) поскольку $i_A \subseteq O$, то $i_A = i_A^{-1} \subseteq O^{-1}$;

б2) если $O * O \subseteq O$, то по теореме 2.3 $O^{-1} * O^{-1} = (O * O)^{-1} \subseteq O^{-1}$;

б3) если $O \cap O^{-1} \subseteq i_A$, то $O^{-1} \cap O \subseteq i_A \supseteq O^{-1} \cap (O^{-1})^{-1}$ из соотношения М1 и теоремы 2.3, что и требовалось доказать.

Отношение частичного порядка O^{-1} называется **двойственным** к отношению частичного порядка O .

Определение 2.3. Отношение частичного порядка \leq^{-1} называется **двойственным** к отношению частичного порядка \leq .

Отношение \leq^{-1} обозначается \geq . Таким образом, $a (\leq^{-1}) b$ означает $b \geq a$. Если $a \leq b$ или $b \leq a$, то элементы a, b называют элементами, **сравнимыми относительно порядка \leq** .

Из принципа двойственности следует, что когда в каком-нибудь утверждении о частично упорядоченном множестве заменить частичный порядок на двойственный к нему порядок, то полученное утверждение тоже будет верным.

Теорема 2.9. *Любое подмножество частично упорядоченного множества также будет частично упорядоченным множеством.*

Доказательство предлагается как простое упражнение.

Определение 2.4. Элемент x из множества (A, \leq) называется **минимальным** (**максимальным**) элементом A , если для всякого элемента a из A , сравнимого с x , имеет место неравенство $x \leq a$ ($x \geq a$).

Определение 2.5. Элемент x из A называется **наибольшим** (**наименьшим**), если $(\forall a \in A) x \geq a$ ($x \leq a$).

Теорема 2.10. В каждом частично упорядоченном множестве существует не более одного наименьшего (a в силу принципа двойственности и наибольшего) элемента.

Доказательство. Предположим, что a и b – два наименьших элемента в множестве A , тогда $a \leq b$, поскольку a – наименьший элемент и $b \leq a$, поскольку b – наименьший элемент. Однако тогда из транзитивности отношения \leq следует, что $a = b$.

Если произвольные два элемента из множества A сравнимы относительно \leq , то такое отношение называется **линейным порядком** на A , а множество A – **упорядоченным** или **линейно упорядоченным**, или **цепочкой**.

Ясно, что если линейно упорядоченное множество A имеет наибольший (наименьший) элемент, то этот элемент будет единственным максимальным (минимальным) элементом.

Определение 2.6. Пусть (A, \leq) – частично упорядоченное множество и $a, b \in A$. Говорят, что элемент b доминирует над элементом a , если $b > a$ и ни для какого элемента x из A неверно, что $b > x > a$.

Следующая простая теорема показывает, что отношение частичного порядка \leq можно однозначно восстановить по отношению доминирования в любом конечном частично упорядоченном множестве.

Теорема 2.11. Пусть $a < b$ в конечном частично упорядоченном множестве (A, \leq) . Тогда в (A, \leq) существует хотя бы одна цепочка $a < x_1 < x_2 < \dots < x_n < b$, в которой каждый x_i доминирует над x_{i-1} , $i = 1, 2, \dots, n$.

Доказательство индукцией по числу n элементов y , удовлетворяющих условию $a < y < b$ (см. ниже метод трансфинитной индукции).

При $n = 0$ b доминирует над a по определению.

Шаг индукции. Предположим, что теорема верна для всех $m < n$. Рассмотрим случай $n = m$, где $n > 0$. Поскольку $n > 0$, то существует такой элемент $c \in A$, что $a < c < b$ и число элементов y, z , удовлетворяющих условиям $a < y < c$ и $c < z < b$, не превышает $n - 1$. По предположению индукции существуют конечные цепочки, связывающие a и c , c и b , средние элементы которых находятся в отношении доминирования. Соединив эти две цепочки, получим искомую цепочку.

В качестве примеров рассмотрим следующие.

1. Множества \mathbf{N} , \mathbf{N}^+ , \mathbf{Z} , \mathbf{RC} и множество \mathbf{D} действительных чисел с их естественным порядком; множество \mathbf{N} часто называют натуральным рядом, а любое его подмножество вида $\{0, 1, 2, \dots, n\}$ – начальным отрезком, либо просто отрезком натурального ряда.

2. Множество точек числовой оси (прямой).

В качестве примеров рассмотрим следующие.

3. Булеан $\mathbf{B}(A)$, то есть множество всех подмножеств некоторого множества A с отношением теоретико-множественного включения \subseteq как отношением частичного порядка;

4. Множество \mathbf{N}^+ с отношением $n \leq n_1 \Leftrightarrow n_1$ делится нацело на n .

Линейно упорядоченное множество A называется **полностью упорядоченным**, если любое его непустое подмножество B имеет наименьший элемент. С понятием полностью упорядоченного множества связан один из основных постулатов теории множеств.

Аксиома полной упорядоченности. *Любое непустое множество можно полностью упорядочить.*

Заметим, что эта аксиома логически эквивалентна другой аксиоме теории множеств – аксиоме выбора [56–58].

Аксиома выбора. *Если дано множество A , то существует функция f , которая ставит в соответствие каждому непустому подмножеству B множества A один определенный элемент $f(B)$ из множества B .*

Логическую эквивалентность аксиом следует понимать так: если одну из них принять за аксиому, то другую можно строго доказать как теорему и наоборот. Так, если принимается аксиома выбора, то аксиома полной упорядоченности становится утверждением, известным в теории множеств как теорема Цермело.

Благодаря аксиоме полной упорядоченности многие свойства полностью упорядоченных множеств можно доказывать методом трансфинитной индукции.

Теорема 2.12 (метод трансфинитной индукции). *Пусть e – наименьший элемент полностью упорядоченного множества A и $P(x)$ – некоторое свойство элемента $x \in A$. Тогда если из истинности $P(e)$ и $P(x)$ для всех $x < a$ следует истинность $P(a)$, то $P(x)$ истинно для всех x из A .*

Д о к а з а т е л ь с т в о. Допустим противное, существует такое непустое подмножество A' элементов из A , что для всех $y \in A'$ $P(y)$ ложно при выполнении условий теоремы. Пусть b – минимальный элемент в A' . Поскольку $P(e)$ истинно, то $b \neq e$ и $b > e$. Из условий теоремы следует, что $P(x)$ истинно для всех $x < b$, но тогда из этих же условий должна следовать истинность и $P(y)$, а это противоречит нашему предположению. Теорема доказана.

Метод трансфинитной индукции даёт возможность не только доказывать свойства по индукции, но и выполнять построение по индукции или давать определение по индукции.

Действительно, пусть A – полностью упорядоченное множество, и пусть мы хотим определить на этом множестве функцию $f(x)$, которая ставит в соответствие каждому элементу x из A некоторый элемент множества B . Допустим также, что $f(x)$ должна удовлетворять некоторым **рекуррентным соотношениям**, то есть соотношениям, которые однозначно задают для каждого $a \in A$ значение $f(a)$ по значениям $f(b)$ для всех $b < a$.

Теорема 2.13. Метод построения по индукции. *Существует единственная функция $f(x)$, которая определена на всем множестве A , удовлетворяет указанным рекуррентным соотношениям и принимает произвольно заданные значения на минимальном элементе множества A .*

Д о к а з а т е л ь с т в о. Покажем сначала единственность такой функции. Допустим, что существуют две разные функции $f(x)$ и $g(x)$ на множестве A , которые удовлетворяют нашим условиям. Пусть существует непустое подмножество элементов x из A , для которых $f(x) \neq g(x)$. Поскольку A полностью упорядочено, то это подмножество имеет минимальный элемент a . Этот элемент не может быть минимальным для всего множества A , потому что тогда, по условию, на этом элементе $f(a)$ и $g(a)$ совпадали бы. Значит существует $b < a$ такой, что $f(b) = g(b)$. По условию теоремы рекуррентные соотношения однозначно задают значения наших функций

для $x = a$ по их значениям для всех $b < a$, следовательно, $f(a) = g(a)$. Полученное противоречие доказывает единственность $f(x)$.

Докажем теперь существование функции. Допустим, что на минимальном элементе множества A значение искомой функции уже задано. Обозначим через P такое свойство: элемент $a \in A$ удовлетворяет свойству P , если на множестве S всех таких x , что $x \leq a$, может быть определена функция $f_a(x)$, которая удовлетворяет рекуррентным соотношениям и принимает заданное значение на минимальном элементе множества A .

По предположению P истинно на минимальном элементе $a \in A$. Далее, если элементы b и c удовлетворяют свойству P и $b < a$, то в силу доказанной выше единственности искомой функции не на множестве A , а на множестве $B = \{x \in A \mid x \leq b\}$, имеем $f_b(x) = f_a(x)$.

Отсюда следует, что, если все элементы b строго меньше элемента a удовлетворяют свойству P , то и сам элемент a удовлетворяют этому свойству. Получаем функцию $f_a(x)$, удовлетворяющую всем требованиям, если для каждого $b < a$ положить $f_a(b) = f_b(b)$, а за $f_a(a)$ взять то значение, которое однозначно задается рекуррентными соотношениями.

На основе метода трансфинитной индукции можно говорить, что для всех a из A истинно $P(a)$. Полагая теперь $(\forall a \in A) f_a(a) = f(a)$, определяем функцию $f(x)$, которая обладает необходимыми свойствами.

Заметим, что метод трансфинитной индукции, как и метод построения по индукции, можно применять и к частично упорядоченным множествам. Ограничимся только формулировками этих фактов.

Теорема 2.14 (условие индуктивности). *Все элементы частично упорядоченного множества A удовлетворяют свойству P , если:*

1) *все минимальные элементы из множества A удовлетворяют свойству P (в том случае, если они существуют);*

2) *из того, что $P(x)$ истинно для всех $x < a$, где $x, a \in A$, следует истинность $P(a)$.*

Теорема 2.15 (построение по индукции). *Существует единственная функция $f(x)$, которая задана на всем множестве A , удовлетворяет указанным рекуррентным соотношениям и принимает произвольные заданные значения на всех минимальных элементах множества A [56, 57].*

2.1.5. Матрицы как пример отображения

Рассмотрим один пример отображения, который называется **матрица**.

Пусть $p, q \in \mathbb{N}^+$, N_p и N_q означают множества чисел $\{1, 2, \dots, p\}$ и $\{1, 2, \dots, q\}$ соответственно, а P – произвольное множество чисел.

Отображение $A: N_p \times N_q \rightarrow P$, где $p, q \in \mathbb{N}^+$, называется прямоугольной **матрицей** над множеством чисел P . Образ $A(i, j)$ часто обозначают a_{ij} и называют элементом матрицы A , а саму матрицу A задают с помощью таблицы элементов-образов из множества P , которые соответствуют отображению A , то есть

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1q} \\ a_{21} & a_{22} & \dots & a_{2q} \\ \dots & \dots & \dots & \dots \\ a_{p1} & a_{p2} & \dots & a_{pq} \end{pmatrix}.$$

При этом говорят, что матрица A состоит из p строк и q столбцов и имеет размерность $p \times q$. Если $p = q$, то матрица A называется **квадратной**. Когда элементы квадратной матрицы таковы, что $a_{ij} = 0$ при $i \neq j$, и $a_{ij} \neq 0$ при $i = j$, то матрица A называется **диагональной**. Диагональная матрица называется **единичной**, если $a_{ii} = 1$ для всех i .

Матрица представляет собой один из удобных способов задания бинарных отношений на конечных множествах.

Пусть R – бинарное отношение, заданное на конечных множествах $A = \{a_1, a_2, \dots, a_p\}$ и $B = \{b_1, b_2, \dots, b_q\}$. Рассмотрим матрицу $A(R): N_p \times N_q \Rightarrow \{0, 1\}$, которая связана с отношением R (с помощью отображения A) таким образом:

$$a_{ij} = \begin{cases} 1, & \text{если } (a_i, b_j) \in R \\ 0, & \text{если } (a_i, b_j) \notin R. \end{cases}$$

Например, если $A = \{a_1, a_2, a_3, a_4\}$, $B = \{b_1, b_2, b_3\}$, $R = \{(a_1, b_2), (a_1, b_3), (a_2, b_1), (a_3, b_1), (a_4, b_2)\}$, то матрица $A(R)$ имеет такой вид:

$$A(R) = \begin{vmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{vmatrix}.$$

Заметим, что в отдельном случае, когда R – бинарное отношение, заданное на конечном множестве A , то:

а) если $R = i$ – отношение тождества на A , то

$$A(i) = \begin{vmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & \dots & 0 & 1 \end{vmatrix}$$

является единичная диагональная матрица (откуда и название отношения). Отсюда легко найти рефлексивное замыкание некоторого заданного отношения R на множестве A . Для этого в матрице $A(R)$ необходимо всюду поставить на диагонали 1 вместо 0;

б) если R – симметричное отношение, то матрица $A(R)$ будет, очевидно, симметричной. Действительно, в силу симметричности отношения R , если $(a_i, b_j) \in R$ ($a_{ij} = 1$), то $(b_j, a_i) \in R$ ($a_{ji} = 1$). Принимая во внимание это обстоятельство, легко построить матрицу симметричного замыкания $A(R_s)$ при условии, что $A(R)$ является матрицей некоторого отношения R . Для этого необходимо в $A(R)$ заменить 0 на 1 в соответствующих местах с учётом симметрии.

Например, если $A = \{a_1, a_2, a_3, a_4\}$, $R = \{(a_1, a_3), (a_1, a_4), (a_2, a_2), (a_2, a_4), (a_3, a_4)\}$, и

$$A(R) = \begin{vmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{vmatrix}, \quad \text{то} \quad A(R_s) = \begin{vmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{vmatrix}.$$

Более того, если отношение R имеет k элементов, среди которых r ($r < |A|$) диагональных, то R_s имеет $2(k - r) + r = 2k - r$ элементов. Следовательно, если

$A(R_\zeta)$ имеет нечётное число элементов равных единице, то R_ζ должно включать хотя бы одну диагональную пару.

2.2. Частично рекурсивные функции. Тезис Чёрча

Понятие алгоритма, как и понятие множества, относится к интуитивным понятиям и поэтому требует уточнения. Одним из такого типа уточнений служит понятие частично рекурсивной функции. Приведём это уточнение понятия алгоритма и определим с его помощью понятие алгоритмической разрешимости.

Пусть \mathbf{N} – множество натуральных чисел и $F = \{f_1, f_2, \dots, f_n, \dots\}$ – совокупность n -арных функций на множестве \mathbf{N} .

Рассмотрим унарные функции o , s и n -арную функцию $I_m^n(x_1, \dots, x_n)$, которые определяются следующим образом:

$$o(x) = 0, \quad s(x) = x + 1, \quad I_m^n(x_1, \dots, x_n) = x_m.$$

Эти функции будем называть **простейшими функциями**. К простейшим относятся также и n -арную функцию $o^n(x_1, \dots, x_n)$, равную 0 для всех $x_1, x_2, \dots, x_n \in \mathbf{N}$.

Рассмотрим $n+1$ функцию из F : функцию f - арности n и функции f_1, \dots, f_n одной и той же арности m .

Говорят, что m -арная частичная функция $g(x_1, \dots, x_m)$ получена в результате **операции суперпозиции** или **подстановки** (S^{n+1}) из функций f^n, f_1^m, \dots, f_n^m , если для любых $x_1, \dots, x_m \in \mathbf{N}$

$$g^m(x_1, \dots, x_m) = f^n(f_1^m(x_1, \dots, x_m), \dots, f_n^m(x_1, \dots, x_m)),$$

где некоторые из x_i могут входить в f_j фиктивно.

В качестве примера рассмотрим n -арную функцию $f(x_1, \dots, x_n) = a$, где a – константа.

Тогда

$$a = \underbrace{s(\dots s(o^n(x_1, \dots, x_n)) \dots)}_{a \text{ раз}},$$

то есть константу a получаем с помощью операции суперпозиции из простейших функций o и s .

Пусть заданы произвольные частичные функции: n -арная функция g и $n+2$ -арная функция h . Говорят, что $n+1$ -арная функция f получена **операцией примитивной рекурсии** из функций g, h ($R(g, h)$), если для всех $x_1, \dots, x_n, y \in \mathbf{N}$ имеем

$$\begin{aligned} f(x_1, \dots, x_n, 0) &= g(x_1, \dots, x_n), \\ f(x_1, \dots, x_n, y+1) &= h(x_1, \dots, x_n, y, f(x_1, \dots, x_n, y)). \end{aligned}$$

Определение 2.7. Пусть Z – некоторая система частичных функций. Функция f называется **примитивно рекурсивной** относительно системы функций Z , если f можно получить из функций системы Z и простейших функций o, s, I с помощью конечного числа операций подстановки и примитивной рекурсии. Примитивно рекурсивная функция f относительно пустой системы функций Z называется **просто примитивно рекурсивной функцией**.

Заметим, что операции подстановки и примитивной рекурсии, применённые к полностью определенным функциям, дают снова полностью определенные функции. В частности, все примитивно рекурсивные функции полностью определены.

В качестве примеров рассмотрим следующие.

1. Для n -арной операции $o^n(x_1, \dots, x^n) = 0$ имеем

$$o^n(x_1, \dots, x_n) = S^2(o, I_1^n) = o(I_1^n(x_1, \dots, x_n)) = o(x_1) = 0.$$

Следовательно, эта функция примитивно рекурсивная.

2. Функцию $f(x, y) = x + y$ можно получить из функций $I_1^1(x)$ и $h(x, y, z) = z + 1$ операцией примитивной рекурсии. Действительно,

$$\begin{aligned} x + 0 &= x = I_1^1(x), \\ x + (y + 1) &= (x + y) + 1 = s(x + y). \end{aligned}$$

3. Функция $x - y$ в множестве \mathbf{N} частична. Для того, чтобы сделать эту функцию полностью определенной на множестве \mathbf{N} , рассматривают усеченную разность

$$x \div y = \begin{cases} x - y, & x \geq y, \\ 0, & x < y. \end{cases}$$

Непосредственно из определения этой функции следуют такие свойства:

$$\begin{aligned} x \div 0 &= x, & x \div (y + 1) &= (x \div y) \div 1, \\ (x \div y) \div z &= x \div (y + z). \end{aligned}$$

Функция $x \div 1$ примитивно рекурсивная, поскольку

$$0 \div 1 = 0 = o(x), \quad (x + 1) \div 1 = x = I_1^2(x, y),$$

то есть её можно получить из простейших функций o и I_1^2 операцией примитивной рекурсии. Из примитивной рекурсивности функции $x \div 1$ и приведенных свойств операции усеченной разности имеем

$$x \div 0 = x = I_1^1(x), \quad x \div (y + 1) = (x \div y) \div 1,$$

т. е. функцию $x \div y$ можно получить с помощью операции примитивной рекурсии из функций $I_1^1(x)$ и $h(x, y, z) = z \div 1$. Отсюда также следует, что $|x - y| = (x \div y) + (y \div x)$ – тоже примитивно рекурсивная функция.

4. Функцию $f(x, y) = x \cdot y$ можно получить из функций $o(x)$ и $h(x, y, z) = z + x$ операцией примитивной рекурсии. Действительно,

$$x \cdot 0 = 0 = o(x), \quad x \cdot (y + 1) = x \cdot y + x,$$

поскольку $x + y$ – примитивно рекурсивная функция.

5. Аналогично предыдущему случаю, функцию x^n можно получить из функций $g(x) = 1$ и $h(x, y, z) = z \cdot x$ операцией примитивной рекурсии. Действительно,

$$x^0 = 1, \quad x^{n+1} = x^n \cdot x.$$

Рассмотрим ещё некоторые свойства примитивно рекурсивных функций, которые понадобятся в дальнейшем.

Теорема 2.16. Пусть n -арная частичная функция g примитивно рекурсивная (относительно системы частичных функций Z). Тогда n -арные функции f , определенные с помощью уравнений

$$\begin{aligned} f(x_1, x_2, \dots, x_n) &= \sum_{i=0}^{x_n} g(x_1, x_2, \dots, x_{n-1}, i), \\ f(x_1, x_2, \dots, x_n) &= \prod_{i=0}^{x_n} g(x_1, x_2, \dots, x_{n-1}, i), \end{aligned}$$

также примитивно рекурсивные (относительно Z).

Доказательство. Из заданных в условии уравнений следует, что

$$\begin{aligned} f(x_1, \dots, x_{n-1}, 0) &= g(x_1, \dots, x_{n-1}, 0), \\ f(x_1, \dots, x_{n-1}, y+1) &= f(x_1, \dots, x_{n-1}, y) + g(x_1, \dots, x_{n-1}, y+1) \\ f(x_1, \dots, x_{n-1}, y+1) &= f(x_1, \dots, x_{n-1}, y) \cdot g(x_1, \dots, x_{n-1}, y+1). \end{aligned}$$

Тезис Тьюринга. Класс функций, которые алгоритмически вычисляются относительно системы функций Z , совпадает с классом частично рекурсивных функций относительно системы Z .

Поскольку понятие вычислимой функции не имеет точного определения, то доказать эти тезисы невозможно. Но благодаря им стало возможным придать необходимую точность формулировкам алгоритмических проблем.

2.2.1. Алгоритмические проблемы

Под алгоритмической проблемой будем понимать вопрос вычислимости некоторых специальным образом построенных функций. Вследствие тезисов Чёрча и Тьюринга вопрос о вычислимости функции равнозначен вопросу о её рекурсивности. Понятие рекурсивной функции строгое, и иногда с помощью обычной математической техники можно непосредственно доказать, что функция, которая решает проблему, не может быть рекурсивной. Если для решения проблемы не существует рекурсивной функции, то проблема называется (алгоритмически) **неразрешимой**. Если же рекурсивная функция существует, то проблема называется (алгоритмически) **разрешимой**.

Всякий алгоритм представляет собой способ решения некоторой *массовой* проблемы, т. е. проблемы преобразования (вычисления) не одного входного слова (одних входных данных), а целого множества входных слов (данных) в соответствующие им выходные слова. Детальный анализ задач показывает, что существуют такие классы задач, для решения которых нет, и не может быть единого универсального способа, т. е. эти задачи относятся к алгоритмически неразрешимым проблемам. Однако это совсем не означает невозможность решения какой-нибудь конкретной задачи из этого класса. Речь идёт о невозможности решения всех задач данного класса одним и тем же способом.

2.3. Формальные логические языки

Когда рассматривается проблема анализа естественно-языкового текста на предмет извлечения знаний, то выразительных свойств языков аристотелевской логики (логики высказываний и предикатов) не хватает. Это связано с тем, что в этих языках нельзя выразить эффективным образом модальности. Например, предложения «всегда, когда к базе данных послан запрос, будет когда-нибудь получен на него ответ» или «мы с ним не знакомы, до тех пор пока он не извинится» нельзя достаточно точно выразить средствами аристотелевской логики. Дело в том, что языки аристотелевской логики «статичны», т. е. в них не учитывается время и формулы этих языков делятся либо на истинные либо на ложные. В естественных языках во множестве встречаются предложения, приведенные выше в примерах.

Другая проблема, которая появляется в такого рода задачах – это представление знаний в формальных логических языках и их обработка. Если допустим, что у нас имеется совокупность фактов, полученных из заданного ЕЯТ и представленных в виде формул того или иного (а часто и того и другого) формального логического языка, то тут возникают вопросы алгоритмического характера. Эти вопросы связаны с проблемами выполнимости, вывода следствий, эффективности проверки выполнимости и выводимости следствий в этих языках. Рассмотрим кратко синтаксис и семантику аристотелевских логик и некоторых модальных логик вместе с

их аксиоматическими системами, а алгоритмические свойства этих логик будут рассмотрены в следующих разделах.

2.3.1. Синтаксис и семантика исчисления высказываний

Исчисление высказываний – один из самых простых формальных логических языков, однако он широко используется в различных предметных областях, в том числе и по тематике исследований данной работы. Исчисление высказываний изучает ЕЯ-предложения, которые могут быть либо истинными, либо ложными [59, 60].

Язык исчисления высказываний состоит из алфавита, который может быть конечным или бесконечным, но не более чем счётным. Символами алфавита выступают большие либо строчные буквы (с индексами или без) латинского алфавита, обозначающие атомарные (неделимые) высказывания, которые квалифицируются либо как истинные либо как ложные, но не те и другие одновременно. Этот алфавит в лингвистике называют *Пропозициональным логическим словарем*. Он пополняется специальными символами, которые называются логическими или пропозициональными связками – отрицание (\neg), конъюнкция (\wedge), дизъюнкция (\vee), импликация (\rightarrow) и логическая эквивалентность (\leftrightarrow). Хорошо известно, что достаточно одной пары логических связок (\neg, \wedge), (\neg, \vee), (\neg, \rightarrow).

Словарь исчисления высказываний даёт возможность строить сложные или составные высказывания из исходных (атомарных), соединяя последние связками. *Правила построения* сложных высказываний называются синтаксисом исчисления высказываний и описывают те выражения, которые являются объектами языка. Такие высказывания называют *правильно построенными формулами* (ППФ). Аналогия с естественными языками очевидна: фраза – это составное высказывание, построенное по определенным правилам.

Синтаксис языка высказываний определяется индуктивно с помощью таких правил [60]:

- Базис: все атомарные высказывания являются ППФ.
- Индукционный шаг: если X и Y – ППФ, то $\neg X$, $(X \wedge Y)$, $(X \vee Y)$, $(X \rightarrow Y)$ и $(X \leftrightarrow Y)$ – ППФ.
- Ограничение: ППФ являются те и только те формулы, которые построены с помощью вышеописанных индукционных правил.

Объектами изучения естественных и формальных языков являются, в частности, синтаксис (который позволяет распознавать фразы среди наборов слов) и семантика (которая придаёт определённое значение фразам). В равной мере это относится к исчислению высказываний.

Как отмечалось, что высказывание либо истинно (И), либо ложно (Л). Для определения значения высказывания необходимо высказывания интерпретировать. Интерпретировать формулу – значит приписать ей одно из двух значений истинности И или Л (часто используют обозначения 1 для И, и 0 для Л).

Семантика (то есть набор правил интерпретации формул), как правило, является композиционной: значение формулы должно быть функцией значений её составляющих. Точнее, значение истинности формулы зависит только от структуры этой формулы и от значений истинности составляющих её атомарных формул.

Таким образом, связки исчисления высказываний представляют функции истинности (или булевы, булевские, функции алгебры логики): например, значение

истинности формулы $(X \wedge Y)$ будет известно, если известны значения истинности X и Y .

Интерпретация – это отображение $i: A \rightarrow \{И, Л\}$, сопоставляющее каждому атомарному высказыванию $p \in A$ некоторое значение истинности. Интерпретация i , заданная на множестве атомарных высказываний, продолжается на множество ППФ (высказываний) посредством таблиц истинности. Соответствующее продолжение I тоже называется интерпретацией. Интерпретация, при которой истинностное значение формулы есть И, называется моделью этой формулы.

Литера (или литерал) – это элементарное высказывание или его отрицание. Литеры p и $\neg p$ называются противоположными или контрарными. Интерпретация определяет разбиение множества L литер на два подмножества $L_{и}$ и $L_{л}$, каждое из которых содержит по одному элементу из каждой пары противоположных литер.

Исчисление высказываний и естественный язык

В исчислении высказываний, как указывалось выше, имеется пять связок. С одной стороны, ясно, что для этих связок имеются «эквиваленты» в естественном языке. Ниже кратко очертим границы сходства между логическими связками и связками естественного языка. С другой стороны, естественный язык был бы сильно обеднён сокращением числа его связок до пяти. Известно, что, напротив, исчисление высказываний не станет богаче, если ввести дополнительные связки. В естественном языке связки «не», «и», «или», «если ..., то» на первый взгляд вполне однозначно описываются приведенными выше функциями истинности.

Например, фразы

Хорошая погода или идет дождь и *Идет дождь или хорошая погода*

кажутся синонимами. Однако иначе обстоит дело с фразами

Ему стало страшно, и он спросил незнакомца и *Он спросил незнакомца, и ему стало страшно,*

так как союз «и» подчеркивает здесь определённый временной и причинный нюанс. Наконец, многочисленные связки естественного языка ни в коей мере не соответствуют функциям истинности. Связка «потому что» служит хорошим тому примером. Чтобы убедиться в этом, предположим, что *идет дождь* (высказывание p) и *что земля мокрая* (высказывание q). Легко принимается как истинное высказывание

q потому что p ,

но нельзя считать истинным высказывание

p потому что q .

Логическая дизъюнкция является неразделительной (или соединительной): она истинна, если истинным является хотя бы один из двух её операндов. В естественном языке союз «или» иногда является исключаящим (разделительным). В предложении *Целое число чётно или нечётно* одна ветвь альтернативы истинна, а другая ложна.

Импликация – очень важная связка; она отражает структуру рассуждений, в частности математических. Первый её операнд называется *посылкой* (или антецедентом), а второй – *заключением* (или консеквентом). Ясно, что если посылка истинна, то импликация принимает значение истинности заключения. Однако может вызвать удивление, что импликация истинна и тогда, когда её посылка ложна. Тем не менее, легко убедиться, что импликация является единственной связкой, удовлетворяющей следующим требованиям:

- если первый операнд истинный, то значение истинности совпадает со значением второго операнда;

- значение истинности зависит от двух операндов;
- связка некоммутативная.

В той мере, в какой эти три критерия удачно очерчивают естественную импликацию, принятое определение служит наилучшим возможным приближением. Чтобы подчеркнуть этот нюанс, смоделированную связкой \rightarrow импликацию иногда называют материальной импликацией. Материальная импликация – основная связка в математических рассуждениях. Она соединяет условие H и утверждение T в теореме. Однако было бы нелогично писать $H \rightarrow T$, так как H и T – не логические формулы, а какие-то произвольные высказывания: неформальные или принадлежащие некоторому нелогическому формализму. В таких случаях лучше писать $H \Rightarrow T$. Аналогично, в таком контексте лучше писать $C_1 \Leftrightarrow C_2$, а не $C_1 \leftrightarrow C_2$.

Исчисление высказываний выражает чисто функционально-истинностные связи между высказываниями. Аксиоматическая теория формального логического языка включает определение четырёх объектов: алфавит языка, синтаксис языка, аксиомы языка и отношения, которые называются правилами вывода.

Аксиоматическая система для языка высказываний имеет вид [59]:

- A – алфавит атомарных формул является алфавитом языка высказываний, а логическими связками являются отрицание (\neg) и импликация (\rightarrow).
- Формулами языка высказываний являются ППФ.
- Схемами аксиом языка высказываний являются формулы
 - $A \rightarrow (B \rightarrow A)$;
 - $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$;
 - $(\neg B \rightarrow \neg A) \rightarrow (\neg B \rightarrow A) \rightarrow B$.

г) Единственным правилом вывода является правило *modus ponens* (MP): из A и $A \rightarrow B$ следует B .

Правило резолюций. В практических применениях формального логического языка наиболее важным является задача выполнимости формул. Не существует общего, по настоящему эффективного критерия для проверки выполнимости конъюнктивных нормальных форм. Тем не менее, известен достаточно удобный метод для выявления невыполнимости множества дизъюнктов. Действительно, множество дизъюнктов невыполнимо тогда и только тогда, когда пустой дизъюнкт L является логическим следствием из него. Таким образом, невыполнимость множества S можно проверить, порождая логические следствия из S до тех пор, пока не получим пустой дизъюнкт.

Для порождения логических следствий используется простая схема рассуждений. Пусть A , B и X – формулы. Предположим, что две формулы $(A \vee X)$ и $(B \vee \neg X)$ – истинны. Если X тоже истинна, то отсюда можно заключить, что B истинна. Наоборот, если X ложна, то можно заключить, что A истинна. В обоих случаях $(A \vee B)$ истинна. При этом получается правило, которое нетрудно доказать, исходя из аксиом и правила вывода в языке высказываний,

$$\{A \vee X, B \vee \neg X\} \models A \vee B,$$

которое можно записать также в виде

$$\{\neg X \rightarrow A, X \rightarrow B\} \models A \vee B.$$

В том частном случае, когда X – высказывание, а A и B – дизъюнкты, это правило называется правилом резолюций. Общезначимость правила резолюций выражается следующей леммой, которую нетрудно доказать, исходя из приведенной аксиоматической системы.

Лемма 2.3. Пусть s_1 и s_2 – дизъюнкты множества формул S , l – литера. Если $l \in s_1$ и $\neg l \in s_2$, то дизъюнкт $r = (s_1 \setminus \{l\}) \vee (s_2 \setminus \{\neg l\})$ является логическим следствием формулы S .

При этом дизъюнкт r называется *резольвентой* дизъюнктов s_1 и s_2 .

2.4. Синтаксис и семантика исчисления предикатов первого порядка

Основными символами языка логики предикатов являются *переменные, индивидуальные константы, предикатные константы, связки* ($\neg, \wedge, \vee, \rightarrow, \leftrightarrow$), *кванторы общности* \forall (для всех) и *существования* \exists (существует). В языке предикатов содержится язык высказываний, так как высказывание – не что иное, как предикатная константа без аргументов или, точнее, предикатная константа с нулевым числом мест. Удобно заменить понятие индивидуальной константы более общим понятием *функциональной константы*. Функциональная константа с определенным числом мест – в точности то же, что предикатная константа. Индивидуальная константа – просто нульместная функциональная константа.

Ввиду того, что исчисление предикатов позволяет оперировать с отдельными объектами (переменными и константами), иногда полезно вводить предикат равенства между этими объектами. Для обозначения этого предиката используется классический символ $=$. Переменные, предикатные и функциональные константы обозначаются буквами или словами, иногда снабженными верхними и/или нижними индексами.

Перечисленный выше словарь позволяет определить *термы, атомы, формулы* и *кванторы*. Ниже приводится индуктивное определение термина и формулы [60].

- *Термом* является всякая переменная и всякая константа.
- Если f – функциональный n -местный символ и t_1, \dots, t_n – термы, то выражение $f(t_1, \dots, t_n)$ – терм.
- Если P – предикатный m -местный символ и t_1, \dots, t_m – термы, то выражение $P(t_1, \dots, t_m)$ – элементарная формула.
- *Атом* – это элементарная формула или некоторое равенство, т. е. выражение типа $(s=t)$, где s и t – термы.
- Если A и B *формулы*, то $\neg A$, $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$ и $(A \leftrightarrow B)$ – формулы и если A – формула и x – переменная, то $\forall x A$ и $\exists x A$ – формулы.

Формулы, построенные по этим правилам, называются *правильно построенными формулами (ППФ)*.

В выражении $\forall x A(x)$ формула A называется областью действия квантора $\forall x$. Если A не содержит вхождений переменной x , то в этом случае считается, что $\forall x A$ и A имеют одинаковый смысл. Вхождение переменной x в данную формулу называется *связанным*, если x является переменной входящего в эту формулу квантора $\forall x$ или находится в области действия входящего в эту формулу квантора $\forall x$. В противном случае вхождение переменной x в эту формулу называется *свободным*. Переменная называется *свободной (связанной)* в данной формуле, если существуют свободные (связанные) её вхождения в эту формулу. Терм t называется *свободным* для переменной x в формуле A , если никакое свободное вхождение x в A не лежит в области действия никакого квантора $\forall x$, где x – переменная, входящая в терм t .

Семантика исчисления предикатов. Формулы исчисления предикатов, как и формулы исчисления высказываний, могут быть интерпретированы, т. е. могут получить значение истинности. Однако формулы исчисления предикатов состоят не только из подформул, но также и из термов. Следовательно, необходимо интерпретировать также термы. Терм интуитивно означает некоторое функциональное выражение, построенное из предметных переменных, констант и операций, определённых в этом языке. Таким образом, интерпретация должна специфицировать множество объектов, называемое областью интерпретации.

Точнее, интерпретация I – это четвёрка (D, I_c, I_v, I_π) со следующими свойствами:

- D – непустое множество, называемое областью интерпретации.
- I_c – функция, которая сопоставляет каждому функциональному m -местному символу некоторую m -арную операцию $I_c(f)$ на D^m .
- I_v – функция, сопоставляющая каждой переменной некоторый элемент из D .
- I_π – функция, сопоставляющая каждому n -арному предикату P n -арное отношение R_p на D .

Отметим, что в логике предикатов вводится операция унификации, являющаяся основным механизмом при выполнении инструкций в логическом программировании. С её помощью метод резолюций, рассмотренный в 2.3, распространяется и на логику предикатов.

Аксиоматическая система для языка предикатов первого порядка имеет вид [59].

- а) A – алфавит атомарных формул является алфавитом языка предикатов, а логическими связками являются отрицание (\neg) и импликация (\rightarrow).
- б) Формулами языка предикатов являются правильно построенные формулы.
- в) Схематическими аксиомами языка предикатов первого порядка являются формулы
 - A1) $A \rightarrow (B \rightarrow A)$;
 - A2) $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$;
 - A3) $(\neg B \rightarrow \neg A) \rightarrow (\neg B \rightarrow A) \rightarrow B$;
 - П4) $\forall x A(x) \rightarrow A(t)$, где $A(x)$ – формула языка предикатов, а t – терм, свободный для переменной x в формуле $A(x)$;
 - П5) $\forall x (A \rightarrow B) \rightarrow (A \rightarrow \forall x B)$, если A не содержит свободных вхождений переменной x .
- г) Правилами вывода являются правило *modus ponens* (MP): из A и $A \rightarrow B$ следует B и правило обобщения (Gen): из A следует $\forall x A$.

Из приведенной аксиоматической системы вытекает, что язык высказываний является подмножеством языка предикатов первого порядка, поскольку аксиоматика языка предикатов включает все схемы аксиом языка высказываний и правило вывода этого языка. Это значит, что все утверждения языка высказываний автоматически являются утверждениями языка предикатов первого порядка.

2.5. Выражение суждений на языке логики предикатов

Известен стандартный способ перевода суждений об отношениях на язык логики предикатов [61]. Он представляет собой следующую последовательность шагов:

1) заменить единичные и общие имена индивидуальными константами и предикатами соответственно;

2) заменить кванторные слова кванторами и выписать кванторы с относящимися к ним переменными в порядке вхождения кванторных слов в предложение, выражающее суждение;

3) выписать формулу, заменяющую первый (по смыслу) предикат, поставив перед ней левую скобку; если индивидуальная переменная формулы, заменяющей первый предикат, связана квантором общности, то поставить после неё знак импликации, после знака импликации или знака конъюнкции поставить левую скобку;

4) если индивидуальная переменная формулы, заменяющей второй (по смыслу) предикат, связана квантором общности, то выписать её и поставить после неё знак импликации, если же она связана квантором существования, то выписать её и поставить после неё знак конъюнкции; после знака импликации или знака конъюнкции поставить левую скобку (если переводится суждение о более чем двухместном отношении) и т. д.;

5) выписать формулу, заменяющую последний предикат;

6) после формулы, заменяющей последний предикат, поставить необходимое число правых скобок (если выявляется логическая форма отрицательного суждения, то перед последним предикатом поставить знак отрицания).

Рассмотрим примеры перевода предложений ЕЯ в язык предикатов, взятые из [59].

1. Пусть заданы следующие факты: некоторые пациенты любят своих врачей; ни один пациент не любит знахарей. Следует ли из этих фактов, что никакой врач не может быть знахарем.

Выясним, является ли данное следствие истинным, если первые два высказывания примем за достоверные факты. Пусть

$P(x)$ значит, что x является пациентом;

$D(x)$ значит, что x является врачом;

$Q(x)$ значит, что x является знахарем;

$L(x, y)$ значит, что x любит y .

Тогда принятые факты и следствие можно записать с помощью следующих формул:

$$F : \exists x(P(x) \wedge \forall y(D(y) \rightarrow L(x, y))),$$

$$G : \forall x(P(x) \rightarrow \forall y(Q(y) \rightarrow \neg L(x, y))),$$

$$H : \forall x(D(x) \rightarrow \neg Q(x)).$$

2. Каждый, кто хранит деньги в банке, получает проценты. Если нет процентов, то никто не хранит деньги в банке.

Пусть

$S(x, y)$ значит, что банк x хранит деньги у;

$M(x)$ значит, что x имеет деньги;

$I(x)$ значит, что x платит проценты;

$E(x, y)$ значит, что x получает проценты от y .

Тогда гипотезу запишем следующим образом:

$$F : \forall x(\exists y(S(x, y) \wedge M(y)) \rightarrow \exists y(I(y) \wedge E(x, y))),$$

а следствие: $G : \neg(\exists x)I(x) \rightarrow \forall x\forall y(S(x, y) \rightarrow \neg M(y))$.

2.6. Пропозициональная модальная логика

Пропозициональная модальная логика (ПМЛ) и линейная темпоральная логика строятся на основе логики высказываний, которая пополняется модальными операторами. Рассмотрим синтаксис и семантику, а также аксиоматические системы этих логик.

Синтаксис. Формулы ПМЛ строятся при помощи тех же самих правил, которые применялись при построении логики высказываний. Но ПМЛ дополнительно использует два новых символа \square и \diamond , которые называются модальными операторами всеобщности и существования, соответственно. Оба оператора являются унарными и действуют на множестве формул. Формулы ПМЛ определяются индуктивно.

Определение 2.9. (синтаксис ПМЛ).

- 1) Все атомарные высказывания являются формулами ПМЛ.
- 2) Если A и B – формулы ПМЛ, то $\neg A$, $A \wedge B$, $A \vee B$, $A \rightarrow B$, $A \leftrightarrow B$, $\square A$, $\diamond A$ тоже формулы ПМЛ.
- 3) Остальные слова, т. е. слова, которые построены не по правилам 1)–2), не являются формулами ПМЛ.

Операторы \square и \diamond , связаны между собой с помощью отношения двойственности:

$$\diamond A = \neg \square \neg A.$$

Это соотношение часто трактуется как определение оператора \diamond при помощи \square . Следовательно, каждую формулу ПМЛ можно записать, пользуясь только оператором \square .

Операторы \square и \diamond могут читаться разными способами. Некоторые из способов чтения этих операторов показаны ниже:

Оператор $\square A$:

Необходимо, чтобы A была истинна.

Всегда A будет истинна.

Требуется, чтобы A была истинна.

Полагаем, что A истинна.

Известно, что A истинна.

Оператор $\diamond A$:

Возможно, что истинна A .

Иногда A будет истинна.

Разрешается, чтобы была истинна A .

Не предполагается противное к A .

Неизвестно противное к A .

Семантика. Для определения семантики формул ПМЛ, необходимо рассмотреть понятие *структуры*.

Определение 2.10. Структурой называется пара $F = (W, R)$, где W – некоторое непустое множество, а $R \subseteq W \times W$ – бинарное отношение на множестве W . Элементы множества W называются *точками*.

Определение 2.11. Пусть P – множество атомарных формул ПМЛ. P -моделью на структуре $F = (W, R, f)$ называется тройка $M = (W, R, f)$, где $f: P \rightarrow$

$V(W)$ – функция из множества формул P в булеан множества W . Для $p \in P$ множество $f(p)$ можно неформально интерпретировать как множество точек из W , в которых высказывание p является истинным.

Если P является фиксированным множеством формул, то символ P в определении модели не указывается, и в этом случае говорят просто о модели.

Пусть $w \in W$ и A есть формула ПМЛ. Выражение $M \models_w A$ означает, что формула A является истинной в точке w модели $M = (W, R, f)$.

Определение 2.12. ПМЛ-формула A называется *выполнимой* тогда и только тогда, когда существует модель M и некоторая точка w этой модели такая, что $M \models_w A$.

Определение 2.13. (семантика формул ПМЛ). Семантика формул ПМЛ определяется на модели $M = (W, R, f)$ следующим образом:

- 1) $M \not\models_w 0$,
- 2) $M \models_w A$, если $A \in P$ и $w \in f(A)$,
- 3) $M \models_w A \rightarrow B$, если из $M \models_w A$ следует $M \models_w B$,
- 4) $M \models_w \Box A$, если для каждого $t \in W$ такого, что wRt , следует $M \models_t A$.

Из последнего правила следует, что формула $\Box A$ должна быть истинной во всех точках t модели M , для которых отношение wRt истинно. Правила 1)–4) называются базовыми. С их помощью можно найти значения формул: 1 , $\neg A$, $A \vee B$, $A \wedge B$, $A \leftrightarrow B$ и $\Diamond A$. Например, постоянную 1 можно записать как $0 \rightarrow A$, а формулу $\neg A$ как $A \rightarrow 0$. Из правила $\neg A = A \rightarrow 0$ следует такое семантическое правило:

- 1) $M \models_w \neg A$, если $M \not\models_w A$,
- 2) $M \models_w \Diamond A$, если $M \models_t A$ по крайней мере для одного $t \in W$ такого, что wRt ,
- 3) $M \models_w A \vee B$, если $M \models_w A$ или $M \models_w B$,
- 4) $M \models_w A \wedge B$, если $M \models_w A$ и $M \models_w B$.

Определение 2.14 Формула A называется *истинной* в модели M , если она истинная в каждой точке этой модели, т. е. $(\forall w \in W) M \models_w A$. Записывается это как $M \models A$.

Формула A называется *истинной* в структуре $F = (W, R)$, если она истинная в каждой модели $M = (W, R, f)$, т. е. $(\forall M = (W, R, f)) M \models A$.

Формула A называется *модальной тавтологией*, если она истинная во всех структурах (W, R) . Записывается это так: $\models A$.

Описанная семантика формул ПМЛ называется *реляционной семантикой* или *семантикой Крипке*, по имени автора, который впервые ввёл её в рассмотрение. Продемонстрируем на примере, как показать истинность или ложность некоторой формулы ПМЛ в модели Крипке.

Аксиоматическая система для ПМЛ. Как уже упоминалось, ПМЛ является расширением классической логики высказываний. Формальная аксиоматическая система для ПМЛ строится по тем самым правилам, что и для логики высказываний.

Формальная аксиоматическая теория для ПМЛ определяется следующим образом:

1. Логическими символами ПМЛ являются \neg , \rightarrow , $(,)$, а также литеры A_i алфавита Al . Символы \neg , \rightarrow называются *пропозициональными связками*, а символы A_i – *пропозициональными переменными* или *пропозициональными литерами*.

2. Все литеры алфавита Al являются формулами ПМЛ. Если A и B являются формулами ПМЛ, то $\neg A$, $A \rightarrow B$, $\Box A$ также являются формулами ПМЛ.

3. Для произвольных формул A, B, C формулы ПМЛ:

- A1) $A \rightarrow (B \rightarrow A)$,
- A2) $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$,

A3) $(\neg B \rightarrow \neg A) \rightarrow ((\neg B \rightarrow A) \rightarrow B)$,

K) $\Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$

являются схемами аксиом.

Пример. Пусть дана формула $\Box A \vee \Box B$ и модель Крипке (рис. 2.1). Для этой модели имеем $W = \{s_0, s_1, s_2, s_3\}$,

$R = \{(s_0, s_0), (s_0, s_1), (s_0, s_2), (s_0, s_3), (s_1, s_0), (s_1, s_2), (s_2, s_1), (s_3, s_2), (s_3, s_1)\}$.

Обозначения состояний показывают, что в

- 1) s_0 истинные формулы A, B ;
- 2) s_1 истинные формулы $A, \neg B$;
- 3) s_2 истинные формулы $\neg A, B$;
- 4) s_3 истинные формулы $\neg A, \neg B$.

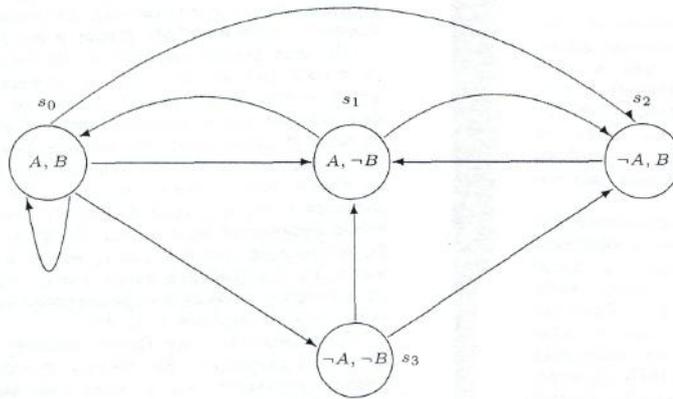


Рис. 2.1. Модель Крипке

По этой интерпретации формула $\Box A \vee \Box B$ является:

- а) ложная в состоянии s_0 , потому что состояния s_0, s_1, s_2, s_3 достигаемы из состояния s_0 и в состоянии s_3 формулы A и B ложные;
- б) истинная в состоянии s_1 , потому что состояния s_0 и s_2 достигаемы из состояния s_1 и формула B истинная в состояниях s_0 и s_2 ;
- в) истинная в состоянии s_2 , потому что состояние s_1 достигаемо из состояния s_2 и в состоянии s_1 истинная формула A ;
- г) истинная в состоянии s_3 , потому что состояния s_1, s_2 достигаемы из состояния s_3 и в состоянии s_1 истинная формула A , а в состоянии s_2 истинная формула B .

4. Правилами вывода являются:

- правило *modus ponens* (MP): из A и $A \rightarrow B$ следует B ,
- правило *модальной необходимости* (RN): из A следует $\Box A$.

Остальные логические связки вводятся с помощью известных тождеств:

$$A \wedge B \leftrightarrow \neg(A \rightarrow \neg B), A \vee B \leftrightarrow \neg A \rightarrow B,$$

$$A \leftrightarrow B \leftrightarrow (A \rightarrow B) \wedge (B \rightarrow A).$$

Отметим, что как и в случае исчисления высказываний, бесконечное множество аксиом ПМЛ задаётся с помощью четырёх схем аксиом A1), A2), A3), K), каждая из которых порождает бесконечное множество аксиом. Конечность числа схем аксиом даёт возможность для произвольной формулы легко проверить, есть ли она аксиомой, итак, ПМЛ является эффективно аксиоматизированной теорией.

Определение 2.15. Доказательством в ПМЛ называется произвольная конечная последовательность формул A_1, A_2, \dots, A_n этой теории, такая, что каждая формула

A ; является или аксиомой, или непосредственным следствием предыдущих формул этой последовательности по одному из правил вывода.

Формула A называется *теоремой* ПМЛ, если существует доказательство в ПМЛ для A такое, что последним элементом в этом доказательстве есть формула A . Такое доказательство называется доказательством формулы A в теории ПМЛ.

Аксиоматическая система ПМЛ, которая включает аксиому K , называется *нормальной модальной логикой*.

Линейная пропозициональная темпоральная логика (ЛПТЛ)

В нормальной логике на отношение R не накладывалось никаких ограничений. В приложениях отношение R часто является отношением частичного или линейного порядка. Если отношение R является отношением линейного порядка, то получим логику, которая называется *линейной пропозициональной темпоральной логикой*. Отношение линейного порядка R имеет такое свойство: для каждого элемента $w \in W$ существует единственный элемент $w' \in W$ такой, что $(w, w') \in R$. Это свойство даёт возможность ввести ещё один модальный оператор, который обозначается как \circ и называется X -оператором или *next*-оператором. Неформально $\circ A$ значит, что формула A истинная в следующей точке модели, которая следует непосредственно за данной точкой согласно отношению R .

Рассмотрим синтаксис и семантику этой логики, а также аксиоматическую систему для неё.

Синтаксис. Пусть $Al = \{A, B, C_1, \dots\}$ – алфавит атомарных формул, а P – множество пропозициональных формул над этим алфавитом.

Определение 2.16 (синтаксис ЛПТЛ).

- 1) Все атомарные формулы являются формулами ЛПТЛ.
- 2) Если A и B являются формулами ЛПТЛ, то $\neg A, A \rightarrow B, \Box A, \circ A$ также являются формулами ЛПТЛ.

Остальные логические связки вводятся с помощью таких тождеств:

$$A \vee B = \neg A \rightarrow B, A \wedge B = \neg (A \rightarrow \neg B), \\ A \leftrightarrow B = (A \rightarrow B) \wedge (B \rightarrow A), \Diamond A = \neg \Box \neg A.$$

Семантика. Модель Крипке для ЛПТЛ имеет вид $M = (T, R, f, t_0)$, где множество точек обозначают T (а не W) и называют множеством моментов времени, $t_0 \in T$ – начальный момент времени, $R \subseteq T \times T$ – бинарное отношение линейного порядка на T , а для каждого $p \in P$ $f(p)$ является подмножеством множества T тех моментов времени, в которых формула p истинная.

Семантика формул ЛПТЛ определяется на модели M следующим образом:

- 1) $M \models p$ тогда и только тогда, когда $t_0 \in f(p)$, где $p \in P$,
- 2) $M \not\models 0$,
- 3) $M \models (\varphi \rightarrow \psi)$ тогда и только тогда, когда из $M \models \varphi$ следует $M \models \psi$,
- 4) $M \models \Box \varphi$ тогда и только тогда, когда $\forall t_1 \in T$ такого, что $(t_0, t_1) \in R$ имеет место $t_1 \models \varphi$,
- 5) $M \models \circ \varphi$ тогда и только тогда, когда $\exists t_1 \in T$ такое, что $(t_0, t_1) \in R$ (т. е. t_1 непосредственно следует за t_0) и $t_1 \models \varphi$.

Аксиоматическая система для ЛПТЛ. Для произвольных формул A, B, C приведенные ниже формулы являются схемами аксиом.

- A1) $A \rightarrow (B \rightarrow A)$,

A2) $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$,

A3) $(\neg B \rightarrow \neg A) \rightarrow ((\neg B \rightarrow A) \rightarrow B)$,

K) $\Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$,

T1) $\Diamond A \leftrightarrow \neg \Box \neg A$,

T2) $\circ(A \rightarrow B) \rightarrow (\circ A \rightarrow \circ B)$,

T3) $\Box A \rightarrow (A \wedge \circ A \wedge \circ \Box A)$,

T4) $\Box(A \rightarrow \circ A) \rightarrow (A \rightarrow \Box A)$,

T5) $\circ A \leftrightarrow \neg \circ \neg A$.

Правилами вывода являются:

– *Modus ponens*: из A и $A \rightarrow B$ следует B ,

– *Правило модальной необходимости (RN)*: из A следует $\Box A$.

Аксиома T4) в данной системе является аксиомой математической индукции.

Шагом индуктивного доказательства есть $A \rightarrow \circ A$, т. е. допускаем, что A истинная "сегодня" и доказываем, что A будет истинной "завтра". Если индуктивный шаг всегда имеет место $\Box(A \rightarrow \circ A)$, то отсюда следует $A \rightarrow \Box A$.

В этой логике можно выразить семантику достаточно точно, например, такие высказывания:

«Я люблю Родину и буду любить её всегда» – $\Box A$, где A означает «я люблю Родину»;

«Я когда-нибудь стану знаменитым» – $\Diamond A$, где A означает «я стану знаменитым».

«Этот повар, как только он будет готовить, всегда будет готовить острые блюда» –

$\Box \Diamond (\text{Этот_Повар_Готовит} \rightarrow \text{Готовит_Блюдо_Острое})$.

Дальнейшая работа с модальными формулами состоит в их трансляции в FOL с последующей обработкой, о которой говорилось выше.

3. ФОРМЫ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

3.1. Логический аспект представления знаний

Проблема извлечения знаний из естественно-языковых объектов, их представление и обработка относятся к области искусственного интеллекта. Этой чрезвычайно важной проблемой занимались многие выдающиеся учёные (Аристотель, Лейбниц, Тарский и др.). В общей проблеме анализа и понимания ЕЯО, во всём многообразии работ [62–66], методов, методологий и терминологии явно просматриваются следующие задачи:

- синтактико-семантический анализ ЕЯО;
- анализ ЕЯО с целью выбора подходящего формально-логического представления в некоторой подсистеме логики и представления в памяти компьютера и дальнейшей эффективной обработки;
- разработка методов извлечения знаний из ЕЯО.

Рассмотрим эти задачи более детально.

Проблема анализа ЕЯО естественным образом связана с *операционной* и *денотационной семантикой* языка. В системах искусственного интеллекта и, в частности, в системах анализа ЕЯО преобладает операционная семантика, так как очевидно преобладание интереса к средствам представления знаний (**как?**), а не к их содержанию (**что?**), которое подлежит формализации [67].

Системы (языки) представления знаний определяют конструкции, поддерживаемые компьютером и обеспечивающие эффективность организации процедур, и как следствие – максимальную унификацию языковых средств.

На информационно-логическом уровне преследуется совершенно иная цель – обеспечение отражения содержания объектов реального мира в языковых конструкциях, согласующихся с восприятием, свойственным человеку. Для пользователя такой системы (схемы) компьютерное представление неудобно, поскольку он общается (или, по крайней мере, должен) с системой на естественном языке.

Однако, для системного аналитика, использующего язык формализации для построения модели предметной области и создающего для этого ориентированные на эту ПдО словари понятийных единиц и на их основе – интеллектуальные системы, наполненные конкретными знаниями, компьютерное представление играет ключевую роль. Аналитику приходится иметь дело с конкретным множеством терминов и ЕЯ-конструкций, именно им давать толкование в терминах языка формализации, устанавливая между ними информационно значимые связи. Поэтому, в случае расхождения языка формализации со строем мышления специалиста, реализация системы обработки данных в конкретном смысловом материале может стать слишком сложной или вообще неразрешимой проблемой.

Наиболее приемлемыми на сегодняшний день являются языки логического типа, в которых понятие "связь по смыслу" формализуется с максимальной полнотой. Одним из главных вопросов при этом является вопрос об уровне абстрактности зафиксированных категорий логического языка. Если нас интересует лишь

общая схема логического следования, то достаточно рассматривать лишь исчисление предикатов первого порядка с правилом резолюций в качестве правила вывода, не конкретизируя ничего, кроме списка аксиом и логических констант. Однако при более богатом содержании ПдО должен быть решён вопрос о типах термовых констант и виде нелогических аксиом (собственных аксиом), используемых при формализации знаний об эмпирических объектах.

Изложенное свидетельствует об актуальности проблемы *разработки языка логического типа для представления знаний, содержащихся в тексте*. Очевидно, что этот язык должен быть расширяемым в сторону способности более полного описания объектов, способности делать предсказания (гипотезы), учитывающие контекст [67].

3.2. Сравнительный анализ формальных логических языков

Общеизвестно, насколько важным звеном в технологической цепи компьютерной обработки ЕЯТ является формально-логическое представление его структуры и правил вывода. Как правило, такое представление использует или определённые подмножества FOL или её вариации. В настоящее время [26] представляются новые виды FOL, с отличным от классической логики набором логических операторов и кванторов.

Если логика нам говорит, что что-либо существует и предоставляет логические операторы и правила манипулирования сущностями, то онтология, во-первых, предоставляет сам словарь этих сущностей, а во-вторых, формализованное представление всех видов сущностей – абстрактных и материальных, которые представляют мир. Высказывание на естественном языке, будучи переведенным в формально-логическое представление, уже может быть "понятным" компьютеру и обработано в соответствии с конкретными задачами пользователя. Поэтому представляется целесообразным проследить эволюцию развития логико-онтологических формальных теорий и их возможностей, в частности, относительно обработки естественно-языковых текстов. Учитывая статус классической двузначной логики [26], ниже приведен краткий анализ подмножеств лишь классической логики.

Силлогистика. "Силлогизмы – умозаключение, в силу которого, признав истинность посылки силлогизма, нельзя не согласиться с истинностью вывода, который вытекает из посылок" [21]. Например, "*Микропроцессоры фирмы Intel составляют ядро персональных компьютеров IBM PC* (главная посылка); *Микропроцессоры серии Itanium 2 производятся фирмой Intel* (вторичная посылка), тогда *Микропроцессоры серии Itanium 2 составляют ядро персональных компьютеров IBM PC* (вывод)". Вывод сделан на основе сходимости средних термов – в данном примере словосочетаний "*Микропроцессоры фирмы Intel*" и "*Микропроцессоры серии Itanium 2*". Понятие силлогизма было введено Аристотелем. Также у него впервые появляются утверждения, которые оперируют не с конкретными объектами, а с их мнемониками (используются буквы латинского алфавита) – таким способом он строит универсальные шаблоны силлогизмов. Аристотель выделил четыре типа силлогизмов

А – «все X есть Y »; Е – «все X не есть Y »;

I – «некоторые X есть Y »; О – «некоторые X не есть Y ».

Средневековый философ Адам, анализируя противоречивость силлогизмов, делит утверждения на такие, которые могут утверждать о себе, и те, которые не могут (например, "*Это высказывание ложное*"). Испанский философ Луллий

обобщает силлогизмы, приводя их к правилам вывода из общего. Однако, силлогизмы не пригодны для представления произвольных научных знаний, так как символическая связь не может полностью выразить смысловой связи, на что обратил внимание немецкий философ-математик Юнг [68]. Силлогизмы также не дают возможности фиксировать отдельные утверждения, например "*Микропроцессоры фирмы Intel составляют ядро персональных компьютеров IBM PC*", а представляют собой знания, которые выводятся. Хотя в силлогистических выражениях используются слова "и", "если... то...", "каждый", "некоторый", "обязательно" и т. п., тем не менее, они не выделяются отдельно, поэтому о логических операторах или кванторах в силлогистике речь не идёт. Формально силлогизмы представляются линейными *текстовыми шаблонами* с неопределённым средним термом. Автоматический вывод в системе силлогизмов может проводиться путём сравнения структуры высказывания с шаблонами, и поиска средних термов в главной и вторичной посылке, и во вторичной посылке и выводе за одну итерацию.

Логика высказываний и алгебра логики. Логика высказываний представляет собой полную (определение [69], доказательство [70]) систему итеративного построения и разбора логических высказываний, в которой атомарной является конструкция, для составных частей которой невозможно установить истинность. Разделение языка на отдельные высказывания провёл французский философ Абельяр – он впервые в логике предложил "начинать изучение с простых, а потом со сложных высказываний" [70], в связи с чем выделил связку (сорула) в утверждениях.

Центральным понятием в логике высказываний и во всей классической логике вообще есть понятие истинности. Разделение высказываний на истинные и ложные впервые ввёл Луллий. Современное применение логики, в частности в инженерии знаний, непременно предусматривает автоматическую обработку, которая может проводиться математическими вычислениями. Так, ещё английский философ-материалист Гоббс впервые высказал мысль, что "...мышление есть ничто другое, как вычисление: при этом прибавлять и отнимать можно и величины, и тела, движения, времена, качества..." [68], а позднее Юнг сделал ударение именно на *математической форме вычисления выражений*. Лейбниц на основе этих течений в науке о логике и своего тезиса, "что все понятия могут быть сведены к элементарным неделимым понятиям" [68], а сложные понятия образуются с помощью операции перемножения (логического) и пересечения объёмов понятий и что набор простых понятий должен быть непротиворечивым, разрабатывает систему математического представления утверждений. За простые утверждения он взял простые числа, а за основные операции – арифметические умножения и деления, имея машину, которая выполняла эти операции. Такой принцип можно отнести как к построению иерархий категорий, так и к логическому представлению естественно-языковых высказываний. Так, если понятие "*Микропроцессоры серии Itanium 2*" обозначить как G , "*Микропроцессоры фирмы Intel*" – как B , то выражение "*Некоторый микропроцессор серии Itanium 2 – микропроцессор фирмы Intel*" будет иметь вид $G = B \cdot x$, а при обозначении, где T – "*Микропроцессор серии SA-111*" и F – "*составляют ядро персональных компьютеров IBM PC*", посылка "*Ни один микропроцессор серии SA-111 не составляет ядро персонального компьютера IBM PC*" будет представлена как $T \cdot x \neq F \cdot y$, где x и y – некоторые числа. Но всегда возможно подобрать такие x и y , чтобы последнее неравенство стало бы уравнением, поэтому Лейбниц вводит положительные числа для утверждения, и отрицательные – для отрицания. Тем не менее, в таком варианте невозможным является комбинация утверждений и отрицаний: выражение "*Микропроцессоры фирмы Intel составляют ядро персональных компьютеров IBM PC, но ни один микропроцессор*

серии SA-111 не составляет ядро персонального компьютера IBM PC" невозможно представить в системе Лейбница, а утверждения для сложных объектов представляли собой числа значительного для вычислений порядка. Однако результатом работы Лейбница была первая логическая система, основанная исключительно на символах и математических операциях, и поэтому способная к вычислениям.

Немецкий философ-математик Ламберт создаёт исчисление с операциями сложения, вычитания, умножения, деления и отношениями равенства и неравенства, обозначая их "+", "-", "×", "÷", "=", "≠" соответственно. При этом он вводит аксиомы для этих операций, среди которых $A \times A = A$ (принцип идемпотентности); $1 + A = 1 - A$, из которой вытекает $A + A = 0$ (исключающее "или"). Такая аксиоматика полностью согласуется с бинарной алгеброй, принятой и теперь за основу логики. Разработчиком такой алгебры стал ирландский математик Буль. На основе принципа идемпотентности в логике $AA = A$ он искал корни уравнения $AA = A$ или же $A^2 - A = 0$. Таким образом Буль установил, что в алгебре логики могут существовать лишь два значения: 1 и 0. Поэтому одной из аксиом его арифметики есть $1 + 1 = 0$. И лишь со временем 1 и 0 начали трактоваться как "истина" и "ложь" (как предлагалось Луллием), умножение – как логическое "и", сложение – как "или" (у Буля исключаящее). Большой вклад в развитие алгебры логики сделал Пирс – он, в частности, изменяет исключаящее "или" на включающее. Вершиной же развития логики высказываний и алгебры логики стала полная аксиоматика немецкого философа-математика Г. Фреге. Таким образом, логика высказываний представляется множеством формул. Для автоматического вывода она использует арифметические вычисления (с правилами арифметики логики) значений истинности составляющих в формулах. Для естественно-языкового текста, оперируя довольно большими по объёму частями текста, автоматический вывод в логике высказываний является довольно эффективным механизмом. Однако каждое высказывание имеет свою внутреннюю структуру, которая не может быть представлена с помощью логики высказываний. Например, атомарные предложения "Микропроцессоры фирмы Intel составляют ядро персональных компьютеров IBM PC" и "Микропроцессоры серии SA-111 не составляют ядро персональных компьютеров IBM PC" не могут быть семантически связаны, даже при наличии того факта, что "Микропроцессоры серии SA-111 произведены фирмой Intel".

Исчисление предикатов. Предикат (лат. Praedicatum) – сказуемое суждения; то, что высказывается (утверждается или отрицается) в суждении о субъекте [21]. Предикатная форма ЕЯТ была подмечена ещё Аристотелем. Одной из главнейших вех в истории предикатного представления естественно-языковых высказываний стало введение Лейбницем понятия символического отношения с переменными: решая проблему невозможности силлогистического представления вывода об отце и сыне, он предлагает рассматривать отношение $R(A, B)$ – "A является отцом для B" и $\tilde{R}(A, B)$ – "B является сыном для A" такие, что $R(A, B) = \tilde{R}(A, B)$. Впервые части высказываний как одноместные предикаты представил немецкий философ Плуке, а вместе с ним Фреге, который считал исчисление предикатов универсальным языком, в котором можно бы было представить систематически и математически точным способом любую возможную форму рационального мышления, которая могла бы стать частью дедуктивного мышления [71]. Фреге вводит в логическое исчисление квантор всеобщности (вместе с операциями импликации и отрицания), причём его система является функционально полной. Тем не менее, считая естественно-языковую форму представления знаний причиной логических противоречий, он разрабатывает собственную двумерную систему определений, которая, однако, не

имела продолжения, так как усложняла запись, не имея при этом наглядности других двумерных систем представления логических знаний (семантические сети, концептуальные графы, и т. п.). Ниже представлен пример изображения ЕЯВ "Микропроцессоры фирмы Intel составляют ядро персональных компьютеров IBM PC" в нотации Фреге (рис. 3.1).

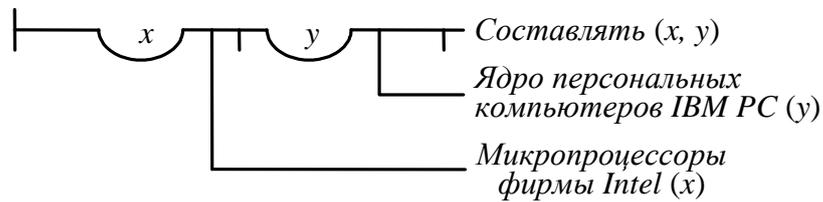


Рис. 3.1. Пример изображения ЕЯВ в нотации Фреге

Как видно из рисунка, малое количество первичных отношений нуждается в перефразировке высказываний в довольно сложную для записи и чтения форму, которые лишь тогда могут быть представлены в нотации Фреге. Форма представления предикатов, как и логических высказываний, – множество формул с переменными, записанными линейно.

Опираясь на арифметические закономерности, Пирс выделил *кванторы всеобщности и существования*: взяв греческий символ Π для обозначения многократного логического умножения, и символ Σ – для сложения, он заметил, что $\Pi = 1$ при истинности (значение 1) всех операндов, а $\Sigma = 1$ – при наличии хотя бы одного истинного (равняющегося 1) операнда. Таким образом, символ Π Пирс назвал квантором всеобщности, а символ Σ – квантором существования. Кроме того, Пирс использует знак " \prec " (модифицированный " \leq ") для следования, так как значение антецедента всегда меньше или равняется значению консеквента. Современное исчисление логики предикатов (и высказываний) наследует систему операций Буля, Фреге и Пирса, кванторы Пирса и нотацию итальянского математика Джузеппе Пеано. Последний во избежание путаницы при совместном использовании логики и арифметики заменил знаки логических операций: квантор существования (от лат. "Est") – на " \exists ", следование (от лат. "Consequentia") – на " \supset ", дизъюнкцию (от лат. "Vel") – на " \vee ", конъюнкцию (от дизъюнкции) – на " \wedge ", отрицание (от "–") – на " \sim ", эквивалентность (от "=") – на " \equiv " [26]. Квантор всеобщности " \forall " (от нем. "Alle") был введен позднее. Также, в настоящее время по обыкновению используется типовая логика предикатов, где вместо конструкций

$(\forall x)(\text{Микропроцессор фирмы Intel } (x) \supset \text{Составляет ядро персонального компьютера IBM PC } (x))$

$(\exists x)(\text{Микропроцессор фирмы Intel } (x) \wedge \text{Микропроцессор серии Itanium 2 } (x))$

используются менее короткие и более читабельные конструкции

$(\forall x: \text{Микропроцессор фирмы Intel})(\text{Составляет ядро персонального компьютера IBM PC } (x))$

$(\exists x: \text{Микропроцессор фирмы Intel})(\text{Микропроцессор серии Itanium 2 } (x))$,

где запись " $(\forall x: \text{Микропроцессор фирмы Intel})$ " читается "для всех x типа микропроцессора фирмы Intel", или просто "для всех микропроцессоров фирмы Intel", а " $(\exists x: \text{Микропроцессор фирмы Intel})$ " – "некоторый микропроцессор фирмы Intel".

Итак, исчисление предикатов, как и логика высказываний, является полной [69, 72] системой итеративного построения и разбора логических высказываний. Однако, атомарность предикатов, вообще, является произвольной, так как кроме

возможности полностью разобрать любое ЕЯВ, например, "*Микропроцессоры фирмы Intel составляют ядро персональных компьютеров IBM PC*" (т. е., если микропроцессор фирмы Intel составляет ядро персонального компьютера IBM PC, то средством для составления ядра является производство фирмы Intel), а именно:

$$(\forall b: \text{Микропроцессор фирмы Intel})(f: \text{Составление})(\exists w: \text{Ядро персонального компьютера IBM PC})(\text{Составлять}(b, f) \supset \text{Средство}(f, w))$$

существует возможность дальнейшей детализации:

$$(\forall b: \text{Микропроцессор фирмы Intel})(f: \text{Составление})(\exists s: \text{Множество}) (\text{Принадлежит}(w, s) \supset \text{Фирма Intel}(w)) \wedge$$

$$(\text{Элементов_в_множестве}(s, 1)) \wedge (\text{Составляет}(b, f) \supset \text{Средство}(f, w)),$$

где объясняется, что под словосочетанием "фирма Intel" следует понимать наличие такой фирмы. Т. е. предикатная модель представления знаний в ЕЯВ способна представить любое утвердительное предложение и, даже некоторые модальности и временные отношения [62]. Автоматический вывод знаний предусматривает предварительный перевод формул в специфическую форму (вывод методом резолюций, например, нуждается в каузальной форме логических формул или нормальной форме Сколема [73]). Поэтому эффективность вычисления знаний, представленных в предикатной системе, ниже, чем у высказываний и быстро уменьшается при детализации знаний.

Экзистенциально-конъюнктивная логика. Дальнейшее изложение касается вариантов, описанных выше подмножеств FOL. Сравнение этих вариантов подмножеств FOL проводится многими авторами работ в области ИИС и ИИ [5, 26, 73 и др.]. Спецификой данного сравнительного анализа является ориентация на ЕЯВ как входную форму представления знаний. В качестве первого из подмножеств рассматривается экзистенциально-конъюнктивная логика (ЭК логика). Её название связано с логическим оператором конъюнкции и квантором существования, которые в ней используются. Широкое применение ЭК логика получила с разработкой и воплощением реляционных БД. Так, каждая БД служит для фиксации наличия некоторого объекта (квантор существования) и перечня неотъемлемых его свойств (соединённых между собой конъюнкцией). Всё, что не входит в БД, рассматривается с логической точки зрения как несуществующее. Такое ограничение позволило создать один из самых употребительных языков запросов – SQL (Structured Query Language) с соответствующими эффективными механизмами выполнения этих запросов, которые сводятся к простому однопроходному перебору. Однако очевидной является неполнота ЭК логики как системы представления знаний. Так, общий факт, что "*Микропроцессоры фирмы Intel составляют ядро персональных компьютеров IBM PC*", нуждался бы во внесении всех микропроцессоров в базу с указанием для каждого из них факта способности составлять ядро персонального компьютера IBM PC. Такая ресурсоёмкая операция должна была бы повторяться для всех записей с каждым появлением новой информации о классе микропроцессоров. Более того, номинально ЭК логика не имеет возможности представить отрицание, например, что микропроцессор серии SA-111 не составляет ядро персонального компьютера IBM PC. В последних диалектах SQL дополнен возможностью написания *триггеров* (процедуры, которые выполняются при истинности некоторого события), и существует возможность перед дополнением в БД порции новой информации, например о микропроцессорах серии SA-111, проверить её и не включать как такую, что противоречит (т. е. отрицает) существующей информации. Тем не менее, использование триггеров, во-первых, расширяет SQL за границы ЭК логики, во-вторых, оставляет невозможным представить информацию с исключениями эксплицитно: "*Микропроцессоры фирмы Intel состав-*

ляют ядро персональных компьютеров IBM PC, но ни один микропроцессор серии SA-111 не составляет ядро персонального компьютера IBM PC", и, в-третьих, написание триггеров как исполняемой процедуры относится к разработке БД, когда внесение факта "микропроцессор серии SA-111 не составляет ядро персонального компьютера IBM PC" в систему с полной логикой является действием на уровне пользователя, что требует меньшей профессиональности и, соответственно, меньших материальных затрат.

Продукционные системы. По своей форме продукционные системы в чём-то наследуют силлогизмы: система продукций состоит из правил $A \supset B$ ("если A , то B "), где в общем виде на месте A и B могут стоять произвольные логические выражения без импликации. Однако в автоматических системах продукций в качестве B используют одиночный факт или ссылку для более простой реализации механизма прямого вывода (от антецедента до консеквента) [73]. Именно системы прямого вывода, по обыкновению, называют продукционными, тогда как системы с обратным выводом по обыкновению являются хорновыми (см. дальше). Составляется система продукций (прямого и обратного вывода) из фактов (предикаты с константными предметами) и правил-продукций. Механизмом обработки продукций с прямым выводом может быть простое вычисление формул с предикатами (в т. ч. и продукций), или же специфические процедуры, ориентированные на постоянство структуры $A \supset B$. Существенным недостатком при использовании только специфических подходов для вывода являются осложнения проверки непротиворечивости продукций [5, 73] – для автоматизации этого процесса необходимо полноценное вычисление предикатных формул, так как методы вывода не предназначены для этого.

Дизъюнкты Хорна. Как уже отмечалось, продукционные системы с обратным выводом в своём большинстве являются хорновыми системами. Причиной этого является использование дизъюнктов Хорна – импликативных формул $A \supset B$, где B является атомарным предикатом. Такой подход даёт ряд преимуществ перед полной системой логики предикатов: эффективный вывод (в общем виде с ростом базы фактов временная сложность возрастает экспоненциально, но в реальных оптимизированных системах вариативность поиска значительно ограничена [74]), простая схема вывода на основе общелогического правила унификации, сведенная к поиску соответствий (детальнее см. [73]), наделение логических импликаций более понятным для пользователя процедурным смыслом: импликация

$$\begin{aligned} & (\text{Ядро_персонального_компьютера_IBM_PC}(x) \wedge \text{Составляет}(x)) \supset \\ & \supset \text{Микропроцессор_фирмы_Intel}(x) \end{aligned}$$

будет читаться не "Из того, что x ядро персонального компьютера IBM PC и x составляет, следует, что x – микропроцессор фирмы Intel", а " x является микропроцессором фирмы Intel – это когда x является ядром персонального компьютера IBM PC и x составляет". Такие принципиальные преимущества сделали язык логического программирования Prolog, который применяет логику дизъюнктов Хорна, одним из известнейших языков проектирования и разработки баз знаний по всему миру. Предыдущее ЕЯВ может быть записано в правиле Prolog как

`Intel's_microprocessor(X) < is_core_of_personal_computer_IBM_PC(X) & makes(X).`

или в другом диалекте

`Intel's_microprocessor(X): is_core_of_personal_computer_IBM_PC(X), makes(X).`

Тем не менее, несущественное для вопросно-ответных систем и баз знаний ограничение данной логики [74] является определяющим при построении систем

представления ЕЯВ с автоматизированной обработкой естественно-языковых структур: выражение логики предикатов

$$(Ядро_персонального_компьютера_IBM_PC(x) \wedge Составляет(x)) \supset$$

$$\supset (Микропроцессор_фирмы_Intel(x) \wedge \sim Микропроцессор\ серии\ SA-111(x)),$$

соответствующее естественно-языковому "Микропроцессоры фирмы Intel, кроме микропроцессоров серии SA-111, являются ядром персонального компьютера IBM PC и его составляют" является недопустимым в Хорновых системах, ведь далеко не все естественно-языковые структуры могут быть представлены в системах с логикой Хорна.

Фреймовые системы и объектно-ориентированные системы. "Термин *фрейм* (от англ. "frame") был предложен Марвином Минским [75]... Эта модель ... имеет глубокое психологическое обоснование. Фрейм – это абстрактный образ для представления некоторого стереотипа восприятия" [5]. Предложение же Минского касалось именно представления знаний. В общем виде, формально фрейм можно описать как структуру, включающую имя фрейма, множество слотов со своими именами и множество присоединённых процедур, связанных с фреймом или со слотами [73]. Такие фреймы называют фреймами-прототипами. Фреймы-примеры являются фреймами-прототипами со значениями слотов и должны отображать определённые реальные объекты, ситуации и т. п. [5, 73]. Из понятия фреймов произошло понятие объектно-ориентированных (ОО) систем, которые сузили трактовку фреймов-прототипов к классам, а фреймов-примеров к объектам класса, сохранив полностью структуру и возможности фреймов. Например, предложение "Микропроцессоры фирмы Intel составляют ядро персональных компьютеров IBM PC" может быть представлено фреймовой структурой как

Микропроцессор фирмы Intel	
Условие вхождения	Быть произведенным фирмой Intel
Вхождение как часть целого	Ядро персонального компьютера IBM PC

или в виде объекта класса в ОО языке программирования (например, C++):

```
class Intel's_microprocessor
{
    const char * condition_of_entering;
    const char * entering_as_part_of_whole;
} some_Intel's_microprocessor;
some_Intel's_microprocessor.condition_of_entering =
"быть произведенным фирмой Intel";
some_Intel's_microprocessor.entering_as_part_of_whole = "ядро персонального
компьютера IBM PC";
```

Во фреймовых системах просто реализуется функция наследования. Так, добавление к примеру информации о том, что "Микропроцессор серии Itanium 2", может быть представлено как

Микропроцессор серии Itanium 2	
фрейм_предок	Микропроцессор фирмы Intel

или

```
class the_microprocessor_of_series_Itanium_2: Микропроцессор_фирмы_Intel
{
    const char *name;
} the_microprocessor_of_series_Itanium_2;
```

the microprocessor of series Itanium 2.name = "микропроцессор серии Itanium 2".

Тем не менее, не считаясь с широким применением фреймового представления знаний благодаря интуитивной понятности принципа прототипов и примеров, простоте реализации наследования атрибутов, высокой степени структурированности и других плюсов фреймовых систем, логической основой фреймов (без использования слотов-процедур) является ЭК логика со всеми вытекающими из этого ограничениями.

Семантические сети и концептуальные графы. Семантической сетью называется ориентированный граф с именованными вершинами, которые представляют объекты, и дугами – отношениями между объектами. Как граф семантические сети используют как минимум два измерения для своей нотации. Первой семантической сетью было небольшое дерево понятий с отношениями род-вид, построенное Порфирием по категориям Аристотеля [26]. Позднее неоднократно строились деревья категорий, однако они использовали очень ограниченный набор отношений. Семантические сети получили дальнейшее развитие в связи с возможностью их автоматической обработки на ЭВМ, расширяется множество используемых отношений: лингвистические (падежные, атрибутивные), логические (конъюнкция, дизъюнкция, отрицание, импликация), квантифицированные (*все, некоторый*) [73]. Логической основой семантических сетей является полная логика предикатов первого порядка – каждое отношение можно рассматривать как предикат от сопредельных ему (отношению) вершин – объектов. Из всего разнообразия семантических сетей особого внимания заслуживают *концептуальные графы* [76]. Об этом свидетельствует хотя бы тот факт, что для КГ комитетом NCITS T2 по информационному обмену и интерпретации были приняты ANSI-стандарты. КГ предназначены для представления знаний на уровне человек-машина, хотя существует форма записи КГ, предназначенная для манипулирования графами между машинами. Для КГ определены базовые универсальные примитивы [76] для построения семантических сетей с произвольными отношениями. Они имеют возможность включать изображения, аудиоинформацию и другие КГ как объекты (вершины графа). Подробное рассмотрение концептуальных графов будет представлено в следующем подразделе.

Язык формализованного представления и обмена знаниями KIF. Одновременно с КГ разработан язык, специально предназначенный для обмена знаниями между гетерогенными системами баз знаний и баз данных. Он носит название *Формат обмена знаниями* (Knowledge Interchange Format или KIF) и также регулируется ANSI-стандартами [77]. Семантика и логическая база языка KIF идентична семантике и логике КГ, но описательная выразительность его меньшая: чтобы упростить компьютерный анализ, KIF имеет упрощенный синтаксис и ограниченное множество символов [26]. Естественно-языковое предложение "*Микропроцессоры фирмы Intel составляют ядро персональных компьютеров IBM PC*" в KIF будет иметь вид

```
(forall (?b "Микропроцессор фирмы Intel") (?f "Вхождение")
  (exists (?w "Ядро персонального компьютера IBM PC")
    (=> ("Входит" ?b ?f)
        ("Составление" ?f ?w)
    )
  )
)
```

Как и КГ, KIF является универсальным языком представления и обработки знаний, поэтому его целесообразно адаптировать для решения конкретной задачи, кое-что модернизируя, но не внося изменений, которые противоречили бы стандартам для KIF, что сохранит возможность обмена представленными таким образом знаниями между системами с поддержкой KIF. Преимуществом KIF над КГ является его линейная текстовая форма записи, подобная LISP, – интерпретаторы последнего имеют возможность обрабатывать KIF-программы при наличии соответствующих модулей, которые делают синтаксис KIF "понятным" для LISP-интерпретаторов [78].

Сравнительный анализ подмножеств FOL для представления ЕЯВ. На основе выполненного анализа подмножеств FOL, в том числе с их реализациями в действующих системах и нотациях, можно сделать вывод, что они дают возможность в той или иной мере представлять сложные утвердительные предложения и вопросы к их частям, заменяя некоторые компоненты представления (силлогизма, формулы, графа) неизвестной переменной, ответы на которые могут быть получены или решением уравнения, где значение формул с переменной должно быть истинным, или поиском фрагмента графа-вопроса в общем графе [5], и установлением значения неизвестных узлов или дуг. ЕЯВ с модальными операторами или пространственными и временными отношениями могут быть представлены в некоторых из рассмотренных подмножеств FOL, но с определенными изменениями подхода к формализации смысла ЕЯВ [62], или привлечением других, *неклассических логик*: модальной, временной, многозначной, нечёткой, интуиционистской и т. п. [69, 79]. Как механизм, описанный в [62], так и специальные кванторы, операции, аксиомы и механизмы вывода неклассических логик, а также их возможности и ограничения относительно представления ЕЯВ, нуждаются в дополнительном основательном исследовании, которое выходит за рамки данной работы.

На основе описанных выше преимуществ и недостатков рассмотренных логических систем и систем представления знаний, можно провести их *нестрогое ранжирование* (описание метода см. в [80]) (от системы с минимальными возможностями к системе с максимальными возможностями) по самым существенным характеристикам с ориентировочными оценками возможностей данных систем относительно использования для анализа ЕЯВ. Список параметров оценки не претендует на полноту и универсальность для задач представления и анализа знаний в целом.

1. ***Описательная выразительность.*** Наименьшие возможности для представления произвольных ЕЯВ (лишь определенное множество логических выводов) имеют силлогизмы. Далее идёт наиболее ограниченное подмножество логики предикатов – ЭК логика, фреймы и ОО системы, в которых взято за основу это подмножество (перечень характеристик произвольного числа существующих объектов). Полной функциональностью обладает система логики высказываний, однако невозможность связывания частей высказываний сильно ограничивает возможности такой системы. Более мощными являются логические системы предикатов с возможностью применения импликации (но с определенными ограничениями) – продукционные системы и дизъюнкты Хорна. Снимая ограничение на применение операций и прибавляя квантор всеобщности, исчисление предикатов и, соответственно, KIF имеют наибольшую описательную выразительность. Такими же возможностями представления ЕЯВ обладают и КГ.

2. ***Наглядность представления.*** Имея большую степень детализации, значительное количество переменных для связывания формул и при этом используя

математические определения для операций и кванторов, исчисление предикатов является наименее наглядной системой представления ЕЯВ. Уменьшая количество операций и кванторов, ЭК логика имеет большую наглядность, чем полная система исчисления предикатов. Импликативная форма представления является более естественной для человека, чем простое перечисление объектов с их характеристиками, поэтому продукционные системы являются следующими по наглядности. Возможность предоставления продукциям процедурного смысла увеличивает наглядность дизъюнктов Хорна. Благодаря применению естественно-языковых слов (с ограничениями), КИФ является более наглядным. Логика высказываний, в отличие от КИФ, разрешает применять произвольные слова и символы, поэтому её наглядность ещё большая. Имея "глубокое психологическое обоснование" относительно человеческой склонности "представления стереотипа восприятия" [5], следующими по наглядности являются фреймовые системы. Будучи, фактически, набором естественно-языковых предложений, силлогизмы являются наиболее наглядными относительно представления ЕЯВ. Однако, поскольку графическое представление не уступает языковому, КГ имеют ту же степень наглядности.

3. **Простота трансляции из естественного языка.** По причинам, указанным для наглядности представления (см. п. 2), трансляция ЕЯВ в предикатное представление является самой трудной. Практически такие же проблемы присутствуют при переводе ЕЯВ в КИФ, кроме разве что, написания на естественном языке кванторов ("forall", "exists") и операций ("and", "or" и т. п.) – поэтому КИФ занимает следующее место по простоте трансляции. Ограничение на операции и кванторы упрощает задачу перевода, сужая круг возможных вариантов представления, поэтому следующим является ЭК логика. Также, для ЕЯВ, которые имеют вид импликаций, продукционные системы имеют такие же характеристики простоты трансляции, что и ЭК логика. Благодаря возможности рассматривать дизъюнкты Хорна в процедурном смысле, Хорновые системы имеют дополнительный механизм перевода ЕЯВ в формулы, и потому занимают следующее за ЭК логикой и продукционными системами место. С точки зрения синтаксиса, фреймы могут представлять структуру предложения по правилам взаимного расположения частей предложения в ЕЯВ – таким образом, имея фреймовое представление возможных схем предложений, задача перевода ЕЯВ во фреймовой системе упрощается. Трансляция в КГ может проводиться по правилам синтаксического разбора предложения, но без требования наличия предварительно составленных схем предложений. Ограниченность логики высказываний даёт ещё более простой механизм представления в ней ЕЯВ – замена оборотов на сложноподчиненные предложения и дальнейшая замена союзов соответствующими логическими операторами. Соответствующие по структуре силлогизмам ЕЯВ могут быть переведены в силлогизмы заменой средних терминов на переменные, что является простейшим из всех возможных способов трансляции ЕЯВ у подмножеств FOL, которые рассматриваются.

4. **Простота представления в оперативной и постоянной памяти ЭВМ.** Самыми сложными для представления в памяти ЭВМ являются структуры, не предусмотренные архитектурой ЭВМ и возможностями языка программирования. Такими структурами являются, в частности, графы. Представление усложняется отсутствием ограничения на типы вершин и связей. Довольно сложным является представление силлогизмов – простое текстовое представление не даёт возможности связывать выражения между собой и выделять силлогизмы из множества других. Использование текстовой формы продукционных систем и систем дизъюнктов Хорна практически нивелирует их преимущества относительно простоты и эффек-

тивности вывода, поэтому вопрос о выборе структуры представления является открытым (в Prolog, например, используются специфические графы [81]). Более простой является задача представления в памяти знаний, которые содержатся в системе с ЭК логикой – следуя принципам БД, таким представлением могут быть блоки записей фиксированной длины. Фреймы и, тем более, ОО системы прямо отображаются структурами (*structure* в терминах С-подобных языков программирования) или записями (*record* в Pascal) и классами в ОО языках соответственно. Простейшей является форма представления в виде текста, которая присуща для логики высказываний, исчисления предикатов и КИФ, тем не менее, отсутствие в алфавите КИФ специальных символов (логических кванторов и операций), делает его представление простейшим из всех.

5. Ёмкость представления в оперативной и постоянной памяти ЭВМ.

Исходя из объёма знаний, представленных силлогизмами (единичный вывод), имея при этом форму как минимум трёх предложений, силлогизмы являются наиболее ёмким способом представления знаний. Системы с ЭК логикой (преимущественно речь идёт о БД), использующие записи фиксированной длины, также являются достаточно ёмкими. Логика высказываний сохраняет только текст, необходимый для установления истинности ЕЯВ, тем не менее, как отмечалось ранее, атомарные элементы логики высказываний являются довольно большими, поэтому нуждаются в значительных объёмах памяти для хранения. Продукционные системы и системы дизъюнктов Хорна более детализированы, чем логика высказываний, поэтому имеют возможность выражать ЕЯВ в менее ёмкой форме. По возможности использования всех операторов и кванторов в любой формуле, исчисление предикатов является более короткой нотацией, чем продукции. Следующим в последовательности стоит КИФ, поскольку включает большее количество операций (арифметика, работа со списками и т. п.). Фреймовые структуры и КГ, имея прямое отображение в память (упорядоченные записи и данные вершин со ссылками на дуги соответственно), имеют наименьшую ёмкость в памяти ЭВМ.

6. Эффективность вывода. Как уже отмечалось, вывод в системах с ЭК логикой сводится к методу прямого полного перебора, который является наихудшим по своей эффективности. КИФу необходимо вычисление предикатных формул, которые являются эффективнее, чем полный прямой перебор, но в отличие от логики предикатов, КИФ имеет больше возможностей представления (функции, определение, цель средства и т. п.), для чего требуются дополнительные вычисления. Поэтому следующим является КИФ, а далее – исчисление предикатов. Применяя перебор фактов, использование продукционных систем и систем дизъюнктов Хорна даёт возможность сократить время вывода оптимизацией поиска (см. выше), в основе которой лежит ограничение данных логических систем. Логика высказываний не включает кванторы, в отличие от исчисления предикатов, и оперирует значительными по объёму элементами (целостные элементарные ЕЯВ), поэтому содержит меньше операций, которые подлежат вычислению. Вывод на КГ может быть сведен к обработке структуры графа (поиск кратчайшего пути, Эйлерового цикла и т. п.), для чего разработаны эффективные алгоритмы. Система фреймов в общем виде является графовой [73], тем не менее, большинство систем используют иерархические структуры фреймов (в частности, ОО системы), что упрощает вывод. Вывод в силлогизмах, как отмечалось раньше, может быть сделан за одну итерацию.

7. Возможность декомпозиции. Силлогизмы являются целостной и неделимой структурой, поэтому вообще не подлежат декомпозиции. Логика высказы-

ваний может разбивать сложные ЕЯВ на простые, не имея возможности представления их внутренней структуры. КГ, как и любые семантические сети, могут быть разбиты на подграфы, однако выделение каждого подграфа разрывает все внешние для него связи, что значительно усложняет проверку целостности графа после декомпозиции. Декомпозиция систем продукций и дизъюнктов Хорна более простая, но структура $A \supset B$ является неразрывной. Фреймовые структуры имеют аналогичные с КГ проблемы относительно декомпозиции, однако представление в виде дерева практически снимает их. Следующими идут исчисление предикатов и KIF, где декомпозиция осуществляется разбивками на элементарные предикаты, которые могут быть далее детализированы их записью как лямбда-выражения. Простейшей является декомпозиция утверждений в ЭК логике, где объекты уже разделены между собой по операции конъюнкции.

8. **Масштабируемость базы знаний.** Прибавляя новый объект к КГ, как и к любой семантической сети, необходимо проанализировать возможность его связывания со всеми другими объектами по всем отношениям, присутствующим в графе, которая является довольно сложной и ресурсоёмкой операцией. Масштабируемость базы знаний систем с ЭК логикой более простая, но нуждается в проверке на полноту и непротиворечивость, которая также решается исключительно полным перебором среди имеющихся записей и существующих отношений, однако количество последних значительно меньше, чем у КГ, поэтому эффективность и простота расширения базы знаний систем с ЭК логикой выше, чем в КГ. Следующими идут продукционные системы и системы дизъюнктов Хорна, недостатком которых при пополнении базы знаний является сложность проверки на непротиворечивость среди правил. Фреймы и ОО системы пополняются добавлением нового фрейма в дерево иерархии, или добавлением нового поля к фрейму-предку. Простейшим является пополнение базы знаний в системе силлогизмов, логике высказываний, исчислении предикатов и KIF, так как единая проблема непротиворечивости новых знаний уже существующим решается единым универсальным методом обработки знаний в этих системах (силами пользователя в силлогизмах и прямым вычислением в других).

9. **Наличие общих методов и средств анализа.** Единой формой представления знаний, для которой не существует общих методов и средств анализа, являются КГ, так как при анализе должна учитываться семантика отношений системы представления, что невозможно из-за отсутствия любых ограничений для КГ в этом плане.

На основе представленного выше ранжирования можно поставить в соответствие каждой системе логического представления соответствующие порядковые номера (баллы), начиная с 1, для системы с минимальными возможностями, представив полученные относительные оценки в таблице 3.1.

Совокупный формальный анализ табличных оценок является проблематичным из-за сложности установления количественных взаимосвязей между относительными оценками, занесенными в каждый отдельный столбец. Однако по данным таблицы могут быть предложены определенные рекомендации относительно задачи представления и анализа знаний, которые содержатся в ЕЯТ, а именно: наибольшей выразительностью обладают *три системы*: исчисление предикатов, концептуальные графы и KIF. Это может служить критерием выбора именно этих систем логики в качестве основных для представления и анализа знаний. По другим характеристикам, можно предложить, что:

- представлять знания для пользователя целесообразно в КГ;

- трансляцию знаний нужно выполнять в три этапа (с ЕЯТ в КГ, с КГ в КИФ, с КИФ в исчисление предикатов);
- сохранять БЗ в оперативной и постоянной памяти целесообразно в КИФ (объединяя оценки простоты и ёмкости представления в памяти);
- выбор формы представления для анализа знаний зависит от задачи анализа (некоторые задачи можно эффективно выполнять навигацией в КГ по алгоритмам работы с графами, но ряд задач не имеет методов и средств их решения на графах – в таком случае должны применяться универсальные общие методы логического вывода на системе исчисления предикатов или на КИФ);
- обработку БЗ (пополнение и декомпозицию) целесообразно проводить в форме исчисления предикатов или КИФ.

Таблица 3.1

Оценки возможностей подмножеств FOL

	Описательная выразительность для ЕЯВ	Наглядность представления	Простота трансляции из естественного языка	Простота представления в оперативной и постоянной памяти ЭВМ	Ёмкость представления в оперативной и постоянной памяти ЭВМ	Эффективность вывода	Простота декомпозиции	Масштабируемость базы знаний	Наличие общих методов и средств анализа
Силлогизмы	1	8	8	2	1	8	1	4	2
Логика высказываний	3	6	7	5	3	5	2	4	2
Исчисление предикатов	5	1	1	5	5	3	7	4	2
ЭК логика	2	2	3	4	2	1	8	2	2
Производные системы	4	3	3	3	4	4	4	3	2
Дизъюнкты Хорна	4	4	4	3	4	4	4	3	2
Фреймы и ОО системы	2	7	5	6	7	7	5	5	2
Концептуальные графы	5	8	6	1	7	6	3	1	1
КИФ	5	5	2	9	6	2	7	4	2

3.3. Формализованное представление ЕЯ-высказываний. Концептуальные графы

Проблема поиска, представления, понимания и компьютерной обработки естественно-языковых объектов является одной из наиболее сложных проблем в свете решения соответствующих задач искусственного интеллекта, в частности построения эффективного человеко-машинного общения на языке, близком к естественному.

Ниже представлено краткое описание метода представления и обработки ЕЯВ с использованием языка предикатов первого порядка. Данный формализм охватывает проблему анализа текстов в сочетании с методом резолюций.

Уместно уточнить, что проблему компьютерной обработки ЕЯВ мы подразделяем на три подпроблемы:

а) представление ЕЯВ;

б) получение следствий, проверки выполнимости, противоречивости или непротиворечивости заданного множества формул на основе выбранного представления;

в) получение цепочки формального доказательства выводимости следствий, выполнимости (невыполнимости) заданного множества следствий.

Анализ перечисленных подпроблем показывает, что использование формальных методов проверки непротиворечивости или выполнимости множества формул (т. е. в системах, в которых доказательства получаются формальным образом из аксиом путём применения правил вывода) для подпроблемы в) не имеет более-менее удовлетворительного решения. Дело в том, что системы гильбертовского типа не являются структурированными, т. е. для сбора необходимой информации по одному единственному объекту необходимо просматривать всё множество логических формул, которые находятся в системе (обычно такой системой является реляционная база данных). К сожалению, этим недостатком страдают все формальные логические системы гильбертовского типа. В целях устранения этого недостатка было предложено графическое представление формул и их аргументов, которое служит глобализации и структурированию информации. Основой графического представления являются *концептуальные графы* и более сложные структуры – *семантические сети* (СС) [26, 82]. Такое представление позволяет визуализировать модель естественно-языковой картины мира и в случае необходимости, весь процесс доказательства или процесс вывода.

Формальные определения и примеры

Введём формальные определения упомянутых выше понятий КГ и СС, а также приведём примеры для иллюстраций этих понятий.

Определение 3.1. *Концептуальным графом называется размеченный двудольный ориентированный граф $G = (V_1 \cup V_2, E)$, где $V_1 \cap V_2 = \emptyset$, вершины из V_1 размечены именами предикатов, а вершины из V_2 – именами аргументов; E – множество дуг (ориентированных ребер). Дуги графа соединяют вершины, помеченные именами предикатов, с вершинами, помеченными именами аргументов. Вершины из множества V_1 называются узлами-предикатами, вершины из V_2 – узлами-концептами, а сами предикаты – концептуальными предикатами.*

Из приведенного определения следует, что концептуальный граф должен удовлетворять таким условиям:

1) количество дуг, связывающих узлы-предикаты с узлами-концептами, равно арности предиката (т. е. числу его аргументов);

2) все узлы-предикаты, помеченные символом одного и того же концептуального предиката, имеют одинаковую арность;

3) все дуги, связывающие узлы-предикаты и узлы-концепты в КГ, упорядочены от 1 до n , где n – арность предиката;

Графически узлы-предикаты обозначаются эллипсами, а узлы-концепты – прямоугольниками.

Если используемый логический язык типизирован, т. е. его объектам приписаны некоторые типы, то с каждым концептуальным предикатным символом связывается кортеж $\langle a, b, \dots, c \rangle$, называемый сигнатурой этого предиката. Или другими словами, сигнатура предиката описывает семантическую модель правильного

употребления аргументов в синтаксической структуре некоторого ЕЯ-предложения с максимальной выразительной мощностью. Под максимальной выразительной мощностью понимается соответствующий кортеж без пустых элементов.

КГ не случайно выбраны в качестве структур данных для представления предикатов и их аргументов. Эти структуры прошли длительную апробацию в системах баз знаний и зарекомендовали себя с наилучшей стороны. Это проявляется, прежде всего, в эффективности выполнения операций на таких структурах данных. Операции на множестве КГ выполняются с целью эффективного построения более сложной структуры – семантической сети.

Определение 3.2. Семантической сетью называется объединение заданного множества концептуальных графов вместе с описанием их взаимосвязей и погружением в контекст области рассуждений.

Для подкрепления этих полуформальных определений КГ и СС рассмотрим примеры, иллюстрирующие эти понятия.

Пример 1. Рассмотрим предложение

“КГ описывает семантику некоторого высказывания” (3.1)

Соответствующий этому предложению концептуальный граф представлен на рис. 3.2.

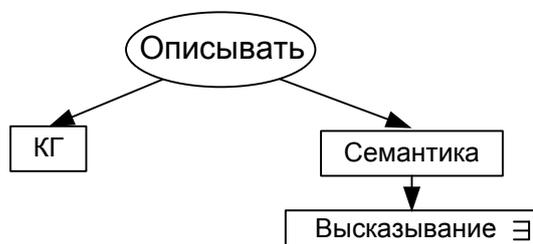


Рис. 3.2. КГ предложения (3.1)

Заметим, что КГ может иметь различные представления, которые включают некоторые уточнения, характеристики контекста или семантики.

В то время как КГ представляют одну логическую формулу, СС представляет совокупность КГ, погружённых во взаимосвязи и общий контекст рассуждений. СС получаются из КГ следующим образом: пусть дано два КГ G_1 и G_2 , тогда КГ G строится из этих графов путём применения правил конъюнкции и упрощения.

Правило конъюнкции (ПК): Если узел-концепт c_1 в G_1 идентичен узлу-концепту c_2 в G_2 , то G получается удалением c_2 и соединением с c_1 всех связывающих узлов, которые были связаны с c_2 в G_2 .

Правило упрощения (ПУ): Если КГ G после соединения содержит два идентичных (соединённых с одними и теми же узлами-концептами) связывающих узла, то можно удалить один из них вместе со связанными с ним дугами.

Пример 2. В качестве первого графа возьмем КГ из примера 1, а второй КГ (рис. 3.3.) соответствует предложению

“КГ описывает семантику предложения с одной грамматической основой” (3.2)

В результате применения ПУ к концептуальным графам высказываний (3.1) и (3.2), получим семантическую сеть для указанных концептуальных графов, представленную на рис. 3.4. Иногда такую СС называют объединенным КГ.

Далее основное внимание будет сосредоточено на КГ и методах их представления, типизации и обработки.

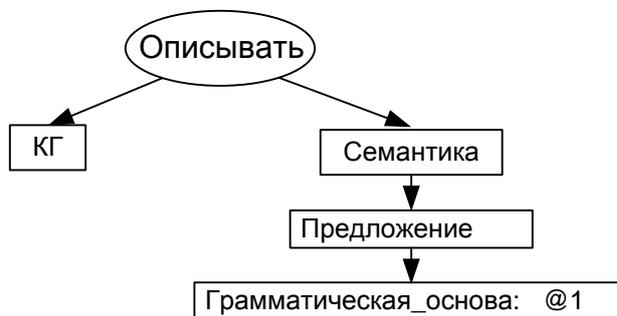


Рис. 3.3. КГ предложения (3.2)

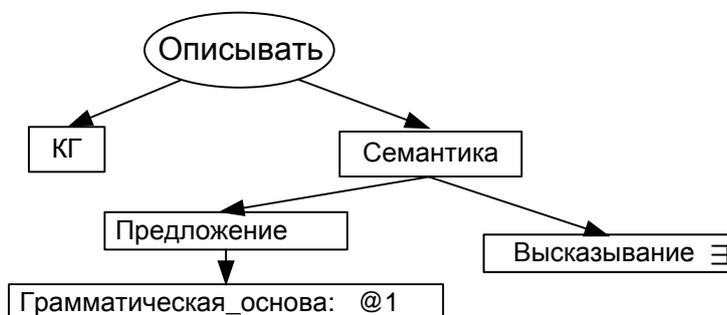


Рис. 3.4. Объединённый КГ для предложений (3.1) и (3.2)

Способы представления КГ и их характеристика

Представление семантической модели ЕЯО в зависимости от типов решаемых задач (и соответствующих КГ) может иметь различную степень детализации. При этом различают две задачи формально-логического представления ЕЯО в процессе компьютерного анализа и интерпретации.

Первая из них относится к внутриязыковой обработке, при которой результат семантического анализа представляется наиболее полными логическими выражениями в соответствующем логическом базисе. Их формирование осуществляется параллельно с процедурой снятия лексической неоднозначности, которая предполагает детализацию КГ и эксплицитное представление соответствующих контекстных зависимостей.

Вторая задача относится к внеязыковой (или постязыковой, машинной) обработке, к этапу создания базы знаний предметной области, а точнее – к созданию базы правил логического вывода. Такая база должна содержать короткие правила, позволяющие реализовывать эффективный вывод (с точки зрения быстродействия) [3].

Из анализа указанных задач следует, что для первой задачи следует использовать детализированные КГ, а для второй – обобщённые КГ и соответствующие СС.

Ниже будут рассмотрены некоторые определения и правила построения детализированных КГ, базирующиеся на концептуальных графах Дж. Соуы [26, 83], отличных от КГ, рассмотренных выше. Также будут рассмотрены основные проблемы, связанные с автоматическим построением КГ (описывающих входной ЕЯО) и предложены пути их преодоления.

Для наглядности описания детализированных концептуальных графов (далее концептуальных графов) будем использовать естественно-языковые высказывания из ПдО “Интеллектуальные информационные системы”, и определение семантической сети, взятое из [3]. Формирование структуры КГ, соответствующей указанному определению, будем проводить в *три этапа*: от простых трёх высказываний (составляющих ядро определения семантической сети) – на первом этапе, к их расширенным аналогам – на втором этапе и их объединению (полному определению) – на третьем.

Рассмотрим содержание **первого этапа** и соответствующий ему КГ (рис. 3.5).

I.1. “Сети представляются в виде графа”.

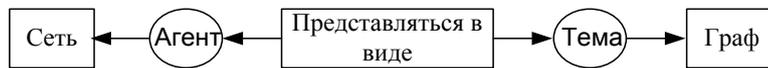


Рис. 3.5. Концептуальный граф для высказывания I.1

Существует и линейная форма записи КГ, при этом прямоугольники заменяются квадратными скобками, а эллипсы – круглыми скобками.

[Сеть] ← (Агент) ← [Представляться в виде] → (Тема) → [Граф].

Обе формы записи, и графическая, и линейная, представляют один и тот же абстрактный граф. Существует алгоритм, с помощью которого можно каждому КГ сопоставить формулу логики предикатов первого порядка. С помощью этого алгоритма КГ на рис. 3.5 преобразуется в нижеследующую формулу логики предикатов. Причём, если никакой другой квантор не определен в поле концепта, то квантором по умолчанию является квантор существования \exists .

($\exists x$: Сеть)($\exists y$: Граф)($\exists z$: Представлять_в_виде)(Агент(z , x) \square Тема(z , y)).

Неформально, КГ – некоторая структура концептов и концептуальных отношений, где каждая дуга связывает некоторое концептуальное отношение R с некоторым концептом S .

Такое представление детализированных КГ несколько отличается от введенного выше описания. Объединение узлов *концептов-предикатов* и узлов *концептов-аргументов* в *узлы-концепты* и введение другого вида узлов – *узлов-концептуальных отношений* (рис. 3.5: агент, тема), является одним из главных моментов сути детализации. При этом узел концепта-предиката назовём *главным узлом-концептом*, а узлы концептов-аргументов – просто *узлами-концептами*. Такое преобразование, по сути, детализирует семантические отношения в синтаксической структуре предложения и представляет их в явном виде.

Компоненты структуры КГ определяются нижеследующими определениями.

- Каждая дуга E в G связывает концептуальное отношение R с концептом C . Дуга E принадлежит отношению R и не принадлежит концепту C .
- Концептуальный граф G может иметь концепты, которые не связаны ни с каким концептуальным отношением, но любая дуга, которая принадлежит некоторому концептуальному отношению в G , должна быть присоединена только к одному концепту.
- Подграфами концептуального графа G могут быть:
 - а) пустой граф, который не имеет ни концептов, ни концептуальных отношений, ни дуг;
 - б) одноэлементный граф, который состоит из одного концепта;
 - в) граф-звезда, который состоит из одного концептуального отношения и концептов, которые присоединены к его дугам.

Ниже представлены следующие два высказывания первого этапа в виде концептуальных графов, записанных в линейной форме.

I.2. “Вершины соответствуют объектам”.

[Вершина] ← (Агент) ← [Соответствовать] → (Тема) → [Объект].

I.3. “Дуги соответствуют отношениям”.

[Дуга] ← (Агент) ← [Соответствовать] → (Тема) → [Отношение].

Второй этап.

Каждый концепт C имеет тип t и референт r . Слева записывается концепт-тип, а через двоеточие справа записывается подтип (элемент, терм, конкретное имя), который присутствует в высказывании. А теперь рассмотрим концептуальный граф для высказывания “Сеть высказывания I.1 представляется в виде концептуального графа”. Для него концепт [Сеть] примет вид [Сеть: Высказывание_I.1]. Такое представление предполагает наличие онтологии ПдО “ИИС” (и базы знаний, содержащей список высказываний, в том числе I.1), фрагмент которой представлен на рис. 3.6.

Каждое концептуальное отношение R имеет реляционный тип t и целое неотрицательное число n , названное арностью.

Число дуг, которые принадлежат R , равно арности n , при этом сами дуги нумеруются от 1 до n .

Для каждого n -арного R существует последовательность, состоящая из n концептных типов $\langle c_1, c_2, \dots, c_n \rangle$, названная *сигнатурой* R . Сигнатура $R \langle c_1, c_2, \dots, c_n \rangle$ является частным случаем сигнатуры концепта-предиката $\langle a, v, \dots, c \rangle$.

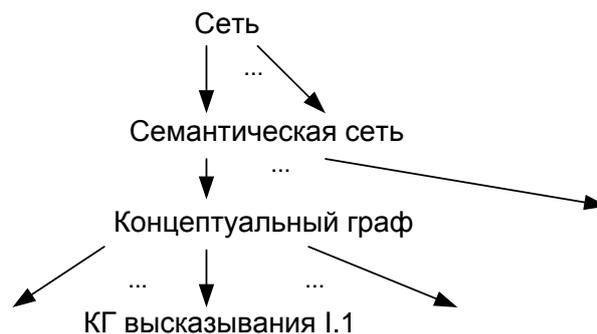


Рис. 3.6. Фрагмент онтологии концепта “Сеть” из ПдО “ИИС”

Сигнатура отношения в общем виде представляет ограничение на типы концептов, которые могут быть связаны с его дугами. Например, для отношения “Агент” сигнатура – <Действие, Сущность>, которая указывает, что тип концепта, связанного с его правой дугой, должен быть “Действие” или некоторым его подтипом в онтологической иерархии (рис. 3.7).

Концепты-типы и концептуальные отношения имеют *иерархические структуры*, которые представляют собой частично-упорядоченные множества T и R соответственно, чьи элементы обозначают типы концептов и отношений. Каждое обозначение концепта в T и отношения в R определено как примитив или как частичное упорядочение по T и R отношением подтипа, с символом \leq – для подтипа, $<$ – для собственного подтипа, \geq – для супертипа и $>$ – для собственного супертипа.



Рис. 3.7. Фрагмент онтологии для концепта-типа “Действие”

Референт концепта определён квантором и указателем. Квантором может быть или квантор существования или квантор всеобщности или обозначения: количества – (@1) и набора {1, 2, 3}.

Квантор существования представлен или символом \exists или отсутствием любого другого символа квантора или выражения.

Квантор всеобщности представлен или символом \forall или выражением в расширенном синтаксисе, которое может быть оттранслировано в концептуальные графы, которые содержат только кванторы существования.

Указатель в поле референта может быть *трех видов: литерал, локатор и дескриптор*.

Референт концепта определяет сущность или множество сущностей, к которым относится концепт.

В КГ под *контекстом* понимается концепт с вложенным КГ, который описывает референт. В свою очередь, вложенный КГ может содержать концепты с вложенными концептуальными графами, т. е. концепты могут быть вложенными.

Кореферентным множеством S в концептуальном графе G называется множество концептов, отобранных из G или из графов, вложенных в контексты G .

В графической системе обозначений элементы кореферентного множества могут быть соединены пунктирными линиями, которые называются *кореферентными связями*.

В линейной системе обозначений два концепта могут быть связаны пунктирной линией только тогда, когда они записаны рядом друг с другом в одной и той же строке.

В ЕЯВ кореферентная связь обычно выражается некоторой формой глагола *быть*. Например, КГ

КГ

 — — — — —

ЕЯВ

 может быть прочитан *Некоторый концептуальный граф есть естественно-языковое высказывание*. В

исчислении предикатов кореферентная связь соответствует равенству: $(\exists x: КГ)(\exists y: ЕЯВ)x=y$.

В теоретическом базисе концептуальных графов определено *понятие базы знаний*, под которым подразумевается контекст типа *KnowledgeBase*, десигнатором которого является некоторый полифункциональный КГ, состоящий из четырёх компонентов: иерархия типов *TypeHierarchy*, иерархия отношений *RelationHierarchy*, каталог используемых индивидов *Catalog of Individuals* и верхний уровень контекста *Assertion*.

Содержание БЗ должно удовлетворять следующим ограничениям:

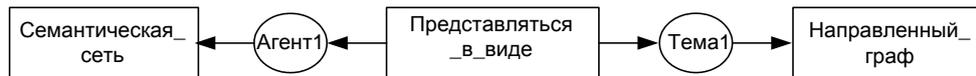
- обозначения типов в любом концептуальном графе должны быть специфицированы в *T*;
- обозначения отношений в любом концептуальном графе должны быть специфицированы в *R*;
- индивидуальные маркеры в любом концептуальном графе должны быть специфицированы в *S*;

Обозначения типов KnowledgeBase, TypeHierarchy, RelationHierarchy, Catalog of Individuals и Assertion являются *метауровневыми обозначениями* и не специфицируются в *T*.

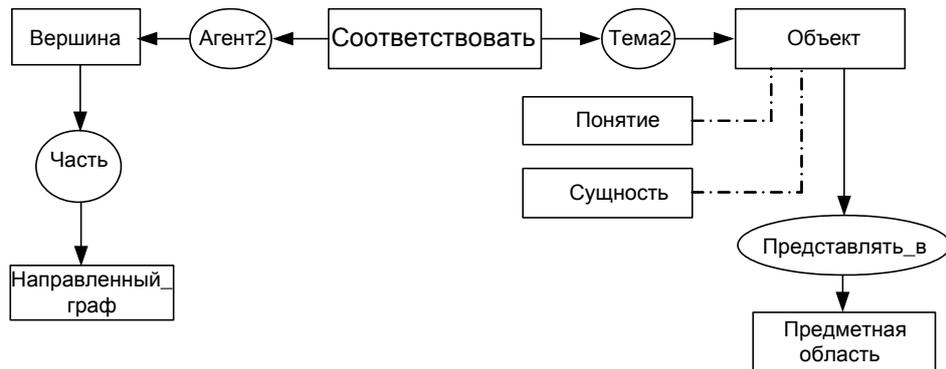
Приведенное определение соотносит БЗ с единственным концептом, который также может быть вложен в другую базу знаний. Каталог индивидов перечисляет все именованные сущности в выбранной для исследования предметной области.

Для дальнейшего рассмотрения основ построения концептуальных графов на рис. 3.8 представлены расширенные высказывания второго этапа и соответствующие им концептуальные графы:

а) “Семантическая сеть представляется в виде направленного графа”.



б) “Вершины направленного графа соответствуют объектам (понятиям, сущностям) предметной области”.



в) “Дуги направленного графа соответствуют отношениям (связям) между объектами”.

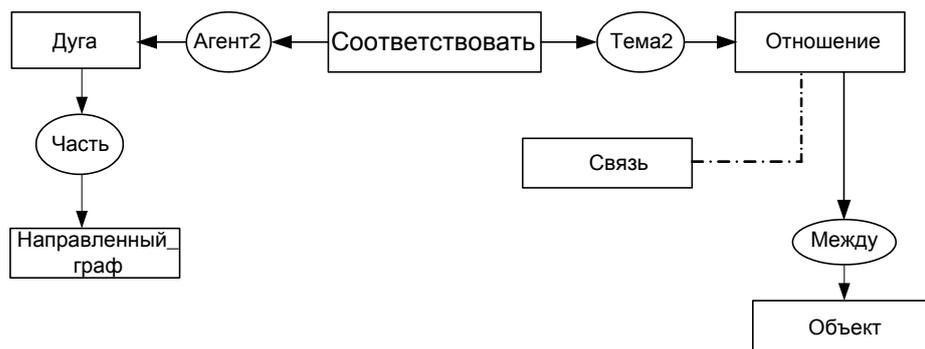


Рис. 3.8. Расширенные высказывания второго этапа и их КГ

В расширенных ЕЯВ второго этапа введенные (на первом этапе) концепты-типы конкретизируются, для некоторых из них – добавляются новые концептуальные отношения и вводятся понятия-синонимы. Указанные расширенные ЕЯВ представляют собой три составляющие (три грамматические основы) одного из принятых формальных определений семантической сети.

На *третьем этапе*, объединив соответствующие КГ, получаем результирующий КГ (или семантическую сеть) для указанного определения.

Несмотря на огромную популярность концептуальных графов, в известной методологии их построения [26, 83] имеется ряд задач, при решении которых у разработчика конкретной знание-ориентированной компьютерной системы возникают серьезные трудности, разрешение которых порою оказывается проблематичным. Например, в случае компьютерной обработки естественно-языковых объектов:

- не определены принципы построения иерархии концептов-типов и концептуальных отношений, а также механизмы их обработки;
- отсутствует механизм функционального анализа ЕЯВ (определения ролевых функций членов предложения);
- не определён механизм выделения терминов из ЕЯО, описывающих некоторую ПдО, причём термины могут состоять из нескольких словоформ различных частей речи (например, “семантическая сеть”, “направленный граф” и др.);
- не определён механизм разрешения индексированных меток (или проблемы анафорических связей);
- не определён механизм фильтрации или отбора из терминов рассматриваемой ПдО концептов и концептов-индивидов;
- не определены механизмы системной интеграции лингвистического и предметного аспектов анализа ЕЯО;
- не определены механизмы квантификации переменных;
- введенное понятие кореферентного множества сформулировано неоднозначно для понимания;
- не определены механизмы формирования шаблонов сигнатур для концептуальных отношений.

Решение проблем, возникающих в связи с устранением описанных выше недостатков, возможно на основе принципов и механизмов построения ЯОИС. Применительно к проблеме построения концептуальных графов в ЯОИС задейст-

вованы компоненты “Языково-онтологическая картина мира”, “Онтология ПдО” и “Грамматический процессор”. ЯОКМ представляет собой общелингвистическую онтологию, компонентами которой являются онтологии лексем таких частей речи, как существительное, глагол, прилагательное и наречие. Онтология ПдО представляет собой онтологическую структуру концептов-типов ПдО, специфичных для неё концептуальных отношений и лексических средств. Системная интеграция указанных онтологий (лингвистической и предметной области) совместно с базой фактов и правил вывода для рассматриваемой ПдО представляет базу знаний *KnowledgeBase*, а грамматический процессор формирует структурные единицы соответствующих иерархий.

Концепты тесно связаны с грамматическими характеристиками ЕЯВ. При построении алгоритма интерпретации ЕЯВ и его КГ важная роль отводится глаголам-сказуемым, поскольку они определяют отношения между подлежащим, дополнением и другими компонентами предложения. Помимо концептов необходимо определить концептуальные отношения, которые будут использованы в КГ. В общем случае для связи компонент в предложении верхний уровень иерархии отношений можно определить следующим образом.

- Отношение *Агент* служит для связывания концепта *Действие* с концептом *Сущность*. Здесь следует различать концепт *Сущность* как одушевлённый объект и неодушевлённый.
- Отношение *Объект* служит для связывания концепта *Событие* или *Состояние* с концептом *Сущность*.
- Отношение *Инструмент* служит для связывания концепта *Действие* с концептом *Сущность* и определяет инструмент, используемый для выполнения этого действия.
- Отношение *Часть* служит для связывания концептов “Часть” и “Целое”.
- Отношение *Тема* служит для связывания концептов *Событие* или *Состояние* с концептом *Сущность*.
- Отношение *Время* маркирует время действия глагола-сказуемого в предложении.
- Отношение *Цель* связывает концепт, представляющий одушевлённую сущность, с другим концептом, представляющим его цель.
- Отношение *Имя* связывает концепт *Действие* с концептом *Одушевлённый* – субъектом действия.
- Отношение *Способ* определяет способ выполнения *Действия* (в предложении обычно выражается наречием).
- Слова, обозначающие предлоги (*В*, *На* и др.) маркируют отношения семантики падежных форм существительных.

Предположим, что указанная обработка ЕЯВ второго этапа уже выполнена. Тогда, применив операцию объединения к концептуальным графам (рис. 3.8), получим синтезированную СС для *третьего этапа* (рис. 3.9).

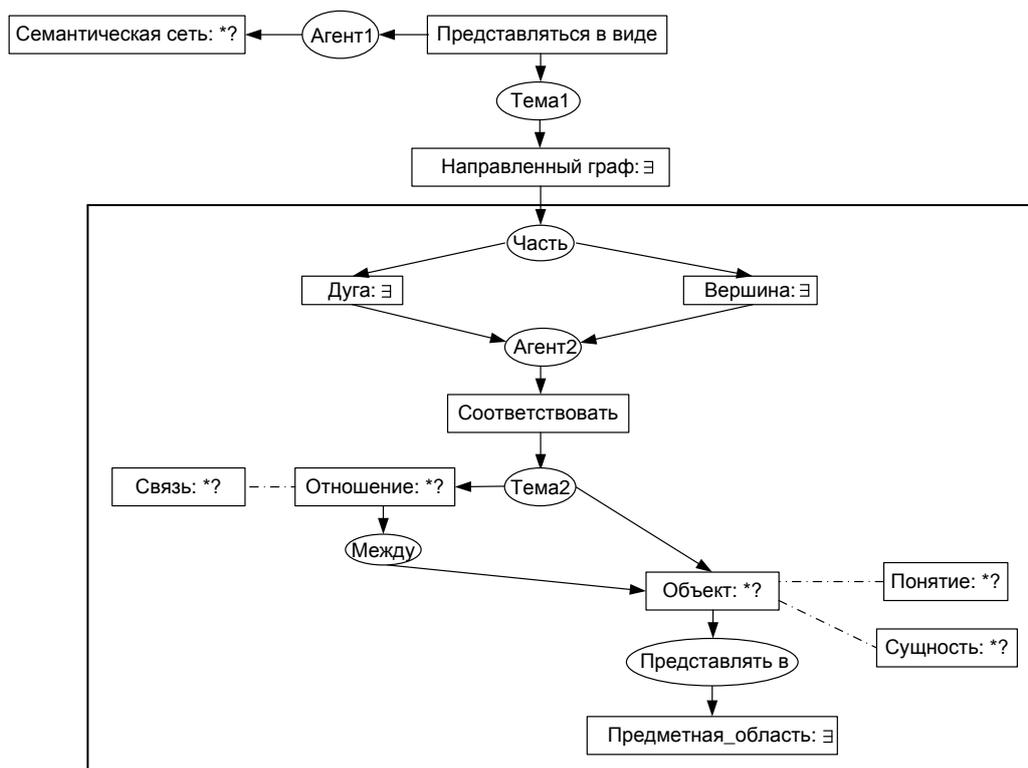


Рис. 3.9. СС ЕЯВ третьего этапа

СС (рис. 3.9) может быть прочитана следующим образом:

Семантические сети представляются в виде направленного графа, вершины которого соответствуют объектам (понятиям, сущностям) предметной области, а дуги – отношениям (связям) между объектами.

Для формирования шаблонов сигнатур концептуальных отношений требуется разработать алгоритм их извлечения из интегрированной онтологии “ЯОКМΛ Онтология_ПдО” (где они представлены в неявном виде) и эксплицитного представления.

Концептуальные графы могут включать концепт-тип [Суждение], объектом ссылки которого является множество концептуальных графов. Этот концепт изображается прямоугольником, содержащим указанное множество концептуальных графов.

КГ позволяют легко представить такие конъюнктивные понятия, как *Граф содержит вершины и дуги*. Сложнее обстоит вопрос с операциями отрицания и дизъюнкции.

Операция отрицания “*Не*” вводится как операция, аргументом которой является концепт-тип [Суждение]. При этом высказывание становится ложным. Так, высказывание *Не существует графов без вершин и дуг* может быть представлено следующим образом.

Используя отрицание и конъюнкцию, можно формировать КГ для представления дизъюнктивных утверждений в соответствии с правилами логики. Предполагается, что в КГ родовые понятия связаны квантором существования. Используя отрицание и квантор существования, можно представить квантор всеобщности.

Например, КГ на рис. 3.10 можно рассматривать как представление логического выражения

$$\forall X1 \forall X2 \forall X3 \forall X4 \left(\neg \left(\left(\text{Граф}(X1) \wedge \text{Вершина}(X2) \wedge \text{Дуга}(X3) \wedge \text{Содержать}(X4) \wedge \right) \right) \right)$$

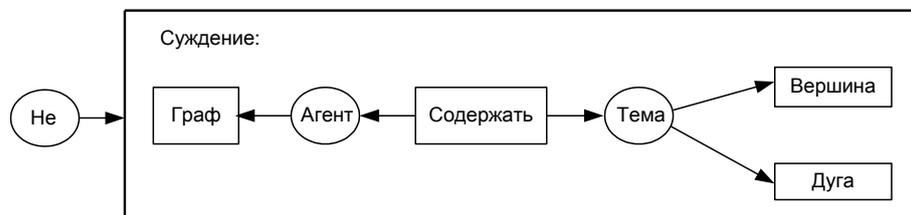


Рис. 3.10. КГ с операцией отрицания

Концептуальные графы по своей выразительной мощности эквивалентны исчислению предикатов первого порядка. Существует прямое соответствие представления КГ системе обозначений исчисления предикатов. Суть его сводится к выполнению таких шагов [83].

1. Каждое из n родовых концептов G связать с отдельной переменной x_1, x_2, \dots, x_n .
2. Каждое конкретное понятие в G связать с отдельной константой. Эта константа может быть просто именем или маркером, используемым для указания объекта ссылки для данного понятия.
3. Представить каждый узел концепта унарным предикатом, имя которого соответствует типу этого узла, а аргументом которого является переменная или константа данного узла.
4. Представить каждое n -арное концептуальное отношение на графе G n -арным предикатом с тем же именем. Каждый аргумент предиката является переменной или константой, соответствующей узлу концепта, связанного с этим отношением.
5. Сформировать тело выражения исчисления предикатов в виде конъюнкции атомарных выражений, построенных в п. п. 3, 4. Все переменные в выражении считаются связанными квантором существования.

Для примера ниже приведены конечные выражения предикатных представлений КГ ЕЯВ этапа (рис. 3.8), полученные в результате применения описанного выше алгоритма. При этом:

– родовые концепты КГ – а) связаны с переменными: $X1$ – “Семантическая_сеть”, $X2$ – “Направленный_граф”, $X3$ – “Представляться_в_виде”;

– родовые концепты КГ – б) связаны с переменными: $X4$ – “Вершина”, $X5$ – “Объект”, $X6$ – “Понятие”, $X7$ – “Сущность”, $X8$ – “Предметная_область”, $X9$ – “Соответствовать”;

– родовые концепты КГ – в) связаны с переменными: $X10$ – “Дуга”, $X11$ – “Отношение”, $X12$ – “Связь”.

а)
$$\exists X1 \exists X2 \exists X3 \left(\left(\text{Семантическая_сеть}(X1) \wedge \text{Направленный_граф}(X2) \wedge \right. \right. \\ \left. \left. \wedge \text{Представляться_в_виде}(X3) \wedge \text{Агент1}(X3, X1) \wedge \text{Тема1}(X3, X2) \right) \right)$$

б)

$$\exists X4 \exists X5 \exists X6 \exists X7 \exists X8 \exists X9 \left(\begin{array}{l} \text{Вершина}(X4) \wedge \text{Объект}(X5) \wedge \text{Понятие}(X6) \wedge \text{Сущность}(X7) \wedge \\ \wedge \text{Предметная_область}(X8) \wedge \text{Соответствовать}(X9) \wedge \\ \text{Агент2}(X9, X4) \wedge \text{Тема2}(X9, X5) \wedge \text{Часть}(X4, X2) \wedge \\ \wedge \text{Представлять_в}(X5, X8) \wedge \text{Быть}(X5, X6) \wedge \text{Быть}(X5, X7) \end{array} \right)$$

в)

$$X2 \exists X9 \exists X10 \exists X11 \exists X12 \left(\begin{array}{l} \text{Дуга}(X10) \wedge \text{Отношение}(X11) \wedge \text{Связь}(X12) \wedge \\ \wedge \text{Соответствовать}(X9) \wedge \text{Агент2}(X9, X10) \wedge \text{Тема2}(X9, X11) \wedge \\ \wedge \text{Часть}(X10, X2) \wedge \text{Между}(X11, X5) \wedge \text{Быть}(X11, X12) \end{array} \right)$$

3.4. Формализованные подходы к извлечению знаний

Вышеприведенный процесс извлечения знаний из ЕЯТ представлен в достаточно общем виде и требует уточнения. Выполним некоторую конкретизацию этого определения применительно к предметной области “Анализ ЕЯТ”. Рассмотрим некоторые примеры.

Лексико-грамматический анализ

Пусть L – язык отношений, которые представляют знания, V – множество грамматических характеристик, включая грамматические разряды ЕЯ, а D – область интерпретации.

Лексико-грамматический анализ приводит к конкретизации интерпретации $\varphi: V \rightarrow D$ и отношений $R_i \in L$. Интерпретация φ в данном случае представляет собой суперпозицию двух функций φ_1 и φ_2 , то есть $\varphi(V) = \varphi_2(\varphi_1(V)) = \varphi_1^* \varphi_2(V)$, где * обозначает суперпозицию функций. Функции φ_1 и φ_2 реализуют процесс синтаксического и семантического анализа предложений текста T , а отношения R_1 и R_2 – это синтаксические (синтаксические правила языка, в котором написан текст T) и семантические ограничения.

Одним из дальнейших возможных уточнений является уточнение отображения φ_1 . Это отображение, в свою очередь, можно рассматривать как суперпозицию двух отображений, которые реализуют морфологический и синтаксический анализ предложений ЕЯТ и вместе с отображением φ_2 образуют целостную систему классического типа, схема которой показана на рис. 3.11 [65].

Для проверки корректности выполненного анализа предусматривается обратный синтез предложения до его записи в обычном орфографическом виде. Такая проверка может выполняться в диалоговом режиме работы системы с пользователем. Возможная структура словарей, которые используются в приведенной схеме и некоторое её обоснование описаны в работах [84–86].

Если детально проанализировать типы общезначимых отношений, с помощью которых строится иерархия понятий, то можно заметить, что они ассоциируются, прежде всего, с *отношением частичного порядка*. А такого типа отношения составляют дистрибутивную решётку, которая имеет ряд полезных свойств, и эти свойства можно использовать для генерации соответствующих следствий, то есть, для поиска (генерации) новых знаний. В качестве примера рассмотрим пример анализа некоторого набора фактов с помощью языка силлогизмов.

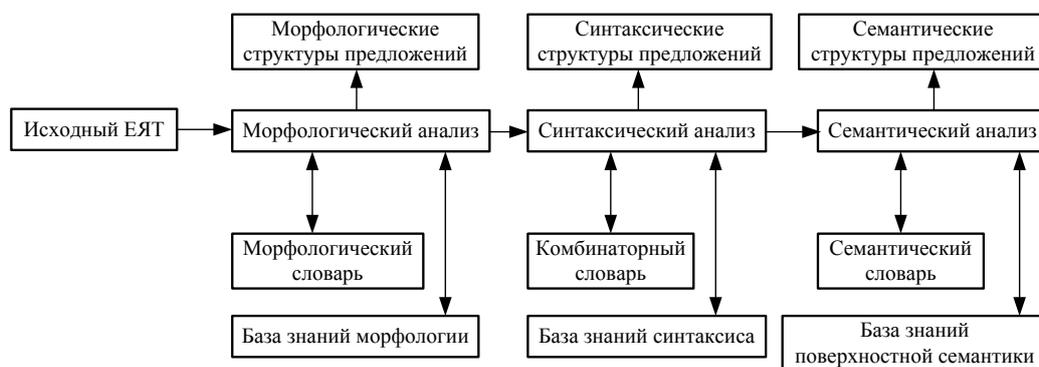


Рис. 3.11 Схема лингвистического анализа классического типа

Силлогистика Аристотеля и её теоретико-множественная интерпретация. Замечено, что силлогизмы Аристотеля легко интерпретируются в алгебре множеств и отношений. А из алгебры множеств и отношений известно, что относительно операций объединения, пересечения и дополнения эти алгебры являются булевыми кольцами, носители которых являются частично упорядоченными множествами относительно теоретико-множественного включения. Законы этой алгебры и свойства частично упорядоченных множеств можно использовать как правила вывода в такой формальной системе. В частности, это общие законы алгебры множеств и отношений (коммутативности, ассоциативности, дистрибутивности, идемпотентности, поглощения и законы де Моргана), три основных свойства отношения включения (транзитивность, контрапозиция и антисимметричность) и закон двойного дополнения. При этом последние два свойства играют роль правил вывода. Проиллюстрируем их действие на примере.

Пример 1. Пусть заданы такие факты (Л. Кэрролл «История с узелками»):

«Все малые дети неразумны»

«Все, кто укрощает змей, заслуживает уважения»;

«Все неразумные люди не заслуживают уважения».

Выясним, что следует из этих фактов. Отметим, что такого типа факты в логике иногда называются *полисиллогизмами* или *соритами*. А силлогизмом называется система, имеющая только две посылки.

Определим основные термины, составляющие систему фактов, введём для них обозначения и выберем универсум U . В данном примере основными терминами являются: «малые дети» (C), «разумные люди» (P), «те, кто укрощает змей» (T), «те, кто заслуживает уважения» (Π). Ясно, что эти термины представляют какие-то множества в универсуме «люди». Их отрицанием будут соответственно такие термины: «не малые дети» ($\neg C$), «неразумные люди» ($\neg P$), «те, кто не укрощает змей» ($\neg T$), «те, кто не заслуживает уважения» ($\neg \Pi$). Теперь наши факты примут вид:

$$C \subseteq \neg P, T \subseteq \Pi, \neg P \subseteq \neg \Pi.$$

Таким образом, определилась решётка (как универсальное множество), которая состоит из элементов ($\emptyset, U, C, P, T, \Pi, \neg C, \neg P, \neg T, \neg \Pi$), где U – универсум. Итак, первыми следствиями данных фактов являются последующие следствия на основании правила контрапозиции (правило контрапозиции в данной интерпретации имеет вид «из $A \subseteq B$ вытекает $\neg B \subseteq \neg A$, где знак \neg означает дополнение множества»):

$$(C_1): P \subseteq \neg C, \quad (C_2): \neg P \subseteq \neg T, \quad (C_3): P \subseteq P.$$

Если перевести полученные следствия на естественный язык, то они соответственно будут означать такие факты: «все разумные люди не являются малыми детьми», «те, которые не заслуживают уважения, не укрощают змей», «заслуживает уважения тот, кто разумный человек».

Пользуясь правилом транзитивности, получаем такие следствия:

$$(C_4): C \subseteq \neg P, \quad (C_5): T \subseteq P, \quad (C_6): \neg P \subseteq \neg T, \quad (C_7): P \subseteq \neg C.$$

$$(C_4): C \subseteq \neg P, \quad (C_5): T \subseteq P, \quad (C_6): \neg P \subseteq \neg T, \quad (C_7): P \subseteq \neg C.$$

Из этих следствий по тому же правилу транзитивности получаем ещё два следствия:

$$(C_8): C \subseteq \neg T, \quad (C_9): T \subseteq \neg C.$$

Если перевести на естественный язык последние следствия, то они будут звучать так:

«все малые дети не являются укротителями змей», «все, кто укрощают змей, не являются малыми детьми».

Таким образом, решение поставленной задачи в приведенном примере получено, но методы генерации следствий являются достаточно сложными процедурами, и хотелось бы иметь какие-нибудь способы упрощения процесса поиска выводов. Если в вышеприведенных обозначениях для формул вместо знака включения \subseteq использовать знак импликации \rightarrow , получим язык, подобный языку математической логики (к сожалению, в этом языке не работает правило *modus ponens*). Одним из способов улучшения ситуации в процессе генерации выводов является использование семантических сетей и их разновидности – концептуальных графов. При таком способе представления фактов поиск и генерация следствий сводится к проблеме достижимости вершин в семантической сети или графе. Рассмотрим детальнее процесс поиска следствий.

Введём некоторые обозначения и нотацию, которой пользуются в математической логике. Будем считать, что каждому термину отвечает некоторое множество или его дополнение, если термин используется вместе с отрицанием.

Определение 3.3. Литералом называется термин или его отрицание, а множество всех литералов некоторого множества терминов будем называть **базовыми литералами**.

Высказыванием называется отношение включения, которое выражено с помощью базовых литералов, в левой части которого находится единый базовый литерал, а в правой части – пересечение множеств, представленных базовыми литералами.

Множество высказываний, которое изображает некоторое множество силлогизмов, будем называть базовым множеством формул.

Определение 3.4. Силлогистической структурой называется множество литералов, отношения между которыми определяются множеством высказываний, которые будем называть **посылками**.

Пример 1 (продолжение). Итак, формализация фактов, приведенных в примере 1, выглядит так: совокупность литералов

$$L = \{C, T, P, \neg C, \neg P, \neg T, \neg P\} \text{ с посылками } C \subseteq \neg P, T \subseteq P, \neg P \subseteq \neg T.$$

Из этого примера следует, что проблема поиска следствий сводится к проблеме построения контрапозиционно-транзитивно-антисимметричного (КТА) замыкания некоторого базового множества формул. При транзитивном замыкании могут возникнуть такие ситуации:

K1) получена формула вида $A \rightarrow \neg A$ или $\neg A \rightarrow A$;

K2) в процессе построения транзитивного замыкания получен хотя бы один цикл.

Выясним, что означают эти ситуации в нашем случае. Первая формула в случае K1) отвечает ситуации $A' \subseteq A$. Из свойства пересечения множества и его дополнения имеем $A \cap A' = \emptyset$, а поэтому такое включение справедливо только тогда, когда A является пустым множеством. Вторая формула в случае K1) означает, что дополнение A' множества A должно быть универсальным множеством. С точки зрения алгебры множеств такие ситуации нельзя назвать противоречивыми, но в нашем случае эта ситуация означает, что некоторый объект должен существовать и в то же время не существовать.

Проанализируем эти случаи более детально. В первом случае причиной возникновения противоречивости типа $A \rightarrow \neg A$ является возникновение в множестве следствий формул вида $A \rightarrow B$ и $A \rightarrow \neg B$. Во втором случае причиной возникновения противоречия $A \rightarrow \neg A$ является возникновение двух формул $A \rightarrow B$ и $\neg A \rightarrow B$.

Пример 1 (продолжение). Если к фактам примера 1 прибавить формулу $P \rightarrow \neg T$ («все разумные люди не укрощают змей»). Такого типа высказывание не кажется странным с точки зрения здравого смысла, но оно приводит к появлению катастрофических последствий. Если построить КТА-замыкание полученного множества фактов, то в этом замыкании появится формула $T \rightarrow \neg T$. Если принимается, что заданное множество фактов правильно, то приходим к выводу, что людей, укрощающих змей не существует.

Заметим, что наличие противоречивости $A \rightarrow \neg A$ не всегда приводит к катастрофическим последствиям. Иногда появление такой противоречивости даёт возможность распознавать факты, которые приводят к противоречию и убирать противоречие.

Ситуация в случае K2) означает (на основании правила антисимметричности (ПА)), что все объекты, находящиеся в цепочке вывода, эквивалентны между собой. Отсюда следует, что в иерархии объектов, составляющих данную информационную систему, действуют два общезначимые фундаментальные отношения – отношение частичного порядка и отношение эквивалентности. Отношение эквивалентности даёт возможность, в случае необходимости, факторизовать объекты в такой информационной системе и этим достигнуть большей компактности в представлении её объектов.

Из рассматриваемых примеров ситуаций для K1) и K2) вытекает ещё один чрезвычайно важный момент. Противоречия типа K1), K2) носят формальный характер, поскольку они появляются только в результате логического анализа заданного множества фактов, но существует ещё один тип противоречия, который существенно отличается от противоречий K1) и K2). Допустим, что в результате построения КТА-замыкания из хорошо обоснованных и проверенных фактов, не являющихся противоречивыми, например, известных и обоснованных теорий, получены противоречивые следствия, то есть, следствия, которые противоречат фактам начальных теорий. Это говорит о том, что **между начальными теориями существуют противоречия, а это и является признаком появления нового знания** или, хотя бы, стимулом для анализа и поиска причин появления полученного противоречия. Всё вышесказанное даёт право ввести такое определение.

Определение 3.5. Информационная система называется корректной, если в ней не возникают противоречия типа K1) или K2).

Известно, что в процессе построения КТА-замыкания, исходя из разных начальных множеств фактов, можно получить одно и то же множество следствий. Это даёт возможность ввести такое отношение эквивалентности на множествах фактов: два множества фактов Φ и Φ' называются эквивалентными, если $KTA(\Phi) = KTA(\Phi')$. На основании этого отношения можно упрощать (путём элиминации) структуру информационной системы, а также начальные множества фактов.

Расширение множества правил вывода силлогической структуры

Вышеприведенные правила вывода в силлогической структуре можно расширить путём использования других свойств отношения включения и законов алгебры множеств отношений. Рассмотрим некоторые из них.

(ЗДД) Закон двойного дополнения (двойного отрицания): $\neg(\neg A) = A$;

(ЗО) Закон объединения для включения (закон дизъюнкции): если $A \subseteq C$ и $B \subseteq P$, то $A \cup B \subseteq C \cup P$, в частности, если $A \subseteq C$ и $B \subseteq C$, то $A \cup B \subseteq C$;

(ЗП) Закон пересечения для включения (закон конъюнкции): если $A \subseteq C$ и $B \subseteq P$, то $A \cap B \subseteq C \cap P$ и $A \cap B \subseteq C$, $A \cap B \subseteq P$;

А когда к результатам применения этих правил применить правило контрапозиции, то приходим к необходимости использования законов де Моргана.

ЗДМ1 Если $A \cup B \subseteq C \cup P$, то $\neg(C \cup P) = \neg C \cap \neg P \subseteq \neg(A \cup B) = \neg A \cap \neg B$;

ЗДМ2 Если $A \cap B \subseteq C \cap P$, то $\neg(C \cap P) = \neg C \cup \neg P \subseteq \neg(A \cap B) = \neg A \cup \neg B$.

Эти правила дают возможность расширить вышеприведенные понятия высказывания и силлогической структуры путём использования правил ЗДД, ЗО, ЗП, ЗДМ1 и ЗДМ2. Применяя эти правила к базовому множеству формул из примера 1, можно получить высказывания типа $T \cap \neg P = \emptyset$ и $C \cap T = \emptyset$, которые говорят о том, что «среди укротителей змей нет неразумных людей», а «среди малых детей нет укротителей змей».

Расширенное таким образом множество правил вывода даёт возможность конструировать дескрипторы, которые включают конъюнкции и дизъюнкции (подобная возможность описывается и в других системах [60]). Следует заметить, что при применении правил ЗО и ЗП, как и ЗДМ1 и ЗДМ2 необходимо следить за некоторыми условиями сохранения смысла. Это связано с тем, что при применении правила ЗП необходима согласованность, состоящая в объединении понятий *родовое* и *конкретное* корректным способом в лексическом и семантическом значении. Отсюда следует необходимость указания отличия в разных моделях, в лексических функциях, в полях устойчивых словосочетаний и т. д. Как правило, в реальных компьютерных системах принимаются ограничения типа: слово представляется не более чем k словоформами, где k – постоянная величина для данной компьютерной системы [63, 84].

Является ли расширенная таким образом силлогическая структура булевой алгеброй, и есть ли добавленное множество правил таким, которое действительно расширяет мощность силлогической структуры – требует дополнительного исследования.

Что касается специальных отношений, то заметим, что они зависят от конкретной ПдО и сформулировать их в общем виде невозможно. Формулирование этого множества отношений целиком зависит от компетентности инженера по знаниям (эксперта в данной ПдО), который эти отношения декларирует и вносит в систему.

4. КОМПЬЮТЕРНЫЕ ОНТОЛОГИИ ПДО

4.1. Системно-онтологический анализ ПДО. Понятие компьютерной онтологии

Под системным *анализом* обычно понимается вид исследования, при котором реальный или мыслимый объект расчленяется на составляющие части (элементы) и затем исследуются эти элементы и связи между ними. Анализ предметной области представляет особый вид научной деятельности, в результате которой строится интерпретационная модель предметных знаний (в широком смысле) [87]. В процессе анализа последние делятся на инвариантные и прагматические знания, концептуальные составляющие которых представляют онтологические знания ПДО.

Некоторые идеи по разработке методологии проектирования онтологии ПДО берут своё начало в литературе по объектно-ориентированному подходу (ООП), возникшему как технология программирования больших программных продуктов [88]. Однако разработка онтологий как иерархической структуры понятий (концептов) отличается от проектирования объектов как классов и отношений в объектно-ориентированном программировании. Последний сосредотачивается главным образом на методах описания классов – программист принимает проектные решения, основанные на *операторных* свойствах класса, тогда как разработчик онтологии принимает эти решения, основываясь на *структурных* свойствах класса. В результате структура понятий и отношения между понятиями в онтологии отличаются от структуры классов объектов подобной ПДО в объектно-ориентированной программе [89]. Кроме того, при разработке онтологии внутреннее содержимое понятия эксплицируется всегда, в то время как в объектно-ориентированном программировании зачастую применяется метод инкапсуляции как способ ограничения доступа к внутреннему содержимому объекта.

Системный подход к познанию ориентирует аналитика на рассмотрение любой ПДО с позиций закономерностей системного целого и взаимодействия составляющих его частей. Системность знаний исходит из многоуровневой иерархической организации любой сущности, то есть все объекты, процессы и явления можно рассматривать как множество более мелких подмножеств (признаков, деталей) и, наоборот, любые объекты можно (и нужно) рассматривать как элементы более высоких классов обобщений.

90-е годы прошлого столетия считаются периодом зарождения *парадигмы компьютерных онтологий* (КО). Она была сформулирована как попытка сгладить (и по возможности устранить) всё чаще проявляющиеся разного рода противоречия при функционировании и внедрении интеллектуальных систем с использованием баз знаний предметных областей. Ярким представителем таких систем на то время были экспертные системы (ЭС). У разработчиков постоянно возникал вопрос: “Как обеспечить их продвижение к конечному пользователю?”. Были предложены разработки оболочек – “пустых” ЭС и ряд других новшеств. Но они не имели решающего значения. Напрашивался вывод, что для эффективного функционирования ЭС на самом важном этапе “жизненного цикла” – функционирования у конечного

пользователя при решении реальных задач – необходимо к каждой ЭС “приложить” эксперта в соответствующей ПдО. Необходимость присутствия эксперта объяснялась, в частности, быстротечностью изменения знаний во многих предметных областях и соответственно необходимостью обновления базы знаний ЭС в “реальном времени”. В тех же ПдО, где знания обладали относительной инвариантностью, ЭС продолжали эффективно функционировать.

Сказанное выше и ряд других факторов “подтолкнули” к разработке парадигмы компьютерных онтологий, основные принципы которой были сформулированы в [7].

1. Доходчивость, ясность (Clarity). Термины (и понятия) онтологии должны отражать реальную действительность. Их символичные обозначения (знаки) должны формироваться на основе общепринятых правил в семиотике и должны выражать общепринятые смыслы реальных объектов. В свою очередь, эти смыслы извлекаются из общепринятых определений терминов (понятий), зафиксированных в толковых словарях, различных глоссариях ПдО. Суждения, входящие в определения, формализуются на основе формального общепринятого аппарата в виде тождественно истинных логических аксиом.

2. Обоснованность, связность (Coherency). Формирование начального набора понятий онтологии и их добавление должно быть обоснованным, определяемым, в первую очередь, требованиями предполагаемой совокупности решаемых задач. Логические аксиомы начального набора понятий должны быть непротиворечивыми. Для этого должен быть предусмотрен механизм логического вывода, который, в том числе, проверяет на непротиворечивость добавляемые аксиомы и выводимые в онтологии утверждения.

3. Расширяемость (Extendibility). Ядром онтологии являются первоначально введенные (спроектированные) понятия и описывающие их аксиомы. В онтологии должен быть предусмотрен механизм расширения (ограничения) совместно используемых словарей понятий без нарушения целостности системы.

4. Минимальное влияние кодирования (Minimal encoding bias). В онтологической системе (ОнС) должен быть реализован принцип совместного использования онтологий, который предполагает: спецификацию онтологии на уровне полноценного представления, а не символического кодирования; запись такой спецификации на общепринятом и платформонезависимом языке описания онтологий можно передать для использования любому программному агенту.

5. Минимальные онтологические обязательства (Minimal ontological commitment). Этот принцип перекликается с принципами обоснованности и расширяемости/ограничения. Важно, чтобы множество понятий онтологии отображало концептуальную структуру ПдО, относительно стабильную на протяжении “жизненного цикла” ОнС. А последняя предоставляла бы возможность расширения или специализации отдельных ветвей онтологического графа. Отделение *концептуальных знаний* от знаний, выраженных фактами, является стратегией построения ОнС, а точнее – онтологических баз знаний.

Остановимся на рассмотрении компьютерной (формальной) онтологии предметной области, онтологии верхнего уровня и онтологии домена предметных областей. Последняя (в том числе) имеет важное значение для объединения (интеграции) концептуальных знаний близких предметных областей или реализации технологии системной интеграции междисциплинарных научных знаний. Мы также выделяем *начальную онтологию ПдО*, являющуюся инициализирующей доминантой при реализации технологии автоматизированного построения онтологии ПдО.

Ниже приведены известные определения понятия онтологии ПдО, начиная с первоначального определения Т. Груббера, сформулированного в [7], с последующим его уточнением: “Онтология – это формальная спецификация согласованной концептуализации”⁴, [8]. Другим важным определением онтологии ПдО является определение Н. Гуарино [90]: “Онтология – это формальная теория, ограничивающая возможные концептуализации мира”. На основе этих определений разные исследователи формулируют свои, частные определения онтологии ПдО – в соответствии с их конкретной областью профессиональных интересов⁵. Тематика и практическая направленность наших исследований (онтология как средство построения баз знаний междисциплинарных научных исследований) предопределяет следующее определение компьютерной онтологии ПдО.

Определение 4.1. Компьютерная онтология ПдО – это:

- 1) иерархическая структура конечного множества понятий, описывающих заданную предметную область;
- 2) структура представляет собой онтограф, вершинами которого являются понятия, а дугами – семантические отношения между ними;
- 3) понятия и отношения интерпретируются в соответствии с общезначимыми функциями интерпретации, взятыми из электронных источников знаний заданной ПдО;
- 4) определение понятий и отношений выполняется аксиомами и ограничениями области действия;
- 5) формально онтограф описывается на одном из языков описания онтологий;
- 6) функции интерпретации и аксиомы описаны в некоторой подходящей формальной теории.

В общем случае онтологию некоторой ПдО формально представляют упорядоченной тройкой [3, 5, 10, 90]:

$$O = \langle X, R, F \rangle, \quad (4.1)$$

где X, R, F – конечные множества соответственно: X – концептов (понятий, терминов) предметной области, R – отношений между ними, F – функций интерпретации X и/или R .

Рассмотрение граничных случаев множеств (4.1): $R = \emptyset$; $R \neq \emptyset$; $F = \emptyset$; $F \neq \emptyset$ во всех четырёх комбинациях значений R и F даёт различные варианты онтологических конструкций, начиная от простого словаря и таксономии до формальной структуры концептуальной базы знаний для высокоинтеллектуальных знание-ориентированных систем. Подробное рассмотрение различных указанных комбинаций с их содержательной интерпретацией выполнено в [91].

По своей функциональной полноте и степени формальности различают три вида онтологий: простая, полная (или строгая) и множество промежуточных или неполных онтологий⁶.

Простая онтология – это такая онтология, в которой $R = \emptyset$; $F = \emptyset$. Она служит (в основном) для однозначного восприятия научным сообществом понятий в соответствующей прикладной области.

Строгая или *полная онтология* ($R \neq \emptyset$; $F \neq \emptyset$) – это такая онтология, в которой множества концептов и концептуальных отношений максимально полные, а к функциям интерпретации добавляются аксиомы, определения и ограничения. При

⁴ Под термином “согласованность” имеется в виду общезначимость, общепринятость.

⁵ Таких определений известно несколько десятков.

⁶ Классификации онтологий по различным основаниям приведены в многочисленной литературе [4, 50, 91 и др.].

этом описания всех компонент представлены на некотором формальном языке, доступном для их интерпретации компьютером. Схема формальной модели полной онтологии описывается четвёркой:

$$O = \langle X, R, F, A(D, R_s) \rangle, \quad (4.2)$$

где X – множество концептов;

R – множество концептуальных отношений между ними;

$F : X \times R$ – конечное множество функций интерпретации, заданных на концептах и/или отношениях;

A – конечное множество аксиом, которые используются для записи всегда истинных высказываний (определений и ограничений);

D – множество дополнительных определений понятий;

R_s – множество ограничений, определяющих область действия понятийных структур.

Полная (компьютерная) онтология является (формальным) выражением концептуальных знаний о предметной области и по своей значимости сопоставима с базой знаний ЗОИС, а её построение является специфической формой творчества. Творческий процесс при этом можно представить совокупностью операций-процедур с суждениями, утверждениями, понятиями и отношениями между ними, а его результат – основой для построения составной части научной теории – онтологической базы знаний в заданной предметной области, описанной в декларативной форме.

Множество *промежуточных* или *неполных онтологий* ($R = \emptyset, F \neq \emptyset; R \neq \emptyset, F = \emptyset$) возникает, когда для каждого концепта (или их большей части) добавлены аксиомы и определения, представленные на ЕЯ.

Одним из распространённых вариантов неполной онтологии является структура вида $O = \langle X, R \rangle$, где множество F в явном виде отсутствует ($F = \emptyset$), в предположении, что концепты $x_j \in X$ общеизвестны (определены по умолчанию) либо (и) достаточно полно интерпретированы отношениями R .

Такая схема классификации по функциональному признаку согласуется с описанием [90]: “Онтология или концептуальная модель предметной области состоит из иерархии понятий предметной области, связей между ними и законов, которые действуют в рамках этой модели”.

Известно, что проектируемые средства информатики в соответствии с их проблемной ориентацией базируются на определённой совокупности фундаментальных принципов, методик и алгоритмов. Парадигма компьютерных онтологий, развиваемая во взаимодействии с методами и средствами системного анализа, положила начало развития новой ветви методов системного анализа ПдО – системно-онтологического анализа (подхода) [2].

Центральной идеей системно-онтологического подхода (СОП) является разработка онтологических средств поддержки решения прикладных задач – *полифункциональной онтологической системы*. Такая система (точнее, её концептуальная часть) описывается двойкой (4.3), включающей онтологию ПдО (состоит из онтологии объектов и онтологии процессов) и онтологию задач [92, 93].

$$O_{HC} = \langle O^{ПоО}(O^O, O^П), O^З \rangle \quad (4.3)$$

На рис. 4.1 представлена схема онтологий-компонентов предметной области и *проблемного пространства*. ПрП – это модель всех таких аспектов или компонент ПдО, с которыми связаны (опосредственно или непосредственно) знания, не-

обходимые для решения различных задач в этой ПдО. Всякое ПрП состоит из двух блоков: инвариантной (относительно неизменной) части и множества изменяемых частей, соответствующих отдельным задачам. В составе инвариантной части, например в методологии SMEE (Structured Methodology for Elicitation of Expertise), выделяют семь типов компонент: объекты, инструменты, операторы, операции, конечные продукты, побочные продукты и ограничения [98]. Эти типы компонент – суть понятия, которые хорошо группируются в онтологии объектов и процессов, представленных на рис. 4.1:

– O^O – онтология множества объектов (понятий, концептов) ПдО, которая рассматривается как иерархическая структура классов, подклассов и элементов классов;

– $O^П$ – онтология множества процессов ПдО, которая рассматривается как иерархическая структура процессов, подпроцессов, действий и операций;

– $O^З$ – онтология совокупности задач (типовых наборов), которые могут быть поставлены и решены в ПдО. Рассматривается как иерархическая структура задач, подзадач, процедур и операторов.

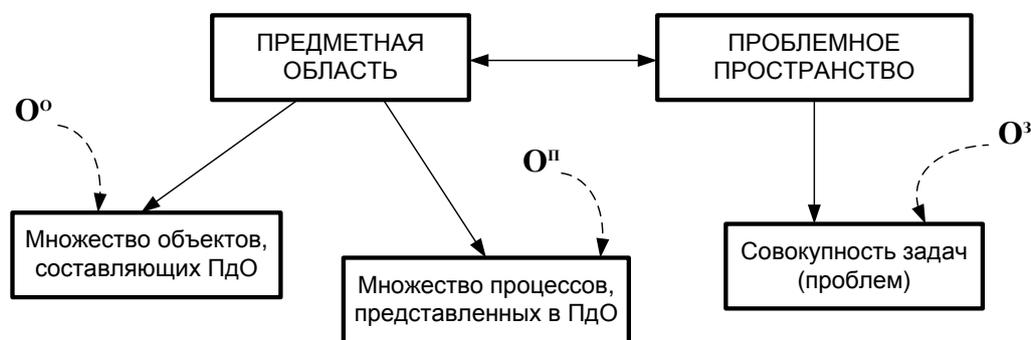


Рис. 4.1. Схема онтологий-компонентов предметной области

4.2. Лингвистические онтологии

Применение онтологического подхода для автоматической обработки текстов на естественном языке предполагает сопоставление понятиям онтологии предметной области (к которой принадлежит множество текстов) языковых выражений (слов и словосочетаний), которыми понятия могут быть выражены в текстах. Процедура сопоставления понятий и языковых выражений может быть реализована различными способами [91].

1. К понятиям уже разработанной онтологии ПдО приписываются языковые единицы. По сложности реализации этот способ является самым простым, но ему присущ ряд очевидных недостатков:

- а) имена понятий достаточно громоздкие, неестественные, с ними трудно оперировать как разработчикам, так и возможным пользователям;
- б) излишняя многозначность лексической единицы, возникающая из-за того, что одно и то же слово может соответствовать слишком большому количеству понятий в зависимости от контекста;
- в) онтологическая гипотеза Сепира-Уорфа (то, что не описывается словами, не может быть отражено в онтологии).

2. Установление соответствия между лексическими ресурсами (например, WordNet) и некоторой (концептуальной) онтологией. WordNet-ресурсы описывают лексические отношения между значениями слов, представленные в виде отдельных единиц в иерархической сети – синсетов. Отношения между лексическими единицами в значительной мере отражают отношения объектов внешнего мира, поэтому такие ресурсы часто рассматриваются как особый вид онтологий – лексические или *лингвистические онтологии*.

Главной характеристикой лингвистических онтологий является то, что они связаны со значениями языковых выражений (слов, именных групп и т. п.). Лингвистические онтологии охватывают большинство слов языка (в идеале – полный набор слов ЕЯ, структурированных в языковую картину мира) и одновременно имеют онтологическую структуру, проявляющуюся в отношениях между понятиями. Поэтому лингвистические онтологии могут рассматриваться как особый вид лексической базы данных и особый тип онтологий.

3. Разработка единого ресурса (смешанной онтологии), в котором были бы сбалансированы обе части: система понятий и система лексических значений, – что достигается разумным разделением этих единиц в создаваемом ресурсе и аккуратным описанием их взаимосвязей. Такой подход реализуется, например, в онтологии Mikrokosmos.

4.2.1. Сравнительный анализ онтологий

В настоящее время в мире известны несколько широко используемых лингвистических, концептуальных и смешанных онтологий, на основе которых разрабатываются информационные системы с обработкой знаний (представленных, в том числе, естественно-языковыми текстами) в различных предметных областях. Ниже приведены общие сведения о лингвистической онтологии WordNet (описания онтологий SUMO и Mikrokosmos представлены в Приложении А).

WordNet. Лингвистическая онтология для лексики английского языка разрабатывалась в середине 1980-х годов в лаборатории когнитологии Принстонского университета (штат Нью Джерси, США) под руководством проф. Дж. А. Миллера. Свободно распространяется в сети Интернет и представляет собой лексикографическую базу данных, структурированную в соответствии с лексико-семантическими отношениями: синонимии, антонимии, гипонимии (гиперонимии), меронимии и морфологическими отношениями [100, 101].

Важнейшим отношением WordNet является подобие значений. В основу понятия “синонимия” положено следующее: два слова являются синонимичными в контексте *S*, если замена одного на другое в *S* не изменяет его истинности.

Итак, синонимические ряды могут образовывать лишь слова одной части речи, поскольку синонимы должны быть взаимозаменяемыми (слова разных синтаксических категорий не могут заменять одно другого).

Антонимия – это лексическое отношение противоположности между формами слов (словоупотреблениями), а не семантическое отношение между значениями слов, т. е. различаются семантические отношения между словоупотреблениями и семантические отношения между значениями слов. Антонимия в WordNet является центральным организующим принципом для прилагательных и наречий.

В отличие от синонимии и антонимии, которые являются лексическими отношениями между формами слов, гипонимия и гиперонимия являются семантическими отношениями между значениями. Поскольку обычно понятие имеет только

один гипероним, оно порождает иерархическую семантическую структуру, в которой считается, что гипоним принадлежит гиперониму. Гипоним вбирает в себя все черты общего понятия и добавляет черты, которые отличают его от гиперонима и от других гипонимов этого же гиперонима (т. е. соблюдается принцип наследования признаков). Это центральный принцип организации существительных в системе WordNet.

Другим типом отношений является "часть-целое", которое в лингвистике известно как меронимия / голонимия. Понятие, представленное набором $\{x, x', \dots\}$, является меронимом понятия, представленного набором $\{y, y', \dots\}$, если носитель языка воспринимает предложение, сконструированное из такой основы, как: *У имеет X (как часть)* или же *X является частью Y*. Отношение меронимии может использоваться для построения иерархии частей (с некоторыми ограничениями, поскольку мероним может иметь несколько голонимов).

Морфологические отношения между формами слов являются важной частью лексических отношений. В системе WordNet сначала интерес ограничивался лишь семантическими отношениями: не было проектов, которые бы учитывали морфологические отношения. Тем не менее со временем программы совершенствовались и стало очевидным, что WordNet должен служить любым запросам, в том числе связанным с морфологическими характеристиками [102].

База данных WordNet разделена на *пять лексико-грамматических классов*: существительное, глагол, прилагательное, наречие и функциональные слова. Части речи организованы в иерархию синонимических множеств (узлов), называемых синсетам. Каждый синсет представляет одно базовое лексическое понятие и состоит из множества слов и устойчивых словосочетаний, равнозначных в некотором контексте. Синсеты связаны между собой отношениями различных типов.

Математической моделью WordNet служит граф $V = (X, R)$, множество вершин в котором разбито на два непересекающиеся подмножества: $X = X_1 \cup X_2$. Вершины из X_1 соответствуют словам и словосочетаниям, вершины из X_2 – их значениям (смыслам, толкованиям). Каждое из значений соотносится с одной из частей речи: существительным, глаголом, прилагательным или наречием. Множество рёбер также разбито на два непересекающиеся подмножества: $R = R_1 \cup R_2$. Рёбра из R_1 связывают слова со значениями, т. е. элементы из X_1 с элементами из X_2 . Эти рёбра представляют отношения, входящие в произведение $X_1 \times X_2$. Рёбра R_2 , принадлежащие второму подмножеству, связывают слова со словами и значения со значениями, т. е. представляют отношения, входящие в произведения $X_1 \times X_1$ и $X_2 \times X_2$ [4]. Объединение слов и словосочетаний в синсеты выражает отношение синонимии. Другие отношения (из перечисленных выше) задают типы рёбер из R_2 . В WordNet выделено *14 базовых типов таких отношений* (помимо них используются обратные отношения для каждого из перечисленных ниже типов): антоним; "имеет отношение к"; глагол, на основе которого образовано причастие; слово, на основе которого образовано наречие; действие, сопровождающее данное действие; глагольная группа; атрибут; "смотри также"; подобие; род-вид, вид-род; целое-часть, часть-целое; "сделан из", "служит субстанцией для"; множество-элемент, элемент-множество; цель-способ, способ-цель.

Технически, WordNet является электронным тезаурусом, определяющим широкий класс значений слов, связанных между собой семантическими указателями. Логическая структура WordNet показана на рис. 4.2.

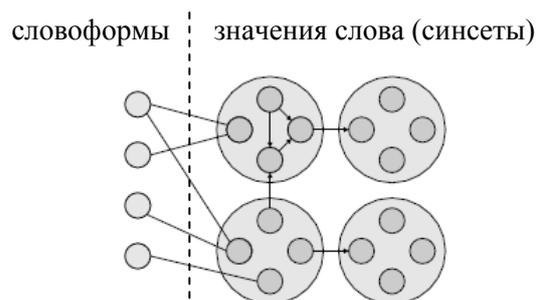


Рис. 4.2. Логическая структура WordNet

Рассмотрим пример записи синсета в базе данных.

**00047131 04 n 02 accession 0 addition 0 001 @
09536731 n 0000 | something added to what you
have already; "the librarian shelved the new accessions";
"he was a new addition to the staff"**

Первая часть записи утверждает, что число 00047131 является уникальным идентификатором синсета существительного {accession, addition}. Часть записи между символами "@" и "|" указывает, что этот синсет непосредственно подчинён синсету с идентификатором 09536731. Последний синсет соответствует значению "acquisition". И последняя часть записи (после символа "|") представляет толкование синсета и некоторые примеры употребления слов, входящих в синсет.

Онтология WordNet очень популярна в научных исследованиях по компьютерной обработке естественного языка и поиску информации в сети (Information Retrieval). Она постоянно обновляется. В настоящее время для Windows-пользователей доступна версия 2.1, для UNIX-подобных ОС – версия 3.0 и в ней существует более 82 тыс. синсетов существительных, более 18 тыс. синсетов прилагательных, более 3600 синсетов наречий и около 14 тыс. синсетов глаголов. Всего она включает более 147 тыс. английских слов и фраз). Другие аспекты описания WordNet и EuroWordNet и большое количество примеров их использования приведено в [91, 100, 102].

Известны и другие (смешанные и концептуальные) онтологии верхнего уровня *Sensus* [103], *Omega* [104], *Cyc* [105], *DOLCE* [106], *PropBank* [107, 108], *FrameNet* [109–111]. Некоторые данные, характеризующие упомянутые онтологии, приведены в таблице 4.1.

Анализ публикаций свидетельствует о том, что наибольший интерес (кроме лингвистической онтологии WordNet) для разработчиков представляют онтологии *Mikrokosmos*, *SUMO* и *Cyc* (онтографы первых двух представлены в расширенном варианте в Приложении А).

Верхний уровень онтологии *Cyc* разрабатывался в соответствии с общей концепцией онтологии – "представить все знания о мире". Он не имеет явного разграничения на общепринятые группы категорий "Материальное и Абстрактное" (M&A), "Продолжительное и Событийное" (C&O) и "Независимое, Относительное и Связанное" (I&R&O) и имеет свою специфическую концептуальную структуру. Кроме того, наиболее разработанная часть проекта *Cyc* является коммерческой и недоступной для общего пользования. Существует и доступная часть проекта онтологии (*OpenCyc*). В качестве положительных моментов *Cyc* можно отметить

наличие формального языка описания СусL, постоянно развивающиеся и доступные (в режиме клиент-сервер) базы знаний и обширную аксиоматизацию [105].

Для разработчиков онтологии ПдО, впервые реализующих такой проект, мы рекомендуем для адаптации одну из двух онтологий, онтографы которых представлены на рис. А.1 и рис. А.2.

Таблица 4.1

Основные характеристики онтологий

№ п/п	Наименование онтологии	Тип онтологии	Разработчик	Браузер	Кол-во концептов (лексем)	Формальная аксиоматизация	Интеракция с другими онтологиями
1	WordNet	Лингвистическая	Princeton University	WordNet	Около 100 тыс.	-	SUMO, FrameNet
2	Sensus	Смешанная	ISI USC	Ontosaurus	более 70 тыс.	-	-
3	Omega	Лингвистическая	ISI USC	Omega	около 120 тыс.	-	WordNet, Mikro-kosmos
4	Mikro-kosmos	Лингвистическая	CLR UNMS	Mikro-kosmos	более 7 тыс.	+	WordNet, Omega
5	OpenCyc	Концептуальная	Cycorp	Opencyc	более 100 тыс.	+	WordNet
6	DOLCE	Концептуальная	LAO ICST	DOLCE	около 4 тыс.	+	-
7	PropBank	Смешанная	University PennState	-	более 4300	-	FrameNet VerbNet
8	FrameNet	Смешанная	ISI, Berkeley, CA	Frame Grapher	около 900 фреймов	+	WordNet PropBank SUMO
9	SUMO	Концептуальная	Tekno-wledge Corporation, SUO WG	SUMO	более 1000	+	FrameNet, WordNet, EMELD

Онтология Mikrokosmos (рис. А.1) имеет достаточно гибкую иерархическую структуру с явным приоритетом группы категорий I&R&O. Две другие группы категорий распределены по ветвям иерархического дерева и не имеют чётко выраженного очертания. Онтология имеет практическую направленность, в которой не все ветви дерева разработаны одинаково. Поэтому, она может быть рекомендована для адаптации в тех предметных областях, понятия и термины в которых наиболее широко представлены в категориях и концептах Mikrokosmos. Информация об онтологии доступна как на веб-портале разработчиков, так и в многочисленных публикациях [113, 114 и др.].

Многие фрагменты онтологии SUMO (рис. А.2) представляют собой варианты адаптации различных ОВУ в одну онтологию верхнего уровня. В ней явно выражен приоритет группы категорий M&A с последующим преобладанием группы категорий C&O. Следует подчеркнуть (что также отмечено разработчиками SUMO), что ветвь онтологического дерева для категории *Материальное* разработана существенно глубже, чем ветвь для категории *Абстрактное*. К преимуществам SUMO можно отнести возможность трансляции описания онтологии на любой из основных языков представления знаний, наличие онтологии среднего уров-

ня, интегрированной с верхним уровнем SUMO, разработанные примеры практического применения, а также связь с WordNet.

Поэтому, данную онтологию можно рекомендовать для адаптации с онтологиями предметных областей, где явно преобладают материальные концепты.

4.2.2. Языково-онтологическая картина мира

Онтология ЯКМ отличается от представленных в табл. 4.1 ОВУ тем, что лингвистическая часть ориентируется на украинский или русский язык. Её рассмотрение было начато в главе 1 и, как определено в [11], основной задачей ЯКМ является фиксация содержания входной информации, причём информационные модели такой фиксации должны адекватно воссоздавать вычислительные процедуры в процессе интерпретации ЕЯТ.

Для решения этой задачи компьютерно-ориентированная ЯКМ должна отвечать следующим требованиям:

- системной полноты как в отношении множества лексем данного языка, так и в отношении формул их толкования;
- иметь в своем арсенале инструментарий для семантико-синтаксического анализа входного текста и, прежде всего, решение проблемы потенциальной многозначности выражения содержания входного текста;
- предоставлять возможности использования в качестве семиотической основы для фиксации содержания входной информации;
- выполнять функции языково-категориальной надстройки баз знаний в конкретных предметных областях.

Определение 4.2. ЯКМ – это развивающаяся, формализованная информационная (лексикографическая) система, которая обеспечивает семантическое толкование элементарных единиц естественного языка и их устоявшихся конструкций различной сложности в процессе анализа и интеграции входной информации, представленной естественным языком.

Как следует из определения, ЯКМ является смешанной онтологией, лингвистическая часть которой названа языково-онтологической картиной мира. Последняя является центральным компонентом языково-онтологической информационной системы. В ЯОКМ включены только общенаучные, общепотребительные (или энциклопедические) понятия, т. е. основные лексические единицы Толкового словаря. При этом множества языковых единиц частей речи структурируются отдельно, соответственно со своими совокупностями базовых отношений (для полных частей речи).

Одной из важнейших функций ЯОКМ считается эффективное формализованное представление синтаксических единиц естественного языка. Причём, формализмы должны быть такими, чтобы результаты представлялись в унифицированной, операционально ориентированной форме, доступной и удобной для компьютерных приложений.

Место онтологии ЯКМ находится ниже метаонтологии категорий верхнего уровня, но выше онтологий прикладных областей. В сущности, она является лингвистической и категориальной метаонтологией предметных знаний наряду с другими метаонтологиями общего характера. Отсюда одно из назначений ЯОКМ – служить цепью, которая связывает лингвосемантические отношения между разными предметными областями. Другим важным назначением является обеспечение

формального требования множественного наследования в онтологической иерархии *категория* (супертип) → *концепт* (тип) → *примитив*.

Развитие знание-ориентированных информационных систем однозначно связано с построением иерархических структур категорий верхнего уровня. Построение системы категорий [115] есть необходимый этап разработки интеллектуальных систем, основанных на знаниях, а также систем, которые обеспечивают автоматизацию понимания смысла текста. При этом существенным является построение языково-онтологической картины мира как основы информационного взаимодействия систем разной материальной природы [3]. С другой стороны, основательная научная разработка языковой картины мира и иерархии сем [32] предполагает построение модели семантических отношений в лексике как некоторого смыслового каркаса, на котором базируется глобальная “сферическая” сеть разнотипных семантических отношений. Смысловой каркас строится исходя из двух аспектов: гносеологического – категории познания (бытие, пространство, время, движение, отдельное, качество, количество, отношение) и онтологического – сферы бытия (человек, природа, общество).

Существующие схемы структуризации ЯОКМ, во-первых – страдают определенной субъективностью подхода к систематизации, структуризации и классификации понятий; во-вторых – неполнотой лексического континуума и отношений между представленными лексическими единицами (понимается, прежде всего, не их теоретический задел, а компьютерная обработка, а точнее – программное моделирование); в-третьих – они отделены от современных интеллектуальных информационных технологий и их приложений.

В данной работе:

– за основу взята иерархическая структура категорий верхнего уровня, синтезированная на логико-философских началах и описанная в главе 1;

– полнота лексического континуума обеспечивается воспроизведением в соответствующих таблицах лексикографической базы данных полного множества лексем Толкового словаря;

– исследование в своей конечной цели ориентировано на разработку аппаратно-программных средств поддержки (грамматического процессора) на современной микроэлектронной базе.

Различие между прикладной онтологией ПдО и общей базой знаний ПдО связано с целью и полнотой онтологии, которая является специфической базой знаний, описывающей только факты, принятые сообществом пользователей как истинные, на основании согласованных значений понятий заданного словаря. Общая база знаний может также описывать факты и утверждения, связанные со специфической конъюнктурой или специфическим эпистемологическим состоянием. Поэтому, в пределах общей базы знаний мы можем отличить два компонента: онтологию (содержащую состояние-независимую информацию) и “ядро” базы знаний (содержащее состояние-зависимую информацию).

Кроме того, следует уточнить интервал истинности состояние-независимой информации (или по другому – концептуальных знаний). Он должен быть как минимум равным “жизненному” циклу разрабатываемой ЗОИС.

ЯОКМ является основным исходным и подготовительным материалом для компьютерной обработки ЕЯТ и одним из главных блоков в языково-онтологической информационной системе. Его основной функцией является поддержка построения формализованного описания отображения входного ЕЯТ с воспроизведением полной акторной составляющей семантики, присутствующей в нём.

Такое отображение можно записать в виде:

$$G : C_o(EЯТ) \rightarrow C_A(EЯТ), \quad (4.4)$$

где $C_o(EЯТ) = \bigcup_{i=1}^N C_{ли}$, где $C_{ли}$ – формализованное описание i -ой компоненты

объектной составляющей семантики слов полнозначных частей речи, которые входят в исходный ЕЯТ, $i = \overline{1, N}$ – количество указанных слов; $C_A(EЯТ) = F_A(C_o)$ – формализованное описание акторной составляющей семантики исходного ЕЯТ.

Программно-аппаратная часть ЯОИС – суть семантический процессор (СП). Задачи пользователя в полном объёме решаются при взаимодействии лингвистического процессора (ЛП), СП и процессора интерпретации базы (мета) знаний и базы знаний конкретной предметной области (построение последней представляет отдельную самостоятельную задачу).

Обязательным условием реализации ЯОКМ является её формализованная компьютерная интерпретация (как программными, так и аппаратными средствами). Такую онтологию иногда называют наивной картиной мира. Знание об окружающей среде в ней исчерпываются системой понятий, сформулированных на некотором языке на уровне здравого смысла, связанных между собой максимально полной системой отношений, которые отображают окружающий мир со всем множеством его объектов и явлений, т. е. представляют собой лингвистическую проекцию бытия, в котором зафиксирован опыт взаимодействия человека с окружающей действительностью. ЯОКМ – составная часть прагматической модели языкового сознания, что является ключевым компонентом современных интеллектуальных ИС с естественно-языковым представлением, обработкой и актуализацией знаний.

Определение 4.3. ЯОКМ – это открытая, эксплицитно заданная на лексико-смысловом континууме лексикографическая система, в которой совокупность понятий формально обоснована и упорядочена в сложную иерархическую структуру по основным типам лексико-семантических отношений.

ЯОКМ как формальное описание общепринятой лексики представляется стандартной схемой

$$O = \langle X, R, F \rangle, \quad (4.5)$$

где X – конечное множество понятий, характеристик, ролей и атрибутов (или контент-слов), выраженных лексическими средствами ЕЯ, прежде всего такими полнозначными частями речи, как существительное, глагол, прилагательное и наречие. В логике они представляются, как правило, одноместными предикатами $P(x)$;

R – полная система отношений, такая как $P(x, y)$ или $P(x, y, z)$ (по утверждению Пирса, отношение с валентностями четыре и больше можно представить композицией двух- и трёхвалентных отношений);

F – множество функций интерпретации, заданных на X и/или R .

Графически ЯОКМ представляет некоторый гиперграф, который является результатом склеивания ациклических ориентированных графов лексических единиц для каждой полнозначной части речи. Служебные части речи учитываются на этапе снятия многозначности и логического представления исходного ЕЯТ.

Развитие абстрактной модели ЯОКМ

Уточним содержание понятия иерархического структурирования в онтологии ЯОКМ относительно лексического континуума естественного языка. Смысл любого высказывания, как таковой, возникает, когда текстовые компоненты вступают в

определенные действующие отношения (объект ↔ действие). Сфера действия определяется количеством объектов, которые она охватывает.

Такое рассмотрение очень упрощенно описывает структуру отображения (4.4) и больше ориентировано на описание объектной составляющей семантики. Очевидно, что для полного отображения построенных лингвистическим процессором синтаксических структур входного ЕЯТ, описания акторной составляющей текста, необходимо учитывать, прежде всего, базовые полнозначные части речи. В обобщенной схеме взаимодействия структурных компонентов ЯОКМ за основу примем предположение относительно представления в ней и семантической интерпретации соответствующих частей речи:

- объекты – как существительные;
- действия – как глаголы;
- атрибуты объектов – как прилагательные;
- атрибуты действий – как наречия.

Такая интерпретация хорошо согласовывается с известной концепцией WordNet, рассмотренной выше. Иерархическая структура ЯОКМ должна представлять собой некоторый гиперграф, как композицию четырех ациклических ориентированных графов иерархических структур существительного, глагола, прилагательного и наречия. Каждый из этих графов строится в соответствии со своими концептуальными и лингвистическими отношениями, присущими каждой из указанных частей речи (отображение семантических особенностей объединения существительного с предлогом учитывается в графе для существительного).

Согласно приведенному выше описанию на рис. 4.3 представлена абстрактная модель языково-онтологической картины мира. На нём приняты следующие сокращения:

– М, П, С, Г, Н, Ч – полнозначные части речи, соответственно местоимение, прилагательное, существительное, глагол, наречие и числительное;

– АО, О, Д, АД – элементы синтаксических структур, соответственно атрибут объекта, объект, действие и атрибут действия;

– $КГ_{АО}$, $КГ_{О}$, $КГ_{Д}$ та $КГ_{АД}$ – элементы семантических структур первой ступени, соответственно концептуальные графы атрибута объекта, объекта, действия и атрибута действия;

– $М^*$, $Ч^*$ – определённым образом упорядоченные морфологические таблицы для частей речи, соответственно местоимения и числительного, в исходном ЕЯТ;

$ЕЯТ^*$ – исходный естественно-языковой текст после интерпретации модулем обработки семантики первой ступени.

Относительно представления абстрактной модели ещё раз укажем, что модель морфологических структур ЕЯ является упрощенной и не раскрывает всех полнозначных частей речи и взаимоотношений между ними. Такое упрощение является справедливым, так как в этой модели сделан акцент именно на онтологическую составляющую.

Как было определено, онтологический граф не включает понятия нижнего уровня. Но формализованного критерия определения границы нижнего уровня не существует, и в каждом случае полностью зависит от интуиции и профессиональных взглядов исследователя.

Одним из этапов *расширения абстрактной модели* является включение в лексикографическую базу данных (ЛБД) дополнительных таблиц, которые включают как полнозначные части речи (причастие и деепричастие), так и служебные

части (предлог, союз и частица) и устойчивые словосочетания. При этом целесообразно таблицу деепричастий реализовать как подтаблицу в таблице “Глагол”, выделить безличную форму глагола в отдельную подтаблицу, таблицу “Существительное” разбить на три подтаблицы “Существительные_общие”, “Существительные_собственные имена” и “Существительные_отглагольные”, а также принять во внимание ряд других грамматических особенностей украинского (русского) языка.

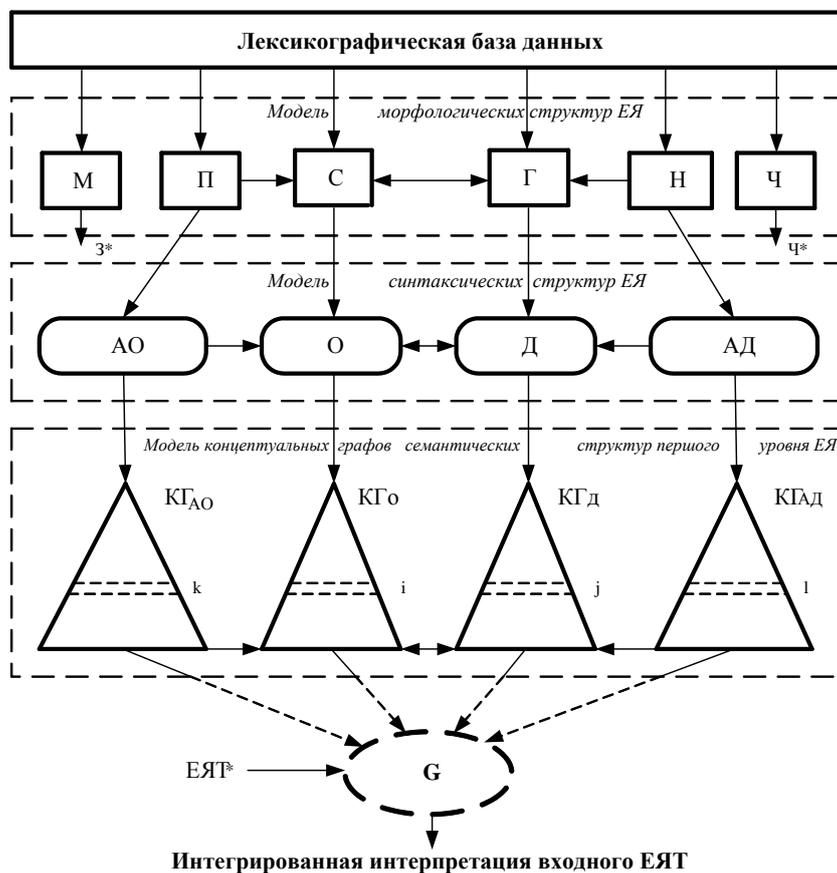


Рис. 4.3. Абстрактная модель ЯОКМ

Следующим этапом является расширение синтаксических структур естественного языка. Добавляются два блока, связанных с таблицами “Причастие” и “Деепричастие”, модели морфологических структур: “Атрибут атрибута объекта” – для причастия и “Атрибут атрибута действия” – для деепричастия. Кроме того, таблица “Предлог” связывается с синтаксической структурой “Объект”, а таблицы “Союз” и “Частица” включаются в фразовые синтаксические структуры для формирования более сложных конструкций.

Модель семантических структур первого уровня (или объектной семантики) сформированная из тех же самых КГ для объекта, действия, атрибута объекта и атрибута действия. При этом в обобщённом виде выполняется их структурирование.

Для КГ объекта – это существительные, которые обозначают общие понятия → существительные, которые обозначают сущности с признаком действия → су-

существительные, которые обозначают обобщённые объекты → существительные, которые обозначают конкретные объекты.

Для КГ действия – глаголы, которые выражают семантические темы, структурируются на верхних уровнях графа; другие глаголы структурируются на нижних уровнях графа.

КГ атрибута объекта разбивается на три подграфа, соответственно качественных, относительных и притяжательных атрибутов объекта.

КГ атрибута действия своими ветвями связывается с соответствующими группами семантических тем глаголов.

Формальное отображение представляет семантическую модель второго уровня (или акторной семантики) и включает подмножества логики первого порядка концептуальных графов, исчисления предикатов и языка представления знаний (KIF), спроецированных на исходный ЕЯТ.

Приведём в виде *Таблицы 4.2* надлежащим образом структурированные формальные основы применения правил, принципов, абстракций и отношений при развитии метаонтологии категорий верхнего уровня и ЯОКМ (отдельно для уровней концептов и примитивов) [5, 26–28, 38, 52, 116–118], и некоторые правила-рекомендации, которые необходимо учитывать при классификации концептов.

Семантические отношения нижнего уровня (для некоторого предложения ЕЯ) определяются на основе работы лингвистического анализатора и зависят от архива лингвистических знаний: словари словоформ для морфологии, грамматические правила для синтаксиса и концептуальные паттерны (или множества слотов фреймовых структур) для семантики. Они могут быть получены из лингвистических категорий и являются основными стандартными блоками для определения более сложных отношений в концептуальных графах, базах данных или экспертных системах.

Приведём некоторые правила-рекомендации, которых (в том числе) необходимо придерживаться при построении онтологии предметной области [52].

1. *Процедура перехода между соседними уровнями должна быть соизмеримой.* Это означает, что объединение объёмов или соединение значений концептов нижнего уровня должно составлять концепт верхнего уровня.
2. *Процедура перехода должна выполняться по одной основе* или характеристика, которая выбирается как основа деления, в ходе деления не должна подменяться другой характеристикой.
3. *Компоненты делимого концепта должны исключать друг друга.* Это означает, что их объёмы не должны содержать общих элементов или их значения не должны иметь общих частей.

Формальные основы структурирования онтологии

I. Метаонтология категорий верхнего уровня	
Виды абстракций	Классификация, обобщение, агрегация и ассоциация
Принципы классификации	Фундаментальные принципы дихотомии Аристотеля, трихотомии Пирса и решёток Лейбница, естественная классификация
Типы классификационного деления	Таксономические, мереологические, пространственные и временные
Наследование признаков:	Множественное
II. Языково-онтологическая картина мира	
А. Уровень концептов	
Концептно-ролевые отношения	Производят два репрезентативные примитива, соответственно названные присоединением и кореферентной связью. На естественный язык переводятся разными грамматическими формами глаголов <i>иметь</i> и <i>быть</i>
Семантические отношения	Классификационные – род-вид, целое-часть, класс-элемент, вышестоящий-нижестоящий, класс-подкласс; Признаковые – объект-атрибут, объект-действия; Количественные – иметь меру
Лингвистические отношения	Гипонимия, меронимия, синонимия и антонимия
Б. Уровень примитивов	
Предикативные отношения	Обобщенно-единичного характера, отдельно-единичного характера и единично-единичного характера
Семантически-ролевые отношения	Классификационные – иметь имя, быть эталоном; признаковые – иметь значение атрибута, иметь значение способа действия; количественные – иметь значение меры; сравнения – равно, сравнимо, больше, больше или равно, меньше, меньше или равно, несравнимое; принадлежности – простой принадлежности; простые временные – быть одновременно, быть раньше, быть позже, совпасть во времени, пересекаться во времени, быть всередине во времени, начинаться одновременно, заканчиваться во времени; простые пространственные – совпадать в пространстве, быть слева, быть справа, быть впереди, быть сзади, быть сверху, быть снизу, быть наискосок, пересекаться в пространстве, торкаться, размещаться на, размещаться в; каузальные – быть целью, быть мотивом, причина-следствие; инструментальные – служить для, быть средством для, оказывать содействие, быть инструментом, быть вспомогательным средством; информационные – быть отправителем, быть получателем, быть источником информации; порядковые – быть следующим, быть текущим, быть ближайшим; модальные – возможности, действительного воплощения, необходимости; модификационные; квантификационные – всеобщности, существования, уникальности, единичности, возможности, необходимости; корреляционные

4. Процедура перехода должна быть последовательной. Это означает, что от родового понятия необходимо переходить к видовым понятиям одного и того же уровня, а от целого к частям, частям частей и т. д.

Одним из возможных вариантов перевода ЕЯ-предложений в формальные структуры может быть перевод в концептуальные графы, рассмотренные в 3.3, так как они с самого начала разрабатывались для этих целей, обеспечивают читабельность, наглядный итерационный процесс выявления ошибок перевода и в то же время являются подмножеством логики первого порядка. Кроме того, концептуальные графы обладают возможностью формирования процедуры вложенности, которая отвечает цепочке синтаксических единиц “фраза → предложение → абзац → параграф → раздел → текст → том” или фрагментам действительности “элементарная ситуация → ситуация → сценарий”.

4.3. Системная интеграция лингвистических и предметных онтологий

Разработка и коммерческое использование баз знаний и соответствующего инструментария во многих прикладных областях (Knowledge based engineering systems) на основе онтолого-управляемых информационных систем во многих научно-исследовательских центрах и корпорациях привели к росту теоретических разработок формализованных методологий проектирования онтологических структур. При этом сущность указанных методологий сводилась к формальному обоснованию структурирования иерархического дерева онтологии (формализованного построения наборов концептов и связывающих их концептуальных отношений или категоризации) и разработке формальных языков представления знаний, которые описывают аксиоматизацию концептов предметной области. Следующим шагом развития теории баз знаний являлась необходимость теоретически обоснованного объединения (или *системной интеграции*) уже разработанных как общедоступных онтологий, так и коммерческих баз знаний для разнообразных прикладных задач, проблем, целых предметных областей и междисциплинарных знаний широкого назначения.

В общем виде процесс, обеспечивающий системную интеграцию множества онтологий, можно описать следующей формулой

$$O = \prod_i^U O_i, i = \overline{1, N}, \quad (4.6)$$

где \prod_i – знак концептуального объединения. Смысл этого знака состоит в системной интеграции исходных онтологических графов с учётом областей определений $O_i (i = \overline{1, N})$ и их взаимосвязи (взаимодействия).

Обобщённая архитектура знание-ориентированной системы, эффективно реализующей процедуры обработки предметных знаний, развития БЗ и системной интеграции, представлена на рис. 4.4.

Реализация технологий представления и обработки знаний и процесса системной интеграции знаний предполагает учёт различных формально-методологических требований, критериев и оценок. Приведём основные из них.

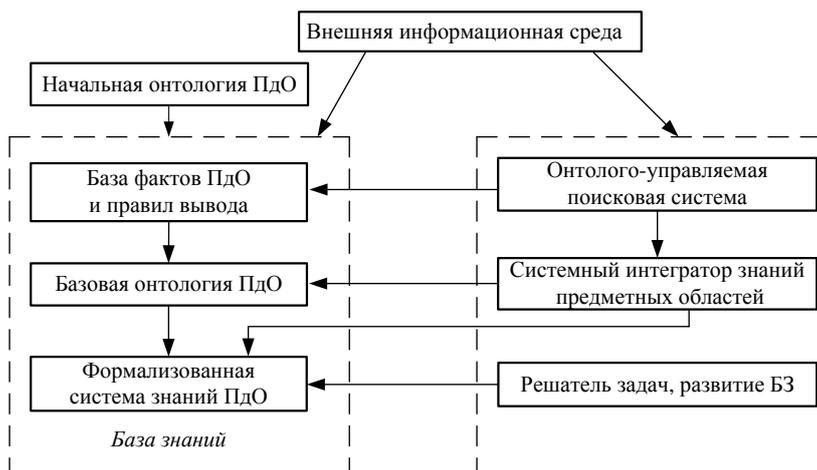


Рис. 4.4. Архитектура развивающихся знание-ориентированных систем

1. Сущность методов онтологического исследования как раздела системного анализа предполагает:
 - а) построение статических и динамических моделей;
 - б) исследование структуры ПдО и взаимосвязи компонент знаний;
 - в) исследование поведения системы, описывающей заданную ПдО – разработка алгоритмов процедур композиции знаний и диаграмм состояний.
2. Влияние предметной области на вид онтологии.
3. Правила формирования достоверных утверждений и выводов, описывающих систему и её поведение.
4. Категориальный уровень онтологии ПдО и соответствующая ему система отношений.
5. Характеристики онтографа (средний коэффициент ветвления i -ой вершины, количество уровней ветвления, количество типов отношений, в том числе по подтипам).

Познание идёт от изучения вещей к изучению процессов (в том числе взаимодействия компонент, включая уровень “субъект–объект”), от устойчиво функционирующей системы – к системе *изменяющейся* и *развивающейся*, а результатом, следствием процесса развития является возникновение *нового качества* [15].

4.3.1. Формализованные аспекты интеграции и её оценка

Объём знаний W в предметных областях можно оценить через характеристики (параметры) их формально-онтологических представлений. В частности, при представлении онтологическим графом (без учёта типов отношений и сложности функций интерпретации) величина W может характеризоваться числом вершин ОГ. В случае простой древовидной структуры это число может быть выражено формулой

$$W = \sum_i \sum_h \sum_l O_i \cdot S_{h,l} \quad (4.7)$$

где O_i – онтограф i -ой предметной области, $i = \overline{1, N}$; $S_{h,l}$ – степень вершины, равная числу исходящих из неё рёбер, $h = \overline{1, H}$ – количество уровней ОГ, $l = \overline{1, L_h}$ – номер вершины на соответствующем (h -ом) уровне ОГ.

При равномерной плотности распределения ОГ, т. е. при $S_{h,l} = S$ ($h, l = 1, 2, \dots$) (4.7) сводится к известной формуле суммы геометрической прогрессии

$$W = \sum_i O_i \frac{1-S^h}{1-S} \quad (4.8)$$

Учёт типов отношений и сложность функций интерпретации приводит к ОГ со взвешенными вершинами и ребрами. Выражение (4.8) при этом преобразуется в вид

$$W = \sum_i \sum_h O_i \cdot \left(\alpha_l + \sum_j \beta_{l,j} \right), \quad (4.9)$$

где α_l и $\beta_{l,j}$ – значения весовых функций соответствующих отношений и функций интерпретации, приписанные вершинам (α_l) и ребрам ($\beta_{l,j}$) ОГ. Выражение

(4.9) даёт полную оценку сложности ОГ, а отношение $\omega = W^O / W$ характеризует среднюю плотность взвешенного ОГ.

Рассмотренные оценки позволяют сравнивать различные варианты представления знаний о предметных областях, а также отслеживать процесс эволюции научных теорий.

Процесс развития знаний в любой ПдО связан с её анализом, концептуализацией и построением формальной теории. При этом формализация в общем случае относится к четырём основным видам представления информации

$$I = I(V, A, T, G), \quad (4.10)$$

т. е. к вербальному (V), аналитическому (A), табличному (T) и графическому (G). Между ними существует взаимно-однозначное соответствие, не всегда на практике реализуемое строго и полностью. Поэтому все они находят своё, вполне определённое место при описании научной теории. Во многих случаях корректным оказывается их ограничение до двух: вербальный и аналитический.

Как правило, процесс развития теорий сопровождается перераспределением объёма информации о предметной области между вербальной и формальной компонентами, т. е. между естественно-языковым описанием предмета исследования и формально-аналитическим A (формульным Φ , табличным T , графическим G представлением существа теории). Очевидно, что формализованное представление является более компактным, а главное, – более строгим и пригодным для компьютерной обработки.

4.3.2. Практическая интеграция онтологий

Накопившийся опыт практического использования SUMO и других онтологий (в первую очередь WordNet и FrameNet) показал необходимость в объединении двух и более онтологий (построении взаимных отображений концептов) в единую

концептуальную структуру. Два вида онтологий (лингвистическая и концептуальная) отражают две стороны концептуализации реального мира. Если в WordNet концептуализация отображается в терминах естественного языка, то в SUMO концепты организованы в логическую структуру. При этом объединение предполагает построение отображения между соответствующими узлами различных онтологий.

Отображение SUMO ↔ WordNet. Рабочая группа SUO WG разработала такое отображение. Оно сопоставляет каждому синсету в базе данных WordNet путём теггирования соответствующего концепта в SUMO. При этом вид отношения между синсетами WordNet и концептами SUMO могут быть: синсет эквивалентен концепту; синсет является более высшим классом, чем концепт; и синсет является элементом концепта. Разработанные файлы отображения позволяют соотнести слова естественного языка в термины SUMO, используя синсеты WordNet как промежуточный уровень [119].

Приведём пример формализованной записи для первого вида отношения между синсетами WordNet и концептами SUMO:

00008864 03 n 03 plant 0 flora 0 plant_life 0 027
@ . . . | a living organism lacking the power of
locomotion &%Plant=

Префикс ‘&%’ указывает на то, что термин взят из онтологии SUMO, а суффикс ‘=’ указывает, что отношение отображения – синонимия.

Мотивацией для построения таких отображений может быть следующее.

Отображения могут функционировать как индексы естественного языка к концептам в онтологии, как мост между этими структурированными концептами и нетекстовой лингвистической структурой. Разработчики создали специальный инструмент и встроили его в SUMO-браузер, который позволяет пользователю задавать термины естественного языка и видеть на экране компьютера соответствующие концепты онтологии SUMO с соответствующей формальной аксиоматизацией. Это позволит значительно упростить процесс инжиниринга знаний и моделирования данных. Кроме того, отображения могут служить важным источником знаний для применений NLP, в частности при реферировании текстов и семантическом поиске документов в сети. И ещё, с помощью соответствующего инструментария можно проверить на полноту степень завершённости проектируемой онтологии.

Отображение SUMO ↔ FrameNet. Онтологии SUMO и FrameNet являются относительно сформировавшимися средствами, но представляется целесообразным объединить их сильные стороны для существенного упрощения автоматической обработки текстов NLP. В частности, NLP-применения, использующие FrameNet, требуют знаний о возможных заполнителях элементов фрейма (FEs). Например, семантическому фреймовому анализатору необходимо знать, является ли определённый фрагмент текста (или именованная сущность) строгим заполнителем для FE, т. е. ему необходимо проверить, является ли тип заполнителя FE совместимым с типом именованной сущности. Поэтому в FrameNet представлено около 40 семантических типов (FT) как ограничения на заполнители FEs и имеющие соответствия с SUMO классами как FTs [120]. Фрагмент связывания SUMO и FrameNet представлен на рис. 4.5.

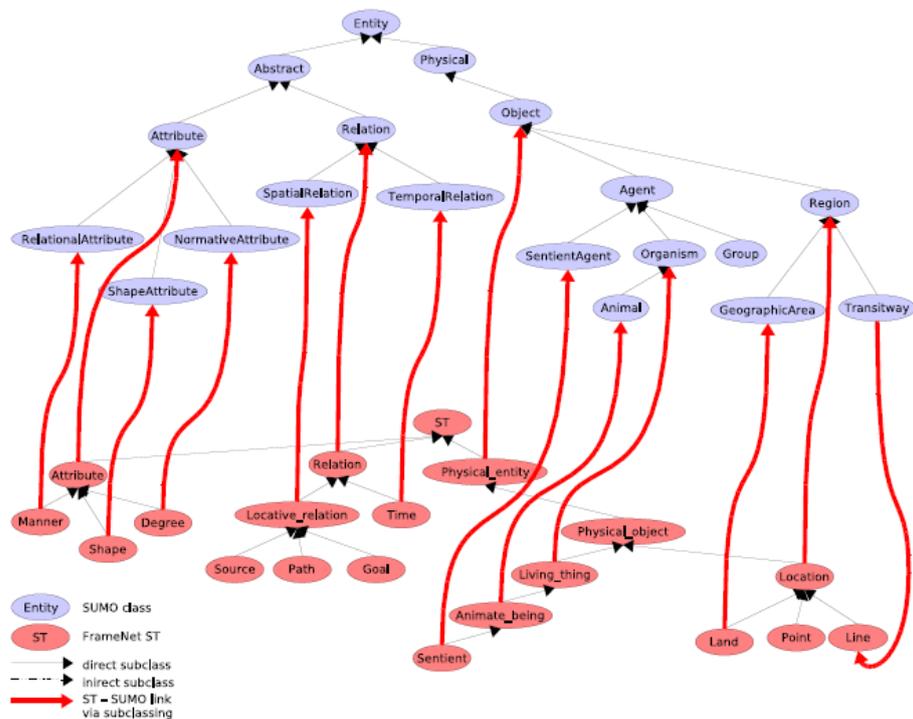


Рис. 4.5. Фрагмент связывания SUMO и FrameNet

Следует отметить, что существуют и другие проекты связывания лексикон-онтологий [121]. Но рассмотренное связывание SUMO и FrameNet имеет ряд преимуществ. Дело в том, что FrameNet, в отличие от WordNet, моделирует семантические и синтаксические валентности, включая высококачественные аннотации. Фреймовые семантики естественно представляют кросс-лингвистические абстракции и нормализацию парафраз. Что касается SUMO, то она значительно шире DOLCE, а в отличие от Сус является открытой для доступа.

При интеграции знаний следует учитывать прикладные аспекты. Принципиально различают области:

- а) взаимодействия с пользователем – когнитивизация представления, поиск информации, создание начальной онтологии и начальной системы знаний;
- б) развития системы знаний на основе имеющихся, генерация новых знаний.

4.4. Методология разработки онтологии ПдО

4.4.1. Общий подход к проектированию

Известна методология структурного анализа и проектирования (SADT) сложных систем в произвольной предметной области [122]. Эта методология породила семейство методик (и соответствующих стандартов) IDEF, ориентированных на разработку моделей ПдО и акцентирующих внимание на каком-то конкретном аспекте проектирования [123, 124]. В частности:

– методика IDEF0 рекомендована для содержательного анализа и функционального проектирования сложных систем управления, в том числе и программного обеспечения. Описание объектов и процессов в методике выполняется в виде иерархической совокупности диаграмм с лаконичным описанием функций. Блоки на диаграммах выражают функции, поэтому их названия – глаголы или отглагольные существительные;

– методика IDEF1x предназначена для информационного моделирования, основана на концепции “сущность-связь”. Обычно отправным пунктом для разработки информационной модели является IDEF0-модель;

– методика IDEF3 описывает поведенческие аспекты конкретных приложений, рассматривает последовательность выполнения и причинно-следственные связи между ситуациями и событиями для структурного представления знаний о ПдО. Если IDEF0 связана с функциональными аспектами и отвечает на вопросы “Что делает система?”, то в IDEF3 детализируются IDEF0-функции. Эта модель отвечает на вопросы “Как система это делает?”;

– методика IDEF5 предназначена для онтологического анализа ПдО, анализа основных терминов и понятий, используемых для описания объектов и процессов, границ использования, а также взаимосвязей между ними. Служит для эффективного исследования и документирования: словаря терминов, используемых при описании характеристик объектов и процессов, имеющих отношение к рассматриваемой ПдО, точных и однозначных определений всех терминов и классификации логических взаимосвязей между ними.

Для каждой из методик семейства IDEF разработаны этапы и стадии построения модели ПдО, языки и диаграммы представления результатов. Анализ получаемых в результате применения методик IDEF моделей ПдО и их описаний с точки зрения формализации и компьютерной обработки показал, что, по сути, они (функциональные модели и описания) представляют собой содержательное техническое задание (ТЗ) на проектирование ОнС, включающей онтологию объектов (сущностей), онтологию процессов и онтологию задач. Описание же самой ОнС имеет несколько иные цели. Оно ориентировано на компьютерное представление с помощью стандартизованных языков предметных знаний с целью широкого использования сообществом пользователей.

Итак: 1) методики IDEF и системно-онтологический подход используют одни и те же множества сущностей ПдО и ПрП, акцентируя внимание на различных совокупностях характеристик и атрибутов; 2) конечные цели этих двух подходов отличаются: для IDEF – функционально-блочная модель, а для системно-онтологического подхода – онтолого-содержательная модель.

Выполненный выше анализ позволяет утверждать, что методики IDEF и СОП решают разные задачи (с определённым “перекрытием” некоторых функций). Весьма очевидно это проявляется для пары “онтология процессов ↔ поведенческая модель ПдО”.

Очевидно, этапы проектирования онтологических систем исходят из принятых в методологии SADT фаз проектирования сложных систем: 1) анализ – определение того, что система будет делать; 2) проектирование – определение подсистем и их взаимодействие; 3) реализация – разработка подсистем по отдельности, объединение – соединение подсистем в единое целое; 4) тестирование – проверка работы системы; 5) установка – введение системы в действие; 6) функционирование – использование системы.

Предварительный анализ предметной области

Во все методологии включён этап предварительного анализа ПдО или составления содержательного ТЗ на проектирование [2, 3, 5, 122–127]. Этот этап (как и процесс проектирования базы знаний ПдО в целом) носит сложный аналитический характер и заключается в многократном абстрагировании, в результате которого из всего многообразия сторон и свойств сущностей предметной области выделяются наиболее существенные, релевантные конкретным задачам. Знание предметной области, понимание сути происходящих в ней процессов, законов, правил и ограничений, управляющих её развитием, является необходимым условием успешного решения задач, стоящих перед исследователем. Более того, наличие таких знаний является необходимым условием постановки, формулировки этих задач, без чего невозможно само решение [41]. Онтологические системы призваны сделать знания коллективным достоянием широкого круга лиц, дать мощный инструмент для фиксации, приобретения и обработки знаний, проверки их на непротиворечивость, полноту и т. п. Кроме того, составляется систематизированное представление знаний о ПдО, выявляются источники формирования элементов множеств и процедур, задач, выполняемых в анализируемой ПдО. Составляется и документируется словарь терминов ПдО.

Этап предварительного анализа ПдО включает:

- обоснованный выбор точного (и достаточного) фрагмента ПрП, относительно которого будут ставиться и решаться задачи пользователя;
- выбор методов и процедур системно-онтологического анализа, которыми, в частности, могут быть абстрагирование и конкретизация, композиция и декомпозиция, структурирование, кластеризация и классификация, тестирование и верификация;
- составление детального словаря терминов и его разбиение на подмножества терминов-объектов, терминов-процессов и терминов, именующих задачи и методы.

Если предметная область (и проблемное пространство) представляет сложную систему, то следует рассмотреть вопрос о предварительном этапе проектирования на основе методик IDEF, которые дополняют описанные выше шаги проектирования. Тем более что известные инструментальные средства анализа ПдО и построения баз знаний (например, KADS-системы, SIMER+MIR) по ряду причин не могут быть использованы [3].

Как правило, методика сводится к алгоритму, который носит итеративный характер. Для процесса разработки необходимо предусмотреть ряд “контрольных точек” для проверки полученных результатов на соответствие выбранным критериям. Указанные критерии должны соотноситься с заданными критериями на проектирование базы знаний ПдО, так как создание последней является целью для разработчиков. Оптимальный результат, как правило, зависит от степени проработки предполагаемых приложений и вариантов использования онтологии.

4.4.2. Онтология объектов ПдО

Определение 4.4. Под *онтологией объектов* предметной области понимается четвёрка [2]:

$$O^o = \langle X, R, F, A(D, R_s) \rangle, \quad (4.11)$$

где $X = \{x_1, x_2, \dots, x_i, \dots, x_n\}$, $i = \overline{1, n}$, $n = \text{Card } X$ – конечное множество концептов (понятий-объектов) заданной ПдО;

$R = \{R_1, R_2, \dots, R_k, \dots, R_m\}$, $R \subseteq X_1 \times X_2 \times \dots \times X_n$, $k = \overline{1, m}$, $m = \text{Card } R$, –

конечное множество семантически значимых отношений между понятиями-объектами ПдО. Они определяют тип взаимосвязи между понятиями. В общем случае, отношения делят на *общезначимые* (из которых выделяют, как правило, отношения частичного порядка) и *конкретные отношения* заданной ПдО;

$F : X \times R$ – конечное множество функций интерпретации, заданных на понятиях-объектах и/или отношениях;

A – конечное множество аксиом, которое состоит из множества определений D_i^l и множества ограничений Rs_i^l для понятия X_i . Определения записываются в виде тождественно истинных высказываний, которые могут быть взяты, в частности, из толковых словарей ПдО. В них могут быть указаны дополнительные взаимосвязи понятий X_i с понятиями X_j . В множестве ограничений Rs_i могут быть заданы ограничения на интерпретацию соответствующих понятий X_i .

Онтология определяет общеупотребительные, семантически значимые “понятийные единицы знаний”, которыми оперируют исследователи и разработчики знание-ориентированных информационных систем. Она отделяет “статические” и “динамические” компоненты знаний ПдО от операциональных знаний. В отличие от знаний, закодированных в алгоритмах, онтология обеспечивает их унифицированное и многократное использование разными исследовательскими группами, на разных компьютерных платформах при решении разных задач.

Построение компонент онтологии

Наиболее существенным компонентом концептуальной модели ПдО является множество понятий заданной предметной области. Напомним некоторые утверждения, непосредственно относящиеся к построению онтографа ПдО [128]⁷.

Все понятия (или концепты) делятся на ряд классов (по семантической зависимости).

1. В зависимости от отображения вида или рода предметов – на *видовые* и *родовые* понятия.

2. В зависимости от отображения части или целого предметов – на *понятия-части* и *понятия-целые*.

3. В зависимости от количества отображаемых предметов – на *единичные* и *общие* понятия.

4. В зависимости от отображения предмета или свойства, абстрагированного от предмета, – на *конкретные* понятия и *абстрактные* понятия.

Онтология ПдО – это концептуальная модель реального мира и её понятия должны отражать эту реальность.

Построение множества X считается наиболее важным моментом при разработке онтологии ПдО. Оно должно быть обязательно не пустым. Соотношения между $\text{Card } X$, $\text{Card } R$ и $\text{Card } F$ характеризуют онтологию по функциональному признаку.

⁷ Описание и определение категории “понятие” рассмотрено в главе 1

Для хорошо проработанных предметных областей за основу множества концептов $\{x_i\}$ может быть взято содержимое подходящих толковых словарей. В противном случае следует составить полный список терминов, в котором указать (причём пересечение объёмов и содержания понятий в таком предварительном списке не существенно):

- чем является каждый термин – понятием-классом предметов или конкретным понятием;
- указать для каждого термина возможные существенные отношения с другими терминами из списка;
- описать возможные существенные свойства понятий.

Известно, что в любой предметной области существуют термины-синонимы. Для них в онтологии отводится только одно понятие, в аксиомах которого может быть указан синонимический ряд терминов. Другими словами, синонимы одного и того же понятия не представляют различные классы.

Далее следует уточнить и определить окончательный список классов-понятий, имена которых будут входить в разрабатываемую онтологию и являться вершинами онтографа. Также следует принять единые *правила присваивания имён* понятиям и их свойствам.

Затем, возможно, следует повторить некоторые фрагменты процесса анализа ПдО (с привязкой к составленному списку понятий), выполненные на предварительном этапе. Отметим, что в число указанных выше “контрольных точек” (точек входа итерации) должно быть включено завершение разработки любого компонента онтологии.

В результате должен быть получен полный список существенных для заданной ПдО (и предполагаемых приложений) понятий и их машинно-интерпретируемые формулировки.

Построение множества R также основано на результатах предварительного этапа анализа ПдО. По сути, требуется установить между элементами $x_i \in X$ семантические k -арные отношения $R_k \subseteq X^k$. Другими словами, необходимо построить множество дуг, связывающих узлы *направленного онтографа*. В качестве узлов онтографа выступает множество понятий ПдО. Вершиной (или вершинами) онтографа является родовое понятие, которое не имеет надкласса, а самый нижний уровень представляют конкретные понятия (примитивы), т. е. не имеющие видовых понятий в заданной ПдО.

На практике множество R на начальном этапе представляют некоторым обобщённым отношением “*выше – ниже*”. Известно несколько подходов для разработки иерархии классов: процесс *нисходящей* разработки, процесс *восходящей* разработки и *комбинированный* процесс разработки. Последний наиболее часто используется разработчиками, так как он является более естественным, сначала оперирует понятиями среднего уровня, к которым наиболее часто обращаются разработчики. Затем эти понятия обобщаются и ограничиваются.

В заключение данного подэтапа следует соотнести разработанные классы и их иерархии с результатами предварительного анализа ПдО. В частности, уточняются зависимости для конкретных пар (x_i, x_j) . В процессе соотнесения (и построения иерархии) следует учитывать, что [89]:

- прямые подклассы в иерархии должны располагаться на одном уровне обобщения;

- класс может быть подклассом нескольких классов, и тогда он может наследовать свойства от всех этих классов;
- если класс имеет только один прямой подкласс, то, возможно, при моделировании допущена ошибка или онтология неполная;
- если у данного класса есть более дюжины (иногда говорят о числе 7) подклассов, то, возможно, необходимы дополнительные промежуточные классы;
- в онтологии число классов соотносится с числом предполагаемых приложений.

И следует помнить, что не существует единственно правильной иерархии классов.

Описанное построение онтографа является специальным видом классификации понятий ПдО – *онтологической классификацией*.

Построение множеств F и A . В зависимости от функциональной ориентации проектируемой онтологии множества F и A могут интерпретироваться по-разному [92, 125–127 и др.]:

1) $A \equiv F$ – множество аксиом тождественно множеству функций интерпретации. В этом случае устанавливаются существенные связи между разрабатываемыми компонентами онтологии и вариантами её использования. Основным назначением такой онтологии является однозначная интерпретация понятий, входящих в онтологию, сообществом пользователей;

2) – множество аксиом $A \subseteq F$ не тождественно множеству функций интерпретации. В аксиомах задаются а) базовые функции (подмножество F) либо б) дополнительные отношения (не являющиеся элементами множества R) между понятиями, ограничения и условия, которые анализируются в машине вывода ОнС и используются в процессе решения задач;

3) – множество аксиом $A \supseteq F$ не тождественно множеству F . Функции интерпретации рассматриваются как специальный вид отношений на множестве понятий $F : x_1 \times x_2 \times \dots \times x_{n-1} \Rightarrow x_n$. В этом случае устанавливаются существенные связи между уже разработанными компонентами онтологии и совокупностью задач предполагаемого приложения (приложений). Онтологии с таким представлением F используются в вопросно-ответных системах, в которых результатом является одно из значений двухэлементного множества {истина, ложь} или имя предиката.

В конечном счёте, независимо от того, какая из этих формулировок будет принята, эффективность разработанной онтологии будет определяться конечными результатами приложений.

Кроме того, из полного списка отобранных в онтологию терминов не все представляют понятия. Существуют термины (например, ролевые), которые соответствуют свойствам определённых классов-понятий. Такие свойства следует привязать к описанию самого общего класса, обладающего ими. А подклассы этого класса будут наследовать указанное свойство (конечно, если между ними установлено некоторое отношение частичного порядка).

Свойства понятий имеют определённые значения, такие как *тип значений*, *мощность значений*, *разрешённые значения* (для данного класса) и другие. Например, значения бывают с единичной мощностью, мощностью без ограничений и мощностью с некоторым допустимым интервалом.

На основе построенных множеств кортежа можно синтезировать концептуальную модель ПдО, например, с помощью известного инструментального сред-

ства Protégé и получить формальное описание разработанной онтологии на одном из языков описания онтологий, а также графическое представление онтографа.

4.4.3. Онтология процессов ПдО

Синонимами онтологий объектов и процессов являются соответственно статическая и динамическая онтологии ПдО. В научно-технической литературе, когда говорят об онтологии ПдО, то подразумевается её статическая составляющая. Именно компоненты последней наиболее разработаны, как в литературе по философии, так и в конкретных описаниях ряда предметных областей. Поведенческое описание сущностей-процессов чаще всего выполняется в виде графических диаграмм и естественно-языковых описаний. Разработка же базы знаний не является прямой целью указанных методик. Поэтому методики разработки онтологии процессов практически неизвестны, хотя в некоторых известных онтологиях верхнего уровня [9, 83, 114, 129] сущность понятия “Процесс” рассмотрена достаточно детально.

На рис. 4.6 представлен синтезированный онтограф, который представляет схему начального участка понятия “Процесс”, а именно той её части, которая соответствует процессам в научно-технических предметных областях (ветви онтографа “Социальный процесс”, “Материальный процесс” и им аналогичные не рассматриваются).

Категория *Процесс* рассматривается как *Действительность* и *Событийность*, в отличие от категории *Объект*, характеризующийся как *Действительность* и *Продолжительность* [27]. В первую очередь *Процесс* рассматривается как зависимая от времени категория и затем подразделяется по видам изменений, наличием начальных и конечных точек и т. д. Далее *Процесс* подразделяется на непрерывный и дискретный. Первый из них характеризуется наличием эксплицитных начальной и конечной точек или без явного указания этих точек. Второй вид процесса указывает, что изменения происходят дискретными шагами, названными *событиями*, которые чередуются с периодами покоя, названными *состояниями*.

Приведенная схема начального развития онтологии процессов не отражает всех характеристик (оснований ветвлений в онтографе) категории “Процесс”, даже для той её части, что представлена на рис. 4.6.

На рис. 4.7 представлена общая схема онтологии процесса ПдО, в которой категория “Процесс” представлена онтографом с p уровнями и n_p подпроцессами (Пп) на каждом уровне. Предпоследний ($p-1$) уровень представлен множеством действий (Д), на которые разбивается каждый Пп предыдущего уровня. В свою очередь, каждое действие на последнем (p -ом) уровне разбивается на последовательность операций $O_{i_j}^p$.

Связи между подпроцессами для соседних уровней соответствуют отношениям “Целое-часть”, а внутри каждого уровня – некоторой смешанной формой организации соединений. На рис. 4.7 показан частный случай такой организации – параллельный. Дальнейшее развитие (конкретизация) онтологии процессов возможно, когда задана конкретная предметная область и соответствующее проблемное пространство, а в более узком смысле – конкретные признаки ветвлений (условия инициирования Пп, условия окончания Пп и ограничения) в онтографе.

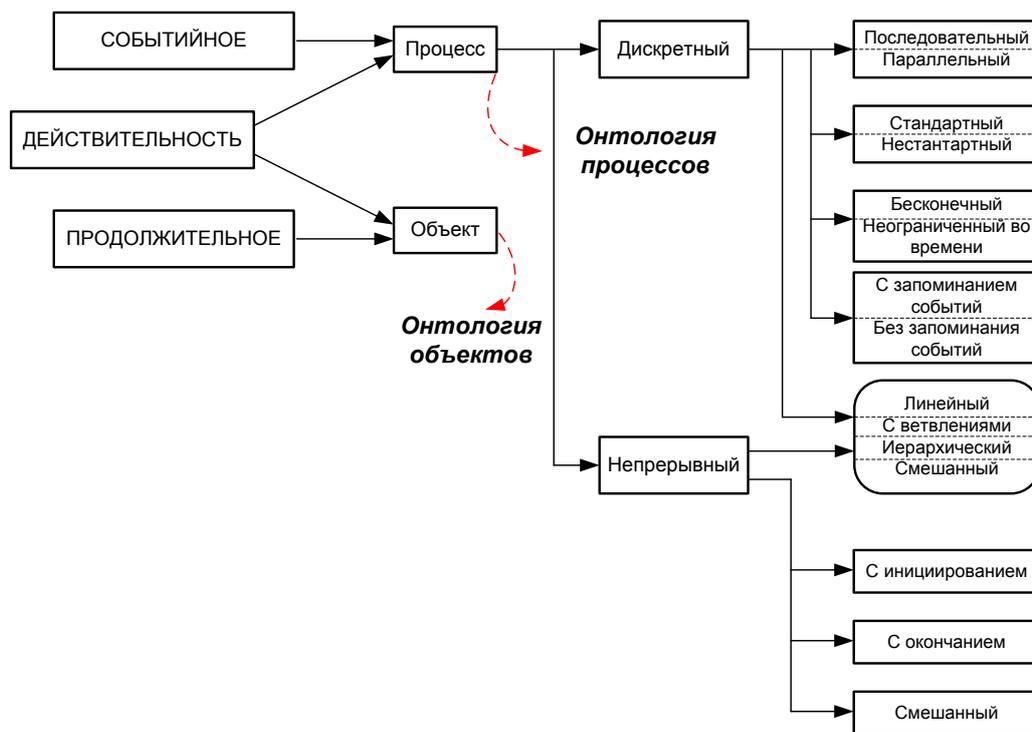


Рис. 4.6. Начальный участок онтологии процессов

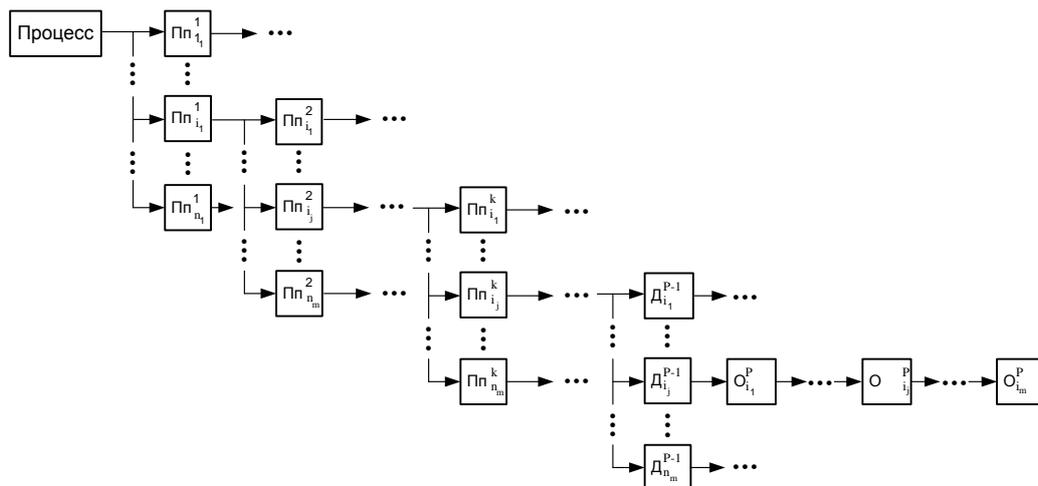


Рис. 4.7. Общая схема онтологии процесса

4.4.4. Онтология задач проблемного пространства ПДО

Из предметно-проблемных знаний в общем случае выделяют понятия и процессы ПДО, понятия и классы задач ПрП, а также методы решения последних. Сюда же относят и алгоритмы, реализующие соответствующие методы. Перечисленные типы знаний по темпоральному признаку делятся на статические и динамические, а по онтологической схеме на:

- онтология объектов ПдО (в некоторых публикациях используется термин “тезаурус”);
- онтология процессов ПдО;
- онтология приложений;
- онтология класса задач ПрП;
- онтология методов решения задач.

Известны различные схемы конструирования (синтеза) онтологических структур из предметно-проблемных знаний [130, 131]. В данной работе в онтологию ПдО включены объекты и процессы (статистические знания), которые сгруппированы в отдельные онтологии, а в онтологию задач включены классы задач, методы их решения и соответствующие алгоритмы. При такой схеме группировки предметно-проблемных знаний, последние строго разделены на статические и динамические. Если изменяется класс решаемых задач, то онтологии объектов и процессов уже готовы к повторному использованию, перепроектируется только онтология задач [92, 93].

В текстовых описаниях (спецификациях) целевых задач выделяются набор объектов и набор процессов (методов), необходимых и достаточных для выполнения конкретных целевых задач. Можно выделить некоторое унифицированное (пополняемое) множество базовых задач (типовых фрагментов задач), на основе которых с помощью определённых логических последовательностей конструируются более сложные задачи.

Полюсом конструктора, описывающего схемы типов решаемых задач, является некоторая семантическая ось с противоположными сторонами “Выбор” и “Построение”, концепты которых описывают простую и сложную схемы решения задач. Под простой схемой (схема “Выбор”) понимается такая схема, в которой для решения задачи онтологической системе необходимо сделать выбор из совокупности известных схем решений. Когда выбор сделан, то известны метод, процесс и его состояния, т. е. решение задачи становится тривиальным. Противоположностью простой схемы является сложная схема (схема “Построение”), когда все составляющие процесса решения задачи неизвестны. На практике, чаще всего, встречаются некоторые промежуточные схемы решения задач. Например, схемы, в которых начальные и целевые состояния, совокупности методов задаются в явном виде. Общей стратегией решения сложных (непростых) задач является многоуровневая декомпозиция исходной задачи до того уровня, на котором полученные подзадачи являются простыми. При этом онтологические знания являются активной основой процесса декомпозиции.

Онтологические знания, описывающие некоторое ПрП, можно разделить на следующие компонентные знания: типы входных и выходных данных, инструменты, операторы (человек или компьютерная программа) и операции (действия оператора или *Решателя задач*) [98].

Для реализации подхода необходимо разработать унифицированный язык представления онтологических знаний и инструментальную среду как набор специализированных и универсальных базовых операций, управляющих процессом решения. Необходимо также разработать *Решатель задач*, который осуществляет выбор средств и методов формирования структуры задачи на основе базовых операций.

Схема модели онтологии задач описывается тройкой

$$O^3 = \langle OZ^{ПрП}, M, PZ \rangle, \quad (4.12)$$

где OZ^{PrP} – обобщённая задача проблемного пространства, состоящая из p задач, которые, в свою очередь, состоят из $w = \overline{1, W}$ фрагментов каждая. Каждый фрагмент представлен процедурой, реализованной на множестве $v = \overline{1, V}$ операций каждая. Кроме того, задача

$$Z^p = \langle D_{in}^p, R^p, C^p, D_{out}^p \rangle \quad (4.13)$$

определяется множествами входных данных D_{in}^p , требований (условий, ограничений) R^p , контекста задачи C^p и выходными данными (или целью решения задачи) D_{out}^p ;

M – множество методов решения задач. Определяется как отображение

$$M^p : (D_{in}^p, R^p, C^p) \rightarrow D_{out}^p \quad (4.14)$$

компоненты которого определены выше;

PZ – решатель задач.

На рис. 4.8 представлена схема онтологии задач.

Одно из преимуществ онтологического подхода, в частности, иерархического представления состоит в том, что сложная задача большой размерности разбивается на последовательно решаемые группы задач малой размерности. Онтология задач в качестве понятий содержит типы решаемых задач, а отношения этой онтологии, как правило, специфицируют декомпозицию задач на подзадачи.

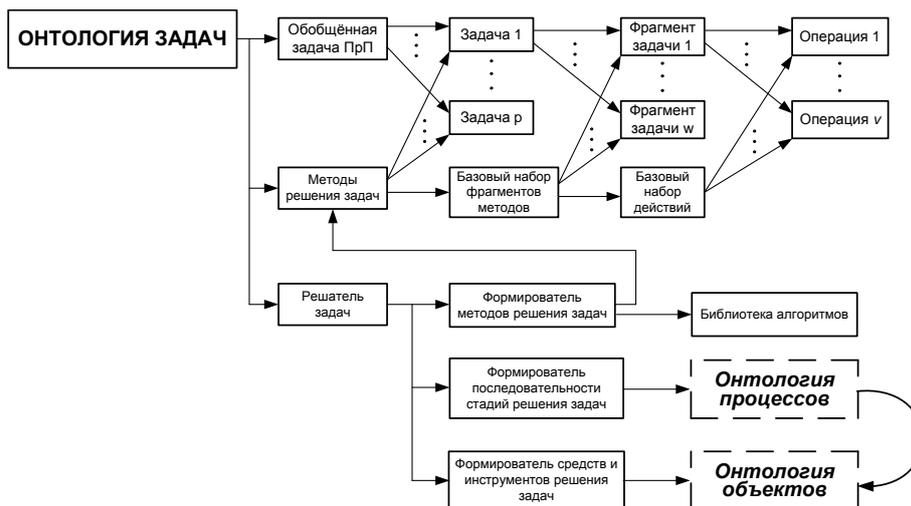


Рис. 4.8. Схема онтологии задач

4.4.5. Алгоритм проектирования онтологий

На рис. 4.9 представлена блок-схема алгоритма проектирования онтологии ПдО и онтологии задач. Онтология ПдО представлена онтологией объектов и онтологией процессов, для которых в блок-схеме предусмотрены отдельные ветви. Предполагается, что множество функций интерпретации в моделях онтологий тождественно множеству аксиом [2]. На рисунке приняты следующие сокращения:

– ПрП – проблемное пространство;

- Т-О, Т-П – термины-объекты и термины-процессы; Т-З – списки задач, процедур и методов их решения;
- X^O, R^O, F^O – конечные множества понятий, отношений и функций интерпретации онтологии объектов;
- $X^П, R^П, F^П$ – конечные множества понятий, отношений и функций интерпретации онтологии процессов;
- $X^З, R^З, F^З$ – конечные множества понятий, отношений и функций интерпретации онтологии задач;
- $OG^O, OG^П, OG^З$ – онтографы соответственно объектов, процессов и задач.

Ниже приведено описание алгоритма проектирования онтологии объектов, онтологии процессов и онтологии задач со ссылками на пункты (номера блоков) на рис. 4.9.



Рис. 4.9. Блок-схема алгоритма проектирования онтологии ПДО и онтологии задач

При выполнении постановки задачи на проектирование онтологии заданной ПДО и связанного с ней ПрП систематизируются методы и средства исследования предметных и проблемных знаний предметной области, а также выявляются источ-

ники их описания. Затем выполняется извлечение и систематизация знаний заданной ПдО и множества задач ПрП.

Алгоритм предусматривает последовательное построение онтологии ПдО (п. 4) и онтологии задач (п. 7).

В результате формируются списки Т–О, Т–П и их свойств с учётом синонимии (п. 4) и списки задач, процедур и методов их решения (п. 7). Следует отметить, что в п. п. 7, 8 выполняется формирование проблемного пространства заданной ПдО, в том числе конкретизируется информационное пространство для решателя задач (п. 8). Далее анализируется полнота сформированных списков (в случае неполноты какого-либо из списков осуществляется возврат к п. 2) и переход к проектированию онтологии объектов, онтологии процессов и онтологии задач, которое состоит из трёх блоков для каждой из них (О, П, З): построение множества X (п. п. 10, 18, 26); построение множества R (п. п. 12, 20, 28) и построение множества F (п. п. 14, 22, 30). После каждого из них проверяется полнота исчерпаемости списка соответствующих терминов.

При построении множества R формируются множества пар (x_i^o, x_j^o) , (x_i^n, x_j^n) и (x_i^3, x_j^3) , $i \neq j$ в соответствии с отношением “выше-ниже”, анализируются и устанавливаются взаимосвязи $x_i^o R_k^o x_j^o$, $x_i^n R_k^n x_j^n$ и $x_i^3 R_k^3 x_j^3$ соответственно и строятся предварительные онтографы для X^o , X^n и X^3 . При построении множества F приписываются свойства для каждого X^o , X^n и X^3 из списка свойств Т–О, Т–П и Т–З соответственно, а также формируются функции интерпретации $\{f_i^o\}$ для каждого X^o , $\{f_i^n\}$ для каждого X^n и $\{f_i^3\}$ для каждого X^3 из определений толкового словаря (или из другого подходящего источника описания типовых наборов задач и процедур и методов их решения).

Далее выполняется построение конечных онтографов OG^o (п. 16), OG^n (п. 24) и OG^3 (п. 32) по элементам множеств понятий X^o , X^n и X^3 соответственно; анализируются на полноту и непротиворечивость онтографы OG^o (п. 17), OG^n (п. 25) и OG^3 (п. 33) и их формализованные описания. Если заданные критерии не выполнены, то осуществляется возврат к п. 10, п. 18 и п. 26 для реинжиниринга онтологии объектов, онтологии процессов или онтологии задач соответственно.

На заключительном этапе работы алгоритма выполняется анализ на предмет построения всех онтологий O^o , O^n и O^3 . Если да – то END, иначе осуществляется возврат к п. 3.

Алгоритм проектирования онтологии ПдО и онтологии задач представляет собой последовательность шагов анализа предметно-проблемных знаний и синтеза онтологических структур объектов, процессов и классов задач. Блоки 1-9 алгоритма описывают процесс анализа, а блоки 10-17, блоки 18-25 и блоки 26-33 – процессы синтеза соответственно онтологии объектов, онтологии процессов и онтологии задач. Алгоритм пригоден как для ручного, так и для автоматизированного проектирования указанных онтологий.

Несомненный интерес представляет автоматизированный способ проектирования, так как даже для сравнительно небольших предметных областей списки терминов-понятий, терминов-процессов, терминов-отношений и терминов, описывающих классы задач, могут составлять несколько тысяч записей. Составление вручную таких списков представляется проблематичным. В главе 7 описывается инструментальный комплекс, программные модули которого реализуют, в частности, представленный на рис. 4.9 алгоритм.

5. АЛГЕБРО-ЛОГИЧЕСКИЕ И ЛИНГВИСТИЧЕСКИЕ МОДЕЛИ ОБРАБОТКИ ЕЯТ

5.1. Общий подход к проблеме анализа ЕЯТ

Степень развития и внедрения компьютерных технологий в значительной мере определяется не столько возможностями традиционной вычислительной техники (ВТ), сколько особенностями предметных областей, и успех эффективной компьютеризации последних существенным образом зависит от глубины исследования и понимания явления, которое моделируется. В соответствии с этим этап обобщённого подхода к формированию информационных технологий исключительно с позиций потенциальных возможностей ВТ постепенно смещается в сторону дифференциации отдельных направлений ИТ, где на первый план выступают особенности (домена) конкретных ПдО. К такому направлению следует отнести ИТ, которые тем или другим образом связаны с языковым моделированием поведения человека, или системы, ориентированные на обработку естественно-языковой информации. Причём под обработкой ЕЯТ понимается реализация ряда ИТ, конечной целью которых является компьютерная обработка знаний как высшей формы умственной деятельности человека.

Разработка формальной теории и методологии компьютерной обработки знаний составляет одну из насущных проблем в общей теории искусственного интеллекта. Сложность указанной проблемы определяется, в частности, необходимостью привлечения целого ряда научных теорий (информатики, математической логики, компьютерной и психологической лингвистики, нейрофизиологии, нейрокибернетики, теории семантических сетей и др.), которые в совокупности, будучи применёнными к решению проблемы формального представления и обработки знаний, составили бы концептуально единую (междисциплинарную) формальную теорию. Составляющие указанной теории должны учитывать сущность этапов языковой и предметной обработки информационных объектов (для первого – ЕЯТ, а для второго – извлечённые из ЕЯТ знания) и формальной связи между ними. В конечном итоге необходима такая теория, которая позволяет разработать комплекс информационных технологий компьютерной обработки ЕЯТ, представления знаний и компьютерной обработки знаний.

При этом возникает несколько проблем.

Первой проблемой (и одной из основных) является проблема анализа текстовой информации, представленной на естественном языке (морфологический, синтаксический, семантический и логический анализ) с целью извлечения знаний [49].

Второй проблемой является проблема проектирования системы поиска, обработки и извлечения знаний, построение её архитектуры и разработка инструментария для пользователя [2, 49].

Третья проблема – это интеграция знаний из нескольких предметных областей для обеспечения эффективности проведения исследований междисциплинарного характера, использования уже существующих алгоритмов, фактов, теоретических положений и практических решений [49].

Третья проблема тесно связана с эффективностью использования систем автоматического поиска доказательства теорем в формальных логических теориях и подобными ей проблемами. Основу систем автоматического поиска доказательств составляют пруверы – программы, выполняющие доказательства. Применение прувера будет успешным только тогда, когда в его распоряжении будет вся необходимая информация для обеспечения процесса доказательства. Например, если пруверу необходимо доказать теорему Лагранжа про делимость порядка конечной группы на индекс её подгруппы, то пруверу недостаточно иметь только аксиоматику теории групп. Ему также понадобится аксиоматика теории делимости, а возможно и аксиоматика Пеано вместе с некоторыми дополнительными фактами из других областей знаний. Решение этой проблемы облегчается, если в распоряжении прувера есть интегрированная система, которая включает в себя необходимые сведения из других областей знаний. Тогда прувер сам найдёт нужную ему информацию и использует её для успешного проведения доказательства.

При решении указанных проблем важной задачей является построение естественно-языковых лингвистических моделей и создание на их основе эффективных лингвистических процессоров, которые в совокупности с ЯОКМ представляют языково-онтологическую информационную систему.

Лингвистические модели – это, в сущности, фундаментальная научно-прикладная область исследований, которая помогает строить системы, обрабатывающие ЕЯТ. Под обработкой ЕЯТ понимается процесс взаимодействия *Система-ЕЯТ-Пользователь*, который включает в себя разные способы влияния на ЕЯТ, такие как анализ, генерация, интерпретация, трансформация, синтез и др. Такое определение лингвистических моделей, основанное на их функциональном аспекте, является полезным с методологической точки зрения, обеспечивая классификацию моделей по их прагматическим признакам, т. е. по цели разработки и сфере применения. Выделим следующие классы лингвистических моделей:

- 1) диалоговые “запрос-ответ” или интерактивные модели;
- 2) концептуально-формальные модели;
- 3) концептуально-функциональные модели;
- 4) когнитивные (семантико-контекстные) модели.

Очевидно, наиболее сложными являются последние модели. Именно они обеспечивают глубинное проникновение в текущий контекст и его трансформацию с сохранением смысла как внутри одной модели, так и между разными моделями.

Таким образом, объектом исследования являются естественно-языковые тексты, представленные в электронном виде и взятые из сети Интернет, монографий, учебников, научно-технических документов и т. п., которые в совокупности составляют исходный лингвистический корпус текстов.

В существующих ИС выделяют пять основных *уровней понимания* ЕЯТ [25].

Первый уровень характеризуется схемой, показывающей, что любые ответы на вопросы система формирует только на основе прямого содержания, вытекающего из текста. В лингвистическом процессоре выполняется морфологический, синтаксический и семантический анализ текста и вопросов, относящихся к нему. На выходе ЛПП получаем внутреннее представление текста и вопросов, с которыми может работать блок вывода. Используя специальные процедуры, этот блок формирует ответы. Другими словами, уже понимание на первом уровне требует от ИС определённых способов представления данных и вывода на этих данных.

Второй уровень. На втором уровне добавляются способы логического вывода, основанные на информации, содержащейся в тексте. Это различные логики текста (временная, пространственная, каузальная и др.), порождающие информа-

цию, явно отсутствующую в тексте. Архитектура ИС, с помощью которой может быть реализован второй уровень понимания, должна иметь дополнительную базу знаний, в которой хранятся закономерности, относящиеся к временной структуре событий, возможной их пространственной организации, каузальной зависимости и т. д., а логический блок – все необходимые средства для работы с неклассическими логиками.

Третий уровень. К средствам второго уровня добавляются правила пополнения текста знаниями системы о среде. Эти знания в ИС, как правило, носят логический характер и фиксируются в виде сценариев или процедур другого типа. Архитектура ИС, в которой реализуется понимание третьего уровня, внешне не отличается от архитектуры ИС второго уровня. Но в логическом блоке должны быть учтены средства не только для чисто дедуктивного вывода, а и для вывода по сценариям.

Три перечисленные уровня понимания полностью или частично реализованы практически во всех действующих ИС.

Четвёртый уровень. На этом уровне происходит изменение содержимого базы знаний. Она дополняется фактами, известными системе и содержащимися в тех текстах, которые введены в систему. Разные ИС отличаются одна от другой характером правил порождения фактов из знаний, опираясь на методы дедуктивного вывода и распознавания образов. Правила могут быть основаны на принципах вероятностей, размытых выводов и т. д. Но во всех случаях база знаний оказывается априорно неполной. В ИС возникают сложности с поиском ответов на запросы. В частности, в базах знаний становится необходимым *немонотонный вывод*.

Пятый уровень. На этом уровне происходит порождение метафорического знания. Правила порождения знаний метафорического уровня, используемых для этих целей, представляют собой специальные процедуры, опирающиеся на выводы по аналогии и ассоциации. Известные в настоящее время схемы вывода по аналогии используют, как правило, диаграмму Лейбница, которая отображает только частный случай суждений по аналогии. Ещё меньше разработаны схемы ассоциативных суждений [25].

Существуют и другие интерпретации феномена понимания. Возможно, например, оценивать уровень понимания по способности системы к пояснению полученного результата. Здесь возможен не только уровень *пояснения*, когда система поясняет, что она сделала, например, на основе введенного в неё текста, а и уровень *обоснования* (аргументации), когда система обосновывает свой результат, показывая, что он не противоречит той системе знаний и данных, которыми она владеет. В отличие от пояснения обоснование всегда связано с суммой фактов и знаний, которые определяются текущим моментом существования системы. И введенный для понимания текст в одних состояниях может быть воспринят системой как истинный, а в других – как ложный. Существующие ИС типа экспертных систем, как правило, способны давать пояснения и лишь частично обоснования [25].

Обобщённый критерий понимания ЕЯТ научно-технического профиля состоит в способности решать на основе содержащихся в них знаний – прикладные задачи.

Особенности анализа ЕЯО определяются направленностью на формирование структуры понятий, то есть, на автоматическое извлечение знаний из текстов и их прагматическую интерпретацию в терминах прикладной задачи. При этом текст рассматривается как объект разных уровней анализа: как знаковая система, как грамматическая система и как система знаний про мир (предметная область). Каж-

дый уровень имеет свои особенности, свои способы выражения и, следовательно, допускает наличие специфических методов обработки.

5.1.1. Методы и модели анализа ЕЯТ

Методы анализа ЕЯТ

Формализация естественного языка с целью автоматизации анализа текстов на естественном языке была начата ещё в начале тридцатых годов 20-го столетия в работах А. Тарского и его учеников, хотя о такой необходимости, как отмечалось выше, говорили ещё Аристотель, Лейбниц и Эйлер.

Если они только говорили о необходимости формализации и делали только первые шаги на этом пути, то более серьёзные попытки такой формализации были сделаны логиком А. Тарским [132, 133], результатом которых стало появление понятия выполнимости (сатисфакции) формул – более общего, чем понятие истинности. Это понятие Тарский применил к открытым и замкнутым формулам (под замкнутой формулой относительно предложений естественного языка понимают фразу). Это дало возможность сформулировать понятие истинности для предложения естественного языка и наложить его на каждую открытую атомарную формулу, которая состоит из примитивного предиката (то есть предметной константы) и стольких переменных, какова арность предиката. Поскольку таких формул существует лишь конечное множество, то такой подход становится конструктивным.

Следующая попытка усовершенствования формализации А. Тарского была сделана Д. Девидсоном [134]. Он предложил добавить к понятиям выполнимости и истины рекурсивное определение истины. Тогда теория T , которая включает рекурсивное определение истины, объясняет, каким образом значение фраз зависит от значения (смысла) слов этих фраз. Но, поскольку слово «значение» не является синонимом слова «истина», то определение истины не всегда будет определением значения (смысла). Таким образом, тезис Девидсона не совсем очевиден, зато его легко обосновать: необходимо сопоставлять смысловые значения повествовательных предложений с условиями их истинности.

Если принимается уравнение «значение повествовательного предложения = условие истинности этого предложения», то нужно наложить следующее условие. Если в определении говорится о том, каким образом условия истинности сложного предложения зависят от условий истинности его составляющих простых предложений, то там же должно говориться о том, как значение сложного предложения зависит от значений его составляющих простых предложений.

Монтегю также верил в то, что методы формальной семантики можно применить к исследованиям семантики естественного языка. Но он, в отличие от Девидсона, отказался от применения логики предикатов первого порядка, отдав предпочтение **категориальным грамматикам**. Эти грамматики включают в себя те категории, которые специалисты в области грамматик традиционно используют при определении естественного языка, например такие категории, как подлежащее или сказуемое. Этот подход противоположен подходу Девидсона. Это дало возможность Монтегю заменить в модели понятие абсолютной истины на понятие относительной истины, потому что в одной модели одно и то же предложение может быть истинным, а в другой – ложным. Такое расширение дало возможность определить понятие логической истинности и логического следствия для более широкого фрагмента естественного языка [135]. Таким образом, Монтегю выделил два элемента: интенцию (смысл) и экстенцию (денотат) и применил их к подлежа-

щим, сказуемым и фразам. Существуют и другие подходы к анализу текстов на естественном языке, которые основываются на понятиях семантической сети, фрейма и т. д.

Если коротко охарактеризовать идеи, которые доминировали последние десятилетия, то они сводятся к таким положениям.

Начиная с 70-х годов в исследованиях естественных и искусственных (формальных) языков доминировали попытки построить теорию, которая бы охватывала как естественные, так и искусственные языки.

Синтаксис вызывает интерес только в связи с семантикой.

Целью синтаксиса является характеристика синтаксических категорий, на которых построены высказывания.

Целью семантики является пояснение понятий истины и логического наследования.

Эти идеи привели к тому, что выработались новые взгляды на такие понятия как грамматика и значение выражений. А именно:

- а) грамматика – это интерпретированное исчисление выражений языка (генеративная лингвистика);
- б) значение выражений вычисляется интерпретатором (интерпретационизм);
- в) очевидна композиционность категорий синтаксиса, семантики и прагматики (категориальные грамматики).

Этапы и модели анализа ЕЯТ

Предыстория обработки естественного языка (ОЕЯ) определена многими факторами, из которых можно выделить два следующих: 1) попытки моделирования нейронов в виде логических устройств; 2) зарождение “информационной парадигмы” – взгляда на числа и на текст как на то, что представляет общее понятие “информации”, имеющее количественные характеристики со всеми вытекающими следствиями для математической теории информации [136].

Собственно историю ОЕЯ иногда представляют состоящей из четырёх этапов [3, 38, 62, 64, 65, 136].

50 – 60-е годы. Идея “информации” привела к концепции “машинного перевода” как установление той информации, которая скрыта в предложении естественного языка, и её выражение другим – целевым языком. Язык информационного содержания назван “языком-посредником”. На этом этапе эксплуатируется идея эвристического поиска, а также языки программирования для решения задач ОЕЯ.

60 – 70-е годы. Рекомендуются использовать ограниченные содержательные области для моделирования ОЕЯ и строить как можно больше систем для обработки ограниченных подязыков, присвоив этим системам статус “экспертов” в определённой предметной области, а потом объединять все эти системы в одну в рамках “большого эксперта”. Последний управляет подключением “малых” экспертов на конкретном этапе обработки при решении конкретной задачи – концепция “большого переключения”. Осуществляется перевод ЕЯ на формальные языки (некоторым из этих языков принадлежат собственные исчисления, типа исчисления предикатов формальной логики; существуют языки поиска в базах данных. Проявлена заинтересованность в “когнитивной науке”, названной позднее когнитологией. Этот период иногда называют эрой “инженерного подхода”: основное ударение делалось на решение конкретных (как правило, прикладных) задач, без попыток моделировать психологическую деятельность. Исследования этого периода выявили целую совокупность недостатков.

70 – 90-е годы. В центре внимания находятся проблемы: соотносительности языковых актов, фокусов внимания и общих (для ЭВМ и человека) презумпций; использования “новаторского” потенциала языка (в частности, интерпретация метафор); построения прикладных систем, понимающих ЕЯ (а не просто его обрабатывающих); обработки высказываний, описывающих пространственные события; введения факторов “здорового смысла” в состав логического вывода систем ОЕЯ, в частности, учёт степени правдоподобия при интерпретации высказываний.

Одной из главных отличительных особенностей настоящего этапа является “онтологический подход” к созданию систем ОЕЯ. Он характеризуется, в частности, широкомасштабным привлечением к интерпретации как языковых, так и предметных знаний, возрастающей активностью бионического направления исследований при моделировании “языкового сознания” человека. Другими словами, исследования принимают явным образом выраженный междисциплинарный характер.

Теоретические и практические исследования создали необходимый описательный аппарат для компьютерного моделирования автоматического анализа текстов. Множество теоретических подходов можно разделить на два основных направления: *формализм* и *функционализм*. Формализм утверждает, что язык является врождённым компонентом человеческого мышления, который может быть представлен в виде абстрактной модели на метаязыке формальной грамматики и не зависит от способов использования языка, а функционализм, наоборот, допускает, что строение языка определяется сферой его использования [137]. Исследования в формальной лингвистике можно также условно разделить на два подхода: *построение универсальной грамматики*, справедливой для всех существующих языков мира, – генеративная модель, и *построение формальной модели*, которая наиболее полно охватывает все множество грамматических явлений конкретного языка – модель “Смысл \leftrightarrow Текст” [138].

В задачах автоматической обработки ЕЯТ, как правило, используются концепции, разработанные в рамках формализма. Соединив две вышеупомянутых парадигмы формальной лингвистики, программные модели являются лишь частичной реализацией теоретических исследований.

Рассмотрим вышеупомянутую комплексную модель формализации семантики ЕЯТ “Смысл \leftrightarrow Текст”, на которой базируется большинство современных теоретических исследований и практических разработок для флективных языков [65, 139]. В основу модели положены такие принципиальные положения:

- синонимия языковых высказываний возникает не только за счёт лексических синонимов – это могут быть синтаксические варианты представления одного значения слова, которые по-разному ориентируют определенную ситуацию относительно её участников;
- коммуникативная организация текста включает: распределение на тему и рему, выделение “старой” и “новой” информации, выделение более или менее важных частей содержания;
- носителями смысла есть не только слова, а и морфемы.

В модели предложены язык семантических множителей для формализации описания сущности и меры семантических тождественностей лексически разных единиц. Под множителями в данном случае понимаются элементарные смыслы, разные комбинации которых отвечают сложным лексическим значениям слов. Значение слова при этом есть не простая совокупность множителей, а определенным образом организованная структура. В модели приняты такие уровни представления текста: семантический, синтаксический, морфологический. Семантическое пред-

ставление включает в себя две компоненты: семантический граф и сведения о коммуникативной организации смысла. *Семантический граф* – это ориентированный граф, вершины которого обозначены символами сем (атомами смысла). Семы классифицируются на два типа: предикаты и имена объектов. Логические связи и кванторы трактуются как предикаты. *Синтаксический уровень* делится на глубоко-синтаксический и поверхностно-синтаксический. Первый включает в себя дерево синтаксического подчинения, узлы которого – словоформы предложения в виде абстрактных или обобщённых лексем с соответствующей морфологической информацией (число, время, вид и т. п.), второй – максимально приближенное к текстовой форме представление синтаксических связей в предложении. *Морфологический уровень* также включает глубинное и поверхностное представления.

Недостатком вышеописанной модели является то, что она не учитывает закономерности описания знаний на знаковом уровне представления текста, что в свою очередь приводит к грубым ошибкам при семантическом анализе текста. Кроме того, прагматический анализ сводится лишь к выявлению общих знаний о мире (предметной области). При этом анализ текста, с точки зрения аналитика, состоит в выявлении таких семантических признаков, которые подчинены целевой установке конечной (прагматической) задачи.

В [139] предложен развитый (по сравнению с моделью “Смысл \Leftrightarrow Текст”) подход к разработке методов распознавания и извлечения знаний из ЕЯТ (которые принадлежат к некоторой ПДО) и который основывается на таких концептуальных положениях:

- входной ЕЯТ – связный текст (т. е. дискурс);
- связность дискурса обеспечивается графемными средствами оформления текста, лингвистическими (грамматическими согласованиями, анафорическими ссылками и т. п.) и экстралингвистическими (например, времени, причинно-следственные связи, которые соответствуют определенной ПДО) средствами;
- все эти средства являются инструментом кодирования знаний о мире (ПДО);
- как элементы реального или абстрактного мира выступают его объекты, отображённые в тексте в форме естественно-языковых понятий, отношений между ними и языковые характеристики понятий и отношений;
- формирование объектов знаний о мире может быть разным и зависит от целевой установки прикладной задачи, проблемной области и логической картины мира как носителя информации, так и того, кто её обрабатывает.

Особенности анализа ЕЯТ определяются направленностью на формирование понятийной структуры, т. е. на автоматическое извлечение знаний из текстов и их прагматическую интерпретацию в терминах прикладной задачи. При этом текст рассматривается как объект разных уровней анализа: как знаковая система, как грамматическая система, система знаний о мире, источник исходных данных для решения прикладных задач. Каждый уровень имеет свои особенности, свои средства выражения, а, следовательно, предполагает наличие специфических методов обработки.

5.1.2. Общие принципы анализа и понимания ЕЯТ

Исследование процессов интерпретации и понимания языковых высказываний, имеет как теоретический, так и прикладной интерес. Работы в области автоматического анализа текста и автоматического решения задач, сформулированных на

естественном языке (или языке, близким к нему), показали актуальность построения интерпретационной теории языка. Понятие “интерпретация” с самого начала лежало в основе общелингвистических теорий, а также в основе логических исследований. Интерпретационный подход представлен как в разнообразных областях чисто лингвистического анализа (в теориях формальных грамматик, в “теории языковых актов” и т. п.), так и в исследованиях по искусственному интеллекту. Согласно интерпретации в основе владения языком и его использованием лежит один и тот же интерпретирующий механизм, обслуживающий разные *сферы языковой деятельности* и использующий разные виды знаний. Среди этих сфер – речь, понимание, редактирование, комментирование, перефразировка, соображение, коммуникация, аргументация, обучение, перевод и др. Сама же интерпретация, через которую и определяются указанные сферы, представляет собой получение на основе одного исходного объекта (объекта, который интерпретируется) другого, отличного объекта, который допускается интерпретатором как равнозначный исходному на фоне конкретной ситуации, набора знаний [136]. Сами же знания не входят в структуру языка непосредственно, а “привлекаются” к интерпретации, и только опосредованно определяют результат интерпретации языковых высказываний.

Одной из основных процедур обработки ЕЯТ является процедура распознавания, в частности когнитивного распознавания. Под *когнитивным распознаванием* ЕЯТ понимается процесс формализации извлечения и представления знаний предметной области, содержащихся в ЕЯТ. Входом процедуры распознавания является ЕЯТ, а выходом (результатом) – формально-логическое представление. Оно является формализованным представлением знаний о мире (предметной области), отражённым в определенном ЕЯТ [139].

Процесс распознавания и извлечения знаний из ЕЯТ базируется на компьютерном моделировании интеллектуальных функций человека, а именно – на моделировании процесса понимания человеком ЕЯТ. При этом термин *понимание* определяется с помощью таких критериев: умение пересказать “своими” словами, т. е. другими (лексическими, синтаксическими) средствами передать содержание входного текста, умение ответить на вопрос относительно определённого текста и др. Процедура распознавания базируется на средствах формализации (т. е. разработки онтологических моделей) знаний об определённом языке и знаний об определённой ПдО. Поскольку процедуры распознавания и понимания являются базовыми при лингвистической обработке ЕЯТ, рассмотрим их более детально с методологической точки зрения.

Отметим, что когнитивное распознавание ЕЯТ имеет свои особенности, а именно:

- внеязыковые ситуации, описанные в текстах, определяются лишь знаниями из определенной ПдО;
- средства вербализации этих знаний ориентированы на определённый уровень профессиональной подготовки;
- механизмы взаимодействия знаний в тексте с когнитивной картиной мира основываются на модели представления человеком знаний об определённом языке (языковая картина мира) и знаниях о фрагментах реальной действительности (базы знаний предметных областей).

5.2. Алгебро-логический подход к анализу и обработке ЕЯТ

5.2.1. Формальная постановка задачи анализа ЕЯТ

Пусть $T = t_1, t_2, \dots, t_n$, естественно-языковой текст в алфавите X , т. е. $T \in L(X)$, где $L(X)$ – язык над алфавитом X , а $t_i \in T$ – предложения, $i = \overline{1, n}$, n – мощность множества T .

Каждое предложение $t_i \in T$, в свою очередь, имеет структуру $t_{i_1}, t_{i_2}, \dots, t_{i_m}$, где t_{i_j} содержательно означают грамматические единицы, из которых построено предложение t_i . Если $t_{i_j} \in t_i$, то $C_L(t_{i_j}) = t_{i_1}, t_{i_2}, \dots, t_{i_{(j-1)}}$ и $C_R(t_{i_j}) = t_{i_{(j+1)}}, t_{i_{(j+2)}}, \dots, t_{i_m}$ будем называть *левым* и *правым контекстом* t_{i_j} соответственно.

С текстом T свяжем такие объекты:

- S – словарь языка $L(X)$, где содержатся слова t_{i_j} со своими определителями (в частности, лингво-семантическими характеристиками единиц словаря);
- $\gamma \subseteq T \times S$ – отношение, определяющее возможные значения и типы слова в словаре S ;
- $A = (D, \Pi)$ – предметная модель, на которой интерпретируется текст T ;
- $\phi \subseteq T \times A$ – отношение интерпретации текста T на области D .

Сигнатура предикатов $\Pi = \{\pi_1^{k_1}, \dots, \pi_r^{k_r}\}$ содержит атомарные предикаты, из которых можно строить сложные формулы. Зафиксируем эту сигнатуру, поскольку она зависит от предметной модели. Так как модель не уточняется, то и сигнатуру уточнить нельзя. Заметим только, что каждый атомарный предикат имеет тип (т. е. речь идёт о некоторой типизированной сигнатуре).

Определим правила вычисления отношений γ и ϕ .

Отношение γ имеет достаточно простой способ вычисления:

$$\gamma(t_{i_j}) = \{(a_1, \tau_1), (a_2, \tau_2), \dots, (a_s, \tau_s)\},$$

где a_i – возможные значения слова t_{i_j} , а τ_i – его возможные типы. Возможен случай, когда $\gamma(t_{i_j}) = \emptyset$. В этом случае значение этого слова считается неопределённым (и это требует пополнения словаря S).

Отношение ϕ определяется несколько сложнее.

$$\phi(T) = \phi(t_1), \dots, \phi(t_n),$$

где

$$\phi(t_i) = \{\phi(\gamma(t_{i_1})\gamma(C_R(t_{i_1}))), \phi(\gamma(C_L(t_{i_2}))\gamma(t_{i_2})\gamma(C_R(t_{i_2}))), \dots, \phi(\gamma(C_L(t_{i_n}))\gamma(t_{i_n}))\},$$

при этом

$$\begin{aligned}\phi(\gamma(t_{i_j})) &= \gamma(\phi(t_{i_j})); \\ \phi(\gamma(C_L(t_{i_j}))) &= C_L(\phi(\gamma(t_{i_j}))); \\ \phi(\gamma(\pi_r^k(p_1, \dots, p_k))) &= \gamma(\phi(\pi_r^k)(\phi(\gamma(p_1), \dots, \gamma(p_k))),\end{aligned}$$

где $\gamma(\phi(\pi_r^k))$ – имя предиката, тип которого согласован с аргументами $\gamma(p_1), \dots, \gamma(p_k)$.

Из этой формальной постановки проблемы анализа ЕЯТ вытекает, что основные задачи сводятся к следующим:

- конкретизировать предметную модель A ; эта задача является основной в связи с тем, что предметная модель является по существу базой знаний (конкретизация состоит в том, чтобы определиться с формальным логическим языком, правилами вывода, аксиоматикой и пр.);

- показать вычислимость отношений γ и ϕ на предметной модели A ;

- построить алгоритмы вычисления отношений γ и ϕ ;

- при вычислении отношений γ и ϕ контролировать соответствия типов аргументов и предикатов;

- определить взаимодействие алгоритмов вычисления γ и ϕ с системами лингвистического анализа текста.

Второстепенными, но также важными, являются задачи, связанные:

- с определением структуры данных для словарей;

- с определением информации, которая должна содержаться в словарях;

- с определением режима взаимодействия с пользователем (автоматический, полуавтоматический или диалоговый).

В качестве удобного языка представления данных, словарей, а также выполнение анализа, предлагается язык списочных структур. Эти структуры являются динамичными, что и определяет их выбор.

5.2.2. Алгебра списочных структур

Пусть $F(X)$ – свободная полугруппа с единицей над некоторым конечным алфавитом $X = \{x_1, x_2, \dots, x_n\}$. Роль единицы играет пустое слово e . Напомним, что словом в алфавите X называется произвольная конечная последовательность символов этого алфавита. Произвольное слово $p = y_1 y_2 \dots y_m$ из $F(X)$ будем называть **списком** элементов $y_1 y_2 \dots y_m$, а сами элементы $y_i \in X, i = 1, 2, \dots, m$, – составляющими этого списка. При этом элемент y_1 называется *началом*, а элемент y_m – *концом списка*. Если $p \in F(X)$, то число составляющих списка p называется его длиной и обозначается $l(p)$. Если p, q – два списка, то список (слово) q называется *началом (концом)* списка (слова) p когда существует такое слово p' , что $p = qp'$ ($p = p'q$). Два списка $p = s_1 s_2 \dots s_k$ и $q = t_1 t_2 \dots t_l$ равны между собой, если $l = k$ и $s_i = t_i, i = 1, 2, \dots, k$.

Из теории известно, что $F(X)$ является алгеброй с одной бинарной операцией конкатенации (*conc*) и одной нулевой операцией (пустое слово e). Введём в рассмотрение ещё несколько функций и операций над списками, то есть над элементами множества $F(X)$ [59].

Пусть N – множество натуральных чисел и $p = y_1 y_2 \dots y_m$ – произвольное слово из $F(X)$, тогда

$$(1) \quad head(p) = y_1 \quad (head: F(X) \rightarrow F(X)).$$

Другими словами, функция $head(p)$ даёт первый символ слова p . Непосредственно из определения этой функции вытекают такие её свойства:
 $head(e) = e$, $head(y) = y$, если $y=X$, $head(head(p)) = head(p)$.

$$(2) \quad tail(p) = y_2 \dots y_m \quad (tail: F(X) \rightarrow F(X)).$$

Очевидно, что $tail(e) = e$, $tail(y) = e$, если $y=X$.

Содержание приведенных ниже функций вытекает из их определения.

$$(3) \quad add(p, i, x) = y_1 \dots y_i x y_{i+1} \dots y_m, \quad 0 \leq i \leq l(p).$$

$$(4) \quad sub(p, i) = y_1 \dots y_{i-1} y_{i+1} \dots y_m, \quad 1 \leq i \leq l(p).$$

$$(5) \quad dist(p, i) = (p_1, p_2), \text{ где } p_1 = y_1 \dots y_i, p_2 = y_{i+1} \dots y_m, \quad 0 \leq i \leq l(p).$$

$$(6) \quad hl(p, i) = y_1 \dots y_i, \quad 0 \leq i \leq l(p) \text{ – префикс слова } p \text{ длины } i.$$

$$(7) \quad tr(p, i) = y_{i+1} \dots y_m, \quad 0 \leq i \leq l(p) \text{ – суффикс слова } p \text{ длины } m-i.$$

$$(8) \quad push(p, x) = px = add(p, l(p), x).$$

$$(9) \quad pop(p) = y_1 \dots y_{m-1} = sub(p, l(p)).$$

Оказывается, что базовыми операциями, то есть такими, через которые выражаются все остальные из перечисленных функций, есть операции e , $conc$, $head$, $tail$, рекурсии и суперпозиции. Другими словами имеет место такое простое утверждение.

Теорема 5.1. Все функции (3) – (9) представляются в виде термов с помощью операций e , $conc$, $head$, $tail$ и операторов рекурсии и суперпозиции.

Доказательство. Рассмотрим последовательно случаи:

$add(p, i, x) = hl(p, i) \text{ } xtr(p, i)$; $sub(p, i) = hl(p, i-1) \text{ } tr(p, i)$; $dist(p, i) = (hl(p, i), tr(p, i))$;

$$hl(p, i) = \begin{cases} e, & \text{если } i = 0, \\ head(p), & \text{если } i = 1, ; \\ head(p)hl(tail(p), i-1), & \text{если } i > 1. \end{cases}$$

$$tr(p, i) = \begin{cases} p, & \text{если } i = 0, \\ tail(p), & \text{если } i = 1, ; \\ tr(tail(p), i-1), & \text{если } i > 1. \end{cases}$$

Из представления функций hl и tr вытекает представление функций add , sub и $dist$, и, следовательно, представление функций $push$ и pop .

Следует отметить, что функции $head$ и $tail$ ($push$ и pop) действительно являются операциями, тогда как остальные функции (3) – (7) операциями не являются. Это позволяет ввести такое определение.

Определение 5.1. Универсальная алгебра $G = (F(X), \Omega = \{conc, head, tail, e\})$, расширенная операторами рекурсии и суперпозиции называется **алгеброй списковых структур (АСС)**.

В этой алгебре легко установить справедливость таких тождеств.

(а) $sub(p, 1) = tail(p) = tr(p, 1)$; (б) $sub(add(p, i, x), i+1) = p$; (в) $hl(p, l(p)) = p$;
 (г) $tr(p, l(p)) = e$; (д) $pop(push(p, x)) = p$.

Имеет место и ряд других соотношений в этой алгебре.

Иногда вместе с множествами функций (3) – (9) рассматриваются и другие полезные функции, такие как

(10) $substr(p, i, j) = x_i \dots x_{i+j-1}$, то есть это полуслово слова p , которое начинается с i -го символа и имеет длину j ;

(11) $conv(p) = x_m x_{m-1} \dots x_2 x_1$, то есть это перестановка символов, составляющих список p в обратном порядке. Очевидно, что $conv(e) = e$ и $conv(y) = y$, если $y \in X$, а представление функций $substr$ и $conv$ в виде термов алгебры списковых структур имеет вид $substr(p, i, j) = hl(tr(p, i-1), j)$ и, $conv(p) = conv(tail(p))head(p)$.

Теперь, пользуясь операциями и функциями (1) – (11), можно вводить и другие функции.

Рассмотрим некоторые из таких функций.

Пусть задан некоторый алфавит $X = \{x_1, x_2, \dots, x_n\}$. Известно, что множество $F(X)$, как было отмечено выше, является полностью упорядоченным относительно лексикографического порядка, минимальным элементом которого есть пустое слово e . Пользуясь этим порядком, определим с помощью индукции такие функции:

а) $l(p) : F(X) \rightarrow N$ – функция длины слова, определение которой было дано выше. Индуктивное определение этой функции такое:

$$l(p) = \begin{cases} 0, & \text{если } p = e, \\ 1 + l(tail(p)), & \text{иначе.} \end{cases}$$

б) $subword(p, q) : F(X) \times F(X) \rightarrow \{0, 1\}$. Эта функция равняется 1, если слово q является подсловом слова p , и равняется 0 – в противном случае. Индуктивное определение этой функции такое: для произвольных слов $p, q \in F(X)$

$$subword(p, q) = \begin{cases} 0, & \text{если } l(p) < l(q), \\ 1, & \text{если } hl(p, l(q)) = q, \\ subword(tail(p), q), & \text{иначе.} \end{cases}$$

Непосредственно из определения этой функции следуют такие её простые свойства: поскольку пустое слово есть подсловом произвольного слова и произвольное непустое слово есть подсловом самого себя, то

$$subword(p, e) = 1, \quad subword(p, p) = 1.$$

в) $substit(p, q, r) : F(X) \times F(X) \times F(X) \rightarrow F(X)$. Результатом этой операции является подстановка слова r вместо первого вхождения слова q в слово p . Определение этой функции такое: для произвольных слов $p, q, r \in F(X)$

$$substit(p, q, r) = \begin{cases} e, & \text{если } p = e, \\ r \cdot tr(p, l(q)), & \text{если } hl(p, l(q)) = q, \\ head(p)substit(tail(p), q, r), & \text{иначе.} \end{cases}$$

Непосредственно с определения следуют такие очевидные свойства:

$$substit(p, e, r) = rp, \quad substit(p, q, q) = p.$$

Теорема 2.13 гарантирует единственность этих функций и предикатов.

Например,

а) $subword(abbcd, cd) = subword(bbcd, cd) = subword(bcda, cd) = subword(cda, cd) = 1$.

б) $subword(abbc, ax) = subword(bbc, ax) = subword(bc, ax) = subword(c, ax) = subword(e, ax) = 0$.

в) $substit(abbcd, cd, a) = asubstit(bbcd, cd, a) = absubstit(bcda, cd, a) = abbsubstit(cda, cd, a) = abbasubstit(a, cd, a) = abbaasubstit(e, cd, a) = abbaa$.

г) $substit(abcd, xa, da) = asubstit(bcd, xa, da) = absubstit(cd, xa, da) = abcsustit(d, xa, da) = abcdsubst(e, xa, da) = abcd$.

5.2.3. Алгебраическая система списочных структур

Расширим АСС предикатом равенства и условным оператором. Расширенную таким образом АСС будем называть **алгебраической системой списочных структур (АССС)**. Для этой алгебраической системы имеет место следующая теорема.

Теорема 5.2. *АССС является алгоритмически полной системой, то есть системой, в которой можно вычислить произвольную частично рекурсивную функцию.*

Доказательство. Для доказательства необходимо показать, что произвольный нормальный алгоритм Маркова можно представить и вычислить его значение в этой системе. Пусть Φ – некоторый нормальный алгоритм Маркова, заданный системой формул подстановки R в алфавите X :

$$\left\{ \begin{array}{l} p_1 \rightarrow [\cdot]q_1 \\ p_2 \rightarrow [\cdot]q_2 \\ \dots \dots \dots \\ p_m \rightarrow [\cdot]q_m \end{array} \right. \quad \left\{ \begin{array}{l} p_1 \rightarrow q'_1 \\ p_2 \rightarrow q'_2 \\ \dots \dots \dots \\ p_m \rightarrow q'_m \end{array} \right.$$

где $q'_i = \cdot q_i$ если формула подстановки заключительная и $q'_i = q_i$, если формула подстановки простая. Далее, пусть $p \in F(X)$ и G – АССС над алфавитом $X' = X \cup \{\cdot\}$. Тогда, используя операции и предикаты АССС, а также функции *subword*, *substit* и функцию длины, можем записать для произвольного $p \in F(X)$:

$\Phi(p)$ = если *subword*(p, p_1) = 1 то
 если *head*(q'_1) = «.» то *substit*($p, p_1, tail(q'_1)$)
 иначе $\Phi(substit(p, p_1, q'_1))$
 иначе если *subword*(p, p_2) = 1 то
 если *head*(q'_2) = «.» то *substit*($p, p_2, tail(q'_2)$)
 иначе $\Phi(substit(p, p_2, q'_2))$
 иначе.....
 иначе если *subword*(p, p_m) = 1 то
 если *head*(q'_m) = «.» то *substit*($p, p_m, tail(q'_m)$)
 иначе $\Phi(substit(p, p_m, q'_m))$
 иначе p .
 иначе $\Phi(substit(p, p_m, q'_m))$
 иначе p .

Покажем с помощью индукции по числу n применённых подстановок к слову p , что функция $f(p)$, полученная с помощью системы R , и функция $f'(p)$, полученная с помощью системы $\Phi(p)$, совпадают.

При $n = 0$ имеем $f(p) = p$, поскольку ни одна из подстановок системы R не применена к слову p . Из системы $\Phi(p)$ следует, что $f'(p) = p$, поскольку все условия вида *subword*(p, p_i) = 0. Значит, в этом случае $f(p) = f'(p)$.

Допустим, что равенство выполняется при всех $m < n$ и пусть n -ой формулой подстановки является формула $p_i \rightarrow q'_i$. Возможны такие случаи: формула $p_i \rightarrow q'_i$ – заключительная и формула $p_i \rightarrow q'_i$ – не заключительная.

Пусть в первом случае $m = n - 1$ и после выполнения m -й подстановки получено слово p' . По допущению индукции $f(p') = f'(p')$. Из того, что $p_i \rightarrow q'_i$ – при-

менима к p' следует, что ни одна из подстановок, предшествующих ей в системе R , не применимы к p' , или что то же самое, что все условия вида $subword(p', p_j) = 0$, где $j < i$, а условие $subword(p', p_i) = 1$ и $head(q'_i) = \langle \cdot \rangle$. Но тогда $f(p') = substit(p', p_i, tail(q'_i)) = substit(p', p_i, q_i) = f'(p')$.

Второй случай аналогичен первому, с той только разницей, что в первом случае вычисления заканчиваются, а во втором – продолжаются.

Алгоритм морфологического анализа, записанный в АССС

Рассмотрим пример использования АССС для представления алгоритма морфологического анализа слов естественного языка. Пусть X означает алфавит некоторого естественного языка, а $F(X)$ – множество слов конечной длины в алфавите X . Входные данные алгоритма морфологического анализа такие: D – словарь языка в алфавите X , F – словарь окончаний, S – словарь суффиксов. Морфологический анализ выполняется для слова $w \in F(X)$ и при этом считается, что принадлежность слова w к определённому классу (существительное, прилагательное, глагол и т. д.) уже известна, как только установлено, что это слово или его корень, принадлежат словарю D , в котором класс явно указан.

МОРФОЛОГИЯ (w, D, F, S)

begin

D : словарь языка в алфавите X ;

F : словарь окончаний;

S : словарь суффиксов;

w : анализируемое слово.

if $subword(D, w) = 1$ then $print(w, e, e)$ else if $l(w) > 1$ then $okonch(w, 1, e, e)$ else $print(\langle \langle w \rangle \rangle)$.

end

$okonch(p, i, q, r)$

begin

if $subword(F, head(conv(p))) = 1$ then

if $subword(D, hl(p, l(p)-i)) = 1$ then $print(hl(p, l(p)-i), q, head(conv(p))r)$

else if $l(hl(p, l(p)-i)) = 1$ then $print(\langle \langle p \rangle \rangle)$

else if $continue(hl(p), l(p)-i) = 1$ then /* окончание имеет продолжение*/

$okonch(hl(p, l(p)-i-1), i+1, q, head(conv(hl(p, l(p)-i))r)$

else $sufiks(hl(p, l(p)-i), i, q, r)$

else if $subword(F, hl(r, l(p)-i)) = 1$ then

$sufiks(hl(p, l(p)-i), i-1, q, hl(r, l(r)-i))$

else $sufiks(p, 1, e, e)$

end

$sufiks(p, i, s, f)$

begin

if $subword(S, head(conv(p))) = 0$ then $print(\langle \langle p \rangle \rangle)$

else if $subword(D, hl(p, l(p)-i)) = 1$ then $print(hl(p, l(p)-i), s, f)$

else if $l(hl(p, l(p)-i)) \leq 1$ then $print(\langle \langle p \rangle \rangle)$

else if $continue(hl(p), l(p)-i) = 1$ then /* суффикс имеет продолжение*/

$sufiks(hl(p, l(p)-i-1), i+1, q, head(conv(hl(p, l(p)-i))s, f)$

else $print(\langle \langle p \rangle \rangle)$

end

5.3. Прикладные аспекты лингвистического анализа ЕЯТ

5.3.1. Информационная модель этапов лингвистического анализа

Архитектура современных знание-ориентированных информационных систем с естественно-языковым представлением и обработкой знаний включает онтологическую составляющую эксплицитно, которую в общем виде можно интерпретировать как концептуальную базу знаний. Такая база знаний представляется в виде ориентированного графа, вершинами которого являются фреймы, описывающие концепты, а дугами – множество концептуальных отношений, связывающих между собой концепты. Другой важной особенностью указанной архитектуры является разделение и отдельная обработка семантики первой и второй ступени [27], что в общем случае означает разделение внутриязыкового и внеязыкового процессинга [62] и переход к формально-логическому представлению исходного текста.

Указанные особенности архитектуры современных ЗОИС трансформируют традиционную модель обработки естественно-языковых текстов в формальную модель следующего вида

$$F = \langle T, W, SS^1, O, S^2, I \rangle,$$

где T – множество обрабатываемых ЕЯТ;

W – множество словоформ, входящих в T ;

SS^1 – множество синтактико-семантических структур первой ступени, описывающих T ;

O – множество онтологических структур, отображающих множества W и SS^1 в S^2 ;

S^2 – множество семантических структур второй ступени, описывающих множество сценариев T ;

I – множество информационно-кодовых представлений S^2 .

Опишем объекты формальной модели.

Множество T представляет совокупность естественно-языковых текстов, характеризующихся стилями делового и научно-технического характера.

Цепочка $W \rightarrow SS^1$ в классическом понимании представляет грамматический анализ ЕЯТ. В отличие от традиционных линейного и сильнокодированного методов анализа здесь использован смешанный метод анализа. Суть его состоит в том, что в лексикографической базе данных полное множество W представлено в таблицах двух типов: таблицами лексем с соответствующими морфологическими, синтаксическими и семантическими характеристиками и таблицами флексий для всех полных, изменяющихся частей речи. При этом алгоритмы формирования парадигмы лексем просты: в таблицах лексики указаны основы лексем и соответствующие коды для выбора записей из таблиц флексий. Нефлексийные изменения учитываются соответствующими алгоритмами.

Описанная структура грамматического анализа однозначно соответствует эффективному отображению функциональных операторов на программно-аппаратный уровень реализации.

Множество O онтологических структур в идеале представляет языково-онтологическую картину мира [11, 27].

Множество SS^1 формируется и интерпретируется итерационно подсистемой синтактико-семантического анализа типа “Конспект” [140]. Основной операцией

синтактико-семантического анализа является распознавание синтаксических и семантических отношений, связывающих слова текста. Распознавание связей между знаменательными словами осуществляется путём анализа флексий и предлогов на основе лексических моделей без использования в явном виде полного множества правил традиционной грамматики. Для каждого предложения исходного текста строится дерево разбора. Разрешение семантической неоднозначности осуществляется путём обращения к множеству онтологических структур O . На основе построенных деревьев разбора фраз строится категориальная сеть, представляющая собой семантическое пространство S^2 текста. В качестве компьютерного представления такого пространства текста удобно использовать, в частности, растущую семантическую сеть множества информационно-кодовых представлений I , организованную на основе пирамидальной сети, рецепторы которой соответствуют именам объектов, классов объектов, свойств, состояний, действий, отношений, семантических падежей и модификаторов [141].

Цепочка преобразования информации $T \rightarrow W \rightarrow SS^1$ и $O \rightarrow S^2 \rightarrow I$, по сути, представляют (соответственно) базовые процедуры анализа и понимания ЕЯТ, средствами интерпретации которых являются грамматический и семантический процессоры.

В приложениях поиска и обработки большого объёма текстовых документов целесообразно использовать знание-ориентированную поисковую систему [142], обеспечивающую начальный и конечный этапы обработки документов – поиска в Интернет и сохранения документов в базе данных в виде их конспектов, сгенерированных подсистемой “Конспект”.

Описанная модель обработки естественно-языковых текстов в знание-ориентированной информационной системе, включающей подсистему “Конспект” как компоненту, представляет перспективное направление развития онтолого-управляемых информационных систем, активно использующих онтологию лексики естественного языка.

5.3.2. Графемно-морфологический анализ

Как начальный этап лингвистического анализа используется графемно-морфологический анализ (ГМА). При выполнении графемного анализа текст рассматривается как разновидность знаковой системы. Задачей данного этапа анализа является исследование свойств единиц языка и правил их соединения в аспекте их знаковой природы. В существующих моделях графемный анализ реализован только как способ решения противоречий анализа текстовых единиц, поступающих на обработку морфологического анализатора, и ограничивается отделением аббревиатур, сокращений и некоторых других классов лексем. Это, в свою очередь, порождает противоречие на этапе семантического анализа текста, что приводит к ошибкам семантического характера, избежать которых пользователю предлагается самому, вручную путём маркировки текста.

Назначение графемного анализа – построение модели графемной структуры входного текста, в которой выделены и связаны отношениями (где это возможно) такие содержательные единицы текста как фрагмент, предложение и лексема. Например, для лексем: выделяются классы лексем, которые отличаются своей графемной структурой и выполняют разные функции в тексте. Кроме того, анализируются закономерности сочетаемости некоторых лексических единиц, которые уже

на этом этапе позволяют объединять несколько лексем в одну на той основе, что они выполняют в тексте единую функцию [143].

Вследствие графемного анализа (точнее интерпретации базы знаний графического изображения ЕЯО с входным текстом) текст превращается в упорядоченный список составляющих ЕЯО.

Под *морфологическим анализом* понимают полную обработку последовательности (без какой-либо связи с контекстом) словоформ (см. алгоритм морфологического анализа), сформированных на подэтапе графемного анализа. Вследствие такой обработки каждая словоформа заменяется грамматической информацией – цепочкой символов, определяющих все те свойства данной словоформы, которые необходимы для дальнейшего анализа. Морфологическая информация совокупности словоформ, в свою очередь, используется на следующем этапе создания программной модели ЛП – во время синтаксического анализа, в результате которого устанавливаются связи между всеми словоформами текста и между частями сложных предложений.

Входными данными процедуры морфологического анализа является графемная структура текста, база знаний морфологии ЕЯ и лексикографическая база данных. ЛБД включает словари лексем, словоизменительную и словообразующую модели входного языка, а также отдельные таблицы для всех частей языка. К каждой лексеме в таблице присоединяется, кроме традиционных морфологических характеристик, наборы синтаксических и семантических признаков [27, 28].

Кроме того, существуют отдельные таблицы падежных окончаний (для изменяемых частей языка) для формирования словоформ лексемы. Все лексические единицы в таблице должным образом проиндексированы и имеют одинаковое интерпретационное значение на всех этапах лингвистического анализа.

5.3.3. Синтаксический анализ

Синтаксическим анализом (parsing) называется процесс структуризации линейной репрезентации соответственно к заданной грамматике. Линейной репрезентацией предложения естественного языка будем называть цепочку элементов, где каждый элемент является минимальной синтаксической единицей. Минимальная синтаксическая единица может быть словоформой или оператором с определённым набором характеристик. *Оператором* называется разделительный знак или сочинительный союз. Процессом структуризации линейной репрезентации предложения называется построение ориентированного графа синтагм и ориентированного графа сегментов. *Синтагма* определяет бинарное синтаксическое выражение вида $R(A, B)$, где A и B – словоформы, а R – тип синтаксического отношения, которое отвечает имени синтагмы, причём A управляет B . Узлами графа синтагм являются терминальные единицы. Связность не является обязательным условием такого графа, поскольку синтагмы опираются только на морфологические репрезентации словоформы, линейный порядок предложения, и, в некоторых случаях, на примитивную модель управления.

Сегмент можно определить как часть предложения, которая выделена в тексте знаками пунктуации и описывает отдельную ситуацию. Каждый такой сегмент имеет как вершину явный предикат, который представляет собой в большинстве случаев финитную форму глагола, или скрытый предикат, который может быть выражен деепричастием, причастием или именем с семантической характеристи-

кой действия; каждый такой предикат и задаёт ситуацию. Узлами графа сегментов являются нетерминальные единицы.

В зарубежной лингвистической традиции понятие «сегмент» эквивалентно термину «клауза». Сегментация предложения является первым этапом полного синтаксического анализа. Целью сегментации есть выделение и классификация сегментов в составе сложного предложения. Задачей второго этапа является устранение грамматической омонимии морфологического уровня и построение словосочетаний. Объектом распознавания являются закономерности взаимодействия лексем в пределах языковой синтагмы. Входными данными есть результаты предыдущих этапов распознавания и база знаний синтаксиса ЕЯ. В ней заложены модели синтаксических правил, которые определяют признаки синтаксического объединения лексем в словосочетания.

Среди языковых средств синтаксического анализа можно выделить следующее.

1. **Словоизменяемые морфологические средства.** Для языков с развитой морфологией, которыми являются украинский и русский языки, – это основной способ материализации синтаксических связей. Показателем морфологической зависимости служит *флексия*, поскольку грамматические словоформы обычно приписаны флексии, что позволяет в некоторых случаях выявить зависимость между двумя словоформами, отсутствующими в словаре. В украинском языке словоформы многих слов омонимичны (например, это может быть омонимия числа и падежа), что создаёт определённые трудности в процессе анализа. Неоднозначность граммем в ходе автоматического синтаксического анализа иногда приводит к возникновению синтаксической омонимии и построению альтернативного синтаксического варианта (графа синтагм). Взаимовлияние двух уровней анализа, морфологического и синтаксического, состоит в том, что граммема, будучи эффективным способом поиска морфологической зависимости, а точнее реализации синтаксического отношения, может быть однозначно проинтерпретирована только путём фиксации этого отношения.

2. **Селективные признаки.** Классифицирующие (селективные) признаки приписываются лексемам в грамматическом словаре. Наиболее важной для синтаксиса является классификация лексем по категориальным признакам: существительное, глагол, прилагательное и т. д. *Категориальные признаки задают потенциальных участников синтаксической связи и определяют, в большинстве случаев, вершину в структуре, а также оговаривают понятия управления и согласования.*

3. **Служебные слова.** Предлоги, союзы, частицы и др. – способы, которые служат опорными точками анализа. Так, например, союз может быть использован для определения поверхностного типа сегмента, предлоги формируют актанта глагола.

4. **Разделительные знаки (операторы).** Запятая, тире, точка, знак вопроса и др.

В первую очередь операторы определяют границы как сегментов, так и всего предложения. Тире является выражением эллиптированного элемента предложения и часто используется в стратегиях поиска неморфологического предиката. Анализ бифункциональности оператора (когда, например, оператор есть одновременно и правой границей сегмента и оператором создания слов) – одно из наиболее трудных заданий сегментации, которая задаёт рекурсивный характер, как грамматических стратегий анализа, так и методов программной реализации.

5. Порядок слов. Линейное расположение слов в предложении вместе с селективными признаками имеет доминирующее значение в проектировании синтаксических анализаторов языков с бедной морфологией (английский). Во многих системах английского синтаксиса порядок слов задаёт направление поиска хозяина или слуги для каждого класса лексем и типа связи. Для флективных языков этот способ анализа имеет второстепенное значение, хотя и используется в ряде случаев для установления синтаксических связей или оценки омонимических структур предложения.

Невзирая на свободный порядок слов в украинском (русском) языке, некоторые синтаксические зависимости имеют обязательным критерием выделение жесткого линейного порядка: генитивное определение должно следовать за словом («ножка стол-а», «сын отц-а»); местоимение предшествует существительному («на стол-е», «у отц-а»). Линейный порядок предложения в автоматическом синтаксическом анализе используется как показатель наиболее вероятного направления поиска слуги или хозяина, и лишь в отдельных случаях – как обязательный критерий установления синтаксической зависимости.

Для проверки и установления возможного синтаксического отношения используются синтаксические правила и стратегии. Каждое грамматическое правило устанавливает один тип синтаксического отношения $R(A, B)$ между двумя единицами анализа и однозначно задаёт вершину. Количество типов синтаксических отношений, а также их названия зависят от прикладной модели и конкретной реализации модели синтаксического разбора.

Набор *универсальных синтаксических отношений* для русского языка (может быть использован и при анализе украинского языка) приведен во многих теоретических работах, например, [137]: предложное (существительное и предлог, который им управляет), определительное (существительное и его согласованное определение), посессивное (существительное и его несогласованное определение), субъективное (сказуемое и подлежащее). В роли единиц анализа на месте A и B , где A – вершина анализа, а B – зависимая единица анализа, могут выступать как отдельные словоформы, так и целые группы (фразовые составляющие). Заполнение A и B во многом зависит от синтаксического аппарата, принятого в анализаторе для описания структуры. *Идеальное грамматическое правило* в автоматическом синтаксическом анализе характеризуется следующими категориями:

- 1) описывает только один тип синтаксического отношения;
- 2) однонаправленность анализа, то есть, зависимое B может находиться только слева или только справа от вершины A ;
- 3) не содержит рекурсивных вызовов или вызовов других правил;
- 4) обрабатывает только компактно расположенные единицы анализа;
- 5) результат не зависит от порядка применения правил.

Использование грамматических правил задаёт прозрачность архитектуры процессора и обеспечивает стойкость системы к изменениям. Перечисленные критерии не являются строгими, но приближают правило к его идеальной форме.

Результатом синтаксического анализа является построенный ориентированный граф сегментов предложения украинского или русского языка.

5.3.4. Семантический анализ

Под автоматическим семантическим анализом будем понимать совокупность методов, реализованных в виде однозначной формальной процедуры, с помощью

которой возможно с достаточной точностью представить содержание любого высказывания на естественном языке в виде последовательности символов, которые образуют некоторую формальную систему.

Наиболее простым и универсальным средством представления содержания некоторого текста в компьютерной системе есть *семантическая сеть*. В общем виде она представляет собой ориентированный граф, вершины которого обозначают сущности (объекты), а дуги – отношения (связи) между ними. Имена вершин и дуг обычно совпадают с именами соответствующих сущностей и отношений, используемых в естественном языке. Дуга и две связанные с ней вершины представляют минимальную значимую информацию – факт наличия связи определённого типа между соответствующими объектами.

Выделяют два подэтапа в проведении семантического анализа. *Первый* подэтап назван поверхностным семантическим анализом. На этом подэтапе осуществляется устранение синтаксической омонимии (если она существует), формирование понятий, отношений и их характеристик (свойств) в пределах одного предложения. Кроме того, на этом этапе происходит замещение анафорических связей (в частности, когда вместо понятия подставляется местоимение), обобщение понятий и отношений, пропусков понятий и отношений, свойственных естественному языку. *На втором* подэтапе семантического анализа строится собственно онтолого-семантическая структура текста.

Согласно с [144], из множества слов русского языка (и украинского языка также) по семантическим признакам можно выделить следующие категории: слова с моделью управления (предикаты) и слова без модели управления (понятия, характеристики и их значения, числители).

Наиболее важной является обработка предикатов действия и их возможных актантов – понятий, характеристик, их значений, а также числителей. Обработка функциональных и пустых предикатов может быть сведена к модификации фрагментов семантической сети, построенных для зависимых от них предикатов действия.

Каждый предикат действия имеет одну или несколько моделей управления. Модель управления налагает синтаксические и семантические условия на возможные актанты (аргументы) данного предиката и отмечает их семантические роли относительно предиката. В упрощённом виде *модель управления* можно представить как таблицу, каждая строка в которой определяет один из возможных актантов. Эта строка содержит: предлог (если есть), часть речи и падеж актанта, его семантическое метапонятие (категорию), семантическую роль в предикате и признак обязательности данной роли. Анализ моделей управления позволяет естественным образом перейти от синтаксического дерева зависимостей к фрагменту семантической сети. Для этого в простом случае достаточно: 1) для каждого предиката отыскать в дереве зависимостей все присутствующие актанты; 2) создать по одной вершине на каждый актант и ещё одну – для самого предиката; 3) провести от вершины-предиката дугу к каждой вершине-актанту; при этом название дуги, (её семантическая роль), выбирается из модели управления в зависимости от актанта.

Если модель управления не допускает существования актанта ни в одной семантической роли, остаётся две возможности: либо данная текстовая единица действительно является относительно предиката не актантом, либо имеет место ошибка синтаксической структуры. Правильно различать эти два случая удаётся не всегда, но можно указать условие, определяющее, с большой долей вероятности, что присутствует именно ошибка синтаксической структуры. Имеется в виду ситуация, когда актант отвечает морфологическим условиям модели управления, а семанти-

ческим – нет. Поскольку синтаксический анализатор при группировании не проверяет семантические признаки, он не может самостоятельно выявлять ошибки такого типа. Таким образом, если ошибка выявлена на этапе семантического анализа, необходимо вернуться назад к синтаксическому анализу и попробовать найти другой вариант синтаксической структуры, который бы не содержал неверного актанта в поддереве зависимостей данного предиката.

Семантический анализ способен выявлять некоторые ошибки синтаксической структуры (например, связанные с неверной моделью управления). Таким образом, существование *обратной связи* между синтаксическим и семантическим анализом – необходимое свойство *лингвистического процессора*, поэтому целесообразно объединить эти два этапа анализа и выполнять их совместно в одном аналитическом блоке.

Как было отмечено ранее, существует обратная связь между семантическим и синтаксическим анализом, поэтому можно объединить эти две процедуры в одном блоке. Конечным заданием процедуры синтаксического анализа есть представление каждого предложения заданного ЕЯТ в виде синтаксического дерева (лексем предложения с семантико-синтаксическими отношениями между ними).

Связывание слов в предложении происходит постепенно – от словосочетания до группы связанных слов и, наконец, к объединению всех групп предложения в одну структуру. Для установления связи между отдельными словами используются естественно-языковые средства выражения семантических и синтаксических отношений. Во флективных языках такими средствами являются изменяемые части полных слов и служебные слова.

Такие сегменты словосочетаний, которые кодируют отношения между полными словами, называются *синтаксическими определителями* [140]. Поскольку одному синтаксическому определителю может отвечать несколько синтаксических отношений, для однозначности определения связей между словами вводится понятие *корреляторов* [140], которые дополнительно учитывают семантические признаки между словами.

Исходные данные для блока семантического анализа:

- результаты предыдущих этапов анализа ЕЯТ (графемного и морфологического);
- словарь основ (содержит основы слов и их семантические признаки);
- список всех возможных флексий слов;
- база данных с определителями (содержит синтаксические определители и списки корреляторов для каждого из них);
- база данных с корреляторами (каждый коррелятор состоит из названия отношения и списка пар семантических признаков, между которыми это отношение может существовать).

Рассмотрим основные этапы работы блока синтактико-семантического анализа ЕЯТ.

Первый этап. Используя словарь основ и список флексий, в каждом слове предложения выделяется его постоянная (основа) и флективная составляющие. Производится классификация слов по семантическим признакам соответствующих основ в словаре. При этом возникает задача возможной неоднозначности выделения основы слова и определения его семантического признака. Возможными путями её решения являются учёт характеристики слов, стоящих рядом в предложении, и расширение словаря основ дополнительными характеристиками.

Постоянная и флективная части слова, его семантический признак могут быть получены и на предыдущих этапах анализа (этап морфологического анализа). Синтаксические характеристики в таблицах ЛБД, например, для глаголов, вклю-

чают в себя лексико-семантические *валентности* глаголов, определяющие признаки наиболее вероятного окружения, и признаки семантических интерпретаций, которые на основе распознанных синтаксических правил определяют семантические отношения между словами в словосочетании. Если предложение состоит из нескольких синтагм, то благодаря интерпретации определённой части базы знаний синтаксиса, происходит их объединение в единое синтаксическое дерево с соответствующими семантико-синтаксическими отношениями.

Второй этап. Связывание слов в предложении целесообразно начинать со словосочетания, определяющего главное отношение (отношение между подлежащим и сказуемым) в этом предложении. В случае, когда такое словосочетание установить невозможно, предложение анализируется слева направо, начиная с первых полнозначных слов. Для выбранного слова формируется синтаксический определитель, состоящий из служебных слов и флективных частей полнозначных слов словосочетания. Например, для выражения «права и свободы» таким определителем будет конструкция типа «-а и -ы». Если сформированный определитель существует в базе данных определителей, ему будет соответствовать список коррелятов. Тогда, учитывая семантические признаки слов в словосочетании, в базе данных корреляторов находится нужный коррелятор, который и установит связь между словами. Однозначность определения такой связи обеспечивается тем, что для отдельного определителя множества пар семантических признаков для корреляторов из его списка не пересекаются.

Дальше к словосочетанию постепенно присоединяются прилегающие полнозначные слова предложения, путём установления связи между новым словом и одним из слов обработанной части предложения. Таким образом, создаётся группа связанных слов. Важным является выбор слова из группы связанных слов, которое будет связываться со следующими словами. Это должно быть или слово с главным отношением, или последнее слово группы. В случае, когда невозможно установить связь между новым словом и словами группы, создаётся новая группа связанных слов. В конце анализа необходимо попробовать соединить все созданные группы в одну, отражающую структуру связей между всеми словами предложения.

Невозможность установления связи между отдельными группами в предложении и невозможность их объединения свидетельствует либо о сложном предложении, части которого не связаны (или связаны неявно), либо о некорректных связях между словами в группах. Для того, чтобы избежать проблемы некорректного связывания слов в принципе необходимо проводить дополнительный анализ возможных связей каждого следующего слова со словами группы и выбирать наиболее вероятный, или же рассматривать все возможные варианты связей (однако, следует учитывать рост количества таких вариантов для каждого следующего слова).

Результатом работы блока семантического анализа является понятийная (логико-семантическая) структура ЕЯТ. Для построения такой структуры вначале необходимо сформировать систему понятий, отношений и определить их характеристики.

Одной из важных является задача *анафорических связей* или замены местоимений в тексте на соответствующие понятия (как правило, существительные). Замещению подлежат некоторые личные, относительные, указательные, притяжательные и обратные местоимения. Алгоритм замещения строится на основе анализа закономерностей (отражённых в базе знаний синтаксиса ЕЯ) использования местоимений в ЕЯ и различается в зависимости от типа и грамматических характеристик местоимения.

Из практики известно, что понятием, которому отвечает местоимение, является согласованное с ним по грамматическим характеристикам ближайшее полное слово, стоящее впереди, или слово, входящее в ядро данного или предыдущего предложения. Поэтому входными данными для проведения замены местоимений являются грамматические характеристики слов (полученные вследствие морфологического анализа) и синтактико-семантические связи между ними (результат работы синтаксического блока). Замещение местоимений позволяет получить дополнительные связи между соответствующими понятиями в предложении и во всём тексте, поэтому после этого целесообразно провести повторный синтаксический анализ отдельных предложений.

Следующей задачей семантического анализа есть *выделение терминов* естественно-языкового объекта, из которых потом формируются понятия. Рассматриваются термины-существительные как однословные, так и словосочетания. При автоматическом выделении терминов анализируются такие данные:

- семантические признаки слов ЕЯТ;
- синтактико-семантические отношения между словами в предложении (для терминов, выраженных словосочетанием);
- отдалённость термина от ядра фразы;
- частота использования термина в ЕЯТ;
- лингвистическая составляющая ПдО, которой принадлежит ЕЯТ.

Качество такого выделения терминов в значительной мере зависит от качества работы синтаксического блока, а именно от корректности классификации слов по семантическим признакам и установленных связей между словами.

Логико-семантические отношения в ЕЯТ

Для выявления логико-семантических отношений в ЕЯТ используются специальные диагностические конструкции фраз и установленные на этапе синтаксического анализа семантико-синтаксические связи между словами. Основные типы отношений логико-семантической структуры следующие: отношения тождества, включения, пересечения и дизъюнкции, иерархические отношения, отношения агрегации и функциональные отношения [145].

Иерархические отношения принадлежат к области, которая может быть выражена семантикой глагола «быть», и делятся на три подвида: род – вид; признак – значение; инвариант – вариант. Отношения *род – вид*, в свою очередь, делятся на структурные, функциональные и семантические.

Для распознавания *родо-видовых отношений* используют следующие основные конструкции фразы:

- «*A* принадлежит *B*», «*A* есть *B* с признаками», «рядом с *A* есть *B*», «*A* принадлежит к классу/семейству *B*» – для поиска родового понятия *B* и видового понятия *A*;
- «*A* имеет следующие виды: *B*₁, *B*₂...», «к видам *A* принадлежат *B*₁, *B*₂...» – для поиска видовых понятий *B* и родового понятия *A*.

Диагностические конструкции отношения *признак – значение* (для поиска *A* – понятия-сущности, и *B* – имени-сущности) следующие: «*A* – *B*», «*A* имеет имя *B*», «*A* именуется *B*», «*A* называется *B*».

Отношения *инвариант – вариант* находят с помощью фраз «к вариантам *A* принадлежат *B*₁, *B*₂...», и «инвариантом *A* есть *B*».

Отношения агрегации принадлежат к области, которая может быть выражена семантикой глагола «иметь». Выделяют четыре подвида: 1) целое – часть (компонент); 2) объект – пространство реализации объекта (локализации, позиции); 3) объект – свойство/признаки; 4) уровень – единица уровня.

Основные диагностические конструкции для поиска отношений агрегации:

Для подвида 1):

– для целого B и части A :

« A входит в состав B », « A составляет часть B », « A есть частью B »;

– для части B и компонента A :

« A состоит из B -в», « A содержит в себе B как часть», « A содержит в своём составе B как часть».

Для подвида 2):

– «место $A - B$ », « A происходит в B », « A находится в B », « A входит в B ».

Для подвида 3):

« A имеет свойство B , имеет B как существенный признак», « A характеризуется наличием B », «для A характерно B »;

Для подвида 4):

« A принадлежит к уровню B », « A – единица уровня B », « A рассматривается на уровне B ».

Функциональные отношения – это отношения, которые принадлежат к сфере описания процессов. Такое отношение представляет собой *многместный предикат*, описывающий некоторый процесс. В этом предикате выделяют следующие переменные:

A – субъект некоторой сущности; B – инструмент, способ, метод или алгоритм выполнения совокупности операций процесса; C – начальный объект или исходные данные; D – конечный объект или конечные данные; E – событие, активизирующее операцию.

Основные диагностические выражения:

– для выявления субъекта: « A осуществляет B »;

– для инструмента, способа, метода или алгоритма – здесь могут быть указаны разнородные соответствующие термины;

– для выявления начального объекта: « A выполняется над B », « A применяется к B », « A изменяет B », « A B -е»;

– для выявления конечного объекта: «вследствие A получается B », «вследствие A образуется B », « A приводит к B », « A преобразует B_1 в B_2 ».

6. МЕТОДОЛОГИЧЕСКИЕ ОСНОВЫ ПРОЕКТИРОВАНИЯ ЗОИС С ОБРАБОТКОЙ ЕЯО

6.1. Онтологический подход к проектированию ЗОИС

К онтологическим аспектам относится широкий круг вопросов: от формального построения и описания компьютерных онтологий до области их применения. Главный вектор исследований направлен на формализацию этапов построения, структурирования и представления знаний предметной области и интегрированно с ней проблемного пространства. В свою очередь, эффективная реализация указанных этапов и получение конечного результата (в виде базы знаний ПдО) невозможно без проведения системно-онтологического анализа заданной ПдО [2].

Онтологический подход (ОнП) к проектированию информационной и архитектурной компонент компьютерной системы возник как междисциплинарная проблема формирования, представления, обработки и анализа знаний, модели которых описывают структуру и взаимосвязи объектов соответствующих предметных областей.

Будем полагать, что ЗОИС с обработкой знаний, содержащихся в ЕЯО, спроектирована на основе *онтологического подхода*, если она обладает следующими характерными чертами:

1. Компьютерные онтологии обеспечивают эффективную машинную обработку языковых и предметных знаний (в [49] такие онтологии названы соответственно языково-онтологической картиной мира и онтологией ПдО).
2. В отличие от эмпирического подхода, являющегося основой “творчества и мудрости”, системно-онтологический подход [2] предполагает строгую (насколько это возможно) систематизацию знаний любого уровня, в том числе категориального. Последний представляется категориальной решёткой, или *онтологией верхнего уровня*. Её проектирование входит в общий алгоритм синтеза ОУИС.
3. Архитектурно-технологические особенности ЗОИС:
 - онтолого-управляемая архитектура (ontology-driven architecture – ODA) характеризуется высоким уровнем формализации представления О ПдО и эффективными механизмами онтолого-управления, в том числе поддерживающими развитие системы;
 - высокая степень интеграции междисциплинарных знаний.
4. Использование средств поддержки автоматизированного построения онтологий ПдО (методика, технология и программно-аппаратная реализация).
5. Прикладная направленность и сильная взаимосвязь технологии обработки информации с архитектурно-структурной организацией ОУИС.
6. Проектирование ОУИС выполняется на основе принципов, методов и механизмов онто-логического подхода (ОП) [49].
7. Функционирование ОУИС в двух режимах: накопления предметных онтологических знаний и их обработки.

6.2. Онто-логический подход к разработке архитектуры ОУИС

Ниже излагаются основы так называемого онто-логического подхода проектирования ОУИС (далее для краткости будем называть его онтологическим). Такой подход учитывает логико-информационную и онтологическую концепции проектирования, а также виртуальную парадигму, при которой архитектура компьютерной системы ориентирована на технологию реконфигурируемого процессинга. Последняя обеспечивает адаптивность системы благодаря наличию в ОУИС архитектурных и технологических возможностей настройки в условиях априорной и текущей неопределённостей на основе обучения и опыта.

Проектирование ОУИС предполагает разработку двух взаимосвязанных подсистем, соответственно для обработки знаний в заданной ПдО и обработки текстов на основе “языковых” знаний (или знаний из ПдО “Компьютерная лингвистика”). Указанные информационные системы на рис. 1.12 и 1.13 представлены соответственно как онтолого-управляемая ИС обработки знаний в заданной ПдО и языково-онтологическая ИС обработки ЕЯ-информации на основе языковых знаний. Взаимодействие между указанными подсистемами осуществляется путём реализации процедуры некоторого полужормального отображения G , использующего базу языковых знаний (в основе которой лежит языково-онтологическая картина мира) [10, 28], подходящую формальную логическую теорию первого порядка и базу знаний заданной ПдО.

Схема обработки предметных знаний, содержащихся в ЕЯО, представлена на рис. 6.1.

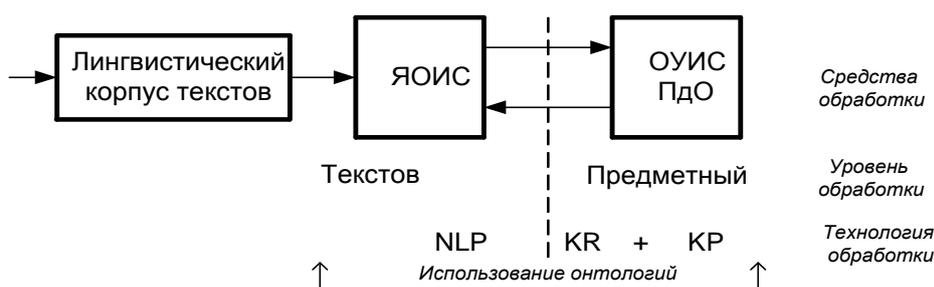


Рис. 6.1. Схема обработки знаний, содержащихся в ЕЯО

Онто-логический подход в онтолого-классификационной схеме (см. 1.4) берёт своё начало от двух подходов – логико-информационного (ЛИП) [18, 19] и онтологического (см. 6.1). Кратко рассмотрим суть *логико-информационного метода* разработки и оценки эффективности системы управления информационной системы с программируемой архитектурой.

В соответствии с логической концепцией логико-информационной модели процессор представляется известной композицией операционного и управляющего автоматов. Последний моделирует иерархическую систему управления современных информационных систем, и, в свою очередь, описывается многоуровневой композицией программируемых автоматов.

В соответствии с информационной концепцией процессор рассматривается как информационная система, вся информация в которой отнесена к трём “сферам” состояний: хранения, транспортировки и преобразования. Очевидно, что при опре-

делённых соотношениях между объектами информации в этих сферах можно получить оптимальные технические параметры информационной системы.

Под уровнем управления $\tau_i (\forall i = 0 \div h)$ понимается совокупность структурных и информационных (программных) средств представления и интерпретации операторов языка соответствующего уровня системы, а также формирования необходимой последовательности их выполнения в соответствии с заданным алгоритмом. Наиболее известными и распространёнными в архитектурах современных процессоров и информационных систем являются микропрограммный, программный и алгоритмический уровни.

Таким образом, формальное описание модели можно представить в виде

$$\forall_{i=0,h} \tau_i = \langle A_i, \Lambda_i, R_i \rangle, \quad (6.1)$$

где A_i – множество алгоритмов, реализуемых и записанных в памяти на i -м уровне; Λ_i – множество операторов программирования i -го уровня, в терминах которых представлены эти алгоритмы; R_i – множество информационно-кодовых представлений операторов i -го уровня.

Оптимальной считается такая структурная реализация модели информационной системы, для которой в соответствии с принятыми критериями найдены оптимальное количество уровней и оптимальные соотношения между определёнными характеристиками компонент на каждом уровне, а также соответствующими характеристиками компонент соседних уровней.

Общий подход к проектированию архитектуры информационной системы в соответствии с моделью (6.1) описывается следующей схемой:

$$\left(\forall i = \overline{0 \div h} \right) \exists \Omega: \begin{array}{c} A_i \Rightarrow \Lambda_i \Rightarrow R_i \\ \downarrow \\ A_{i-1} \Rightarrow \Lambda_{i-1} \Rightarrow R_{i-1} \\ \vdots \\ \downarrow \\ A_0 \Rightarrow \Lambda_0 \Rightarrow R_0 \end{array} \quad (6.2)$$

где Ω – символ отображения алгоритмического описания информационной системы на структурное. Среди всех вариантов представляет интерес такое Ω , при котором оптимизируется интегральный критерий θ , учитывающий заданные характеристики качества архитектуры (аппаратные затраты, объём информации, быстродействие, функциональные возможности и пр.)

Схема (6.2) основывается на следующих утверждениях. При заданных критериях, во-первых, существует оптимальная связь между алгоритмами и операторами, в которых они представлены на каждом уровне, а также операторами и их информационно-кодовыми эквивалентами; во-вторых, оператор любого уровня можно представить в виде алгоритма, заданного на множестве операторов низшего уровня; в-третьих, от информационно-кодовых представлений операторов на всех уровнях существует непосредственный переход к структурным компонентам и параметрам информационной системы.

Схема (6.2) является ядром формализованной методики проектирования, которая представляет собой систему взаимосвязанных алгоритмов проектирования. Основные из них относятся к системному, алгоритмическому и логическому этапам проектирования.

В данной работе логико-информационный подход модифицируется в соответствии с принципами и механизмами онтологической концепции.

По сути, онтологическая модель ОУИС ориентируется на предварительно “онтологизированные” (на основе системно-онтологического анализа [2]) сферы обработки, хранения и транспортировки информации, являющиеся базовыми понятиями в логико-информационной модели. При этом условно можно считать, что ОНП ответственен за “операционную часть” ОУИС, а ЛИП – за “управляющую”.

“Онтологизация” сфер переработки информации имеет следующие характерные черты:

- в *сфере хранения* – память ОУИС проектируется с учётом онтологической информации – сегментируется на память библиотеки онтологий предметных областей, библиотеки онтологий текстовых документов, общую базу фактов и правил вывода предметных областей;
- в *сфере транспортировки* речь идёт о сложной траектории естественно-языковой информации, как носителя знаний, которой присущи разного рода неоднозначности (неполнота, неточность) и, прежде чем перейти к машинной обработке этих знаний, информация “фильтруется” (итерационно) лингвистическими онтологиями языковых и предметных знаний до тех пор, пока не будут устранены все виды неоднозначности;
- в *сфере обработки* существенной является ориентация на формализованные предметные знания. Верхним уровнем формализации являются онтологии проблемных доменов, в общем случае междисциплинарного характера, основанные на системной интеграции онтологических знаний исходного множества предметных областей.

Важной отличительной особенностью ОП по сравнению с ЛИП (кроме онтологизации) является добавление ещё одного (предметного) уровня управления ОУИС.

Входной информацией для ОУИС является множество ЕЯО, и первым этапом её обработки является графемно-морфологический анализ. На этом этапе выполняется совокупность программных процедур, алгоритмически не сложных, но занимающих львиную долю времени обработки. Существенно повысить производительность (на два, три порядка) системы для указанного вида лингвистического анализа возможно, осуществив переход с программного уровня интерпретации на аппаратный уровень. И далее, анализ показал, что принципиально возможно, учитывая современные архитектурные характеристики микроэлектронной базы (ПЛИС) и возможности САПР для проектирования вычислительных устройств на ПЛИС, реализовать все три этапа лингвистического анализа (графемно-морфологический, синтаксический и поверхностно-семантический) на аппаратно-микропрограммном уровне. При этом выполняется анализ входной ЕЯ-информации не пословно, а по предложениям. Предложение является минимальной синтаксической единицей, для которой может быть выполнен полный синтаксический и поверхностно-семантический анализ.

Одним из вариантов аппаратной поддержки наиболее трудоёмких процедур является реализация системы в виде автоматной сети, включая вариант комбинационных автоматов без памяти [146] на базе адаптивных логических сетей. ЛИП, а

также известные его модификации доказал свою эффективность при решении задач архитектурного проектирования компьютерных систем [146, 147].

В схеме (6.2) для онтолого-управляемых архитектур $\tau_0, \tau_1, \tau_2, \tau_3, \tau_4$, – уровни управления знаниями:

τ_0 – физический уровень;

τ_1 – микропрограммный уровень;

τ_2 – программный уровень;

τ_3 – алгоритмический уровень;

τ_4 – предметный уровень;

Под *уровнем управления* понимается совокупность информационных и структурных средств представления и интерпретации операторов языка соответствующего уровня, а также средств формирования необходимой последовательности их выполнения в соответствии с заданным на этом языке алгоритмом. Каждый из уровней управления, таким образом, соответствует определённому уровню языка:

$$\tau_1 \rightarrow \Lambda_1, \tau_2 \rightarrow \Lambda_2, \tau_3 \rightarrow \Lambda_3, \tau_4 \rightarrow \Lambda_4.$$

Структурное представление онто-логической (О) модели

На рис. 6.2 приведено структурное представление О-модели, поясняющее её логическую и информационную концепции. Каждый из уровней управления ($\tau_1, \tau_2, \tau_3, \tau_4$) в соответствии с информационной концепцией представлен преобразователем информации $Pr_i, \forall i = \overline{1 \div 4}$ (сфера обработки) и узлом памяти P_i (сфера хранения). Совокупность указанных четырёх уровней управления представляет управляющий автомат A . Операционный автомат B (исполнительный уровень τ_0) представлен двумя составляющими Pr_0 и P_0 , что соответствует современной тенденции построения операционных устройств.

Известно, что в процессе создания ОУИС главными задачами являются разработка методов и подходов систематизации знаний, что предполагает разработку формальной теории, в том числе языка представления знаний (ЯПЗ), средств обработки (интерпретации) знаний и механизмов вывода новых знаний.

Анализ процесса исследования ПдО [3, 41] начинается, прежде всего, с фиксации некоторой части объективной реальности (предметной области), с последующим построением её обобщённой модели (класса IDEF0). На следующем этапе – выявление наиболее существенных черт, явлений ПдО, определение элементарных понятий и их взаимосвязей, аксиом и ограничений, управляющих развитием данной ПдО, т. е. создание некоторого образа реальности, отражающего “нестрогое” субъективное её восприятие исследователем. Для фиксации приобретаемых знаний в виде текста в рамках некоторой знаковой системы необходимо определить подходящий формализм, обеспечивающий наглядное и удобное их представление. Он включает собственно язык, по возможности более строгий, чем естественный, и в то же время не слишком отличающийся по восприятию от последнего, а также метод (методы), позволяющий осуществить эту формализацию. В качестве такового может быть сочетание алгебро-логического и аксиоматического методов.

Следующий этап процесса проектирования заключается в построении формальной теории исследуемой ПдО. Эта теория подлежит интерпретации, то есть каждому выражению теории, иначе, каждому понятию необходимо сопоставить

конкретный объект реального мира, а каждому суждению – связи между реальными объектами, соответствующими входящим в данное суждение понятиям, и т. п. После этого можно определить истинность или ложность высказываний теории, то есть судить о правильности рассуждений о предметной области. Далее можно исследовать построенную модель в строгом научном смысле, ставить по отношению к ней вопросы относительно её полноты, непротиворечивости, разрешимости, проследить её эволюцию и т. п. и, наконец, интерпретировать полученные знания в заданной ПдО.

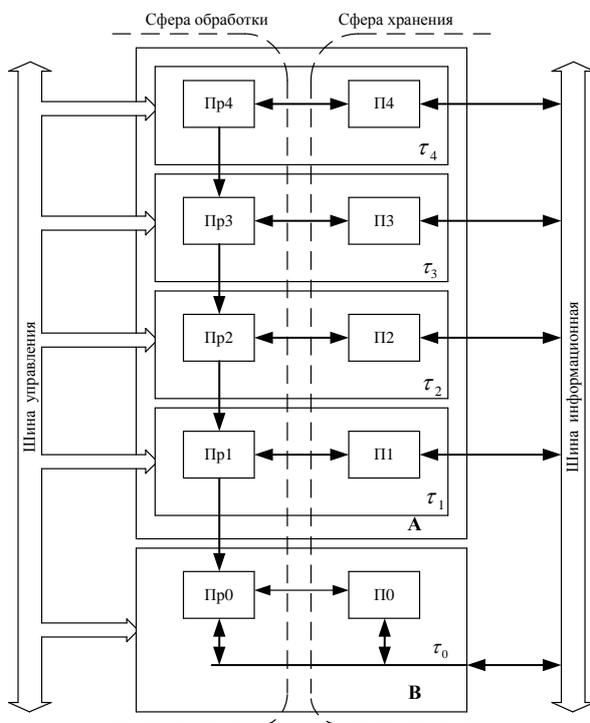


Рис. 6.2. Структура О-модели с программируемой архитектурой

Процесс синтеза структуры ОУИС описывается блок-схемой, представленной на рис. 6.3.

Неотъемлемой компонентой любой ОУИС является наличие в ней базы знаний, в которой в формализованном виде представлены знания предметной области. Именно наличие этих знаний в системе позволяет ей успешно решать как традиционные, так и новые (проблемные) задачи, которые ранее были исключительно прерогативой научного творчества. В процессе проектирования БЗ всё чаще применяется онтологический подход и онтологическое описание предметной области. Возникло самостоятельное ответвление в исследованиях по построению баз знаний – “*онтологические базы знаний*”. Онтологические знания одновременно могут выступать как информационная структура концептуальных знаний ПдО и как один из главных компонентов ОУИС. Указанное функциональное сочетание отображает указанную выше двойственную парадигму и два аспекта проектирования ОУИС – информационный объект и инструментальное средство обработки.

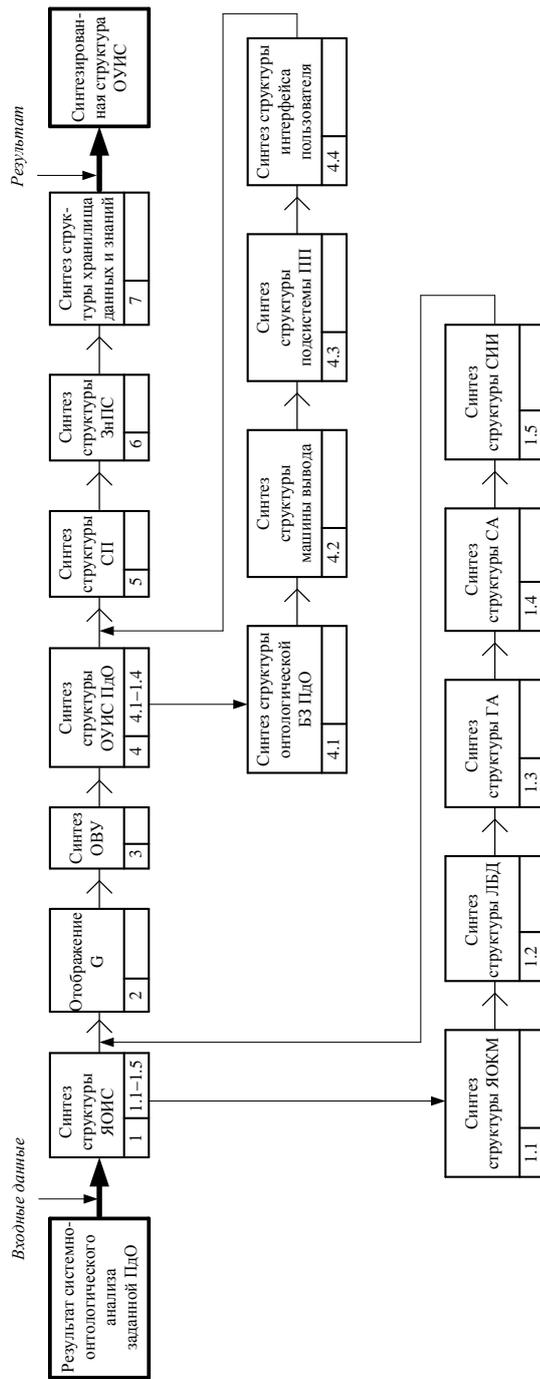


Рис.6.3. Блок-схема процесса синтеза структуры ОУИС

Схема модели ОУИС и критерии её эффективности

Модель проектируемой ОУИС для решения типового набора задач в заданной предметной области можно представить следующей системой

$$S = \langle M, A, X, P \rangle, \quad (6.3)$$

где: M – множество математических методов, лежащих в основе решения типового набора задач в заданной ПдО;

A – множество алгоритмов, реализующих множество методов M ;

X – множество объектов и процессов, описывающих заданную ПдО и участвующих в реализации алгоритмов $\{A_i\}$ решения m -го типового набора задач;

P – процедура онтологического описания объектов, процессов и задач заданной ПдО.

Абстрактная архитектура ОУИС описывается тройкой

$$A^{ОУИС} = \langle A^{ЯОИС}, A^{ОУИС ПдО}, G \rangle,$$

где $A^{ЯОИС}$ – архитектура подсистемы обработки ЕЯО на основе языковых знаний;

$A^{ОУИС ПдО}$ – архитектура подсистемы машинной обработки знаний в заданной ПдО;

G – отображение результата обработки ЕЯО в формализмах некоторой подходящей теории (например, логики первого порядка).

Для ОУИС процесс реализации О-метода и обобщённого алгоритма решения типового набора задач носит итеративный характер. Критериями эффективности указанного метода (алгоритма) являются обобщённые характеристики производительности, программно-аппаратных затрат, точности решения задачи, сложности алгоритма, надёжности проектируемой системы либо специальные критерии, такие как трудоёмкость реализации О-метода и обобщённого алгоритма, полнота обоснованных рекомендаций для принятия решений, уровень автоматизации построения онтологической базы знаний ПдО, ограничения реального времени получения результата и др.

Результат проектирования образуется на основе комплексного набора методов анализа ЕЯО, классической логики первого порядка для формально-логического представления ЕЯО, структуризации онтологических знаний, а также процедур прикладного процессинга и соответствующих архитектур проектируемых ОУИС, выбора подходящей пары (метод – архитектура) для конкретного круга (типичного набора) проблемных задач.

6.3. Свойства онтолого-управляемой архитектуры

В многочисленных зарубежных публикациях под *онтолого-управляемой* (ОУ) – *архитектурой* понимают такую архитектуру ЗОИС, в которой формальная онтология ПдО совместно с другими компонентами формирует механизм реализации конкретного приложения, например онтолого-управляемой системы информационного поиска, онтолого-управляемого программного комплекса, ОУ электронного обучения и др. При этом, О ПдО описана на некотором языке описания онтологий (список таких языков определён, в частности, консорциумом W3C, например (OWL, RDFS и др.).

ЗОИС иногда классифицируют [90] по функциональному признаку на онтолого-знающие информационные системы (ОЗНИС) и онтолого-управляемые информационные системы. ОЗНИС характеризуется “пассивной” ролью онтологии,

т. е. выполнением ею роли справочного пособия. Любая другая подсистема только “знает” о существовании (возможно отдалённо) онтологии и может использовать её для любой специфической цели приложения. В ОУИС онтология играет “активную” роль, кооперируется с другими подсистемами в текущем времени в соответствии с “высшей” общей целью ЗОИС. Важным следствием использования онтологии в реальном времени является способность коммуникации между программными агентами. Последние поддерживают связь друг с другом через сообщения, которые содержат выражения, сформулированные в терминах онтологии (*ОУ-коммуникация*). Для того чтобы программному агенту “понять” значения этих выражений, ему необходим доступ к онтологии [3].

Известна и другая трактовка онтолого-управляемой архитектуры. Это новая концепция разработки программ – MDD (Model Driven Development – разработка, управляемая моделями) и вариант её реализации – MDA (Model Driven Architecture – архитектура, управляемая моделью, или модельно-управляемая архитектура), предполагающая разработку архитектуры программного комплекса, ориентированную на предметную область и не зависящую от конкретной платформы программирования. Развитием данной концепции является предлагаемый консорциумом W3C подход к разработке программного обеспечения, в частности, серверных приложений. В результате образуется архитектура, управляемая онтологией [148].

Использование онтологии в качестве компонента интерфейса пользователя не столь очевидно. При этом онтология предоставляет семантическую информацию относительно ограничений, наложенных на классы и отношения. Пользователь может ознакомиться с содержанием онтологии, чтобы лучше понять словарь, используемый ЗОИС, и сделать корректные запросы к базе данных (знаний).

Использование онтологии в качестве компонента прикладной программы также является важной частью многих ЗОИС. В этих компонентах закодированы фрагменты знаний ПдО, которые, по различным причинам, явно не сохранены в базе данных. Возможно, в принципе, генерировать статическую часть программы, опираясь на содержимое онтологии.

Кроме того, онтологии, интегрированные с лингвистическими ресурсами, могут быть использованы для поддержки развития объектно-ориентированного программного обеспечения. Можно также представить все знания предметной области неявно закодированными в прикладной программе, преобразуя программу в систему, основанную на знаниях. В этом случае база знаний содержит ядро, представленное в виде онтологии.

Выше приведено общепринятое (по крайней мере, в зарубежных публикациях) описание онтолого-управляемой архитектуры для ИС различного назначения. Для ЗОИС с обработкой ЕЯО оно является фундаментальным, основано на свойстве активности знаний, в том числе онтологических. Специфика обработки языковых и предметных знаний предполагает включение в архитектуру ОУИС соответствующих онтологий, которые в свою очередь упорядочиваются в соответствии с онтологической иерархией. И главной особенностью архитектуры ОУИС, подчёркивающей её онтолого-управляемость, является явное представление процедурных знаний в виде онтологии задач и процедур их решения [2].

На рис. 6.4 представлена схема фрагмента архитектуры ОУИС, отображающая только её “онтологическую” компоненту. Это “ядро” системы, между которым устанавливаются эксплицитные связи с процессами и процедурами (или технологической цепочкой) ИИТ обработки знаний, содержащихся в ЕЯО, с приложениями (через онтологию задач), с общей (категориальной – ОВУ) и языковой (ЯОКМ) картинами мира. Здесь: $ОД_k$ – онтология k -го домена предметных областей, где

$k = \overline{1, K}$, $K = \text{Card } \text{ОД}$ – мощность множества онтологий доменов, интегрированных в библиотеку; O_i – онтология, где $i = \overline{1, I}$, $I = \text{Card } \text{ПдО}$ – мощность множества онтологий предметных областей, интегрированных в библиотеку; OZ_i^j – онтология j -го типового набора задач i -ой предметной области, где $j = \overline{1, J}$, $J = \text{Card } \text{ОЗ}$ – мощность множества типовых наборов задач i -ой предметной области; $ЛО_i$ – лингвистическая онтология i -ой предметной области (или общая); 1 – обработка омонимичных форм; 2 – построение поверхностных семантических структур; 3 – обработка семантических неоднозначностей; 4 – обработка концептуальных структур.

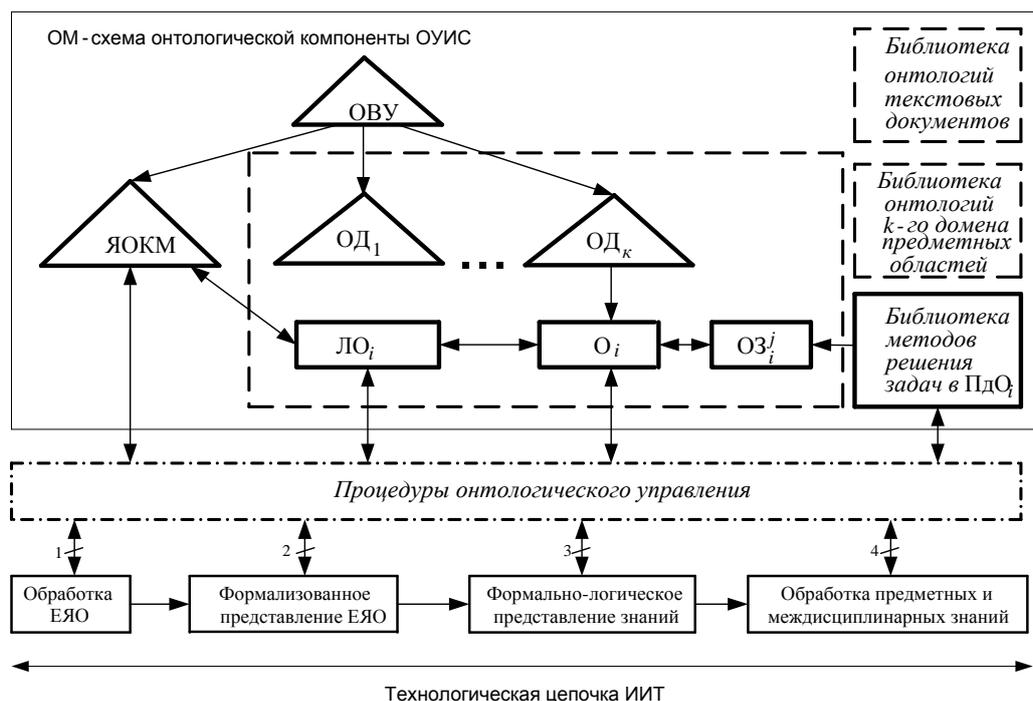


Рис. 6.4 Фрагмент архитектуры “онтологической” части ОУИС

Итак, ЗОИС с онтолого-управляемой архитектурой присущи следующие характерные черты:

- композиция онтологий разного уровня и назначения, как по вертикали, так и по горизонтали. По вертикали интегрируются онтология верхнего уровня, онтология домена предметных областей и онтология предметной области. По горизонтали интегрируются онтологии предметных знаний и знаний проблемного пространства. В свою очередь, предметные знания взаимодействуют с языковыми знаниями через лингвистическую онтологию предметной области;
- эффективное многократное использование онтологии предметной области и онтологии задач для разных наборов типовых задач. Будучи один раз построенными указанные онтологии пригодны для решения произвольного набора задач. При этом для новой задачи только составляется её расширенная спецификация на естественном языке, которая затем анализируется лингвистическими средствами ОУИС. Результат анализа передаётся в блок решателя задач, где

формируются обобщённый алгоритм решения задачи, наборы понятий-объектов и понятий-процессов, участвующих в решении задачи, и выбираются методы решения задачи;

- следствием композиции принципов и механизмов двойной парадигмы предметных знаний и онтологического управления является интеграция и эффективное применение компонентов архитектуры ОУИС в архитектурно-структурную организацию инструментального комплекса автоматизированного построения онтологических баз знаний предметных областей;
- применение произвольной онтологии (простой, смешанной или формальной) эффективно, прежде всего, в обучении. Действительно, гораздо эффективнее показать новому сотруднику предприятия онтограф представления структурных подразделений предприятия, чем долго рассказывать о той же самой структуре. Только ИС с онтолого-управляемой архитектурой, имеющая в своём составе формальную онтологию некоторой предметной дисциплины, позволяет реализовать все преимущества электронного образования, в том числе и дистанционного обучения. Наиболее полно указанные свойства проявляются при использовании и взаимодействии онтологий двух уровней – онтологии домена предметных дисциплин (это может быть множество дисциплин, читаемых на кафедре или факультете) и онтологий курсов самих предметных дисциплин. При этом естественным образом находят решение проблемы открытости и закрытости систем обучения, единообразия представления понятийных структур обучения, автоматизированного построения новых курсов обучения с учётом накопленных преподавателями кафедры знаний, замена кадрового состава преподавателей и др.

6.4. Языково-онтологическая информационная система

Подсистема обработки ЕЯО в ЗОИС представляет собой языково-онтологическую информационную систему, одной из отличительных особенностей которой от известных лингвистических процессоров является наличие языково-онтологической картины мира. Последняя, по сути, представляет собой онтологическую базу данных лексики естественного языка, компьютерно-ориентированную информационную структуру, которая является необходимой компонентой любой знание-ориентированной системы, обрабатывающей естественно-языковые тексты [3, 11, 27].

Онто-логическая модель ЯОИС имеет вид:

$$\bigvee_{i=0,p} G_i = \langle T_i, W_i, C_i, S_i^1, O_i, S_i^2, I_i \rangle, \quad (6.4)$$

где p – количество уровней аналитической обработки ЕЯТ (основными из которых являются графемный, морфологический, синтаксический, поверхностно-семантический, глубинно-семантический и информационно-кодовых представлений объектов текста). Эти уровни, в совокупности, обеспечивают эффективное функционирование и настройку ЯОИС на заданный класс задач (по обработке текстовой информации).

G_i – отображение i -ого уровня обработки ЕЯТ;

T_i – множество обрабатываемых на i -ом уровне объектов ЕЯ-текстов;

W_i – множество слов, составляющих текст i -го уровня;

C_i – множество синтаксических структур, связывающих W_i на i -ом уровне;
 S_i^1 – множество поверхностно-семантических структур, соответствующих C_i на i -ом уровне;
 S_i^2 – множество глубинно-семантических структур, отображающих T_i на i -ом уровне;
 O_i – множество фрагментов онтологических структур, участвующих в формировании S_i^2 на i -ом уровне, композиция которых в идеале составляет языково-онтологическую картину мира;
 I_i – множество информационно-кодовых представлений S_i^2 (индексов) на i -ом уровне.

Из всего множества задач, возложенных на ЯОИС и задекларированных в [10, 27, 116], одной из самых важных является задача снятия лексической неоднозначности, присутствующей в представленном на обработку ЕЯТ.

Однако, несмотря на большие трудозатраты, до сих пор не существует надежных механизмов определения лексического значения языковых компонент. Решение указанной проблемы позволит существенно увеличить качество машинного перевода, релевантность поиска (по оценкам специалистов – от 30 до 70%), понизить общий трафик в сети и время поиска конкретной информации. Даже частичное решение этой проблемы способно обеспечить прорыв сразу по нескольким направлениям [149]. Одним из методов решения проблемы предлагается частичное построение языковой картины мира, или “онтологии класса слов”. Другой метод [27, 143] базируется на индивидуальном формировании блоков синтаксических и семантических характеристик для каждой лексемы, построении формализованной структуры ЯОКМ. Лексическая неоднозначность в ЯОИС преодолевается на уровне семантики первой или второй ступени путём построения кортежей расстояний в замкнутых семантических пространствах соответствующих текстовых фрагментов и нахождения их минимальных значений. При этом существенную роль играет унифицированная система индексации языковых единиц на уровнях морфологического, синтаксического и семантического анализа.

Практическая ценность получаемых результатов при обработке ЕЯТ, в основном, зависит от полноты интерпретационных моделей семантических структур ЕЯТ и их формального представления. Под полнотой понимается включение в модель как составной семантики первой ступени (или объектной составляющей), так и составной семантики второй ступени (или акторной составляющей). Такое распределение семантики хорошо согласовывается как с онтологической иерархией концептуальных категорий, так и со сложностью выполнения вычислительных процедур при компьютерной обработке ЕЯТ.

С точки зрения лингвистики, семантическая составляющая первой ступени описывается на уровне грамматики отдельных частей речи, в то время как составляющая второй ступени уже описывается синтаксическими конструкциями таких единиц синтаксиса, как предложение, абзац, параграф, раздел и текст. С точки зрения математической логики, если первую ступень можно описать (довольно условно) исчислением высказываний, то вторая ступень должна описываться исчислением предикатов с квантифицированными переменными.

Наибольшей полноты (и соответственно наибольшей степени сложности) приобретают модели, которые описывают ЕЯТ в целом. Такие модели описывают, в частности, некоторый сценарий (как высшую категорию, которая описывает яв-

ление как взаимоотношение объектов, которые находятся в постоянном движении), который отображает содержание ЕЯТ. В свою очередь, как ЕЯТ делится на синтаксические единицы, так и общий сценарий раскладывается на отдельные сценарии, ситуации и элементарные ситуации.

Описанное важное различие между объектной и акторной составляющими семантики, а также морфолого-синтаксическим анализом, в частности, в сложности их интерпретационных моделей, обусловила выделение для моделирования и интерпретации семантики отдельного функционального модуля – семантического процессора. Морфологический и синтаксический анализ при этом выполняется грамматическим процессором, а точнее отдельными его блоками (морфологического и синтаксического анализа). Он содержит также лингвистическую СУБД реляционного типа и синтаксическую базу знаний.

Архитектура ЯОИС, разработанная в соответствии с О-моделью для предметной области обработки ЕЯТ, представлена на рисунке 6.5. На нём приняты следующие обозначения: ЛБД – лексикографическая база данных; ЯОКМ – языково-онтологическая картина мира.

ЛБД представляет набор таблиц, соответствующих грамматическому словарию для каждой части речи ЕЯ, таблиц падежных окончаний для формирования словоформ лексем, уникальных идентификаторов лексем ЕЯ и их синтаксических и семантических характеристик [27, 150]. Все лексические единицы в таблицах соответствующим образом проиндексированы и имеют одинаковое интерпретационное значение как для грамматического, так и семантического процессора [27]. Заметим, что функции ЛБД значительно расширены по сравнению с традиционными грамматическими словарями, и они эффективно реализуются аппаратными средствами.

ЯОКМ представляет лингвистическую онтологию, одну из центральных компонент ЯОИС [27] и совместно с ЛБД представляет базу знаний лексики ЕЯ. ЯОКМ – это формализованная онтология, в которой аксиомы и определения входят в состав баз знаний синтаксиса и семантики модуля грамматического анализатора (на рисунке не показаны), а ограничения являются составной частью синтаксических и семантических характеристик ЛБД.

Грамматический анализатор – это компонента, реализующая процедуры графемного, морфологического и синтаксического анализа. Она взаимодействует с лексикографической базой данных, а результаты анализа (сформированные в виде итоговой морфологической таблицы текста и синтаксических деревьев предложений, входящих в текст) передаются на вход семантического анализатора.

Семантический анализатор – это компонента, реализующая процедуры семантического анализа предложений текста, решения задачи грамматической и лексической неоднозначности и построения формально-логического представления предложений текста. При этом первые две процедуры могут выполняться итерационно. Семантический анализатор взаимодействует с ЛБД и ЯОКМ, а результаты анализа (сформированные деревья и формально-логические представления семантически связанных фрагментов текста) передаются на вход семантико-информационного интерпретатора (СИИ).

СИИ реализует процедуры построения информационно-кодowego представления семантики текста и его интерпретационной модели (знание-ориентированной компоненты). Причём, если построение первой составляющей обеспечивает прикладной процессинг непосредственно входного текста (реферирование, классификацию, построение простой онтологии документа и др.), то в совокупности со второй составляющей обеспечивается прикладной процессинг для различных про-

цедур обработки не полностью формализованных знаний (извлечение, интеграция, накопление знаний и др.).

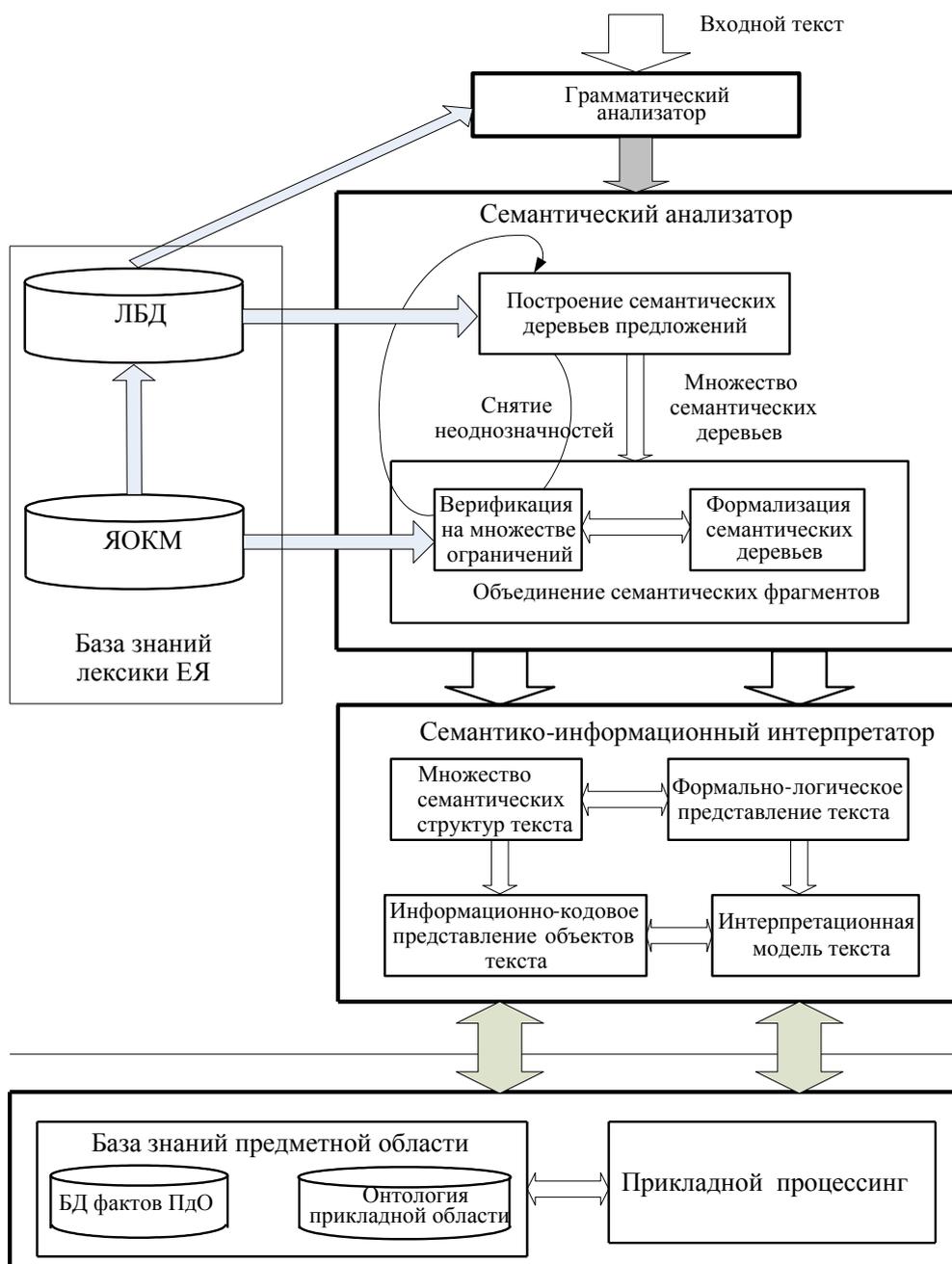


Рис. 6.5. Архитектура ЯОИС

Цепочки преобразования информации $T \rightarrow W \rightarrow S^1$ и $O \rightarrow S^2 \rightarrow I$ модели (6.4), по сути, представляют (соответственно) базовые процедуры анализа и понимания ЕЯТ, средствами интерпретации которых являются грамматический и семантический анализаторы и семантико-информационный интерпретатор.

Модули прикладного процессинга (ПП) и базы знаний предметной области предназначены для решения конкретных задач пользователя. Причём для простых задач обработки ЕЯТ реализующие их алгоритмы могут войти в состав ОУИС с обработкой ЕЯТ.

6.5. ОУИС с обработкой знаний предметной области

Ряд идей, выдвинутых в процессе реализации проекта по созданию ЭВМ пятого поколения [151] в новом качестве могут быть использованы и в настоящее время для архитектур обработки ЕЯ, в частности, применение интеллектуальной многопортовой памяти, сортировки, параллельной обработки синтаксических единиц ЕЯО и др.

Выполненный анализ когнитивного процесса познания, основных источников знаний, известных технологий обработки языковых и предметных знаний, архитектурных особенностей информационных систем, реализующих указанные технологии, позволяет сделать некоторые, общего характера выводы относительно архитектурно-структурной организации современной ОУИС. В частности:

- архитектуру ОУИС целесообразно строить на базе двух, относительно независимых информационных подсистем – ЯОИС и ОУИС ПДО (см. рис. 6.1);
- необходимы формализованные средства (интерпретатор), позволяющие осуществить переход от обработки текстовой информации на основе языковых знаний к обработке формализованных представлений предметных знаний;
- экспертные знания, представленные в виде естественно-языковых спецификаций должны эффективно обрабатываться в такой ОУИС;
- необходимо интегрировать в единую технологию четыре, относительно независимые информационные технологии (ИТ):
- NLP – Natural Language Processing,
- KR – Knowledge Representation,
- KP – Knowledge Processing;
- создание и использование онтологий предметных областей.

Блок-схема ОУИС ПДО представлена на рис. 6.6.

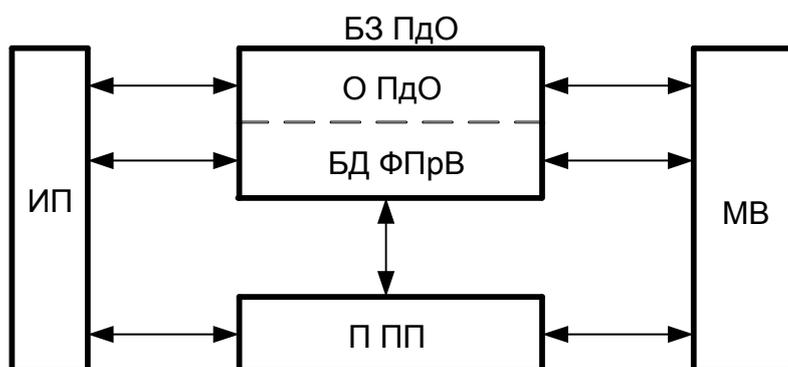


Рис. 6.6. Блок-схема ОУИС ПДО

Компоненты ОУИС ПдО и их функции

На основе анализа известных положений об архитектурно-структурных особенностях и потребительских признаках ЗОИС [10, 39, 40], определим состав и назначение основных компонентов онтолого-управляемой информационной системы обработки знаний в произвольной предметной области.

ОУИС ПдО содержит онтологическую базу знаний (в общем случае конечное множество системно интегрированных онтологических баз знаний), машину вывода, подсистему прикладного процессинга и интерфейс с пользователями и/или внешней средой. Компоненты ОУИС ПдО характеризуются совокупностью следующих признаков:

- *онтологическая БЗ* – О ПдО является компьютерной формой представления модели ПдО. Она состоит из концептуальной надстройки (иерархической структуры концептуальных понятий – О ПдО) и фактографической базы (БД Ф ПрВ), включающей как компоненты базу фактов и базу правил вывода новых фактов. В свою очередь, концептуальная надстройка связана концептуальными отношениями с понятиями более высокого уровня абстракции, входящими в т. н. метаонтологию домена прикладных областей (например, информатика (приложение Б), медицина, материаловедение, право и т. д.);
- *машина вывода* (МВ) реализует внутренний алгоритм интерпретации структур знаний применительно к входным данным, взаимодействует с подсистемой прикладного процессинга (П ПП), принимает от него входные данные и выдаёт полученный в процессе интерпретации результат;
- *подсистема ПП* осуществляет в диалоге с пользователем и/или внешней средой следующие функции: принимает задания и формирует обобщённый алгоритм их реализации в виде последовательности процедур; формирует задания для МВ и получает от неё результаты; выдаёт результаты решения задач через интерфейс пользователя (ИП); осуществляет ввод, редактирование и пополнение знаний в БЗ ПдО;
- *интерфейс пользователя* осуществляет перевод текущих заданий/результатов в их внутреннее (машинное)/внешнее (язык пользователя) представление.

Одним из главных назначений такой ОУИС (прежде всего с точки зрения пользователя) является фиксация всей релевантной информации об интересующей его (пользователя) части реального мира для манипулирования ею, моделирования и рассуждения о правилах, законах и ограничениях, действующих в этой части реального мира (предметной области) и управляющих её развитием, о других её свойствах и характерных особенностях.

Режимы функционирования ОУИС:

- а) обработка целевых заданий (внешних и внутренних), в частности, активизация процесса, актуализация информации, релевантной одной или несколькими ПдО, и размещение её в памяти, решение поставленной проблемы (задачи), выработка, систематизация и выдача результирующих продуктов (в случае знание-ориентированной деятельности – приращение знаний);
- б) развитие ОУИС как информационной системы согласно общей стратегии развития: инвентаризация и систематизация знаний (расширение метазнаний), формализация и когнитивизация представлений, интерпретационное расширение системы знаний, увеличение объёма реакций и ассоциативных связей.

Особенности заданной ПдО полностью определяют подход к проектированию компонент подсистемы обработки знаний – БЗ ПдО и алгоритмов ПП.

Для БЗ ПдО необходимо создать базу данных суждений и фактов и разработать алгоритмы её взаимодействия с онтологией ПдО, спроектированной в подсистеме полной онтологии ПдО. Указанное проектирование выполняется в соответствии с известными методами.

Проектирование компоненты ПП в общем случае представляется разработкой совокупности алгоритмов, реализующих заданное множество задач пользователя и их взаимодействие с БЗ ПдО и знаниями, полученными в подсистеме ЯОИС.

Полная онтология ПдО в процессе реализации цепочки технологий играет различную роль:

- при NLP основную роль играет внутриязыковое обеспечение ПдО – интеграция лингвистической онтологии с соответствующим фрагментом ЯОКМ. Оно поддерживает выполняемые ЯОИС процедуры анализа и распознавания знаний, содержащихся во входном ЕЯО;

- при KR выполняется переход к формальному описанию знаний на некотором языке L , при котором активную роль играют онтологические знания ПдО, представленные онтографом. При этом формируются онтограф и интерпретационная структура, соответствующие знаниям, содержащимся во входном ЕЯО;

- при КР активную роль играет база знаний ПдО, в состав которой входит формальное описание онтологических знаний и её аксиоматизация.

6.5.1. Архитектурно-структурная организация ОУИС

Архитектурно-структурная организация онтолого-управляемой информационной системы с обработкой ЕЯО представлена на рис. 6.7. Здесь приняты следующие обозначения: ЗнПС – знание-ориентированная поисковая система, ЯПЗ – язык представления знаний, ЛО ПдО – лингвистическая онтология предметной области.

Приведенная архитектурно-структурная организация ОУИС, с одной стороны детализирует обобщенную блок-схему обработки знаний, содержащихся в ЕЯО (рис. 6.1), а с другой – объединяет компоненты архитектурной организации ЯОИС (рис. 6.5) и ОУИС ПдО (рис. 6.7). Такое объединение позволяет реализовать цепочку обобщенных процедур: *“обработка естественно-языковой информации → формальное логико-онтологическое представление ЕЯ-информации → компьютерная обработка знаний”*, которые в известной степени соответствуют интегрированной информационной технологии работы со знаниями.

На рис. 6.7 к компонентам, входящим в состав ЯОИС и ОУИС ПдО, добавлены следующие компоненты: семантическая память, знание-ориентированная поисковая система, хранилище данных и знаний, лингвистическая онтология в блоке БЗ ПдО.

ЗнПС выполняет (в том числе) поиск документов в различных источниках ЕЯО и организует их хранение в виде онтологий текстовых документов в семантической памяти и/или хранилище данных и знаний [26, 28, 49].

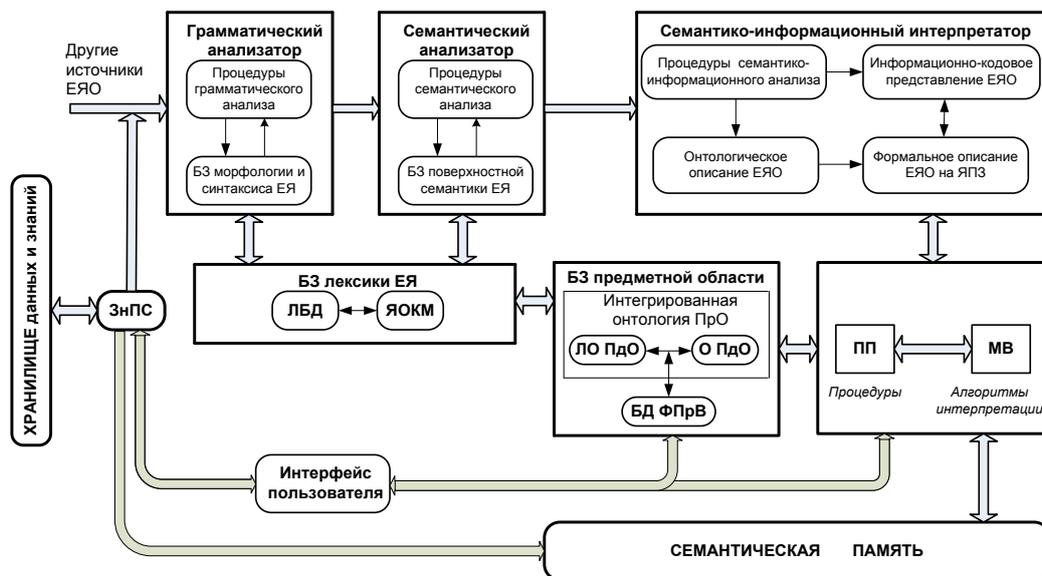


Рис. 6.7. Архитектурно-структурная организация ОУИС

Семантическая память включает следующие виды информации:

- *онтологии* домена прикладных областей;
- *определения*: необходимые и достаточные условия, которые определяют новые типы концептов и отношений в терминах подтипов нижних уровней;
- *ограничения*: общие принципы и/или аксиомы, тождественно истинные для элементов концептов;
- *значения по умолчанию*: информация, которая является предпочтительной для элементов концептов;
- *поведение*: правила, которые управляют сценариями для каждого объекта и взаимодействия наборов объектов.

В общем случае семантическая память и хранилище данных и знаний (частично) реализуют функции подсистемы “Информационный ресурс” ИКОН (гл. 7).

Лингвистическая онтология ПдО является частью лингвистического обеспечения ОУИС и связующим звеном между БЗ лексики ЕЯ и БЗ ПдО.

Отметим следующие особенности архитектуры ОУИС:

- ЗнПС для найденных множеств текстовых документов строит (совместно с ЯОИС) соответствующие им онтологии, которые хранятся в семантической памяти;
- интерфейс служит для распознавания запросов пользователей, составленных в виде естественно-языковых предложений;
- семантическая память, в общем случае, ориентирована на широкий спектр хранимой информации и настраивается на круг задач прикладного процессинга. Можно выделить следующие, наиболее важные её сегменты:
 - библиотека онтологий прикладного домена;
 - онтология прикладного домена;
 - корпоративная память, ЭБ или коллекция документов, хранящихся в виде онтологий;
 - память прикладного процессинга и др.

6.5.2. Обобщённый алгоритм проектирования ОУИС

Основные положения обобщённого алгоритма

В известных методах формализованного проектирования информационных систем [152] сам процесс проектирования представляется в виде последовательности этапов (в основном системного, алгоритмического и логического), на каждом из которых проект представлен совокупностью математических моделей, описывающих различные её части. Указанная совокупность математических моделей тесно связана с системой взаимосвязанных алгоритмов, которые, в свою очередь, описывают соответствующее множество решаемых задач и в своей совокупности представляют общий алгоритм проектирования ИС.

Описанные выше положения онтологического подхода, онто-логическая модель и оценки характеристик ОУИС положены в основу обобщённого алгоритма проектирования онтолого-управляемой ИС с обработкой ЕЯО. Он представляет собой систему взаимосвязанных алгоритмов, часть из которых формализованы и сведены к постановке и решению задач анализа и синтеза, остальные используют эвристические методы [152, 153]. Каждый алгоритм является отдельным фрагментом проектирования, где он представлен в виде формализованной методики проектирования или содержательным описанием метода решения задач анализа, синтеза и оптимизации отдельных компонентов и архитектурно-структурной организации в целом.

Системный и **алгоритмический** этапы проектирования знание-ориентированных ИС связаны с анализом предметной области и её проблемного пространства, синтезом концептуальных структур знаний и построением базы знаний ПДО, выбором архитектуры проектируемой ЗОИС, определением её подсистем и состава функциональных блоков, их параметров и способов взаимодействия, а также разработкой структурного описания вычислительных процедур (в своей совокупности составляющих интегрированную информационную технологию) в терминах числовых характеристик, устанавливающих связи между временными, структурными и информационными параметрами ЗОИС [3, 5, 152]. На основе предварительных оценок временных параметров реализации вычислительных процедур решается вопрос аппаратной поддержки наиболее трудоёмких из них. Этим этапам свойственны системный анализ и априорная оценка характеристик предметной области, проблемного пространства и архитектурно-структурных компонентов проектируемой ЗОИС, в частности:

- онто-гносеологический анализ особенностей компьютерной обработки предметных знаний;
- анализ особенностей онтологического подхода к знание-ориентированной обработке ЕЯО;
- анализ концептуальной парадигмы работы со знаниями;
- исследование особенностей системно-онтологического анализа предметной области и проблемного пространства (типовых наборов прикладных задач);
- исследование особенностей архитектурно-структурной организации современных ЗОИС;
- анализ и классификация методов и средств знание-ориентированной обработки ЕЯО;
- анализ и выбор формальных теорий и формально-логического представления ЕЯВ;

- разработка онто-логической модели структуры ОУИС с обработкой ЕЯО.

Основной задачей синтеза, решаемой на этих этапах, является построение многоуровневой композиции программируемых автоматов – основы архитектуры и структуры ОУИС, разработка эффективных средств связи между уровнями программирования (и в особенности для предметного уровня).

К задачам анализа, решённым на *логическом этапе* проектирования, относятся:

- анализ средств реализации и разработка модели этапов лингвистического анализа ЕЯТ;
- построение обобщённой модели ОУИС с обработкой ЕЯО;
- построение схемы обобщённой модели инструментальных средств автоматизированного построения онтологии ПдО и др.

Наиболее важные задачи синтеза логического этапа проектирования:

- разработка методики проектирования ОУИС с обработкой ЕЯО;
- разработка методики синтеза онтологий ПдО;
- синтез онтологической базы знаний ПдО.

Методика проектирования, опираясь на данную систему алгоритмов и лежащую в её основе онто-логическую модель ОУИС, позволяет решить главную задачу проектирования – формализовать процесс поиска оптимальной пары “алгоритм-аппаратно-программная реализация”.

Методика предназначена для проектирования:

- онтолого-управляемых информационных систем с обработкой знаний, содержащихся в ЕЯО, и работающих с заданным набором алгоритмов прикладного процессинга;
- онтологической базы знаний заданной ПдО.

Она может видоизменяться и развиваться в зависимости от исходной предметной области, типового набора задач, элементарно-технологической базы и т. п.

Описание блок-схемы общего алгоритма проектирования ОУИС

Блок-схема алгоритма синтеза ОУИС с обработкой ЕЯО представлена на рис. 6.8.

Исходной для проектирования является информация о предметной области (в том числе и о естественном языке) и типовых наборах задач, для решения которых проектируется ОУИС.

Ниже описано содержание блоков алгоритма.

Системный и алгоритмический этапы (п. п. 1–6)

Анализируется проблемное пространство заданной ПдО – множество алгоритмов типовых наборов задач с целью определения их статистических характеристик (которые могут быть заданы). При проектировании ОУИС для решения задач типа Text Processing непосредственно задаётся лингвистический базис общеязыковых знаний, на основе которого формируется лексикографическая база данных и языково-онтологическая картина мира. Кроме того, для произвольной ОУИС задаётся толковый словарь заданной ПдО, на основе которого разрабатывается лингвистическая онтология ПдО.

Анализируя множество сущностей заданной ПдО определяются наиболее существенные, релевантные конкретным наборам типовые задачи. Одновременно с этим формируется список понятий-категорий ПдО, которые, с одной стороны, служат связующим звеном с онтологическими категориями домена предметных областей, а с другой – основой для формирования начальной концептуальной структуры

знаний ПдО. Компоненты языково-онтологической информационной системы формируются на основе априорных знаний о заданном естественном языке.

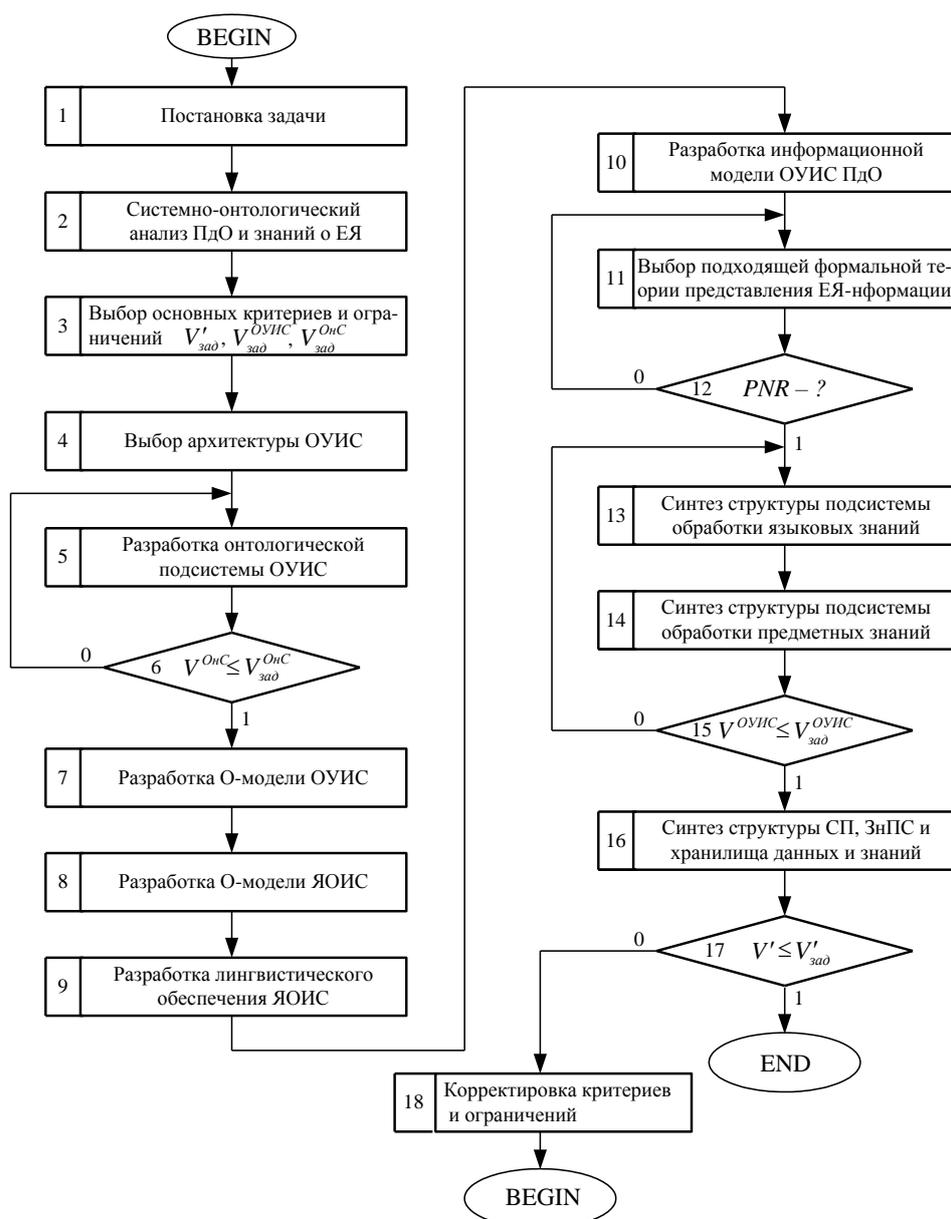


Рис. 6.8. Блок-схема обобщённого алгоритма проектирования ОУИС

В результате анализа технического задания и заданной ПдО (п. п. 1, 2) определяется множество допустимых параметров ОУИС, включающих:

$V'_{зад}$ – допустимые обобщённые (временные, структурные, информационные, производительности, программно-аппаратных затрат,

- а) точности решения задачи, сложности алгоритма и др.) параметры ОУИС и её функциональных узлов: семантической памяти, знание-ориентированной поисковой системы и хранилища данных и знаний;

- б) $V_{зад}^{ОнС}$ – допустимые временные (интерпретации информационных структур) и информационные (объём памяти для хранения информационных структур) параметры онтологической подсистемы;
- в) $V_{зад}^{ОУИС}$ – допустимые временные (реализации алгоритмов решения типовых наборов задач) и структурные (для подсистем обработки языковых и предметных знаний) параметры. Учитываются такие критерии как полнота, непротиворечивость и разрешимость (PNR) формально-логического представления ЕЯ-информации в соответствии с выбранной логической теорией.

Далее определяются функциональные блоки онтологической подсистемы, подсистем обработки ЕЯ-информации и предметных знаний. Кроме того, определяются общая архитектура, взаимосвязи и общие параметры семантической памяти, знание-ориентированной поисковой системы и хранилища данных и знаний. Разрабатываются подсистемы ОУИС на основе композиции общелингвистических знаний (ЯОКМ), статических и динамических знаний (лингвистическая онтология, онтология объектов, онтология процессов и онтология задач) заданной ПдО. При этом формируются количественные оценки в соответствии с принятыми критериями и ограничениями.

Этап логического анализа (п. п. 7–12)

Далее разрабатываются онто-логические модели ОУИС ПдО и ЯОИС, для которых:

а) в результате анализа алгоритмов A_4 решения типового набора задач (п. 1) на основе выделенных фрагментов стандартных процедур, реализующих метод (методы) решения задач, определяется полный набор операторов Λ_4 для предметного уровня управления τ_4 . Здесь же:

а) выполняется анализ процедур, реализующих механизм онтолого-управления, определяются соответствующие пары (n -ки) уровней управления и их взаимодействие;

б) в результате анализа алгоритмов интерпретации этапов лингвистического анализа (п. 2) определяются наборы операторов Λ_3, Λ_2 для алгоритмического и программного уровней управления. Определяются:

- оптимальный итеративный цикл обработки неоднозначностей, присутствующих в исходной ЕЯ-информации;
- на основе выделения наиболее трудоёмких (по временным затратам) процедур реализации этапов лингвистического анализа разрабатываются структурные решения их интерпретации на аппаратном (физическом) уровне τ_0 и механизмы взаимодействия между соответствующими уровнями управления.

Для разработки лингвистического обеспечения ЯОИС определяются модель управления, структура и наполнение лексикографической базы данных, а также информационная модель ОУИС ПдО в соответствии с информационной концепцией O -модели для предметного уровня управления τ_4 .

На основе полученных в п. п. 7-10 результатов осуществляется выбор подходящей формальной теории представления ЕЯ-информации. Причём таких теорий может быть несколько, каждая из которых наиболее эффективна для определённо-

го этапа представления ЕЯ-информации. Обязательным является требование однозначного формального перехода от одной теории к другой без потери семантики ЕЯ-информации. Затем анализируется множество наборов формально-логических представлений ЕЯ-информации в соответствии с требованиями полноты, непротиворечивости и разрешимости. Если комплексный критерий *PNR* выполняется, то осуществляется переход к п. 13. В противном случае осуществляется возврат к п. 11.

Этап логического синтеза (п. п. 13–18)

На этом этапе выполняется синтез структур подсистем обработки языковых и предметных знаний, семантической памяти, знание-ориентированной поисковой системы и хранилища данных и знаний, сравнительный анализ полученных и заданных количественных оценок для заданных критериев и ограничений. Если $V^{OYIC} \leq V_{зад}^{OYIC}$ и $V' \leq V'_{зад}$, то осуществляется переход к п. 16 или “END” соответственно. В противном случае, осуществляется возврат к п. 13 и выполняется реинжиниринг ЯОИС и ОУИС ПдО или п. 18, в соответствии с которым необходима корректировка технического задания (в том числе пересматриваются критерии и ограничения).

7. ИНСТРУМЕНТАЛЬНЫЙ КОМПЛЕКС ОНТОЛОГИЧЕСКОГО НАЗНАЧЕНИЯ

7.1. Анализ функциональных возможностей средств построения онтологий

Методология проектирования онтологии ПдО (глава 4) предполагает формирование множеств концептов, отношений, функций интерпретации и аксиом. Построение указанных множеств вручную является трудоёмким процессом, как по времени, так и по количеству вовлечённых в процесс проектирования высококвалифицированных специалистов. Ручное проектирование онтологий мало чем отличается от проектирования экспертных систем.

Понимание важности создания инструментальных средств поддержки процесса проектирования онтологии заданной ПдО пришло практически одновременно с принятием парадигмы компьютерных онтологий. В настоящее время известно более ста инструментальных программных систем⁸ [5, 52, 91, 93, 154], которые имеют следующие характеристики.

1. *Поддерживаемые формализмы и форматы представления.*

Под формализмом понимается некоторая формальная теория, лежащая в основе способа представления онтологических знаний (логика предикатов, фрейм-овые модели, дескриптивная логика, концептуальные графы и др.). Формализм существенно влияет на внутренние (компьютерные) структуры данных и может определять их формат представления.

Формат представления онтологий задаёт вид их хранения в библиотеке и способ передачи онтологических описаний другим потребителям. В качестве форматов разработаны языки представления онтологий, наиболее известными из которых являются OWL, RDFS, KIF.

Некоторые из известных редакторов онтологий поддерживают работу с несколькими формализмами представления, но только один формализм и формат являются предпочтительными для данного (конкретного) редактора [91].

2. *Функциональность.*

Является одной из самых важных характеристик редакторов онтологий, под которой понимается множество предоставляемых пользователю сценариев работы с онтологическими структурами.

Базовый набор функций обеспечивает:

- работу с одним или несколькими проектами одновременно;
- графический интерфейс с пользователем;
- редактирование онтологии (создание, редактирование, удаление концептов, отношений, аксиом и прочих структурных элементов онтологии).

3. *Архитектура приложения, место хранения онтологий, язык программного обеспечения, интерфейс пользователя, доступность.*

4. *Дополнительные возможности.*

⁸ Такие системы в литературе обычно именуется редакторами онтологий.

К ним относят поддержку языка запросов, анализ целостности, использование механизма логического вывода, поддержку удалённого доступа через Интернет, документирование.

Известны три группы ИнС онтологического инжиниринга [154].

К первой группе относят инструменты создания онтологий, которые предполагают поддержку совместной разработки и просмотра, создание онтологии в соответствии с заданной (произвольной) методологией, поддержку рассуждений.

Ко второй группе относят инструменты объединения, отображения и выравнивания онтологий.

Объединение предполагает нахождение сходств и различий между исходными онтологиями и создание результирующей онтологии, которая содержит элементы исходных онтологий. Для этого ИнС автоматически определяют соответствия между концептами или обеспечивают графическую среду, в которой пользователь сам находит эти соответствия.

Процедура отображения заключается в нахождении семантических связей разных онтологий.

Процедура выравнивания онтологий устанавливает различные виды соответствия между двумя онтологиями, информация которых сохраняется для дальнейшего использования в приложениях пользователя [155].

К третьей группе относят инструменты для аннотирования Web-ресурсов на основе онтологий.

Содержательный обзор известных инструментов инженерии онтологий, в котором рассмотрены основные функции и возможности ИнС, их достоинства, недостатки, сравнительный анализ и описание известных доступных онторедкторов приведен в [154, 156–158].

Общими недостатками известных инструментальных средств являются:

- отсутствие процедур автоматического (автоматизированного) формирования компонент онтологии;
- англоязычный интерфейс с пользователем, в котором (для большинства ИнС) не предусмотрено присвоение имён компонентам онтологии на русском или украинском языке;
- структуризация концептов выполняется только по одному типу отношений;
- для большинства общедоступных ИнС не предусмотрена работа с большими по объёму онтологиями (например, для OntoEditFree – до 50 концептов);
- большинство инструментов хранит свои онтологии в текстовых файлах, что ограничивает скорость доступа к онтологиям;
- задекларированные функциональные возможности для общедоступных инструментов зачастую так и остаются нереализованными;
- недостаток информации для пользователей в инструкциях.

Описанные выше недостатки известных англоязычных ИнС отразились и в разработке аналогичных инструментов для российского сегмента, в частности “Многоуровневый редактор онтологий” [159] “SIMER + MIR”, “OntoEditor+”⁹.

Указанные инструментальные средства являются закрытыми системами, поэтому оценить их функциональные возможности в полном объёме не представляется возможным, хотя последний инструмент освещён в литературе достаточно и

⁹ “OntoEditor+” разработан группой исследователей Казанского университета (Россия) под руководством проф. Невзоровой О.А.

представляет собой наиболее близкий прототип к описанному в следующих разделах инструментальному комплексу автоматизированного построения онтологий предметных областей.

Инструментальная система визуального проектирования “OntoEditor+” [160] является специализированной СУБД. Система предназначена для ручного редактирования онтологий, хранящихся в реляционной базе данных, а также обслуживания запросов пользователей и внешних программ. Расширенные возможности системы обеспечиваются функциональным набором “Лингвистический инструментарий”, посредством которого реализуется встраивание прикладной онтологии в лингвистические приложения. Наиболее типичными задачами, решаемыми с помощью инструментария системы “OntoEditor+”, являются: задачи корпусного исследования (загрузка корпуса, сегментация на предложения, автоматическое ведение статистики по различным объектам корпуса), изучение структурных свойств прикладной онтологии с помощью исследовательского инструментария, построение лингвистической оболочки прикладной онтологии, покрытие онтологическими входами, построение выводов по прикладной онтологии и поддержка протоколов информационного обмена системы с внешними программными модулями, в том числе с внешними информационными ресурсами.

Для решения структурно сложных задач (например, извлечение из текстов информационной модели алгоритмов) в определенной проблемной ситуации, контроля структурной и информационной целостности выделенной схемы алгоритмов, создана специализированная система обработки текстовых документов “ЛОТА” [161], в которую входят три взаимодействующие подсистемы: подсистема лингвистического анализа технических текстов “Анализатор”, подсистемы “Интегратор” и “OntoEditor+”.

Подсистема “Анализатор” реализует основные этапы лингвистической обработки текста (графематический, морфосинтаксический и частичный семантический анализ).

Подсистема “Интегратор” исполняет внешний запрос на извлечение знаний из текста. Структура внешнего запроса содержит компоненты информационной модели алгоритма. Внешний запрос интерпретируется при взаимодействии с подсистемой “OntoEditor+” как структура, привязанная к прикладной онтологии. Выделение компонент информационной модели происходит на основе механизмов отождествления элементов дерева сегментов входного текста (взаимодействие с подсистемой “Анализатор”) и элементов структуры запроса (взаимодействие с подсистемой “OntoEditor+”).

Основными требованиями при выборе редактора онтологий для практических целей являются:

- бесплатное распространение;
- локальное приложение Web-интерфейса;
- расширение функциональных возможностей приложения;
- возможность подключения плагинов, разработанных пользователями.

Анализ многочисленной литературы по онторедакторам показал, что приведенным выше требованиям удовлетворяет онторедактор Protégé. Он, кроме того:

- наиболее широко используется разработчиками онтологий [156];
- версии приложения постоянно обновляются. Для версии 4.0 основными преимуществами являются: доступность большего числа плагинов, в том числе понимание и генерация (англоязычного) ЕЯТ, скрипты и др.; конфигурируемая (постоянная) схема компонентов; создание, импорт и экспорт конфигурируемых пользователем ярлыков;

- реализация произвольной методологии проектирования онтологии;
- наличие плагина Prompt, исполняющего различные процедуры манипулирования двумя онтологиями (в частности, процедура Merge может служить прототипом процедуры системной интеграции онтологий предметных областей). На рис. 7.1 показаны две простые онтологии из домена “Информатика”, которые объединяются в онторедакторе Protégé плагином Prompt, а на рис. 7.2 – результат указанного объединения;
- языком программного продукта является Java, используемый также в оригинальных программах инструментального комплекса, описанного ниже;
- наличие машины вывода;
- возможность присвоения имён концептам онтологии на русском или украинском языках;
- содержательное соответствие схем компонент проектируемой онтологии в Protégé и описанной в главе 4. Указанное соответствие представлено на рис. 7.3, в котором приняты следующие сокращения: R_1 – отношение “класс-подкласс”, R_2 – отношение “класс-свойство”, R_3 – отношение “класс-элемент”, R_4 – отношение “элемент-свойство”, X, R, F, A – компоненты формулы (4.2).

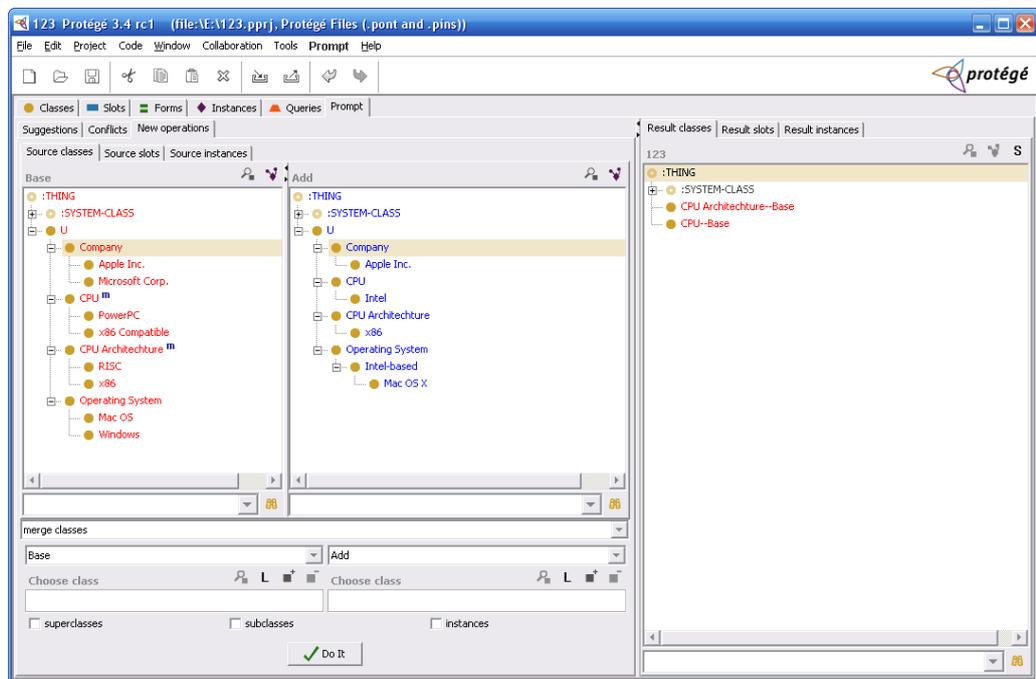


Рис. 7.1. Интегрирующая (base) и интегрируемая (add) онтологии в Protégé

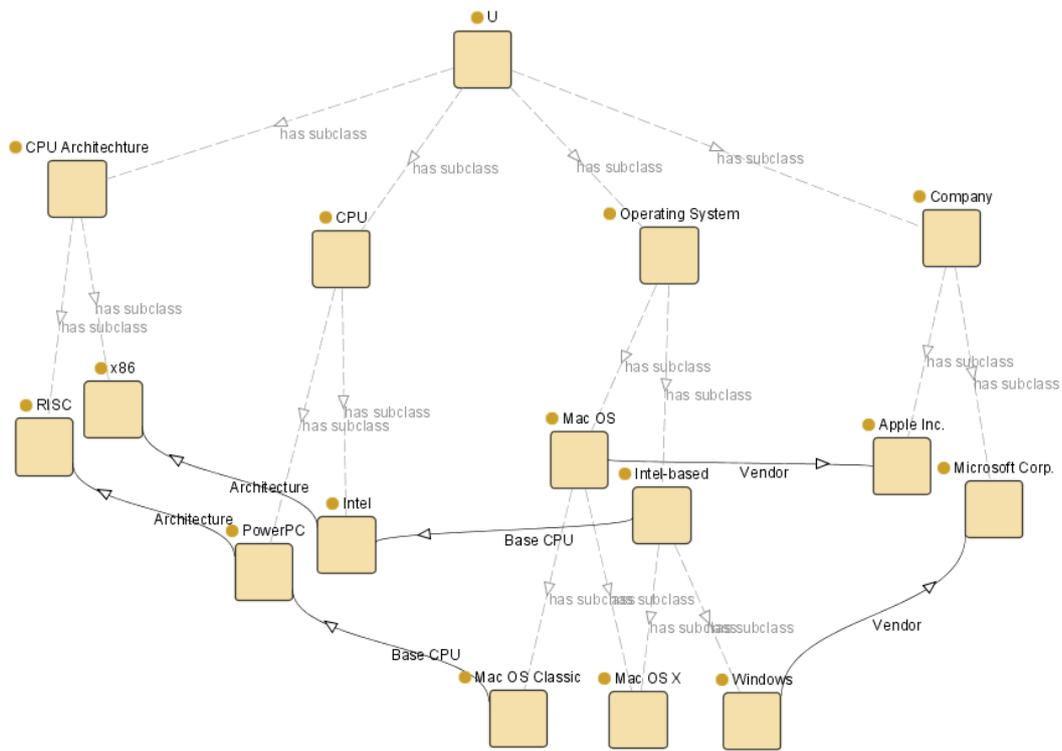


Рис. 7.2. Результат объединения онтологий в Prompt

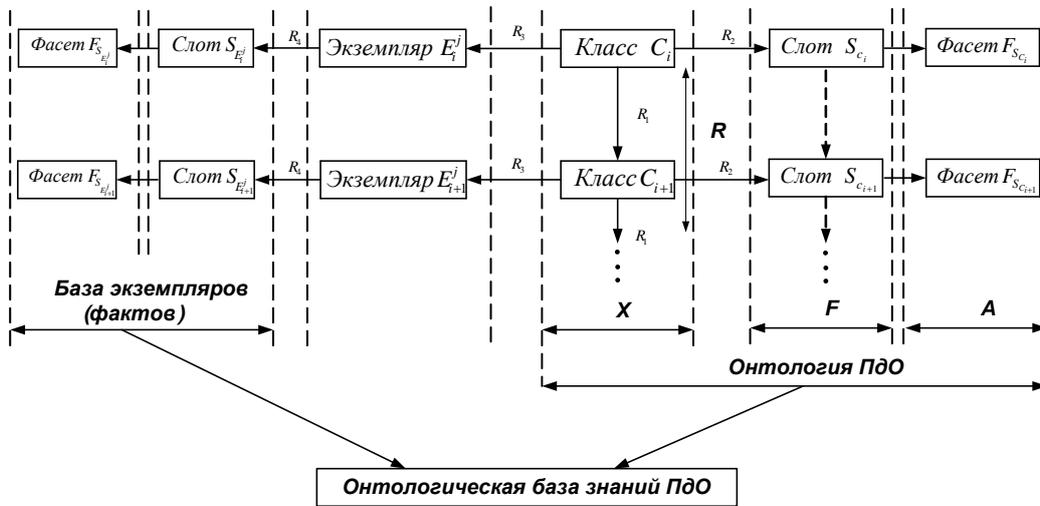


Рис. 7.3. Соответствие схем компонент проектируемой онтологии в Protégé и (4.2)

7.2. Инструментальный комплекс автоматизированного построения онтологий ПдО

Инструментальный комплекс онтологического назначения для автоматизированного построения онтологии в произвольной предметной области является системой, реализующей одно из направлений комплексных технологий *Data & Text Mining*, а именно – анализ и обработку больших объёмов неструктурированных данных, в частности лингвистических корпусов текстов на украинском и/или русском языке, извлечение из них предметных знаний с последующим их представлением в виде системно-онтологической структуры или онтологии предметной области. ИКОН предназначен для реализации множества компонентов единой информационной технологии:

- поиск в сети Internet и/или в других электронных коллекциях (ЭлК) текстовых документов (ТД), релевантных заданной ПдО, их индексацию и сохранение в базе данных;
- автоматическая обработка естественно-языковых текстов (Natural Language Processing);
- извлечение из множества ТД знаний, релевантных заданной ПдО, их системно-онтологическая структуризация и формально-логическое представление на одном (или нескольких) из общепринятых языков описания онтологий (Knowledge Representation). Кроме того, внутри этой технологии реализуется процедура построения, визуализации и проверки семантических структур синтаксических единиц ТД и понятийных структур заданной ПдО в виде несильно связанного онтографа, названного *начальной онтологией ПдО* (НО ПдО);
- создание, накопление и использование больших структур онтологических знаний в соответствующих библиотеках;
- системная интеграция онтологических знаний как одна из основных компонент методологии междисциплинарных научных исследований;
- другие процедуры, связанные с автоматизацией приобретения знаний из множества естественно-языковых объектов.

ИКОН состоит из трёх подсистем и представляет собой интеграцию разного рода информационных ресурсов (ИР), программно-аппаратных средств обработки и процедур естественного интеллекта (ЕИ), которые, взаимодействуя между собой, реализуют совокупность алгоритмов автоматизированного, итерационного построения понятийных структур предметных знаний, их накопления и/или системной интеграции. Обобщённая блок-схема ИКОН представлена на рис. 7.4.

Подсистема **Информационный ресурс** включает блоки формирования лингвистического корпуса текстов, баз данных языковых структур и библиотек понятийных структур. Первый компонент представляет собой различные источники текстовой информации, поступающей на обработку в систему. Второй компонент представляет собой различные базы данных обработки языковых структур, часть из которых формируется (наполняется данными) в процессе обработки ТД, а другая часть формируется до процесса построения О ПдО и, по сути, является ЭлК различных словарей. Третий компонент представляет собой совокупность библиотек понятийных структур разного уровня представления (от наборов терминов и понятий до высокоинтегрированной онтологической структуры междисциплинарных знаний) и является результатом реализации некоторого проекта (проектирования онтологии ПдО и/или системной интеграции онтологий).

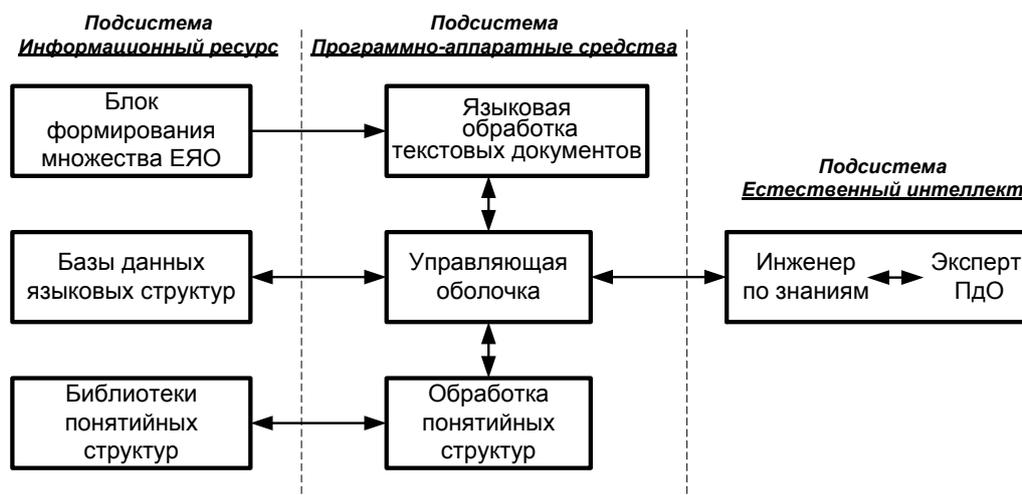


Рис. 7.4. Обобщённая блок-схема ИКОН

Подсистема *Программно-аппаратные средства* включает блоки обработки языковых и понятийных структур и управляющую графическую оболочку (УГО). Последняя, во взаимодействии с инженером по знаниям, осуществляет общее управление процессом реализации связанных информационных технологий.

Подсистема *Естественный интеллект* осуществляет подготовку и реализацию процедур предварительного этапа проектирования, а на протяжении всего процесса осуществляет контроль и проверку результатов выполнения этапов проектирования, принимает решение о степени их завершённости (и в случае необходимости – повторении некоторых из них).

7.2.1. Архитектурно-структурная организация и описание компонент ИКОН

Архитектурно-структурная организация ИКОН представляет собой композицию трёх подсистем, связанных между собой управляющей графической оболочкой и общей информационной шиной (ОИШ), и показана на рис. 7.5.

Подсистема *Информационный ресурс* подразделяется на внешний и внутренний ИР. В свою очередь, внешний ИР подразделяется на два вида: первый из них создаётся на основе первоисточников, предшествуя процессу автоматизированного построения онтологии ПдО (ЭлК энциклопедических словарей, толковых словарей и тезаурусов), второй – подключается в процессе создания О ПдО (сеть Internet, другие источники ТД и распределённые базы знаний).

Внутренний ИР также подразделяется на два вида. Первый из них создаётся как результат реализации алгоритмов автоматизированного построения О ПдО (библиотеки онтологий ПдО, онтологий ТД и терминов и понятий домена ПдО), второй – создаётся и используется в процессе построения О ПдО и представлен базой данных ТД.

Электронные коллекции энциклопедических и толковых словарей являются (по определению) общезначимыми ресурсами знаний в заданном домене прикладных областей. Они содержат, как правило, несколько определений для каждого понятия с учётом смысловых оттенков и ориентированных на широкий круг

потребителей информации. Такое представление знаний соответствует основным принципам построения формальной онтологии, а их смысловая интерпретация является основным источником формирования онтологии ПдО для инженера по знаниям.

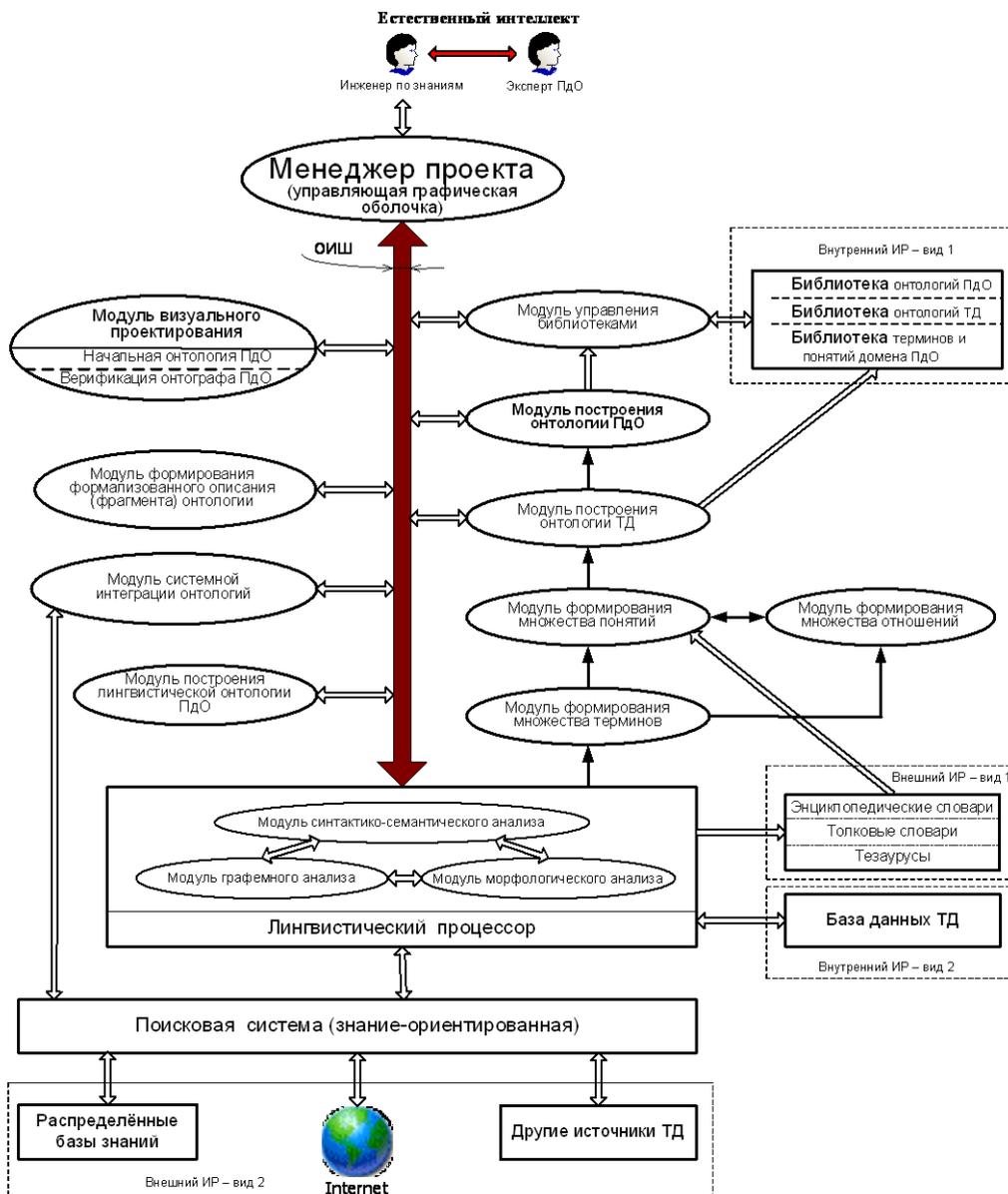


Рис. 7.5. Архитектурно-структурная организация ИКОН

Онтология ПдО разрабатывается безотносительно к конкретной прикладной задаче и, поэтому, представляет собой некоторую обобщённую (с точки зрения иерархии понятий) и усреднённую (с точки зрения множеств функций интерпретации понятий) понятийную структуру предметных знаний. С одной стороны (преимущество) онтология ПдО является открытой, общезначимой системно-онтологической структурой, доступной к пополнению новыми знаниями и/или

адаптации к конкретному кругу задач. С другой стороны (недостаток) общезначимые структуры не являются оптимальными для конкретных прикладных задач и требуют приложения серьезных усилий разработчиков, чтобы стать таковыми. Следствием указанного выше преимущества формальной онтологии ПдО является её многократное использование для разных наборов типовых задач. Другими словами, один раз построенная формальная онтология ПдО пригодна для решения произвольного набора задач. Когда же речь идёт о поиске оптимальной онтологической системы, необходимо адаптировать онтологическую базу знаний к целевому применению (исключить не используемые цепочки понятий в онтографе, конкретизировать множества функций интерпретации и ограничить значения признаков).

Другая составляющая внешнего ИР представляет собой совокупность источников с естественно-языковой и аналитической (формально описанные онтологические базы знаний) формами представления информации. Они сгруппированы применительно к задаче системной интеграции технологий соответственно: автоматизированного построения О ПдО и междисциплинарных научных знаний. Причём, часть блоков являются общими для обеих технологий.

Такой источник как *сеть Internet* не требует дополнительного пояснения, специфика обработки извлечённой из сети информации проявляется в организации работы поисковой системы и лингвистического процессора. В качестве *других источников ТД* используются электронные библиотеки, содержащие файлы монографий, научно-технических статей и других описаний предметных знаний.

Распределённые базы знаний, технология их использования и обмена между заинтересованными участниками, входящими в подсистему ЕИ, представляют самостоятельную научно-техническую проблему. Концепция развития междисциплинарных научных исследований, затрагивающая, в том числе, и указанные выше вопросы, приведена в [99].

Библиотеки внутреннего ИР являются конечным результатом проекта (проектов), представляют собой глубокоструктурированное хранилище знаний домена прикладных областей, которое путём применения технологии системной интеграции превращается в *общее интегрированное пространство трансдисциплинарных знаний*.

Библиотеки онтологий ПдО и ТД состоят из разделов, каждый из которых, в свою очередь, разбит на две части – для хранения онтографа и формально-логического описания. Библиотека терминов и понятий структурирована по предметным областям. В соответствующих записях понятий проставлены индексы их принадлежности к соответствующему уровню иерархии (метаонтология, онтология домена или онтология ПдО).

С накоплением объёмов информации в библиотеках расширяются и их функциональные возможности. В частности, системы разнообразных научных знаний, представленных системно-онтологическими структурами, могут существенно улучшить качество наполнения электронных курсов (дистанционного) обучения, спроектированных на основе онтологического подхода.

В дальнейшем, при достижении в библиотеках представительного объёма информации, следует реализовать “обратную связь” для библиотеки онтологий ТД и библиотеки терминов и понятий домена ПдО, когда результаты работы ИКОН становятся источником исходной информации, занимающей (по структурированности) промежуточное место между источниками онтологий и источниками ЕЯТ.

Второй вид внутреннего ИР связан с обработкой текстовой информации в лингвистическом процессоре и поисковой системе.

База данных ТД подключена к лингвистическому процессору (и поисковой системе) и хранит оригиналы текстовых документов, проиндексированных во внутреннем индексном формате ИКОН. Она также предназначена для хранения множества ЛКТ разных ПдО домена.

Подсистема **Естественный интеллект** представлена процедурами поддержки работы инженера по знаниям и эксперта в заданной ПдО. Инженер по знаниям во взаимодействии с управляющей графической оболочкой (менеджером проекта) выбирает и запускает на исполнение основные процедуры информационной технологии обработки текстовой информации. Эксперт участвует в построении категориальных знаний (начальной онтологии) заданной ПдО и принятии решения об окончании (или возврате к соответствующему итерационному циклу) процесса построения онтографа ПдО.

Подсистема **Программно-аппаратные средства** представляет собой совокупность как оригинальных программных модулей, реализующих соответствующие алгоритмы этапов проектирования О ПдО, так и известных инструментальных средств, свободно распространяемых в сети Internet, допускающих расширение и встраивание в пользовательские системы, в частности, инструментальная система Protégé [163]. Система Protégé используется как один из основных компонентов модулей визуального проектирования онтографа начальной онтологии, проверки онтографов, формирования формализованного описания спроектированной онтологии ПдО.

Кроме указанного выше деления программно-аппаратных средств на оригинальные и известные, эта подсистема включает программную оболочку *Менеджер проектов*, в состав которой входит ОИШ, поисковую систему, когнитивный лингвистический процессор [138] и совокупность программных модулей, реализующих процедуры обработки языковых и предметных знаний из заданного домена прикладных областей.

Менеджер проектов – прикладная программа, управляющая всеми проектными процедурами. Он интегрирует все компоненты ИКОН в одну объединённую среду и выполняет следующие функции:

- во взаимодействии с инженером по знаниям осуществляет предварительное наполнение среды внешними электронными коллекциями энциклопедических, толковых словарей и тезаурусов, описывающих домен предметных знаний;
- обеспечивает запуск и последовательность исполнения прикладных программ, реализующих составные компоненты информационной технологии проектирования онтологии ПдО и системной интеграции междисциплинарных знаний;
- отображает ход процесса проектирования;
- обеспечивает интерфейс с блоком ЕИ (содержит позиции меню для запуска как последовательностей, так и отдельных прикладных программ, используемых в процессе проектирования);
- индицирует сообщения о текущем состоянии проекта, его наполнения информационными ресурсами;
- обеспечивает обмен информацией между прикладными программами и базами данных через ОИШ.

Поисковая система предназначена для поиска и извлечения из различных внешних источников текстовой информации и формирования ЛКТ в базе данных ТД. В качестве поисковой системы могут быть использованы как известные программы поиска, так и знание-ориентированная поисковая система [142]. В первом случае отбор релевантных текстовых документов заданной ПдО осуществляется традиционным способом задания ключей. Во втором случае указанная процедура

реализуется в самой ЗОИС, где эта функция возлагается на когнитивный лингвистический процессор. При этом достигается существенное повышение релевантности найденных документов по запросу пользователя за счёт снятия разного рода неоднозначностей и учёта “фоновых” знаний при обработке ТД. Запрос инженера по знаниям на поиск формируется в виде онтографа запроса, построенного на основе соответствующего тезауруса.

Когнитивный лингвистический процессор (или процессор, основанный на знаниях) (КЛП) выполняет процедуры лингвистического анализа с учётом контекстных знаний [138, 143, 164, 165]. КЛП включает модули графемного, морфологического и синтактико-семантического анализа. С одной стороны, КЛП соединён с поисковой системой, от которой к нему поступает множество ТД. С другой стороны КЛП соединён с ОИШ Менеджера проектов, через которую обработанная лингвистическая информация поступает в соответствующие программные модули и/или базы данных. Кроме традиционных лингвистических процедур КЛП является ведущим компонентом при наполнении базы данных ТД (для случая, когда используется известная поисковая машина), выполняющим отбор релевантных заданной ПдО текстовых документов, составляющих ЛКТ.

Программные модули представляют собой набор прикладных программ, обрабатывающих экстралингвистическую информацию, и формирующих понятийные структуры разного уровня представления.

Отдельным программным модулем является **модуль управления библиотеками**. Он выполняет множество функций (по сути, является системой управления некоторой БД), основными из которых являются:

- запись, накопление и хранение информации в соответствующих библиотеках;
- чтение по запросу менеджера проектов из библиотек запрашиваемой информации и её передачу на ОИШ;
- корректное представление интегрированной онтологии ПдО совместно с модулем системной интеграции и др.

Следующие три модуля манипулируют экстралингвистической информацией, полученной после обработки в когнитивном лингвистическом процессоре. По сути, эти три модуля формируют списки элементов множеств X , R и F , являющихся компонентами схемы формальной модели онтологии ПдО.

Модуль формирования множества терминов. Для автоматического выделения терминов из множества ТД, описывающих заданную ПдО, используется поиск и выделение субстантивных именных словосочетаний, выражаемых схемой – “согласуемые слова + существительное”. В этой модели существительное является главным словом, а согласуемые слова – зависимыми и могут выражаться как прилагательными, так и существительными. Словосочетания могут включать в свой состав также предлоги и союзы. Процедура выделения терминов из текста включает два основных этапа.

На первом этапе происходит непосредственный поиск в тексте слов и словосочетаний – кандидатов в термины. В качестве однословных терминов выбираются существительные и аббревиатуры. Процедура автоматического выделения однословных и многословных терминов использует результаты синтактико-семантического анализа текста. Распознавание поверхностных семантических отношений осуществляется с помощью анализа флексий полных слов, учитывая предлоги и союзы, без полного предварительного грамматического разбора.

Многословные термины формируются с помощью определенных типов отношений между словами предложения, путем постепенного присоединения слов к

однословному термину-существительному. Для терминов – именных словосочетаний используются следующие основные типы отношений между словами: объектное, принадлежность (между двумя существительными), определительное (между прилагательным и существительным), однородные слова (между двумя существительными или двумя прилагательными). Выделенные группы слов проверяются на соответствие заданным *шаблонам терминов*. Порядок расположения в предложении слов, образующих термин, может точно не соответствовать заданному шаблону, но обязательным условием выделения термина является соответствие отношений между словами определенным типам отношений.

На втором этапе список кандидатов в термины фильтруется: учитывается значимость выделенных словосочетаний (приближенность в дереве разбора к подлежащему или сказуемому предложения) и частота, с которой они встречаются в тексте.

По результатам работы этого модуля в *модуле построения лингвистической онтологии ПдО* формируется лингвистическая онтология ПдО, являющаяся связующим звеном при переходе от обработки языковых знаний к формализации и обработке предметных знаний. Следует отметить, что построение ЛО ПдО выполняется при разработке онтолого-управляемой информационной системы заданной ПдО.

Модуль формирования множества понятий. Под понятием будем понимать обобщённую модель некоторого класса предметов по определённым, общезначимым, специфическим для него признакам. Основой для формирования множества понятий являются выделенные термины. Автоматическое формирование понятий выполняется путем поиска *определений терминов*, которые в явном виде приведены в текстовых документах и электронных коллекциях энциклопедических, толковых словарей и тезаурусов. По окончании процедуры формирования множества понятий в соответствующую библиотеку записываются множества понятий и терминов.

Модуль формирования множества отношений. Он использует выделенные синтактико-семантические отношения и информацию из лексикографической базы данных об онтологических признаках для слов с грамматическими характеристиками глагола, причастия, деепричастия и существительного. Для различных типов отношений определены лингвистические способы их описания в тексте. Пересечение множеств онтологических признаков и синтактико-семантических отношений позволяет сформировать множество отношений между понятиями.

Модуль построения онтологии ТД манипулирует элементами множеств терминов, понятий и отношений, полученными после обработки некоторого текстового документа в предыдущих модулях и записанными в соответствующую библиотеку. Сначала из множеств терминов и понятий выделяются характеристики.

Далее последовательно выполняются две основные процедуры этого модуля – построение онтографов синтаксических единиц ТД и поочерёдное их склеивание в онтограф ТД. И в заключение к понятиям последнего формируются множества функций интерпретаций с использованием приписанных понятий-свойств.

В случае необходимости модуль построения онтологии ТД может запросить информацию о синтаксических деревьях предложений соответствующего ТД или даже обратиться к его оригиналу. Затем онтограф ТД передаётся в модуль визуального проектирования, где инженер по знаниям выполняет его проверку.

Особенностью следующих двух модулей является то, что они реализованы на основе открытого программного кода известной инструментальной системы Protégé.

Модуль визуального проектирования состоит из двух частей – разработки начальной онтологии ПдО и проверки онтографа ПдО. Этот модуль обеспечивает этапы ручного, неавтоматизированного проектирования онтологии ПдО. Управляющие процедуры и информационное наполнение поступают к нему через управляющую графическую оболочку. Модуль реализует две важные функции ИКОН – ручное проектирование начальной онтологии ПдО и проверку онтографа спроектированной онтологии ПдО.

Начальная онтология ПдО, по сути, является “размытой”, когнитивно-графической схемой базовой структуры понятий ПдО. От качества её составления зависит количество итерационных повторений отдельных этапов проектирования, что в конечном итоге влияет на экономические показатели проектирования.

Схема названа “размытой”, потому что структура связей между понятиями не является однозначно определённой соответствующими отношениями. Два связанных понятия могут не отображать связи типа “класс–подкласс”, “род–вид”, “целое–часть” и др., а быть связанными лишь отношением меронимии. И между ними (вершинами-понятиями ОГ) в процессе проектирования онтологии ПдО предполагается встраивание произвольных цепочек вершин-понятий, но с конкретизированными связями или дугами ОГ (конкретными именами отношений). Когнитивно-графическая схема названа потому, что блок ЕИ при её составлении руководствуется “глубинными” знаниями ПдО, выбирает из них базовые, определяющие понятия и представляет их в графической (графовой) форме.

Проверка (evaluation) онтографа ПдО. Вторая часть модуля предназначена для интерактивной оценки (проверки и коррекции) онтографа ПдО, в общем случае представляющем ациклический граф. Ведущая роль при выполнении этой процедуры закреплена за экспертом в заданной ПдО, так как только он в полной мере владеет совокупностью предметных знаний. Проверка может быть выполнена как “сверху–вниз”, так и “снизу–вверх”. Однако на практике чаще всего применяется комбинированная методика, проверка начинается со средних вершин ОГ (так как именно они наиболее часто встречаются в описаниях ТД) и расширяется параллельно вверх и вниз по дугам онтографа. Визуализация и корректировка вершин и дуг ОГ производится в графическом окне системы Protégé. Меню графики содержит достаточный набор команд¹⁰, обеспечивающих реализацию указанных функций, оперирующих ОГ. Процедура проверки завершается просмотром (и корректировкой) всех цепочек (путей) в онтографе. Частным случаем процедуры оценки онтографа ПдО является проверка онтографа ТД.

Описанная процедура проверки является заключительной перед принятием решения инженером по знаниям совместно с экспертом ПдО об окончании процесса проектирования. Далее к онтографу присоединяются функции интерпретации соответствующих вершин-понятий. Указанное присоединение выполняется в окне индикации фреймов и слотов системы Protégé. При этом формируется внутрисистемное представление онтологии ПдО, которое передаётся в модуль формирования формализованного описания онтологии ПдО.

Модуль формирования формализованного описания онтологии в автоматическом режиме формирует формализованное описание онтологии ПдО на одном

¹⁰ Недостающие команды реализуются отдельными программными процедурами (плагинами) и встраиваются в систему.

из общепринятых языков описания онтологий (и рекомендуемых консорциумом W3C), поддерживаемых системой Protégé (OWL, RDFS) и передаёт на хранение (под управлением менеджера проектов и модуля управления библиотеками) в библиотеку онтологий ПдО.

Модуль построения онтологии ПдО является одним из основных компонент инструментального комплекса и выполняет следующие функции:

- считывает из модуля визуального проектирования начальную онтологию ПдО. Из ЭлК считывает определения понятий, входящих в начальную онтологию, и формирует множество функций интерпретации;
- считывает из соответствующей библиотеки онтографы и формализованные описания онтологий ТД, проверяет их на непротиворечивость;
- анализирует на полноту множество функций интерпретации. Просматривает ЭлК энциклопедических, толковых словарей и тезаурусов и пополняет последние в отношении соответствующих понятий-объектов и понятий-процессов. Кроме того, в процессе просмотра могут быть уточнены и дополнены соответствующие отношения между понятиями;
- в процессе построения онтологии, по необходимости, обращается к любому информационному хранилищу;
- выполняет поочерёдное “склеивание” онтографов и функций интерпретации онтологий ТД;
- выполняет “склеивание” общего онтографа ТД и множеств функций интерпретации с онтографом и множествами функций интерпретации начальной онтологии соответственно;
- передаёт полученный результат в модуль визуального проектирования на верификацию.

Модуль системной интеграции онтологий является основным компонентом при реализации технологии системной интеграции междисциплинарных научных знаний. Он поддерживает процедуру интеграции двух онтологий, одна из которых является исходной, задаётся инженером по знаниям и находится в библиотеке онтологий ПдО, а другая поступает к нему от некоторой распределённой базы знаний¹¹. В результате будет построен компьютерный прототип общего интегрированного пространства трансдисциплинарных знаний.

7.2.2. Алгоритм автоматизированного построения онтологии ПдО

ИКОН предназначен для реализации ряда информационных технологий, которые интегрируются в две большие группы – автоматизированное построение онтологий ПдО и системная интеграция междисциплинарных научных знаний. Эти группы технологий синтезированы из концептуальной модели интегрированной ИТ обработки знаний, содержащихся в естественно-языковых объектах (см. 1.1.2). Приведём цепочки компонент-технологий, входящих в указанные выше группы, и опишем их обобщённые алгоритмы [2, 5, 49, 66, 142, 143, 165, 166].

Интегрированная технология автоматизированного построения онтологии ПдО ::= Поиск в сети Internet и/или других ЭлК текстовых документов (Т1) → Автоматическая обработка ЕЯТ (Т2) → Извлечение, структурирование и формально-логическое представление знаний (Т3) → Создание, накопление и использование баз онтологических знаний с большим объёмом информации (Т4).

¹¹ Такая же схема объединения онтологий принята в системе WonderWeb [106].

На рис. 7.6 приведен обобщённый алгоритм процесса автоматизированного построения онтологии ПдО, реализующего приведенную выше цепочку технологий.

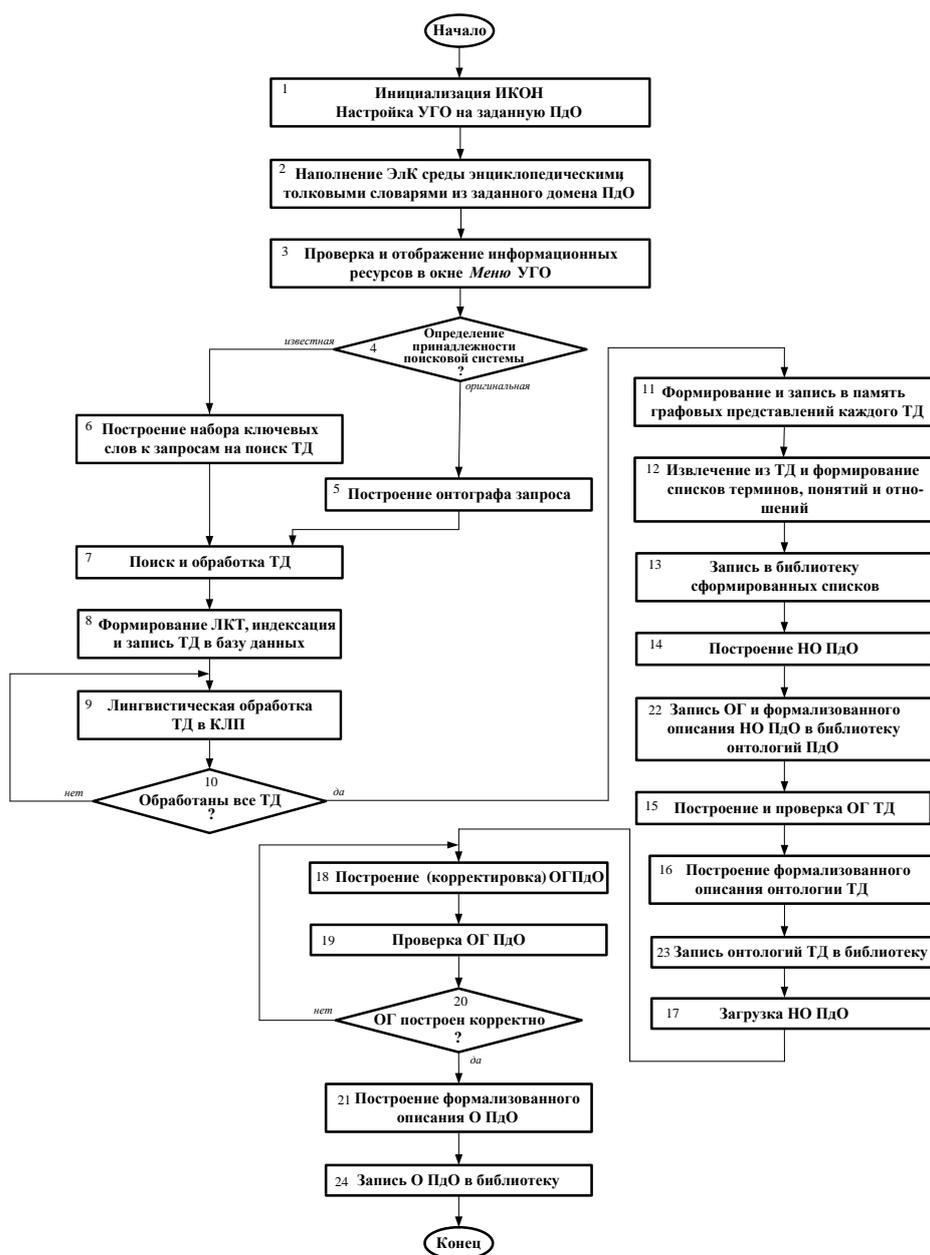


Рис. 7.6. Обобщённый алгоритм процесса автоматизированного построения онтологии ПдО

Опишем этапы обобщённого алгоритма.

Предварительный этап (п.п. 1–4).

Инструментальный комплекс предназначен для автоматизированного построения онтологии для произвольной предметной области и в его состав входят как оригинальные, так и известные программные модули. Поэтому требуется предварительная настройка компонент системы, что (в основном) выполняется вручную инженером по знаниям. Он указывает системе, для какой ПдО будет строиться

онтология, какой выбран формальный язык описания онтологий и др. Большинство настроек выполняется в центральной программе – управляющей графической оболочке. В графическом окне *Меню* УГО задаются виды и способы реализации ручных шагов проектирования, предпочтительные варианты их реализации, предельные значения (интервалы) некоторых параметров и др.

Электронные коллекции энциклопедических и толковых словарей (и тезаурусов) ИКОН являются внешним ИР, создаваемым на предварительном этапе процесса проектирования. Поэтому реализация этого шага не регламентирована методикой проектирования.

Инженер по знаниям самостоятельно выбирает источники указанных информационных ресурсов и пути реализации этого шага проектирования.

Во взаимодействии инженера по знаниям и УГО выполняется сбор информации об имеющихся информационных ресурсах. Менеджер проектов через ОИШ обращается или непосредственно, или через программные модули, КЛП и поисковую систему к соответствующим базам данных и памяти. Результаты сбора информации отображаются в окне *Меню* УГО.

Далее выполняется идентификация имеющейся в распоряжении комплекса поисковой системы (известной или оригинальной).

Конечно, на предварительном этапе осуществляется и ряд частных настроек, относящихся к конкретным компонентам комплекса и алгоритмам их работы.

Поиск во внешних источниках текстовых документов (п.п. 5–8).

П. 5 реализуется в случае использования в среде оригинальной, знание-ориентированной поисковой системы. При этом запрос на поиск ЛКТ ПдО задаётся инженером по знаниям в виде расширенной спецификации, составленной на естественном языке. Затем эта спецификация передаётся в КЛП и ЗнПС. Последние, во взаимодействии, формируют онтографы запросов на поиск ТД.

При использовании в ИКОН традиционной поисковой системы инженер по знаниям формирует набор ключевых слов, совокупность которых представляет собой запрос на поиск ТД.

Далее выполняется поиск во внешних источниках текстовых документов и их предварительная обработка (для случая использования ЗнПС).

Из всего набора найденных текстовых документов формируется лингвистический корпус текстов, непосредственно относящихся (релевантных) к заданной ПдО. Считается, что он является представительным. Далее когнитивный лингвистический процессор (для случая с ЗнПС – совместно) выполняет индексацию ТД во внутреннем формате ИКОН и запись в базу данных ТД.

Автоматическая обработка ЕЯТ (п.п. 9–10).

Эта технология по временным затратам является наиболее трудоёмкой. С другой стороны, она хорошо проработана [65, 66, 85, 86, 143, 165 и др.].

Извлечение, структурирование и формально-логическое представление знаний (п.п. 11–21).

К знаниям в данной технологии причислена вся экстралингвистическая информация, выделенная из ЛКТ и полученная после обработки в когнитивном (знание-ориентированном) лингвистическом процессоре. Структурированная в понятийные структуры, она переходит в виды онтографового и формально-логического представлений. Результаты работы КЛП, представленные в виде синтаксических, семантических и концептуальных графов, записываются в соответствующую память. Сформированные на предыдущем шаге графовые представления ТД уже содержат предварительные наборы терминов и отношений. На этом шаге происходит их окончательное формирование в виде списков в модулях формирования мно-

жеств терминов и отношений. Результаты работы указанных модулей являются исходной информацией для модуля формирования множества понятий. При этом используются определения и описания терминов из энциклопедических, толковых словарей и тезаурусов домена прикладных областей.

Далее выполняется запись сформированных на предыдущем шаге списков терминов и понятий в соответствующую библиотеку. Эти списки используются на этапе структурирования знаний и могут уточняться в процессе обработки.

Затем выполняется построение начальной онтологии ПдО в интерактивном режиме. Оно может быть выполнено и параллельно с предыдущими шагами технологии Т4, если для подсистемы ЕИ нет необходимости в просмотре сформированного списка понятий.

Далее выполняется построение онтографов для каждого текстового документа (такие онтографы являются первой компонентой в онтологии ТД) и их формализованные описания. Для этого, внутреннее представление онтографов ТД передаётся в соответствующий модуль. Сформированный в последнем результат передаётся обратно в модуль формирования онтологий ТД, в котором формируется окончательное представление онтологий ТД. Отметим, что промежуточные результаты, полученные в модуле системы Protégé, доступны для просмотра в подсистеме ЕИ. После этого выполняется загрузка начальной онтологии ПдО в модуль формирования онтологии ПдО.

Следующий шаг является основным во всём обобщённом алгоритме. В нём выполняется построение (а в случае необходимости – и корректировка) общего онтографа О ПдО. Исходным является онтограф НО ПдО. К нему поочередно присоединяются (“склеиваются”) онтографы текстовых документов. По запросу подсистемы ЕИ процесс “склеивания” может быть визуализирован в системе Protégé и откорректирован.

В заключение этого этапа выполняется окончательная оценка онтографа О ПдО и, если онтограф построен корректно – формируется формализованное описание О ПдО.

Создание, накопление и использование баз онтологических знаний с большим объёмом информации (п.п. 22–24).

Эта технология по своему содержанию является ёмкой и в рассматриваемом обобщённом алгоритме представлена только созданием и накоплением информации в известных СУБД. Структурирование и извлечение из неё информации регламентируется соответствующими описаниями.

На этом этапе выполняется запись онтографа, формализованного описания начальной онтологии ПдО, онтологий текстовых документов и онтологии ПдО в соответствующие библиотеки онтологических информационных структур.

7.3. Методология системной интеграции междисциплинарных научных знаний

Как отмечалось ранее, разработка методов и средств построения формальных компьютерных онтологий и их системной интеграции интенсивно развивается и нацелена на повышение эффективности *научных исследований* (НИ) сложных предметных областей и в первую очередь, междисциплинарных НИ. Вместе с тем следует отметить, что теория познания как основа формирования методологии и процессов НИ претерпевает сегодня радикальные изменения. Во-первых, в системе научного знания проходят интенсивные процессы дифференциации и интеграции

знаний, развиваются комплексные и междисциплинарные исследования, новые способы и методы познания, методологические установки, появляются новые элементы картины мира, выделяются новые, более сложные типы объектов познания, характеризующиеся историзмом, универсальностью, сложностью организации, которые раньше не поддавались теоретическому (математическому) моделированию. Одно из таких новых направлений в современном естествознании представлено синергетикой.

Следующим шагом развития междисциплинарных научных исследований, и теории баз знаний в частности, является теоретически обоснованное объединение (или *системная интеграция*) уже разработанных как общедоступных онтологий, так и коммерческих баз знаний для разнообразных прикладных задач, проблем, целых предметных областей и трансдисциплинарных знаний общего характера. В [99] сущность системной интеграции сформулирована следующим образом: “Устойчивые знания совокупности научных дисциплин можно представить в форме интегрированной иерархической сети научных теорий (разного уровня развития, содержательности и охвата действительности), составляющие которых, возможно, связаны посредством общих объектов действительности”. Там же говорится о цели междисциплинарных исследований – приближение к построению общенаучной картины мира – а также о системной интеграции знаний (онтологий) как одной из важных задач в достижении указанной цели.

Системная интеграция двух и более онтологий представляется далеко не тривиальной задачей. Её алгоритмы не проработаны, требуют тщательной проверки и апробации на представительном множестве онтологий. Перечислим его (алгоритма) основные шаги.

1. Считывание исходной онтологии ПдО из соответствующих информационных источников (библиотеки).
2. Разложение исходной онтологии на списки множеств концептов, отношений и функций интерпретации.
3. Повторение шагов 1, 2 для интегрируемой онтологии.
4. Для сравниваемых элементов множеств понятий проверка на соответствие множеств отношений и функций интерпретации. При этом следует учитывать совпадение/несовпадение имён понятий-объектов и понятий-процессов.
5. Построение общих онтографа и функций интерпретации двух онтологий.
6. Передача полученной общей онтологии в модуль визуального проектирования на верификацию. Далее процедуры проверки, формирования формализованного описания и записи в библиотеку аналогичны для описанной ранее схемы построения онтологии ПдО.

Системная интеграция междисциплинарных научных знаний ::= Поиск в сети Internet и/или других ЭК онтологий ПдО (Т1) → Системная интеграция онтологических знаний (Т5) → Извлечение, структурирование и формально-логическое представление знаний (Т3) → Создание, накопление и использование баз онтологических знаний с большим объёмом информации (Т4).

На рис. 7.7 приведен обобщённый алгоритм работы ИКОН для процесса системной интеграции междисциплинарных научных знаний, реализующего приведенную выше цепочку технологий. На нём приняты следующие сокращения: $O_{исх}$ – исходная онтология; $O_{ин}$ – интегрируемая онтология; O^* – интегрированная (результатирующая) онтология.

Опишем этапы обобщённого алгоритма.

Предварительный этап (п.п. 1–3).

Шаги предварительного этапа для обоих алгоритмов, в основном, аналогичные. Ниже отметим только их особенности для описываемого алгоритма.

На этом этапе инженер по знаниям задаёт исходную онтологию (онтологию, которая считается правильно построенной (с математической точки зрения) и предпочтительной, когда необходимо сделать выбор между компонентами двух онтологий $O_{исх}$ и $O_{ин}$) и выбирает источники ЭлК для предполагаемых к интеграции онтологий, а также выполняется сбор информации о содержимом баз данных словарей и библиотеки онтологий ПдО, которое затем отображается в окне *Меню* УГО.

Поиск во внешних источниках онтологий ПдО (п.п. 4–6).

Реализация этого шага полностью зависит от профессиональных навыков инженера по знаниям и осуществляется в ручном режиме. Он инициирует процедуру поиска во внешних источниках онтологий предметных знаний и их предварительную обработку. Последняя может заключаться, в частности, в просмотре графического и аналитического представлений в соответствующих окнах системы Protégé. Далее подсистема ЕИ принимает решение о выборе интегрируемой онтологии из совокупности найденных.

Системная интеграция онтологических знаний (п.п. 7–14).

Сначала в модуль системной интеграции загружаются исходная и интегрируемая онтологии, после чего выполняется их декомпозиция на исходные кортежи $\langle X, R, F \rangle$ (при этом понятия-объекты и понятия-процессы представлены общим списком) и поэлементное сравнение компонент кортежей. Главными компонентами для сравнения являются $X_{исх}$ и $X_{ин}$. После сравнения имён указанных компонент, далее сравниваются компоненты $F_{исх}$ и $F_{ин}$, и затем $R_{исх}$ и $R_{ин}$. В процессе сравнения составляются соответствующие списки, необходимые для выполнения последующих действий.

Далее выполняется формальное объединение двух онтографов в один, общий онтограф OG^* интегрированной онтологии O^* и ручная корректировка содержаний онтографов $OG_{исх}$ и $OG_{ин}$ в модуле визуального проектирования.

Извлечение, структурирование и формально-логическое представление знаний

На этом этапе выполняется окончательная проверка построенного OG^* и в подсистеме ЕИ принимается решение об окончании процесса интеграции онтологий. В случае корректного построения результирующего онтографа составляется формализованное описание интегрированной онтологии O^* в соответствующем модуле.

Создание, накопление и использование баз онтологических знаний с большим объёмом информации (п. 15).

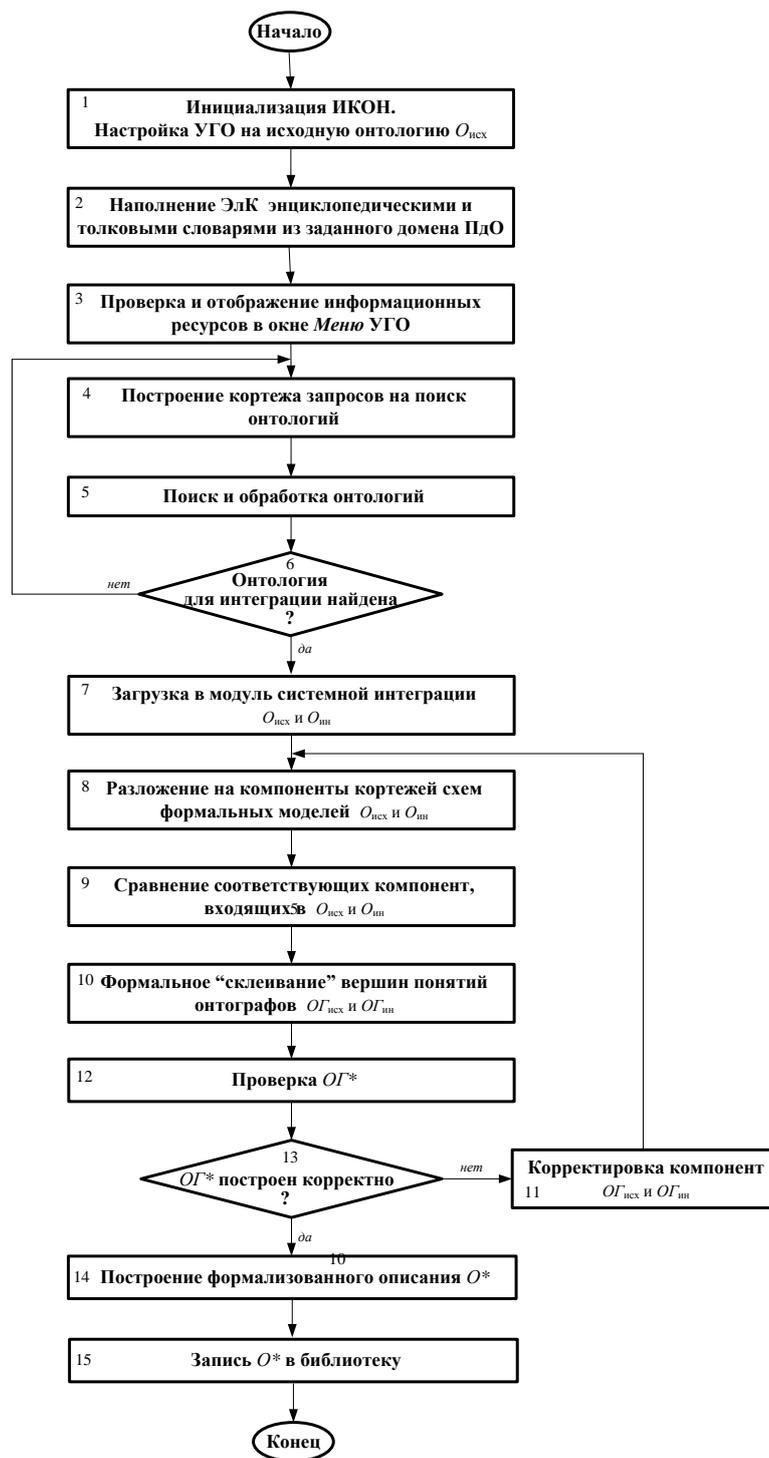


Рис. 7.7. Обобщённый алгоритм работы ИКОН для процесса системной интеграции онтологических знаний

В заключение выполняется запись окончательного результата работы обобщённого алгоритма – интегрированной онтологии O^* – в соответствующий раздел библиотеки онтологий ПдО.

7.4. Перспективы исследования проблем системной интеграции онтологических знаний

Исследование проблем системной интеграции онтологических знаний следует рассматривать в контексте междисциплинарных научных исследований (МНИ), хотя они имеют и самостоятельное научно-практическое применение. Концепция развития МНИ изложена в [99], в которой методология системной интеграции онтологических знаний, построение онтологизированных систем знаний предметных областей являются одними из главных составляющих компонентов. При этом основная роль отводится формальным (компьютерным) онтологиям предметных областей.

Онтологии позволяют формализовать и компактно представлять накопленные знания, одновременно определяя и объединяя терминологию различных предметных областей, строить единую научную картину мира как результат комплексных трансдисциплинарных исследований [167].

При этом онтологии можно представить не просто в виде составных компонентов, а как универсальный инструмент междисциплинарных научных исследований в виде пирамиды наиболее важных обобщенных функций онтолого-управляемых информационных систем (рис. 7.8).



Рис. 7.8. Обобщённые функции ОУИС

Общая характеристика междисциплинарных исследований

Знания совокупности научных дисциплин можно представить в форме интегрированной иерархической сети научных теорий (разного уровня развитости, формальности, содержательности и охвата действительности), составные части которых, возможно, связанные через общие объекты действительности (онтологические категории).

В наиболее общем виде внутренне-дисциплинарный генезис науки имеет две составные части: частные теории, решающие специфические проблемы относительно части объектов дисциплины, и фундаментальные, фиксирующие наиболее общие знания относительно всей области дисциплины: от фундаментальной до отдельных теорий, которые углубляют её. Всякое междисциплинарное исследование состоит в выявлении новых отношений между понятиями исходных дисциплин, в установлении новой системы законов, которые их связывают, и синтезе прагматики решения новых, всё более сложных задач [99]. При этом знания исходных дисциплин могут оставаться неизменными (простейший, линейный, случай междисциплинарного взаимодействия), включенными целиком (или выборочно) в новую иерархическую структуру или модифицироваться, развиваться благодаря процессам обмена парадигмальными установками, понятиями и методами разных наук (нелинейное междисциплинарное взаимодействие), являющееся отображением сущности междисциплинарной проблемы. Нелинейное междисциплинарное взаимодействие в совокупности с проблемной ориентацией – общепризнанная норма эволюции современной науки [39].

При этом де-факто формируется новая трансдисциплинарная теория, возможно, и научная область, и бесконечно повторяется итерационный процесс (от одних дисциплин – через междисциплинарное исследование – к новым дисциплинам с новым предметом исследования) в направлении приближения к целостной *общенаучной картине мира* [99].

Проблемные ситуации, возникающие на пути бесконечного процесса познания, постоянно усложняются, охватывают каждый раз всё больше явлений действительности, и преобладающая их часть нуждается в проведении междисциплинарных исследований.

Итак, сущность проблемы оптимизации информационной поддержки МНИ проявляется в обеспечении эффективности проведения научных исследований благодаря преодолению препятствий в *междисциплинарном взаимодействии* в современной науке.

Решение этой проблемы является ключом, который ускорит темпы развития науки и научно-технического прогресса общества в целом. Это возможно лишь в результате создания методологии, информационных систем и технологий, адекватных современным процессам профессиональной деятельности учёных.

Решение этой проблемы зависит от комплекса условий (научных, организационных, материально-технических, финансовых и кадровых).

Каждую из реальных проблем, с формальной точки зрения, можно представить как определённую иерархическую сеть составных частей проблем, принадлежащих к разным научным теориям (дисциплинам) или её разделам, теориям.

В мировой науке *отсутствует системная методология* и определенная форма общепринятой или *ведущей конструктивной научной теории*, которые бы обеспечили постановку и эффективное решение произвольных научных проблем не только междисциплинарных исследований, но и отдельных разделов науки. Основными факторами такой ситуации есть её (методологии) *сверхвысокая слож-*

ность и комплексный характер. Большие трудозатраты приходится на поиск, экстракцию и формализованное конструктивное представление существующих в мировом пространстве знаний, последующее их преобразование в удобную для исследователя форму. Такая форма, которая отвечает требованиям конструктивности, унифицированности, интеграционной полноты и когнитивного совершенства названа в [99] *канонической*.

Оптимизация междисциплинарных научных исследований

Исследование и разработка комплекса информационных технологий и систем поддержки МНИ, создание условий для существенного роста производительности профессиональной деятельности учёных, повышение эффективности, уровня и места науки (прежде всего отечественной) в государстве и мировом сообществе позволяют сформулировать и предложить широкому кругу научных исследователей общую концепцию оптимизации системно-информационной поддержки междисциплинарных научных исследований.

Во-первых, учитывая естественное развитие науки и возрастание требований к ней со стороны общества основой управления междисциплинарными исследованиями должны стать интегрированные информационно-технологические системы, обеспечивающие организационные процессы, мониторинг научных исследований, регламентацию всех этапов их жизненного цикла (включая электронный документооборот), анализ и оценку результатов исследований и принятие на этой основе решений и определение актуальных направлений и т. п. На этих началах необходимо создать *общее интегрированное пространство трансдисциплинарных знаний*, где синергетически будут взаимодействовать многочисленные коллективы профессионалов разных предметных областей, которые сосредоточат внимание на решении важнейших трансдисциплинарных научно-практических проблем. Их работу в реальном времени и пространстве будут поддерживать эффективные информационные технологии, позволяющие превратить имеющиеся знания в конечный товар, измеряемый в показателях количества и качества и имеющий определённую цену.

Во-вторых, (один из главных векторов концепции) – определение и воплощение необходимых и достаточных условий для эффективного функционирования *конструктивных знаний*.

Для этого необходимо перейти от режима недетерминированного продуцирования и использования знаний субъектами научного процесса к режиму *эффективного управления знаниями* и их применения (knowledge management). При этом актуализируются не только знания и их качество, но и *метазнания* (т. е. знание о знании), обеспечивающие функционирование агентных технологий и рыночных механизмов (прежде всего их товарный статус).

Таким образом, эффективные информационные технологии междисциплинарных научных исследований должны строиться на основе научной теории и конструктивной системной методологии, разработке новых информационных технологий, быть пригодными для постановки и решения научных проблем МНИ высокой сложности и обеспечиваться эффективной инфраструктурой управления процессами МНИ, а следовательно и Knowledge Management.

Создание информационных технологий базируется на компьютерно-сетевых системах нового поколения (интеллектуальных корпоративных сетях, грид-системах и т. п.), которые интегрируют высокоскоростную телекоммуникационную систему, знание-ориентованную архитектуру, средства обработки и представления информации, математическое обеспечение.

Ядро интегрированных информационных технологий для МНИ составляют системно-интегрированные базы конструктивно представленных знаний, распределённые знание-ориентированные услуги, обеспечивающие высокоорганизованный доступ к информационным и вычислительным ресурсам, выполнение таких функций, как: выявление закономерностей, сортировка данных и поддержка принятия решений, подготовка задач, планирование, решение трансдисциплинарных задач, виртуализация кооперативного взаимодействия, аутсорсинг, применение современных методов обработки мультимедийных информационных ресурсов в виртуальном гиперпространстве, проведение видеоконференций и т. п.

Ожидаемые результаты

В результате реализации указанной концепции будет сформировано *концептуально единое интегрированное информационное пространство знаний*, представленных в канонической форме, и совокупность адекватных взаимно согласованных компьютеризированных информационных технологий эффективной поддержки всех составных частей профессиональной деятельности исследователя: от постановки и конструктивного решения произвольных научных проблем к непосредственному практическому внедрению результатов научной деятельности.

Создание *интегрированных информационных технологий*, которые обеспечат выполнение научно-исследовательских работ не только в конкретной предметной области, но и во время решения *сложных междисциплинарных проблем*, – реальный путь решения поставленной проблемы.

Новый арсенал пополнят высокоэффективные методы и средства создания нового поколения компьютеров, кооперативного взаимодействия исследователей, трансдисциплинарной интеграции знаний, интеллектуального поиска информации, конструктивизации и когнитивизации знаний и т. п.

Благодаря объединению наиболее важных креативных признаков (унифицированная, конструктивная форма) при условиях всесторонней их поддержки средствами информационных технологий (и законодательством) научные знания приобретут признаки конечного товара: готовность к *непосредственному применению*, *измеряемость* показателей объёма и качества, и *стоимости*.

В заключение, отмечая системообразующую роль информатики, перечислим основные компоненты оптимизации информационно-технологической поддержки междисциплинарных НИ.

1. Разработка системологии и математических основ МНИ.
2. Поддержка основных процессов МНИ: мониторинг и управление знаниями, регламентация процессов и документов, принятие решений (выявление актуальных направлений), управление научно-техническими программами.
3. Формализация знаний, конструктивизация научных теорий: построение онтологических систем знаний предметных областей (методика, глоссарии и индексированные компьютерно-онтологические толковые словари предметных знаний, энциклопедий и метаонтологий); поиск и формализация знаний из различных информационных источников, создание баз (сетей) знаний (включая коммерческие с решением проблем экономизации знаний и средств поддержки интеллектуальной собственности (в частности, персональных баз знаний)).
4. Разработка прикладных систем: системы управления стратегического уровня (управление государством, экономикой, наукой); виртуальные научно-инновационные и научно-учебные центры и лаборатории, электронные курсы и книги; системы медико-экологического мониторинга и пр.

8. ПРИКЛАДНЫЕ ЗАДАЧИ

8.1. Пример автоматизированного построения онтологической базы знаний и онтологии задач ПдО “Проектирование вычислительных устройств и систем на ПЛИС”

Одним из главных требований к онтологической системе (с извлечением знаний из ЕЯО) является наличие формализованной методики автоматизированного построения онтологии произвольной ПдО, действенной интегрированной информационной технологии и соответствующих инструментальных средств реализации последней. При этом предполагается, что в качестве источника онтологических знаний выступает лингвистический корпус текстов из заданной ПдО и знания специалиста-проектировщика.

Принципы и механизмы онтологического подхода могут быть применены при проектировании архитектурной и информационной компонент знание-ориентированной компьютерной системы [168]. Данный параграф посвящён разработке средств представления знаний в предметной области “Проектирование вычислительных устройств и систем на ПЛИС”, создания информационной технологии и поддерживающего её алгоритмическо-программного комплекса для автоматизации постановки и решения задач пользователей.

Рассмотрим *пример* проектирования *онтологии ПдО* и *онтологии задач* “Проектирование вычислительных устройств и систем на ПЛИС” на основе методологии проектирования (гл. 4). В качестве множества ЕЯО использовались монографии [146, 169–171], а для формирования множеств функций интерпретации и аксиом для концептов онтологии – толковые словари и энциклопедия [172–175]. Мы не приводим отдельно глоссарий определений понятий, вошедших в онтологию ПдО и онтологию задач, так как большая их часть приведена в приложении В. Такой глоссарий по содержанию должен быть эквивалентен тезаурусу ПдО (т. н. Когнитивному тезаурусу), предназначенному [130] для: классификации и унификации понятий ПдО; классификации методов и задач, решаемых в заданной ПдО; построения описаний методов и типовых наборов задач; автоматизации информационного поиска в базах знаний; классификации, индексирования и поиска библиографической и справочной информации по данной тематике; автоматизации обучения в заданной ПдО.

В соответствии с (4.3) проектирование состоит из двух этапов – проектирования онтологии ПдО (состоящей из онтологии объектов и онтологии процессов) и онтологии задач.

Проектирование онтологии ПдО **“Проектирование вычислительных устройств и систем на ПЛИС”**

Проектирование такой онтологии описано в [176]. Полное формализованное описание спроектированной онтологии заданной ПдО занимает более тысячи страниц, поэтому приведём только фрагмент начальной онтологии заданной предметной области ПдО (НО ПдО).

На рис. 8.1 представлен онтограф НО ПдО, который в соответствии с методикой спроектирован вручную инженером по знаниям и экспертом. Первые семь уровней НО ПдО отображают фрагмент онтологической структуры, включающей понятия верхнего уровня домена “Информатика” (приложение Б). Эти понятия непосредственно связаны отношениями с понятиями верхнего уровня заданной ПдО. Структура НО ПдО представляет онтологический каркас (некоторый ориентир), в который в процессе проектирования онтологии объектов и онтологии процессов происходит “встраивание” соответствующих цепочек понятий. При этом “встраиваются” и дополнительные связи между понятиями.

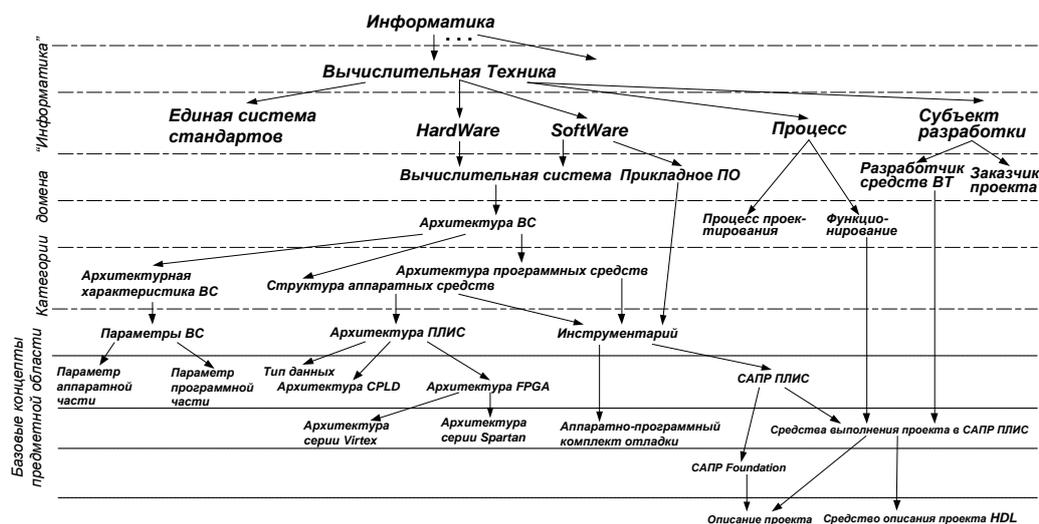


Рис. 8.1. ОНТОГРАФ начальной онтологии ПдО
«Проектирование вычислительных устройств и систем на ПЛИС»

Верхние уровни НО ПдО предназначены для системной интеграции онтологических знаний в домене “Информатика”, а на нижних уровнях представлены базовые концепты заданной ПдО.

Онтологическая структура ПдО считается построенной полностью, если в неё вошли все объекты и процессы, задействованные при решении типового набора (наборов) задач пользователя. Поэтому, такой компонент как *онтология задач* является неотъемлемой частью разрабатываемой *онтологической системы*.

На рис. 8.2 а) – 8.2 в) представлен онтограф онтологии объектов заданной ПдО. Связи между вершинами, расположенными на разных листах (рисунках) обозначены следующим образом:

- (1.2.3) – это трёхпозиционный код, который обозначает переход (выход) с первого листа (рис. 8.2 а), второго уровня, третий по порядку слева – направо;
- [1.2.3] – обозначает вход, соответственно с первого листа, второго уровня и третьего по порядку.

Цифры возле дуг онтографа онтологии ПдО соответствуют следующим семантическим отношениям: 0 – категорное отношение; 1 – целое-часть; 2 – род-вид; 3 – класс-подкласс (множество-подмножество); 4 – множество-элемент; 5 – атрибутивное отношение; 6 – отношение эквивалентности; 7 – отношение инициирования проекта; 8 – отношение протекания процесса; 9 – отношение завершения процесса.

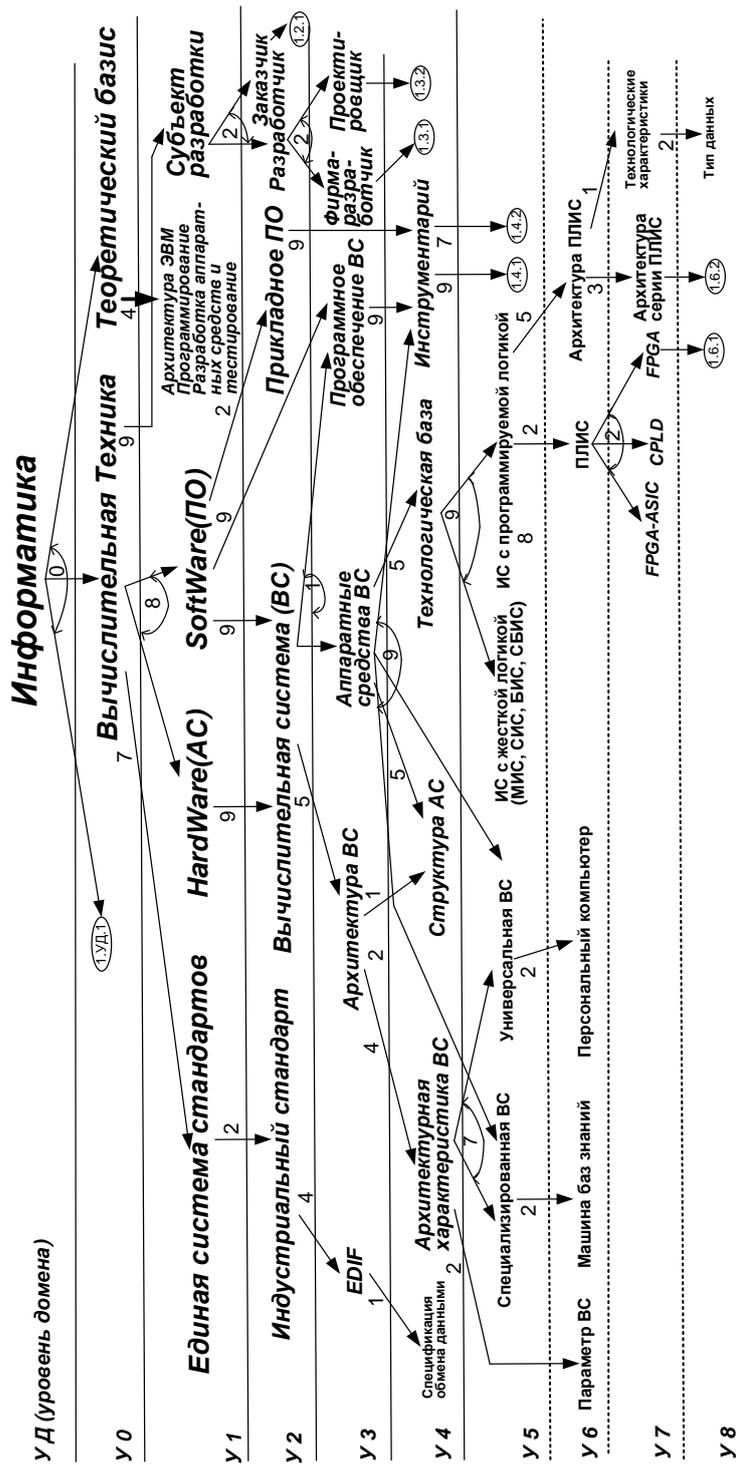


Рис. 8.2 а. Онтограф объектов ПдО “Проектирование вычислительных устройств и систем на ПЛИС”

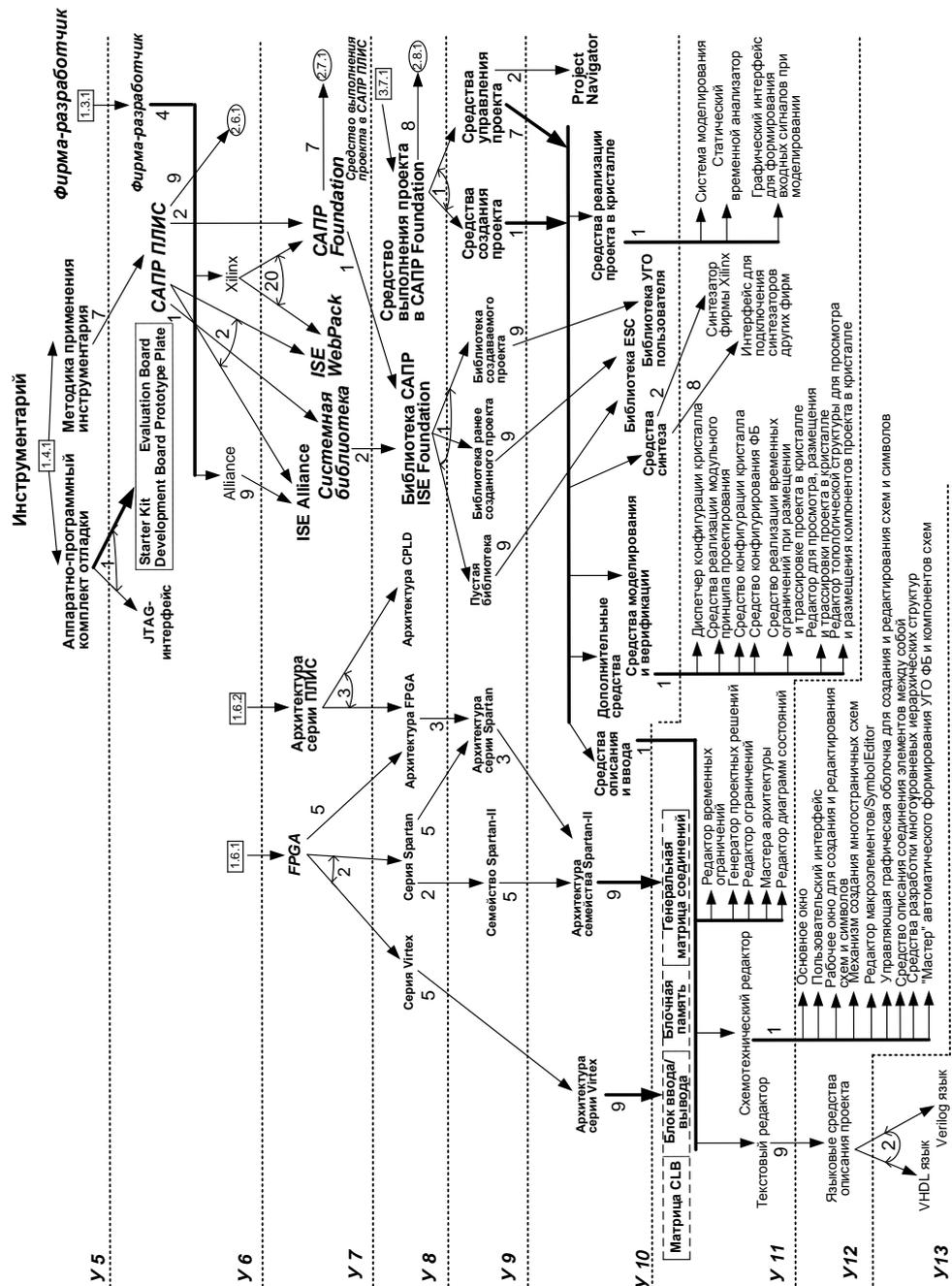


Рис. 8.2 б. Онтограф онтологии объектов ПдО "Проектирование вычислительных устройств и систем на ПЛИС" (продолжение)

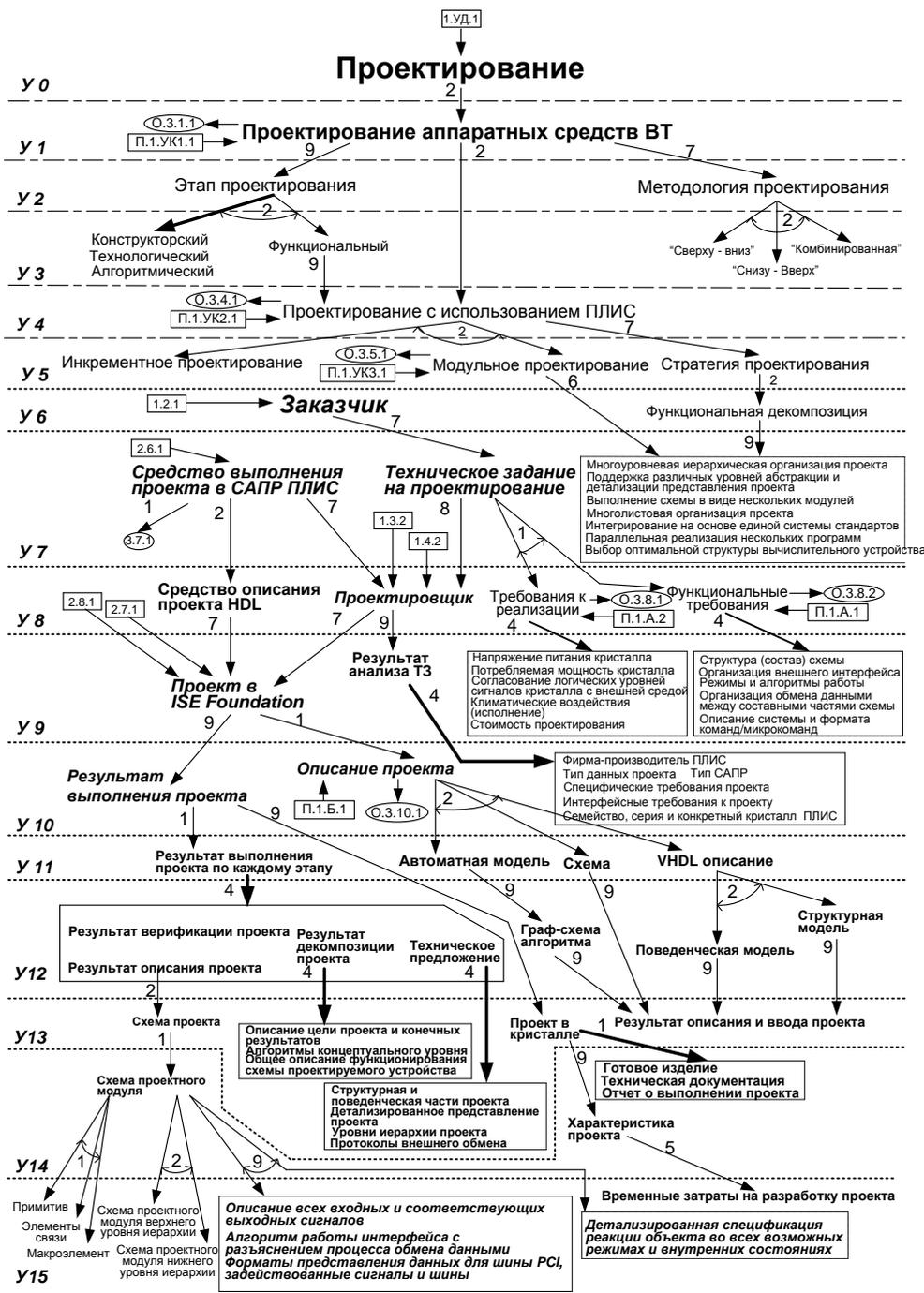


Рис. 8.2 в. Онтограф онтологии объектов ПдО «Проектирование вычислительных устройств и систем на ПЛИС» (продолжение)

На рис. 8.3 а) – 8.3 б) представлен онтограф онтологии процессов ПдО “Проектирование вычислительных устройств и систем на ПЛИС”. На нём переходы между листами обозначены аналогично принятым в онтологии объектов, за исключением того, что на второй позиции вместо уровня иерархии указан этап процесса

проектирования (А, Б, В, Г или Д). Этапы процесса проектирования очерчены пунктирными линиями.

Переходы между онтографами онтологии объектов (рис. 8.2) и онтологии процессов (рис. 8.3) обозначены следующим образом:

– **О.1.2.3** – это четырёхпозиционный код, который обозначает переход (выход) с онтографа онтологии объектов, с первого листа (рис. 8.2 а), второго уровня, третий по порядку слева – направо;

– **П.1.УК2.3**, **П.1.А.3** – это четырёхпозиционные коды, которые обозначают переходы (выходы) с онтографа онтологии процессов, с первого листа (рис. 8.3 а), соответственно второго уровня категорий или этапа проектирования А, третий по порядку сверху – вниз и слева – направо;

– **О.1.2.3**, **П.1.УК2.3**, **П.1.А.3** – обозначают входы, соответственно с онтографа объектов и онтографа процессов, первого листа, соответственно второго уровня, второго уровня категорий прикладной области или этапа проектирования А и третьего по порядку сверху – вниз и слева – направо.

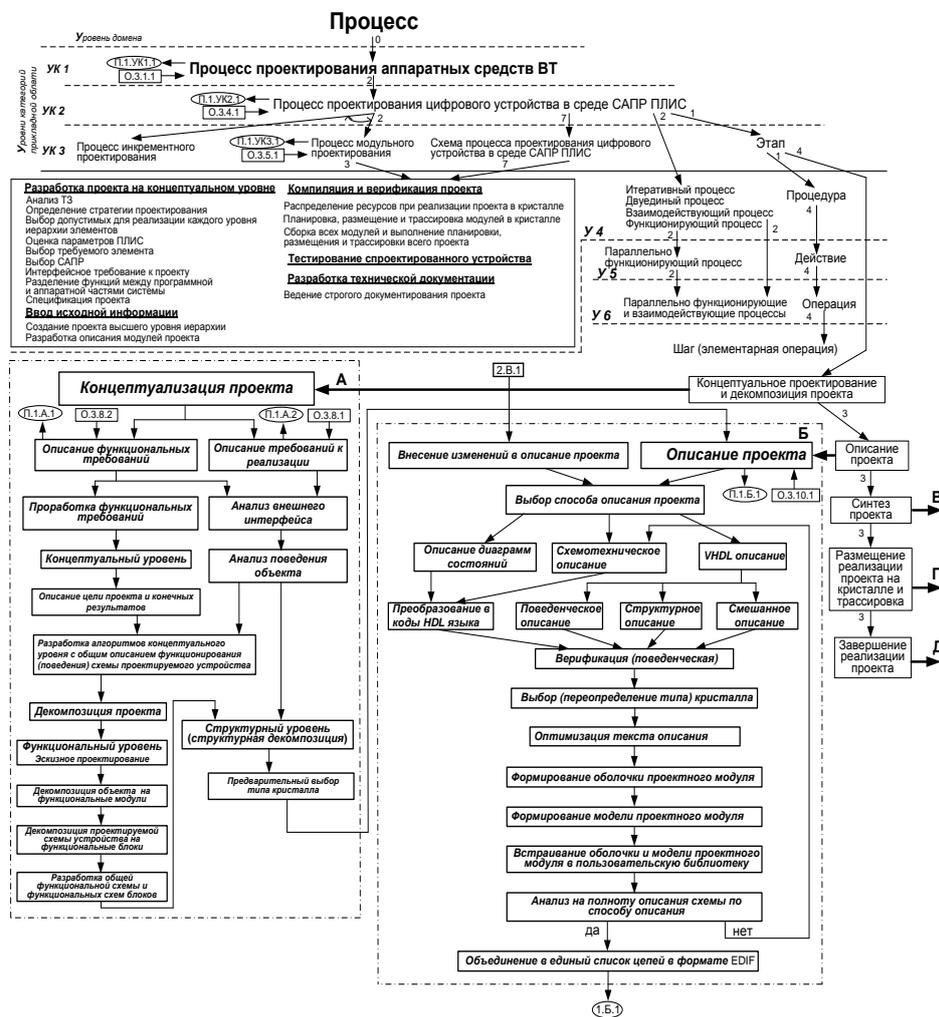


Рис. 8.3 а. Онтограф онтологии процессов ПдО «Проектирование вычислительных устройств и систем на ПЛИС»

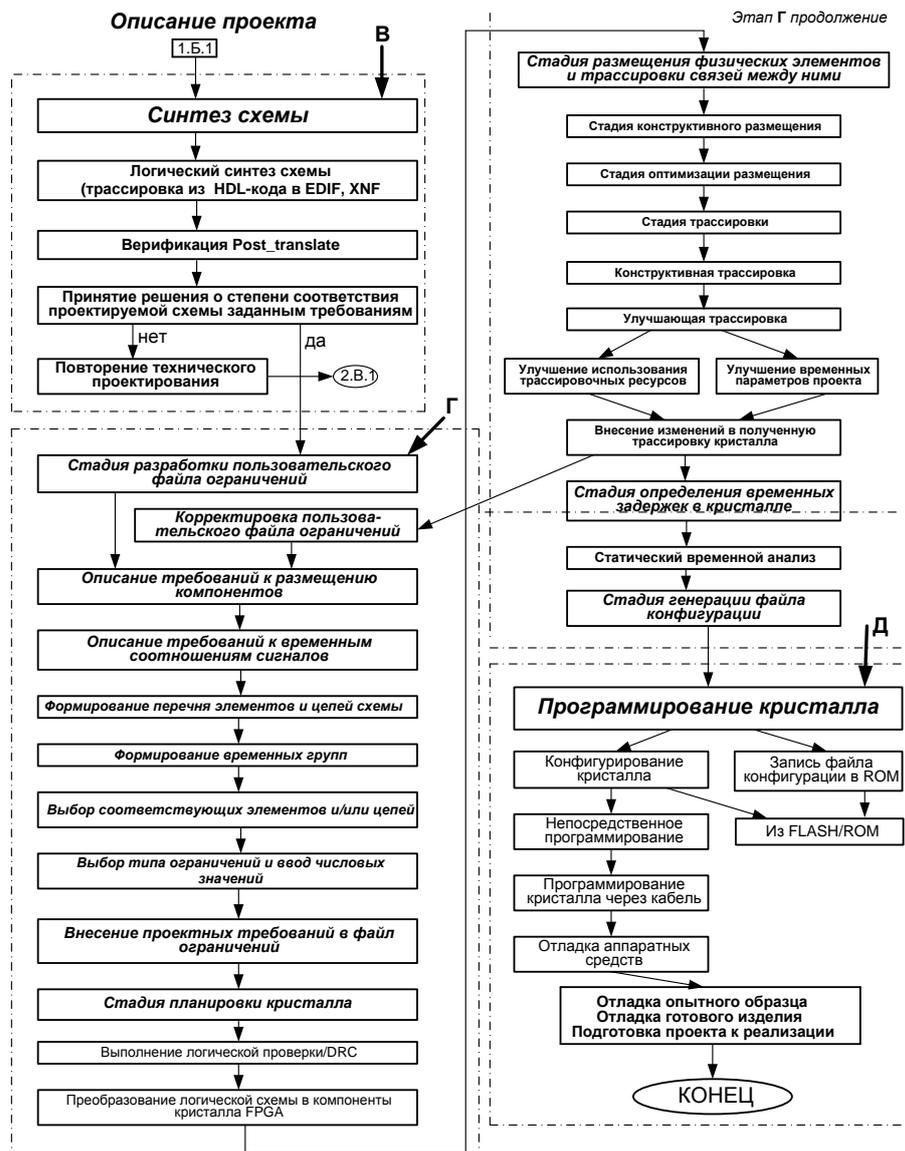


Рис. 8.3 б. Онтограф онтологии процессов ПДО «Проектирование вычислительных устройств и систем на ПЛИС» (продолжение)

На рис. 8.4 представлен фрагмент онтографа начальной онтологии заданной ПДО, спроектированный в инструментальной среде Protégé.

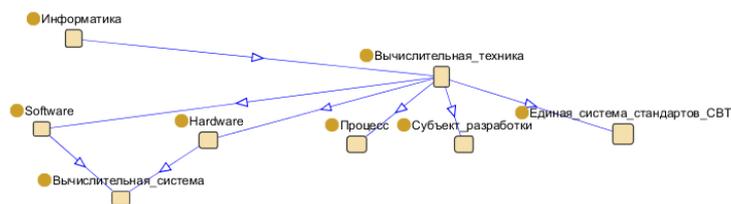


Рис. 8.4. Фрагмент онтографа начальной онтологии заданной ПДО, спроектированный в инструментальной среде Protégé

OWL-описание фрагмента онтографа начальной онтологии

```
<!-- Created with Protege (with OWL Plugin 3.4, Build 526)
http://protege.stanford.edu -->
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
  xmlns="http://www.aduis.com.ua/ПЛИС#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xml:base="http://www.aduis.com.ua/ПЛИС">
  <owl:Ontology rdf:about="">
  </owl:Ontology>
  <owl:Class rdf:ID="Информатика"/>
  <owl:Class rdf:ID="Вычислительная_техника">
    <rdfs:subClassOf rdf:resource="#Информатика"/>
  </owl:Class>
  <owl:Class rdf:ID="Software">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Вычислительная_техника"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Hardware">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Вычислительная_техника"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Процесс">
    <rdfs:subClassOf rdf:resource="#Вычислительная_техника"/>
  </owl:Class>
  <owl:Class rdf:ID="Субъект_разработки">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Вычислительная_техника"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Единая_система_стандартов_СВТ">
    <rdfs:subClassOf rdf:resource="#Вычислительная_техника"/>
  </owl:Class>
  <owl:Class rdf:ID="Вычислительная_система">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Вычислительная_система"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#Вычислительная_система">
    <rdfs:subClassOf rdf:resource="#Hardware"/>
    <rdfs:subClassOf rdf:resource="#Software"/>
  </owl:Class>
  <owl:ObjectProperty rdf:ID="Commondef"/>
  <owl:DatatypeProperty rdf:ID="axiom">
```

```
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
```

"Определения для понятий, приведенные ниже, взяты из энциклопедии и толковых словарей [172–175]"

```
<Информатика rdf:ID="_Информатика">
  <axiom rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >(наука) and (общие свойства) and (общие закономерности) and (информации)</axiom>
```

```
<hasCommon rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >наука об общих свойствах и закономерностях информации</hasCommon>
```

```
<Definition rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >большая область теоретических и прикладных знаний, связанных с получением, сохранением, преобразованием, пересылкой и использованием информации</Definition>
```

```
<Definition rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >научная дисциплина, изучающая структуру и общие свойства научной информации, а также закономерности всех процессов научной коммуникации</Definition>
```

```
<Definition rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >наука об общих свойствах и закономерностях информации, а также методах её поиска, передачи, хранения, обработки и использования в различных сферах деятельности человека</Definition>
```

```
<Definition rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >научное направление, которое занимается изучением законов, методов и способов накопления, переработки и передачи информации с помощью компьютера и других технических средств; группа дисциплин, которые занимаются разными аспектами использования и разработки ЭВМ: прикладная математика, программирование, программное обеспечение, искусственный интеллект, архитектура ЭВМ, вычислительные сети</Definition>
```

```
</Информатика>
```

```
<Вычислительная_техника rdf:ID="_Вычислительная_техника">
```

```
<axiom rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >(наука) and (вычислительные машины) and (построение) and (использование)</axiom>
```

```
<hasCommon xml:lang="ru">наука, изучающая вычислительные машины, принципы их построения и использования</hasCommon>
```

```
<Definition rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >наука о принципах построения, действия и проектирования этих средств.</Definition>
```

```
<Definition rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >область техники, объединяющая средства автоматизации математических вычислений и обработки информации в различных областях человеческой деятельности</Definition>
```

```
<Definition rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >дисциплина, изучающая вычислительные машины, принципы их построения и использования. Включает исследование таких аспектов, как: программирование, информационные структуры, разработка программного обеспечения, языки программирования, компиляторы и операционные системы, разработка аппаратных средств и тестирование, архитектура вычислительных систем, сети ЭВМ и сопря-
```

жение вычислительных машин, системный анализ и проектирование, теория информации, систем и вычислений, прикладная математика и электроника, вычислительные методы, методы машинной графики и приложения</Definition>

</Вычислительная_техника>

<Software rdf:ID="_Software">

<Definition rdf:datatype="http://www.w3.org/2001/XMLSchema#string"

>общий термин для обозначения "неосязаемых", в отличие от физических частей вычислительной системы</Definition>

<Definition rdf:datatype="http://www.w3.org/2001/XMLSchema#string"

>является одним из видов обеспечения вычислительной системы, наряду с техническим (аппаратным), математическим, информационным, лингвистическим, организационным и методическим обеспечением</Definition>

<axiom rdf:datatype="http://www.w3.org/2001/XMLSchema#string"

>("неосязаемые" части вычислительной системы) and ¬(физические части вычислительной системы)</axiom>

<hasCommon rdf:datatype="http://www.w3.org/2001/XMLSchema#string"

>общий термин для обозначения "неосязаемых", в отличие от физических частей вычислительной системы;</hasCommon>

</Software>

<Hardware rdf:ID="_Hardware">

<hasCommon rdf:datatype="http://www.w3.org/2001/XMLSchema#string"

>технические средства, используемые при выполнении программы</hasCommon>

<axiom rdf:datatype="http://www.w3.org/2001/XMLSchema#string"

>(технические средства) and (используемые) and (выполнение программы)</axiom>

<Definition rdf:datatype="http://www.w3.org/2001/XMLSchema#string"

>технические средства, в отличие от программ, процедур, правил и соответствующей документации</Definition>

<Definition rdf:datatype="http://www.w3.org/2001/XMLSchema#string"

>средства, используемые при выполнении программы</Definition>

</Hardware>

<Процесс rdf:ID="_Процесс">

<axiom rdf:datatype="http://www.w3.org/2001/XMLSchema#string"

>(последовательность действий) and (определяется объектом) or (определяется явлением) </axiom>

<hasCommon rdf:datatype="http://www.w3.org/2001/XMLSchema#string"

>последовательность действий, которая определяется объектом или явлением</hasCommon>

<Definition rdf:datatype="http://www.w3.org/2001/XMLSchema#string"

>некоторая последовательность действий, составляющих задачу</Definition>

<Definition rdf:datatype="http://www.w3.org/2001/XMLSchema#string"

>последовательность предсказуемых событий, которая определяется объектом или явлением и выполняется в заданных условиях; течение событий, которые происходят в соответствии поставленной цели или результата</Definition>

</Процесс>

<Разработчик rdf:ID="_Субъект_разработки">

<hasCommon rdf:datatype="http://www.w3.org/2001/XMLSchema#string"

>разработчик средств вычислительной техники или заказчик проекта разработки некоторого вычислительного устройства или системы</hasCommon>

```

<axiom rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>(разработчик средств ВТ) or (заказчик проекта) and ((разработка вычис-
лительного устройства) or (разработка вычислительной системы)</axiom>
<Definition rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>разработчик средств вычислительной техники или заказчик проекта разработки
некоторого вычислительного устройства или системы</Definition>
</Разработчик>
<Единая_система_стандартов_СВТ
rdf:ID="_Единая_система_стандартов_СВТ">
<hasCommon rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>система стандартов, которая распространяется на стационарные средства вы-
числительной техники (СВТ), и устанавливает требования к СВТ</hasCommon>
<Definition rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Единая система стандартов СВТ – система стандартов, которая распростра-
няется на стационарные средства вычислительной техники (СВТ), применяемые в
автоматизированных системах управления различного назначения всех уровней, в
системах обработки данных, сетях ЭВМ, на вычислительных центрах автономно, а
также встраиваемые в машины, оборудование и приборы, и предназначенные для
сбора, подготовки, ввода, накопления, обработки, вывода, отображения, приёма и
передачи информации, и устанавливает требования к СВТ, изготовляемым для
народного хозяйства и экспорта</Definition>
<axiom rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>(система стандартов) and ( распространяется на стационарные средства
вычислительной техники) and (устанавливает требования к СВТ)</axiom>
</Единая_система_стандартов_СВТ>
<Вычислительная_система rdf:ID="_Вычислительная_система">
<Definition rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>взаимосвязанная совокупность средств вычислительной техники, включающая
не менее двух основных процессоров либо вычислительных машин, из которых
роль основного процессора выполняет хотя бы одна</Definition>
<Definition rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>собственно вычислительная машина с подключёнными к ней внешними
устройствами </Definition>
<axiom rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>(взаимосвязанная совокупность средств вычислительной техники) and
(включающая не менее двух основных процессоров) or (вычислительных
машин) or (собственно вычислительная машина) and (с подключёнными
внешними устройствами).</axiom>
<hasCommon rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>взаимосвязанная совокупность средств вычислительной техники, включающая
не менее двух основных процессоров либо вычислительных машин; собственно вы-
числительная машина с подключёнными к ней внешними устройства</hasCom-
mon>
</Вычислительная_система>
<owl:AnnotationProperty rdf:ID="Definition"/>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 3.4, Build 526) http://protege.stan-
ford.edu -->

```

Формально-логическое описание концептов начальной онтологии

Информатика: (наука)&(общие свойства)&(общие закономерности)&(информации);

Вычислительная техника: (наука)&(вычислительные машины)&(построение)&(использование).

Hardware: (технические средства)&(используемые)&(выполнение программы).

Software: ("неосязаемые" части вычислительной системы)&-(физические части вычислительной системы).

Процесс: (последовательность действий)&(определяется объектом)or(определяется явлением).

Субъект разработки: (разработчик средств ВТ)or(заказчик проекта)&((разработка вычислительного устройства)or(разработка вычислительной системы)).

Единая система стандартов СВТ (система стандартов)&(распространяется на стационарные средства вычислительной техники)&(устанавливает требования к СВТ).

Вычислительная система: (взаимосвязанная совокупность средств вычислительной техники)&(включающая не менее двух основных процессоров)or(вычислительных машин) or(собственно вычислительная машина)&(с подключенными внешними устройствами).

Онтология задач ПДО

“Проектирование вычислительных устройств и систем на ПЛИС”

Онтология объектов некоторой ПДО с хорошо развитой теоретической базой обязательно включает такое понятие как “Теоретический базис”. Концептом нижнего уровня этого понятия является “Метод”. С помощью методов, входящих в этот концепт, выполняется постановка и решение задач в соответствующем ПрП. Концепт “Метод” является абстрактным понятием, которое на нижнем уровне классифицируется (имеется в виду онтологическая классификация) по признаку “Класс задач”. Алгоритмы, реализующие соответствующие методы для решения определённого класса задач, не рассматриваются ни в онтологии объектов, ни в онтологии процессов. Описания указанных алгоритмов (в принятом представлении для описаний онтологий объектов и процессов) хранятся в блоке *Формирователь методов решения задач* (рис. 4.8), входящем в онтологию задач.

Следует отметить, что базовыми онтологическими задачами являются: классификация понятий, построение таксономий с входящими в них отношений, интеграция онтологических структур, отображение и модификация онтологий, навигация (информационный поиск и запросы), наконец решение прикладных задач предметной области на уровне сервисной архитектуры и др. [5].

Для построения онтологии задач некоторого ПрП в общем виде обычно задаётся следующая информация.

1. Предметная область с областью интерпретации D .

2. Онтология ПДО.

2.1 Онтология объектов ПДО, формальная модель которой описывается тройкой

$$O^o = \langle X, R^o, F^o \rangle, \quad (8.1)$$

где $X = \{x_1, x_2, \dots, x_i, \dots, x_l\}$, $\forall i = (1 \div l)$,

$$R^o = \{r_1^o, r_2^o, \dots, r_j^o, \dots, r_J^o\}, \quad \forall j = (1 \div J),$$

$$F^o = X \times R^o = \{f_h^o\} = \{x_i\} \times \{r_j^o\}, \quad \forall h = (1 \div H).$$

2.2 Онтология процессов ПдО, формальная модель которой описывается тройкой

$$O^{\Pi} = \langle \Pi, R^{\Pi}, F^{\Pi} \rangle, \quad (8.2)$$

где $\Pi = \{nn_1, nn_2, \dots, nn_l, \dots, nn_L\}, \quad \forall l = (1 \div L),$

$$R^{\Pi} = \{r_1^n, r_2^n, \dots, r_k^n, \dots, r_J^n\}, \quad \forall k = (1 \div K),$$

$$F^{\Pi} = \Pi \times R^n = \{f_g^n\} = \{nn_l\} \times \{r_k^n\}, \quad \forall g = (1 \div G).$$

3. В ПрП задан типовой набор задач $OZ_n^{\text{ПрП}} \in OZ^{\text{ПрП}}$, где $OZ^{\text{ПрП}}$ – множество задач, решаемых в проблемном пространстве; $\forall n = (1 \div N)$ – множество типовых наборов задач.

4. В ПрП задано множество методов $\{M_{n_q}^{\text{ПрП}}\}$, с помощью которых решаются задачи $\{Z_{n_p}\}$, где q – множество методов для n -го типового набора задач; p – множество задач в n -ом типовом наборе.

4.1 Если $q < p$, то для некоторых (или всех) задач $\{Z_{n_p}\}$ существует не более чем один метод их решения.

4.2 Если $q = p$, то для каждой задачи Z_{n_p} существует только один метод её решения $M_q^{\text{ПрП}}$.

4.3 Если $q > p$, то для некоторых (или всех) задач $\{Z_{n_p}\}$ существует больше одного метода их решения.

Необходимо.

Для типового набора задач $\{Z_n\}$ необходимо сформировать подмножества задач $\{Z_{n_p}\}$, фрагментов задач W и операций V .

Конкретизировать *Решатель задач* (рис. 4.8) для заданного типового набора, для чего:

- разработать *Формирователь методов решения задач* – для множества методов $\{M_{n_q}^{\text{ПрП}}\}$ необходимо сформировать подмножества базовых наборов фрагментов методов и действий;
- сформировать последовательности процессов, задействованных в алгоритмах соответствующих методов решения задач, и объектов, участвующих в решении задач, с возможными значениями, ограничениями и условиями;
- указать в явном виде какие конкретные объекты из онтологии объектов участвуют в соответствующих процессах (алгоритмах) решения задач.

Разработать обобщённый алгоритм решения заданного типового набора задач;

Разработать алгоритмическо-программный комплекс, который, кроме реализации обобщённого алгоритма, связывал бы в единую интеллектуальную информационную технологию все компоненты *Онтологии задач* и обеспечивал интерфейс с пользователями.

Специфика заданной ПдО “Проектирование вычислительных устройств и систем на ПЛИС” такая, что в ней практически решается два набора типовых задач –

“1) спроектировать и 2) перепроектировать некоторое вычислительное устройство (ВУ) или систему на ПЛИС в соответствии с техническим заданием на проектирование”. Первый набор ориентирован на проектирование нового ВУ или системы, второй же набор – на перепроектирование уже существующего ВУ или системы в элементный базис ПЛИС. Кроме того, есть разные способы описания проекта (объекты “Автоматная модель”, “Схема” и “VHDL-описание” на рис. 8.2 в, уровень 11), для которых известны методики работы и подробные алгоритмы их реализации.

Постановка задачи для *второго набора задач* согласно описанной выше схеме может быть выполнена следующим образом.

Перепроектировать уже существующее ВУ, например интерпретатор информационных структур, представляющий собой классический микропрограммный процессор [177, 178]. Такое перепроектирование, выполненное ручным способом, описано там же.

Входными данными для перепроектирования являются:

- тип ВУ или системы и его (её) назначение;
- спецификация элементной базы, на которой реализовано ВУ или система;
- схема электрическая принципиальная ВУ или системы;
- интерфейс с “внешним миром”;
- технические требования и критерии, которым удовлетворяет перепроектируемое ВУ или система.

Кроме того, обычно ещё задаётся тип САПР, на которой будет выполняться перепроектирование, и диапазон доступных кристаллов ПЛИС.

Выходными данными решения задачи являются:

- готовый проект в кристалле, включающий готовое изделие (или файл конфигурации кристалла);
- характеристика проекта;
- техническая документация;
- отчёт о выполнении проекта.

Задано ТЗ на проектирование, в котором указаны критерии, которым должно удовлетворять решение задачи.

Методом решения для второго типового набора задач (как правило) является метод модульного проектирования со структурной декомпозицией ВУ или системы.

Следует отметить, что существенную помощь проектировщику в постановке задачи должен оказывать ИКОН (гл. 7), который в автоматическом режиме извлекает из ТЗ и технической документации по заранее составленным шаблонам требуемую информацию. В частности, для упомянутого выше интерпретатора информационных структур это могут быть также научно-технические статьи [177, 178].

Как следует из контекста *Онтологии задач*, упомянутый выше обобщённый алгоритм состоит из двух частей – *проблемной* (в т. ч. и предметной) и *прикладной*. На рис. 8.5 и рис. 8.6 представлены обобщённые алгоритмы проектирования проблемной и прикладной компонент *Онтологии задач* соответственно.

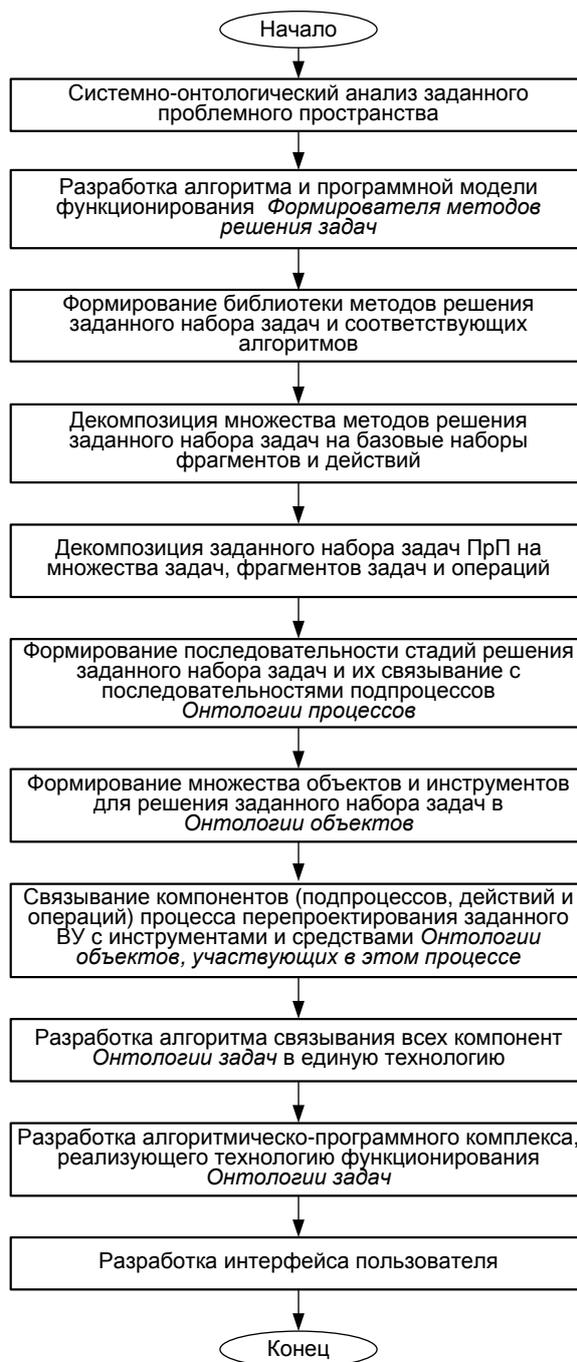


Рис. 8.5. Обобщённый алгоритм проектирования проблемного компонента *Онтологии задач*

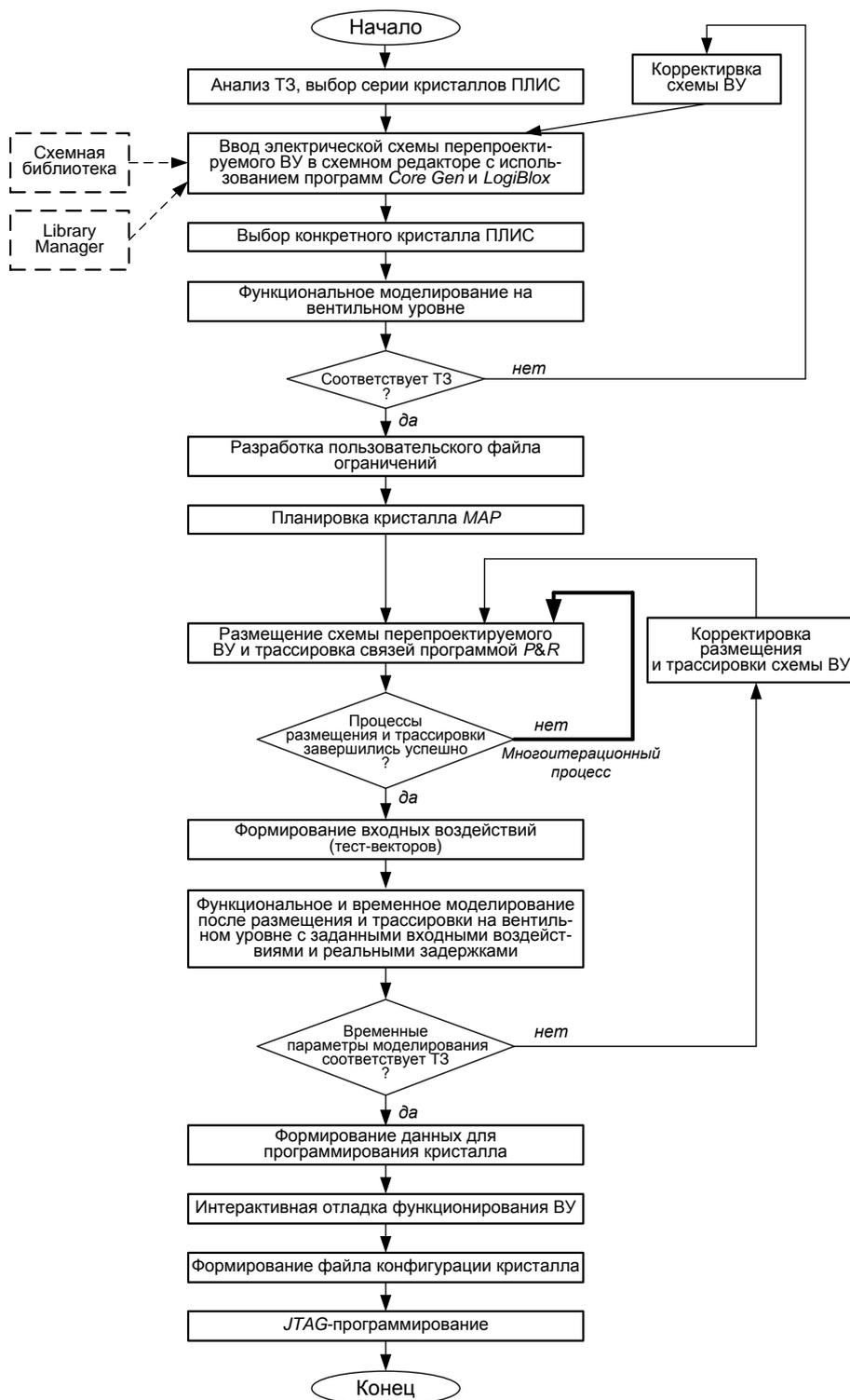


Рис. 8.6. Обобщённый алгоритм проектирования прикладного компонента *Онтологии задач*

8.2. Применение онтологического подхода к разработке электронных курсов обучения

8.2.1. Проектирование онтологии электронного курса предметной дисциплины “Базы данных”

На современном этапе развития передовых IT-технологий такие термины, как “электронное правительство”, “электронные деньги”, “электронное образование” (e-learning) и др. уже прочно вошли в обиход соответствующих сообществ. Сейчас происходит содержательное наполнение соответствующих парадигм “электронной деятельности”, в частности “электронного образования”, направленное на улучшения качества и доступности образования, расширения его форм и методов, интеграцию в общеевропейские образовательные программы и др.

Доступность к знаниям является важным показателем экономического, культурного и социального развития общества. Для всех тех, кто (в частности) ещё не охвачен официальной системой образования, существует возможность его получения по технологии e-learning [179].

В наши дни объём технической информации удваивается (и обновляется) каждые два года. Таким образом, половина из того, что студент изучил на первом курсе, может устареть к третьему. E-learning позволяет сократить время на замену устаревших учебных программ, или даже просто на их обновление. Кроме того, интенсифицируется процесс переквалификации специалистов, уже имеющих одну специальность. В далёком зарубежье их процент достигает до 80 процентов. Без электронного образования достичь таких показателей проблематично [180].

В Украине e-learning, наряду с традиционным обучением, по праву заняло свою нишу и имеет (по сравнению с традиционным обучением) определенные преимущества. Наиболее очевидным является временной фактор, в частности – гибкость, возможность обучения в удобное время с перерывами при необходимости, нет необходимости тратить время на дорогу. Есть также целый ряд менее очевидных преимуществ – возможность самоподготовки в любом месте и времени, освоение материала быстрее, чем с традиционными курсами, модернизация курса дешевле, чем переиздание учебников и переподготовка преподавателей, возможность вернуться к разделам обучения, которые не были усвоены с первого раза и др. [181].

В настоящее время e-learning развивается параллельно с World Wide Web, совершенствуя свои методики и формы обучения, ориентируясь на последние достижения Интернета [182]. Во многих странах, включая Украину, с повсеместным распространением Интернет вводятся новые формы обучения, базирующиеся на компьютерных и телекоммуникационных технологиях, в том числе такие формы e-learning, как дистанционное обучение (ДО) [183].

Объективность процесса информатизации общества и образования выводит дистанционное обучение в ряд равноправных форм получения образования. Соответственно ВУЗы расширяют формы и методы обучения. Традиционные педагогические теории, методы и модели сочетаются с передовыми информационными технологиями. В учебных планах многих ВУЗов в обязательном порядке предусматривается перевод определенной части учебных курсов на ДО.

На кафедрах, осуществляющих обучение по предметным дисциплинам (ПдД) внедряется технология ДО, которая потенциально допускает высокое качество подготовки специалиста за счёт усиления социально-значимых мотивов (самореа-

лизация, самоутверждение и коммуникативность, ускорение усвоения материала) и повышения эффективности использования необходимых ресурсов – телекоммуникационных и иных средств. Такие технологии могут быть реализованы в виде электронных курсов (ЭК). Создание каждого варианта ЭК предметной дисциплины требует больших ручных затрат на разработку их логико-организационной структуры, и в особенности наполнения. Для каждого варианта меняется логическая структура изложения учебного материала, объём учебной информации, уровень наглядности. С объёмом учебного материала связаны также временные затраты на подготовку варианта ПдД. Эффективность и самого варианта ПдД и затрат на его составление неизвестна до апробирования на определённом контингенте обучающихся.

Известные системы e-learning не ориентированы на автоматизированную подготовку ЭК. Например, одна из наиболее известных и распространённых систем электронного обучения модульная объектно-ориентированная динамическая учебная среда Moodle [184] ориентирована, прежде всего, на организацию взаимодействия между преподавателем и обучающимися. Она переведена на десятки языков, используется во многих учебных организациях различных стран мира, хорошо подходит как для организации дистанционных курсов, так и для поддержки очного обучения. Однако при внедрении технологий дистанционного образования необходимо разрабатывать несколько реализаций ЭК. Например, для специальностей “информатика” и “деловодство” для одной и той же предметной дисциплины “Базы данных”, необходимы реализации электронных курсов обучения, отличающиеся своей направленностью и объёмами. В Moodle всю эту работу приходится делать вручную.

Помимо большого объёма работ по созданию, электронный курс, построенный вручную, субъективен.

С другой стороны, уже более 10 лет успешно развивается направление онтологического инжиниринга, связанное с концептуальными структурами знаний. Ожидается, что онтологии будут влиять на следующее поколение систем e-learning [185].

Онтология [2, 49] обладает когнитивностью, структурирует знания некоторой предметной области в соответствии с онтологическими принципами.

Только информационная система с онтолого-управляемой архитектурой, имеющая в своём составе формальную онтологию некоторой ПдД, позволяет реализовать все преимущества электронного образования, в том числе и дистанционного обучения. Наиболее полно указанные свойства проявляются при использовании и взаимодействии онтологий двух уровней – онтологии домена предметных дисциплин (множество дисциплин, читаемых на кафедре) и онтологий самих предметных дисциплин. При этом естественным образом находят своё решение проблемы открытости и закрытости систем обучения, единообразия представления понятийных структур обучения, автоматизированного построения новых курсов обучения с учётом накопленных преподавателями кафедры знаний, замена кадрового состава преподавателей и др.

Постановка задачи

Выше уже отмечались о больших трудозатратах и субъективности ЭК, созданного вручную. Это объясняется слабой формализацией моделей, методов и технологий, которые применяются при разработке ЭК. Практически не используются модели для оптимизации и рационализации ЭК, также не используются моде-

ли для их индивидуализации. Уже созданные ЭК эксплуатируются годами, без учёта динамики модификации ПдД.

Автоматизация позволит модифицировать ЭКО с учётом развития ПдД, что особенно актуально в области информатики. Таким образом, можно поддерживать динамические электронные курсы обучения (ДЭК).

Онтологическое представление знаний ПдД позволит из имеющихся ЭК и учебников выделить проблемные участки, эффективно переработать их. Электронный курс, построенный на основе онтологического подхода автоматизированным способом по данной ПдД, интегрирует общие концепты и их связи и в этом смысле является общезначимым, общепринятым, построенным с существенным уменьшением субъективного фактора.

В автоматизированной технологии формализация описания техпроцесса обучения должна обеспечить математическое моделирование, решить оптимизационные задачи по трудозатратам и получить ЭК, удовлетворяющий заданным критериям, например, ЭК, настроенный на психофизиологические параметры обучающегося. При этом появляются возможности не просто разработать ЭК, а создать рациональный техпроцесс обучения по данной ПдД с учётом сочетания и последовательности подачи информационных, итоговых и тестовых блоков.

После автоматизированного построения онтологии ПдД на базе имеющихся ЭК и учебников по данной ПдД и выбора преподавателем понятий для ЭК в соответствии с онтологией заполняются (возможно, в автоматизированном режиме) необходимые БД и в автоматизированном режиме воспроизводится ЭК. При этом можно ожидать существенное снижение трудозатрат разработки и гарантии заданного уровня качества ЭК.

Решение задачи

Методология автоматизированного построения онтолого-управляемых ЭК включает в себя задачу разработки методики автоматизированного построения онтологий ЭК на базе имеющихся ЭК и учебников по данной ПдД [186].

Как известно [2], знание-ориентированные информационные системы с онтолого-управляемой архитектурой обеспечивают эффективное многократное использование онтологий ПдД для разных классов задач. Будучи один раз построенными онтологии пригодны для решения произвольного класса задач, в том числе задач обучения.

Композиция методов и механизмов *парадигмы предметных знаний* и онтологического управления приводит к интеграции и эффективному применению компонентов архитектуры онтолого-управляемой ИС в архитектурно-структурную организацию инструментального комплекса онтологического назначения (гл. 7), ориентированного на автоматизированное построение онтологических баз знаний предметных областей.

Концептуальная часть обучающей онтологической системы описывается известным кортежем (8.3) и представляет онтологию ПдД, состоящую из онтологии объектов, онтологии процессов (которые для домена “Образование” могут быть объединены в одну) и начальной онтологии, и метаонтологию домена предметных дисциплин [186]

$$O_{ОНС} = \langle MO^Д, O^{ПдД}(O^{О-П}, NO) \rangle, \quad (8.3)$$

где *NO* – начальная онтология заданной ПдД, которая рассматривается как “размытая”, когнитивно-графическая схема базовой структуры понятий ПдД. Напри-

мер, онтограф начальной онтологии ПдД “Базы данных” включает на уровнях категорий домена “Информатика” следующие понятия – “Информационные технологии”, “Формальная логика”, “Вычислительная система” и др. На уровнях “Базовые концепты предметной области” к НО относятся понятия – “Реляционная БД”, “Реляционная модель данных”, “Система управления БД”, “Исчисление предикатов”, “Основные функции СУБД”, “Язык SQL” и др. На уровнях “Методическое обеспечение” в НО присутствуют понятия – “Методы сериализации транзакций”, “Методы организации индексов”, “Методы журнализации” и др.;

– O^{O-P} – онтология множества понятий-объектов и понятий-процессов ПдД, которая рассматривается как иерархическая структура классов, подклассов и элементов классов. Например, фрагмент онтографа онтологии ПдД “Базы данных” (рис. 8.7) на уровнях (U_0, U_1, U_2, \dots) категорий домена “Информатика” уточняет понятия иерархических структур классов “Информационные технологии”, “Вычислительная техника” и др.;

– MO^D в (8.3) – метаонтология домена предметных дисциплин, рассматривается как иерархическая структура (в основном) дисциплин из учебных планов кафедры или факультета;

– онтология O^{O-P} описывается четвёркой (4.2).

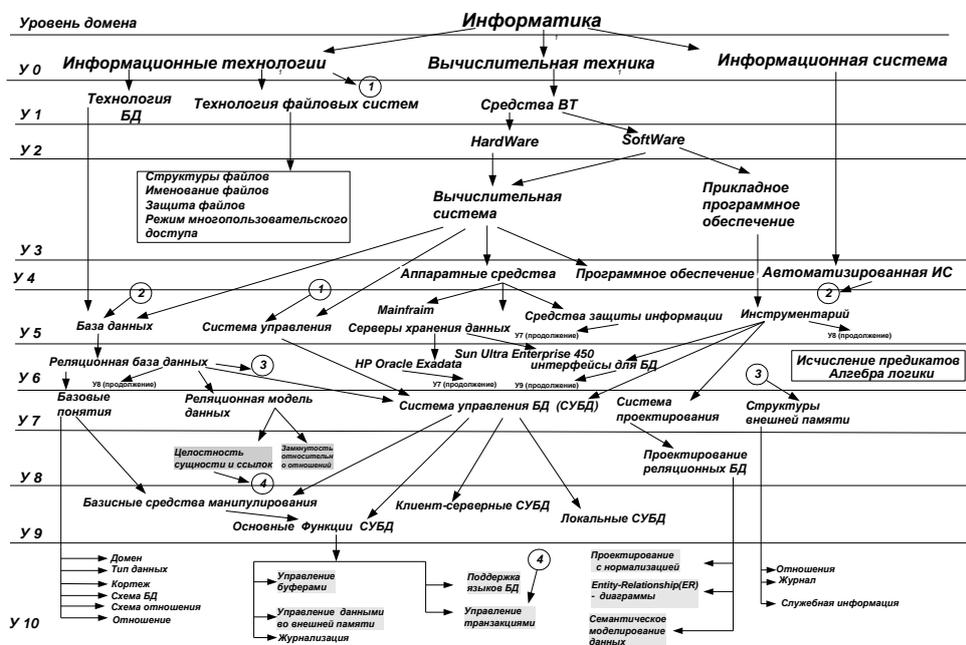


Рис. 8.7. Фрагмент онтографа онтологии ПдД “Базы данных”

Методология автоматизированного построения онтолого-управляемых ЭК предполагает разработку преподавателем в автоматизированном режиме полного онтографа ЭК на базе имеющихся ЭК и учебников по данной ПдД. Следует отметить, что онтограф является первым компонентом онтологии, а вторым является его формализованное описание на одном из общепринятых языков описания онтологий.

Методологически наиболее сложной и важной частью при разработке ЭК является проектирование онтологии ПдД. Учитывая выше изложенное, для автома-

тизированного построения ЭК для произвольных предметных дисциплин необходимо разработать формализованную методику проектирования онтологии ПдД, соответствующие алгоритм и инструментальные средства. Последние представляют собой инструментальный комплекс онтологического назначения.

Алгоритм построения онтологии ПдД

Обобщённый алгоритм автоматизированного построения онтологии ПдД на базе имеющихся ЭК и учебников по данной ПдД представлен на рис. 8.8. Входными данными являются уже созданные ЭК и учебники по данной ПдД.

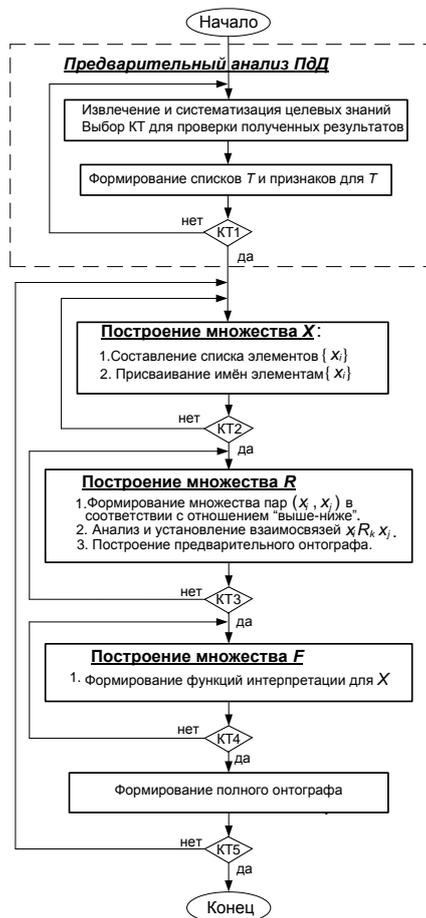


Рис. 8.8. Обобщённый алгоритм автоматизированного построения онтологии ПдД на базе уже созданных ЭК и учебников

В алгоритме последовательно строятся:

- $X = \{x_1, x_2, \dots, x_i, \dots, x_l\}$ – конечное множество понятий заданной ПдД;
- $R = \{r_1, r_2, \dots, r_k, \dots, r_K\}$, $R : x_1 \times x_2 \times \dots \times x_l$, $k = \overline{1, K}$, $K = Card R$ – конечное множество семантически значимых отношений между понятиями ПдД. Отношение r_k связывает понятия $x_i \in X$ и $x_j \in X$, где $x_i R_k x_j$, $i, j = \overline{1, n}$, $i \neq j$, $k = \overline{1, m}$. Другими словами, необходимо построить множество дуг, связывающих узлы *направленного онтографа*;

– $F : X \times R = \left\{ f_h \right\} : \left\{ x_i \right\} \times \left\{ r_k \right\}, h = \overline{1, H}, H = \text{Card } F$ – конечное множество

функций интерпретации, заданных на понятиях и/или отношениях.

На рисунке приняты следующие обозначения:

- T – термины предметной дисциплины;
- X, R, F – конечные множества понятий, отношений и функций интерпретации;
- КТ – контрольные точки;
- КТ1 – анализ на полноту сформированных списков терминов;
- КТ2 – выполняется анализ “Список X исчерпан?”;
- КТ3 – выполняется анализ “Отношения установлены для всех пар (x_i, x_j) ?”;
- КТ4 – выполняется анализ “Функции интерпретации определены для всех X?”;
- КТ5 – анализ на полноту и непротиворечивость онтологии ПдД методическим советом кафедры.

Заключение

В этом подразделе рассмотрена задача автоматизированного построения онтологии предметной дисциплины. Такая онтология является основной единицей в общей совокупности онтолого-управляемых электронных курсов кафедры. Следующим этапом развития рассмотренных методологических основ является разработка интегрированной информационной технологии жизненного цикла функционирования совокупности электронных курсов обучения на кафедре, включая разработку соответствующих методов, алгоритмов и программно-аппаратных комплексов.

8.2.2. Разработка ЭК “Общие основы построения знание-ориентированных компьютерных систем” в системе Moodle

В подразделе описаны методические указания по созданию ЭК для произвольной предметной дисциплины в системе Moodle в соответствии со следующим содержанием.

Введение

1. Подготовка учебного материала
 - 1.1 Подготовка теоретического материала
 - 1.2 Подготовка практического материала
 - 1.2.1 Списки терминов для «Глоссария»
 - 1.2.2 Материалы для «Базы данных»
 - 1.2.3 Задания для практической и самостоятельной работы
 - 1.2.4 Вопросы для тестов
2. Внесение учебных материалов в систему Moodle
 - 2.1 Создание и настройка курса
 - 2.2 Создание ресурсов курса
 - 2.2.1 Автоматическое связывание ресурсов
 - 2.2.2 Создание внутренних ссылок в элементах курса
 - 2.2.3 Формулы в текстовых документах
 - 2.3 Создание элементов (активностей) курса
 - 2.3.1 Наполнение «Глоссария»
 - 2.3.2 Построение SCORM-иерархии электронного курса

Введение

В данных методических указаниях кратко описан один из подходов к созданию электронного онтолого-управляемого курса в системе дистанционного обучения Moodle. Созданный курс можно легко переносить на другие машины и сайты с системой Moodle.

Информацию по вопросам инсталляции системы Moodle и работы с ней можно найти в документации и на форумах сайта [184] (доступна также русскоязычная поддержка).

Подготовка учебного материала

Для организации эффективной работы с курсом заранее следует продумать его структуру и разбить весь учебный материал на отдельные элементы, по разделам, подразделам и типом представления знаний (создать онтологию курса).

Онтология электронного курса содержит две составляющие: первая представляет собой структурированное семантическое пространство всех форм знаний электронного курса, а вторая – онтологическую иерархию концептов – терминов, связанных одним концептуальным отношением «выше-ниже». Системная интеграция указанных составляющих должна обеспечить построение полной онтологии электронного курса, в которой реализуется онтолого-информационная связь, как внутри самого курса, так и с другими, близкими по тематике курсами.

Рассмотрим детальнее указанные онтологические составляющие.

Первая составляющая, по сути, представляет собой содержание всех форм знаний (аналитической, табличной, графической и текстовой), которые содержатся в ЭК. Первые три формы представлены в виде отдельных полных списков аналитических выражений, таблиц и рисунков, имена которых соответствуют именам-обозначениям в тексте ЭК.

Текстовая форма представления знаний формируется следующим способом. Вершина структуры соответствует имени ЭК. Ниже дерево разветвляется соответственно делению текста на разделы, подразделы и т.д. При этом возможны три варианта:

- один раздел соответствует одной лекции (а имя раздела соответствует имени одной лекции);
- один раздел соответствует нескольким лекциям, тогда имена подразделов будут соответствовать именам лекций;
- несколько подразделов соответствуют одной лекции, тогда следует сформировать новый узел, который будет соответствовать имени лекции.

Формирование структуры первой составляющей онтологии в основном учитывает требования SCORM-стандарта к наполнению электронных курсов. Оттого она названа SCORM-иерархией. Пример SCORM-иерархии электронного курса «Общие основы построения знание-ориентированных компьютерных систем» приведен на рис.8.9.

Вторая составляющая онтологии ЭК представляет собой простую онтологию концептов (составленных из общих терминов), которые связаны между собой обобщённым концептуальным отношением «выше – на_одном_уровне – ниже». По сути, это есть начальная онтология концептуальных знаний электронного курса. Для среднего по объёму электронного курса количество таких концептов равно приблизительно 20-ти.

Далее вокруг каждого такого концепта формируется множество концептов-терминов, элементы которого могут принадлежать как данному ЭК, так и другим

курсам. При этом необходимо обеспечить семантическую связь как с SCORM-иерархией ЭК (данного и других, к которым входит выбранный термин-концепт), так и с глобальным глоссарием (в котором наиболее полно приведено определение термина). Таким образом, вторую составляющую О ЭК назовём начальной онтологией О ЭК электронного курса. На рис. 8.10 показан пример начальной онтологии для ЭК «Общие основы построения знание-ориентированных компьютерных систем».

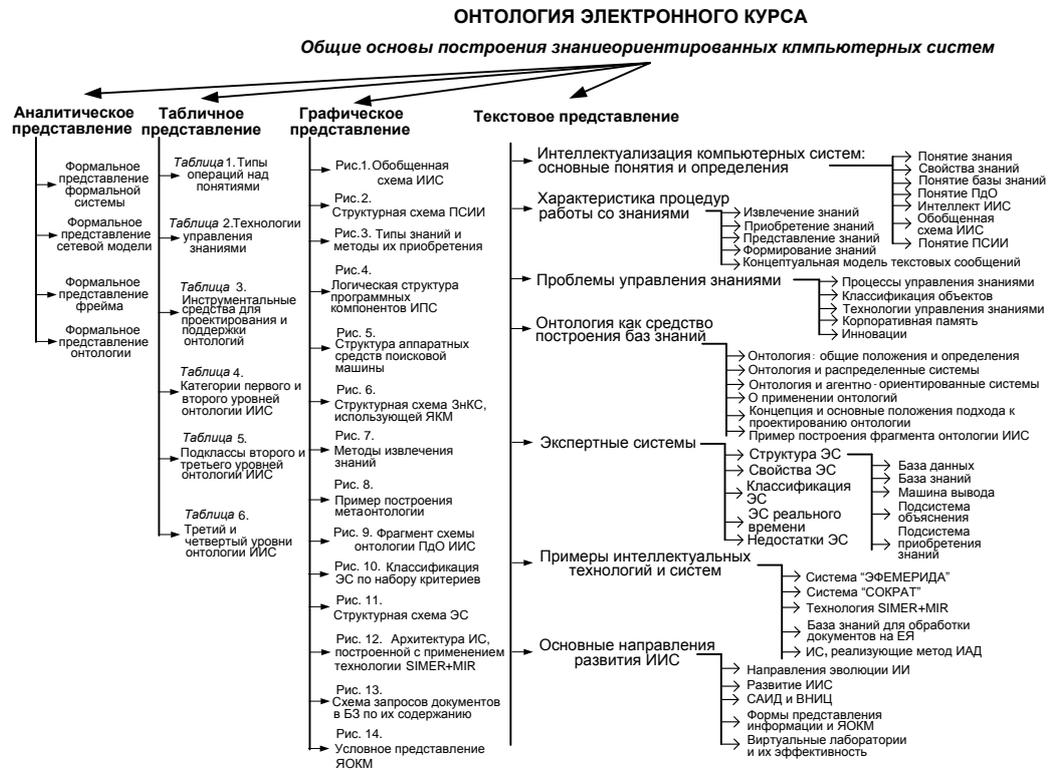


Рис. 8.9. Пример SCORM-иерархии электронного курса

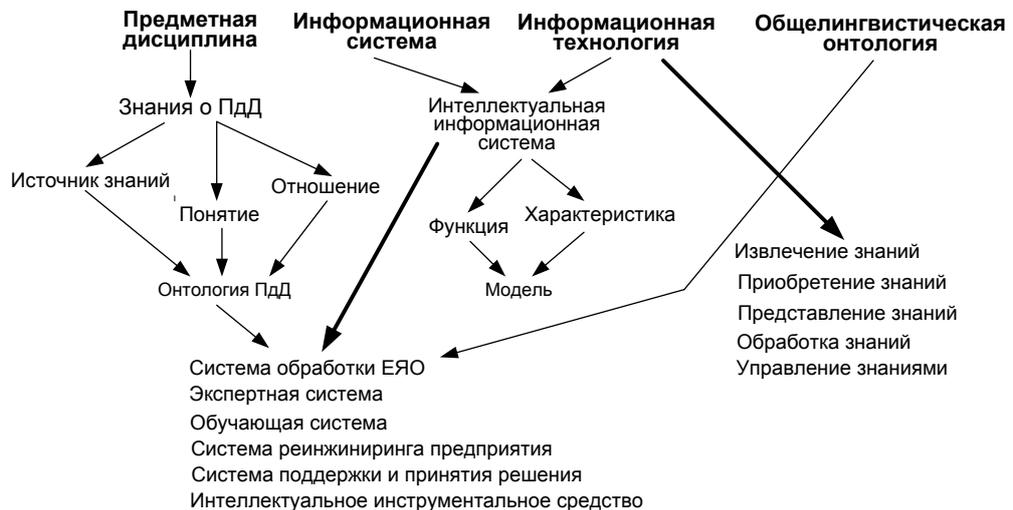


Рис. 8.10. Пример начальной онтологии электронного курса

Отдельно следует выделить разработку мультимедийной формы представления знаний электронного курса, что полностью зависит от профессионального и педагогического уровня преподавателя.

Подготовка теоретического материала

Для создания курса в Moodle доступны два инструмента: ресурсы курса и элементы (активности) курса. Ресурсы курса представляют теоретический материал, а элементы курса объединяют средства для интерактивного общения и контроля знаний (практический материал).

Учебный материал может содержаться в файлах разных форматов. Moodle позволяет также создавать внутренние ресурсы в виде текстовых и веб-страниц (см. п. 12.2.1 файла «MOODLE. Виртуальная обучающая среда»). Такие ресурсы имеют больше возможностей для внутреннего связывания элементов курса между собой.

В системе Moodle могут создаваться автоматические связи между ресурсами. Например, в тексте *Пояснения данного вопроса содержит Текстовый ресурс 1.1.2.* под фрагментом *Текстовый ресурс 1.1.2* будет создана автоматическая ссылка на упомянутый ресурс. Следует отметить, что автоматические ссылки создаются только в текстах внутренних ресурсов Moodle. Если, например, данный текст встретится во внешнем doc- или rtf-файле, автоматического связывания не будет.

Можно также вручную создать связи между отдельными частями материала (создание таких связей описано в п. 2.2.2 данных методических указаний).

Подготовка практического материала

Организовать практическую работу студентов, интерактивный процесс обучения и систему контроля знаний позволяют такие элементы Moodle как «Глоссарий», «Задания», «База данных», «Форум», «Чат», «Тест», «Урок» и другие (детально описано в пунктах 13-15 файла «MOODLE. Виртуальная обучающая среда»).

Списки терминов для «Глоссария»

Элемент «Глоссарий» позволяет создать автоматическое связывание терминов в тексте с соответствующими статьями этого глоссария (на уровне курса или на уровне всей системы, т. е. и других курсов). Например, если глоссарий содержит статью «*Граф – конечное множество вершин, соединённых рёбрами*», то далее в произвольном тексте внутренних ресурсов курса слово *Граф* (или его словоизменение) автоматически будет связано ссылкой на эту статью. (В отдельных фрагментах текста можно запретить такое связывание).

Для реализации этих возможностей необходимо создать списки терминов, которые будут вноситься у «Глоссарий» (начальная онтология курса).

Для каждого раздела можно создать отдельный «Глоссарий». Выделенные термины, которые принадлежат не только данному курсу, а и другим курсам системы, вносятся у «Глоссарий» глобального типа (статьи которого доступны из других курсов сайта).

Списки терминов удобно оформить в виде таблицы, как показано ниже.

Таблица 8.1.

Термины глоссария

Термин	Определение	Ключевые слова
<i>Граф</i>	<i>конечное множество вершин, соединённых рёбрами</i>	<i>Графу Графом Графы</i>

В поле «**Ключевые слова**» можно вносить словоизменения термина, синонимы и произвольные слова, которые необходимо связать с данной статьёй «Глоссария».

Материалы для «Базы данных»

Некоторую информацию удобно представить в виде элемента «База данных» (например, список использованной литературы). Здесь также можно организовать автоматические ссылки, аналогично изложенному выше для элемента «Глоссарий».

Для интерактивной работы можно разрешить студентам дополнять, редактировать и комментировать статьи «Глоссария» и «Базы данных».

Задания для практической и самостоятельной работы

Практическую и самостоятельную работу организуют с помощью элемента «Задания». Для отдельных разделов курса необходимо подготовить тексты заданий, которые могут содержать графические изображения, медиафайлы, ссылки на другие ресурсы и файлы, предварительно загруженные на сайт. Ответ студента на такое «Задание» может быть в виде простого текста или загруженного файла произвольного формата. Можно организовать защиту «Задания»: явно или с помощью элементов «Форум» и «Чат» (в режиме он-лайн). Детальнее – см. п. 13.2.2.1 файла «*MOODLE. Виртуальная обучающая среда*» [184].

Можно также подготовить задания для работы в режиме семинара и в режиме урока. Элемент «Семинар» позволяет студентам отвечать на поставленное задание, а также оценивать и комментировать ответы один другого. Детальнее – п. 15.2.2.12 файла «*MOODLE. Виртуальная обучающая среда*».

Элемент «Урок» даёт студенту по частям некоторый текстовый материал с вопросами по тексту. Причём, последовательность подачи этих частей зависит от ответов студента на вопросы (см. п. 13.2.2.7 файла «*MOODLE. Виртуальная обучающая среда*») [184].

Вопросы для тестов

Для контроля знаний студентов в системе Moodle предусмотрены «Тесты». Оценка результатов тестирования производится системой автоматически, по заданным преподавателем критериям. Для создания таких тестов необходимо подготовить базу вопросов. Причём, вопросы необходимо разбить на группы в зависимости от раздела, тематики, степени трудности и др.

Например, создано такие группы вопросов:

- Лекция 1
 - Вопросы к разделу 1.1
 - ◆ Трудные вопросы
 - ◆ Вопросы средней трудности
 - ◆ Лёгкие вопросы
 - Вопросы к разделу 1.2
 - ◆
- Лекция 2
-

Тогда при создании некоторого Теста1 можно выбирать соответственно к тематике теста конкретные вопросы отдельной группы, а можно задать системе такие условия:

- добавить к тесту произвольные 2 вопроса средней трудности из раздела 1.1;
- добавить к тесту произвольный лёгкий вопрос из раздела 1.2 и т. д.

Качество внесенных вопросов потом можно проверить с помощью статистических характеристик тестов, которые система Moodle рассчитывает автоматически (индекс лёгкости вопроса, процент правильных ответов на вопросы, среднеквадратическое отклонение, коэффициент дискриминативности и др.). Для конечного варианта теста оставляют вопросы с наилучшими характеристиками.

Вопросы можно создавать с помощью готового шаблона (файл Moodle-question.dot). Детальнее про типы вопросов в системе Moodle и примеры см. п. 14.2.2.0 файла «MOODLE. Виртуальная обучающая среда» [184].

Внесение учебных материалов в систему Moodle

Создание и настройка курса

Если курс создаётся в рамках уже существующей учебной среды Moodle, необходимо, чтобы администратор сайта предоставил права для создания и редактирования курса.

Если доступа к такой среде нет, необходимо установить систему Moodle на свой компьютер (последние версии инсталляционных файлов свободно доступны на сайте [184], или можно воспользоваться файлом Moodle.zip).

После инсталляции на титульной странице необходимо выбрать пункт меню *Управление* ► *Курсы* ► *Добавить/редактировать курсы*.

Далее для существующей или новой категории курсов нажать «Добавить курс». Откроется страница с настройками курса (Рис. 2.1.1). Символ  даёт пояснения для каждого из пунктов настроек. (Для того, чтобы интерфейс системы был русскоязычным, необходимо файл ru_utf8.zip разархивировать в каталог moodle\lang.

Заполните необходимые поля и нажмите на кнопку «Сохранить» в конце страницы. Возвратиться к этой странице и изменить настройки можно будет на любом этапе работы с курсом.

Далее появится окно с назначением ролей для курса и их описания (рис. 2.1.2). Если курс будет использоваться на других сайтах, эту страницу можно пропустить.

Назначить роли можно и позже на любом этапе работы с курсом. Если на сайте уже есть зарегистрированные пользователи, которые будут принимать участие в создании элементов курса, следует назначить им роль *Administrator*, *Course creator* или *Teacher*. (Нажать на соответствующее имя роли и выбрать необходимого пользователя из списка зарегистрированных пользователей).

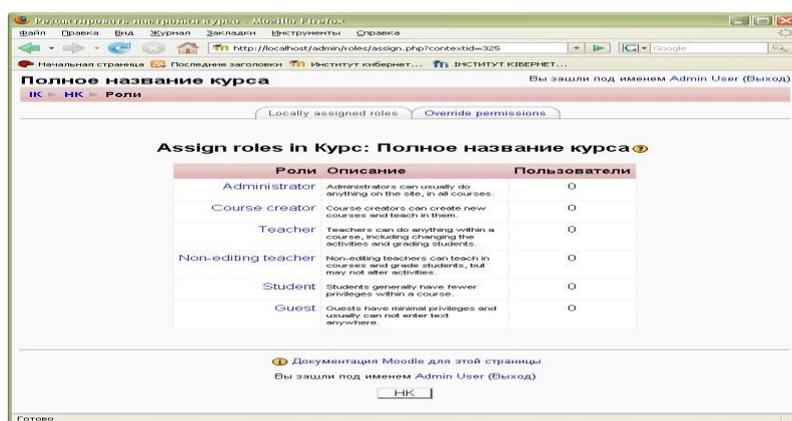


Рис. 2.1.1. Создание нового курса

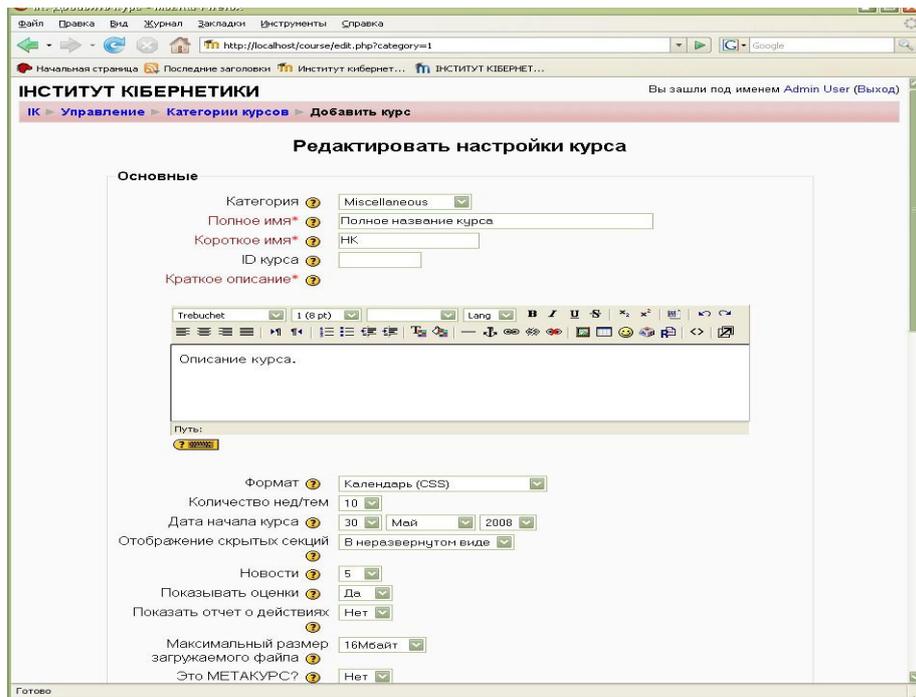


Рис. 2.1.2. Назначение ролей

Далее следует внизу окна нажать на кнопку с кратким названием курса («НК») или в верхнем меню на это же самое имя. Появится титульная страница курса. Если в настройках при создании курса был выбран формат «Структура» (рис. 2.1.1), то вид страницы будет таким как на рис. 2.1.3. Детальнее о назначении блоков и пунктов меню титульной страницы курса см. п. 6.1.1-11 файла «MOODLE. Виртуальная обучающая среда».

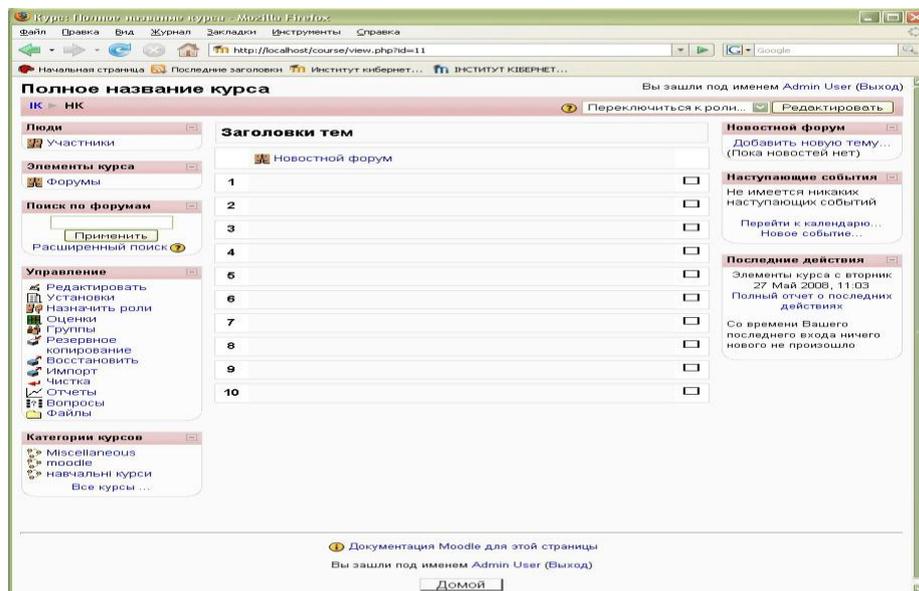


Рис. 2.1.3. Титульная страница курса

Возвратиться к настройкам курса и изменить его формат можно с помощью пункта «Установки» меню «Управление» (слева на рис. 2.1.3).

Создание ресурсов курса

Для этого предварительно необходимо перейти в режим редактирования (справа верхняя кнопка «Редактировать» или одноимённый пункт меню «Управление» на рис. 2.1.3). В п. 12 файла «MOODLE. Виртуальная обучающая среда» [184] детально описаны все ресурсы и процесс их создания. Рассмотрим отдельные вопросы.

Автоматическое связывание ресурсов

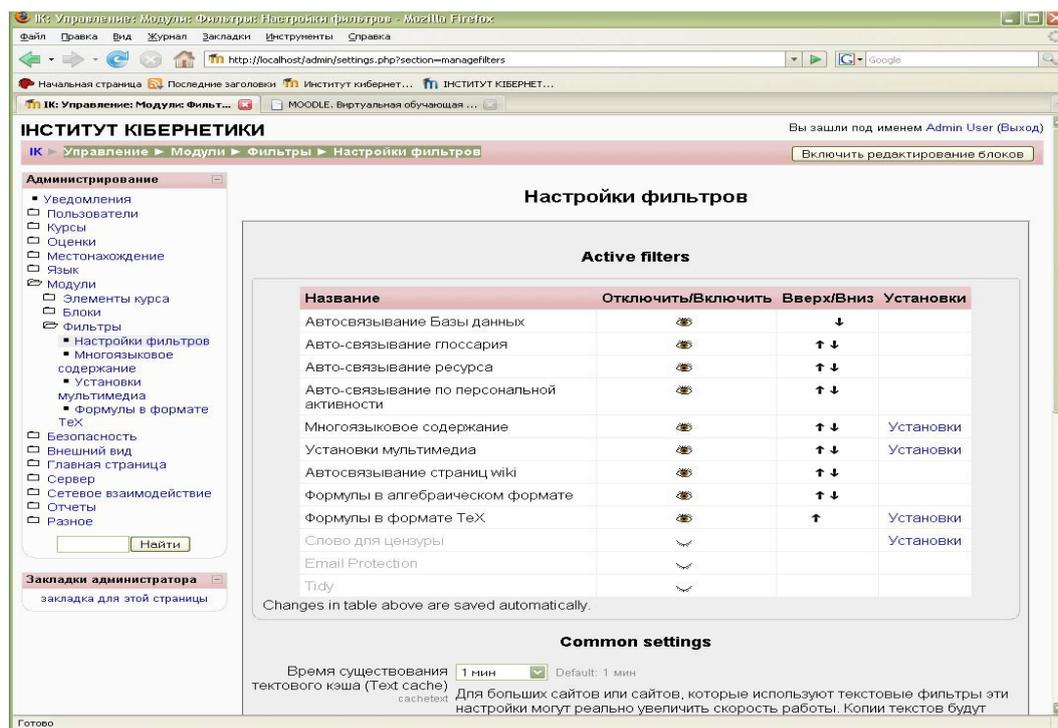
Для автоматического связывания ресурсов курса по их названию (так, как было описано в примере пункта 1.1 настоящих методических указаний), необходимо сначала разрешить такое связывание в настройках на главной странице сайта (доступно только администратору сайта): *Управление* ► *Модули* ► *Фильтры* ► *Настройки фильтров*.

На рис. 2.2.1: знак  обозначает, что функция разрешена.

Создание внутренних ссылок в элементах курса

Гиперссылки на ресурсы курса можно создавать с помощью встроенного визуального HTML редактора. Если элемент, на который выполняется ссылка, находится на текущей странице, необходимо сначала создать закладку с помощью

кнопки  визуального редактора. В окне, которое появится (рис. 2.2.2), ввести имя закладки.



The screenshot shows the Moodle administration interface for filter settings. The page title is "Настройки фильтров" (Filter Settings). It features a table of active filters with columns for Name, On/Off status, Up/Down arrows, and Settings. Below the table are common settings, including a text cache timeout set to 1 minute.

Название	Отключить/Включить	Вверх/Вниз	Установки
Автосвязывание Базы данных		↓	
Авто-связывание глоссария		↑ ↓	
Авто-связывание ресурса		↑ ↓	
Авто-связывание по персональной активности		↑ ↓	
Многоязыковое содержание		↑ ↓	Установки
Установки мультимедиа		↑ ↓	Установки
Автосвязывание страниц wiki		↑ ↓	
Формулы в алгебраическом формате		↑ ↓	
Формулы в формате TeX		↑	Установки
Слово для цензуры			Установки
Email Protection			
Tidy			

Common settings: Время существования текстового кэша (Text cache) Default: 1 мин

Для больших сайтов или сайтов, которые используют текстовые фильтры эти настройки могут реально увеличить скорость работы. Копии текстов будут

Рис. 2.2.1. Настройка для фильтров



Рис. 2.2.2. Окно добавления имени закладки

Затем необходимо выделить элемент текста, который будет ссылаться на закладку, и нажать кнопку . В окне, которое появится (рис. 2.2.3), выбрать имя закладки

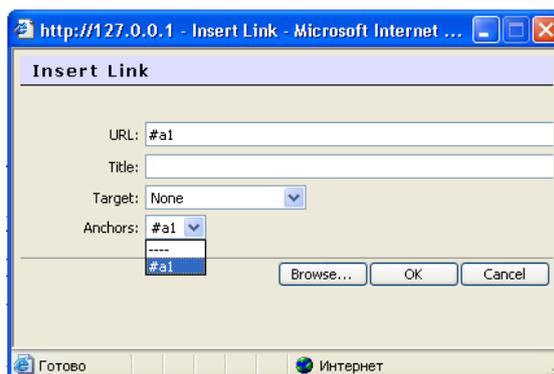


Рис. 2.2.3. Окно добавления гиперссылки в закладку

Гиперссылки на ресурсы курса, которые находятся на разных страницах, можно, например, создавать следующим образом:

- 1) создать закладку (рис. 2.2.2);
- 2) определить адрес страницы с закладкой (обычно адрес отображается в соответствующем поле браузера

Адрес:  <http://127.0.0.1/mod/resource/view.php?id=17>);

- 3) выделить элемент текста, который будет ссылаться на закладку, и нажать кнопку . В окне, которое появится (рис. 2.2.4), в поле URL ввести адрес страницы и имя закладки, например `href="http://127.0.0.1/mod/resource/view.php?id=17#a1"`.

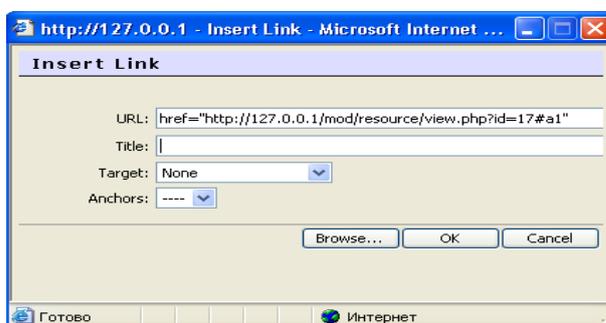


Рис. 2.2.4. Окно добавления гиперссылки на закладку на другой странице

Формулы в текстовых документах

Для отображения формул в системе Moodle, они должны быть в формате TeX, алгебраическом формате или в формате графического изображения (тогда каждая формула вставляется в текст как рисунок). Использование математических формул алгебраического формата и формата TeX описано в п. 18 файла «MOODLE. Виртуальная обучающая среда». Для отображения таких формул, возможно, будет необходимо установить дополнительные программы (см. в меню главной страницы сайта *Управление* ► *Модули* ► *Фильтры* ► *Формулы в формате TeX*)

Создание элементов (активностей) курса

Создание элементов курса детально описано в пунктах 13-15 файла «MOODLE. Виртуальная обучающая среда» [184].

Наполнение «Глоссария»

Если списки терминов были подготовлены в формате Таблицы 1.2.1, тогда текст полей таблицы заносится в соответствующие поля на рис. 2.3.1:

- «Слово» соответствует полю таблицы «Термин»;
- «Определение» – «Определение»;
- «Ключевое(ые) слово(а)» – «Ключевые слова».

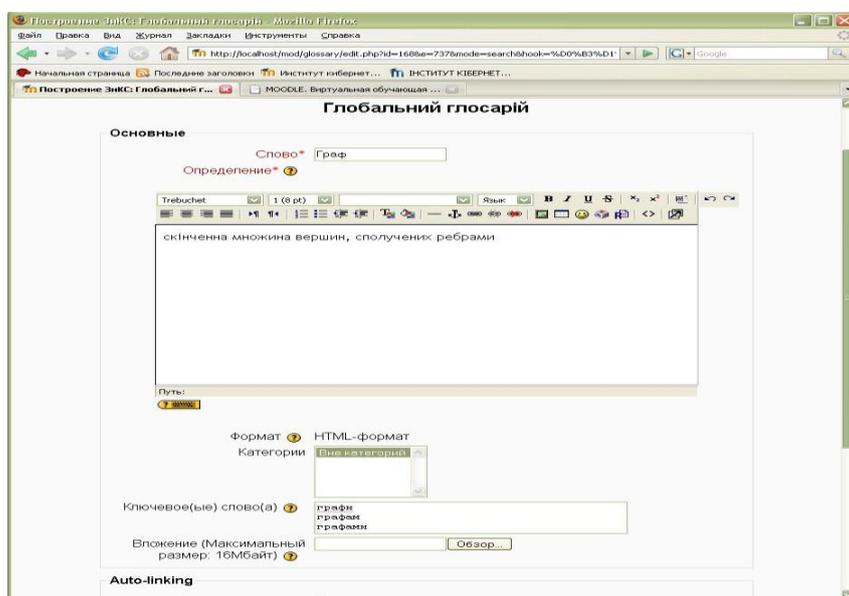


Рис. 2.3.1. Наполнение «Глоссария»

Для создания автоматической ссылки на статьи «Глоссария», они должны быть разрешены в настройках термина (рис. 2.3.1), настройках «Глоссария» (локальных и глобальных – рис. 2.3.2) и настройках фильтров курса (рис. 2.2.1). Аналогично для настройки работы «Базы данных» и других элементов курса.

Построение SCORM-иерархии электронного курса

Для построения SCORM-иерархии электронного курса необходимо добавить новый ресурс – «web page», выбрав соответствующий пункт меню «Compose web page» из поля со списком, как показано на рис. 2.3.3.

Самым простым вариантом представления иерархии является таблица, которую можно добавить к странице с помощью кнопки «Insert table» (выделена жирной линией на рис. 2.3.4) встроенного визуального HTML редактора.

Создаются гиперссылки между названиями элементов электронного курса и необходимым местом текста курса соответственно пункту “Создание внутренних ссылок в элементах курса”.

Оболочка Moodle предоставляет разнообразные инструменты для организации эффективного общения преподавателя и студента в процессе дистанционного обучения. Соблюдение приведенных рекомендаций при создании электронного курса в системе Moodle позволит упростить создание электронного курса, повысить качество представления учебного материала и организацию обучения.

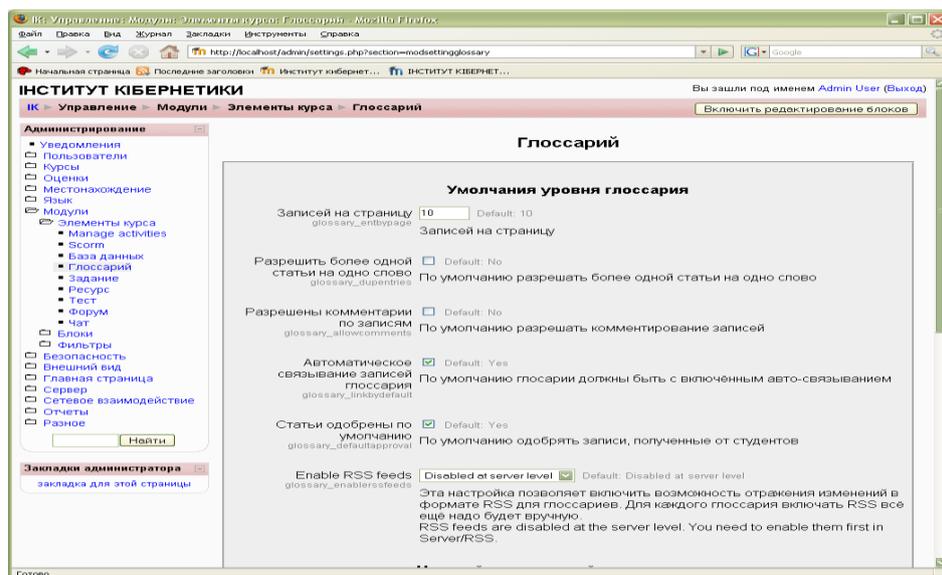


Рис. 2.3.2. Глобальные настройки для «Глоссария»

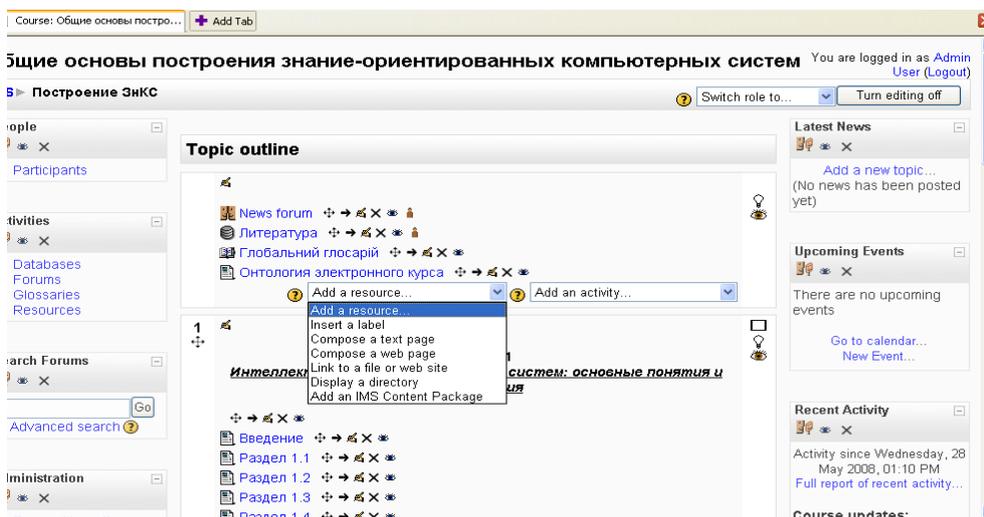


Рис. 2.3.3. Добавление иерархии электронного курса

Аналитическое представление	Табличное представление	Графическое представление	Текстовое представление
Формальное представление формальной системы	Таблица 1. Типы операций над понятиями	Рис1. Обобщенная схема ИИС	Интеллектуализация компьютерных систем основные понятия и определения
Формальное представление сетевой модели	Таблица 2. Технологии управления знаниями	Рис2. Структурная схема ПСИИ	

Рис. 2.3.4. Табличное представление иерархии электронного курса

8.3. Пример формализованного представления онтологии текстового документа

При анализе лингвистических корпусов текстов средствами интеллектуальных информационных систем, целесообразно хранить обработанный документ в виде его онтологии, что позволит включить его в общую базу данных для последующего анализа, построения схемы зависимостей между документами, упрощения анализа больших объёмов информации и повышения релевантности поиска информации при запросе пользователя. Данная задача является комплексной, и в данном разделе освещена только часть, касающаяся представления онтологий в реляционной СУБД и правила выборки части онтологии из БД в соответствии с определенными критериями для использования в прикладных задачах пользователей.

Для анализа информации на больших массивах данных целесообразно использовать знание-ориентированную поисковую систему [142], задача которой состоит из четырёх подзадач:

- построение онтологии отдельного текстового документа;
- интеграция построенных онтологий ТД в онтологию предметной области;
- поиск документов по запросу пользователя;
- развитие самой системы.

Следует отметить, что любые работы, связанные с автоматическим анализом текстов, требуют определенного набора лингвистических средств анализа, основу которых составляют машинные словари, программы морфологического, синтаксического и семантического анализа, выделения терминологической лексики и т. д. Для ЗнПС указанные средства предоставляет языково-онтологическая информационная система [27]. Кроме лингвистических ресурсов, для анализа естественно-языковых текстов необходимы базы данных с “внешней” информацией, которая предоставляется экспертом в заданной ПдО. В данном случае инструментом взаимосвязи между экспертом и программными моделями анализа естественно-языкового текста является О ПдО.

Одной из задач анализа ЕЯТ является построение его онтологии в виде формализованных структур данных и правил их обработки. При этом каждая построенная онтология ТД представлена традиционной схемой: $O = \langle X, R, F \rangle$.

Такая модель даёт возможность анализировать знания в примитивах нижнего уровня и организовать последние в структуры высшего уровня. В ЕЯТ основной единицей является слово, а основной структурой – предложение. Для классификации и организации знаний используются такие структуры высшего уровня как параграфы, разделы, главы и тома.

При анализе текста средствами ЗнПС и ЯОИС происходит его разбиение на отдельные, логически связанные блоки, средним звеном которых является абзац, а низшим – предложение. Анализируя каждое предложение определённого абзаца, можно построить модель данного предложения в виде фрейма. Такое представление удобно тем, что строго декларативная информация во фрейме может быть оттранслирована в логику первого порядка [26], что позволит проводить дальнейший анализ при интеграции представлений отдельных предложений в более общие структуры. Обработанная текстовая информация, а также её представление в виде фреймов, заносится в базу данных и используется для обновления и изменения начальной О ПдО. Машина логического вывода, выполняющая данные функции, реализована как программная надстройка над базой данных ЗнПС.

Для наполнения баз данных, составления правил для машины логического вывода и обмена данными между системами ЗнПС и ЯОИС, в последнее время нашел широкое применение язык представления знаний KIF, который можно использовать как расширение синтаксиса языка программирования LISP.

Приведём пример представления простого предложения в виде LISP-программы с поддержкой представления данных в формате KIF. Рассмотрим следующее предложение: «ЗнПС включает в себя базу данных и лингвистический процессор».

Сначала построим строчную запись концептуального графа [26] для данного предложения в следующем виде:

Object:ЗнПС <-- Включает в себя -->Attr --> база данных --> лингвистический процессор

Наряду с разработкой онтологического подхода к построению онтологии ТД, возникает необходимость поиска определённой ветви онтологии в БД. При представлении онтологии в виде KIF-структур вместе с дополнительными программами-процедурами на языке LISP возникают затруднения в использовании традиционных реляционных БД. Возможно, наиболее рациональным подходом может оказаться организация отдельной БД для хранения только «образа» онтологии: $O = \langle X, R \rangle$, то есть усечения онтологии до онтографового представления. В этом случае онтология представляется в виде дерева с произвольным ветвлением, которое легко можно преобразовать в двоичное дерево. Такое представление упростит проведение выборки определённой ветви дерева. При использовании данной БД возможно использование языка доступа к базам данных SQL для выбора интересующей пользователя ветви онтологии.

Двоичные деревья поиска позволяют выполнять следующие операции с динамическими множествами [187]: Search(Поиск), Minimum(Минимум), Maximum(Максимум), Predecessor(Предыдущий), Successor(Следующий), Insert(Вставить), Delete(Удалить). Время выполнения основных операций пропорционально «высоте» дерева. Если все уровни в двоичном дереве имеют максимальное число вершин, то его «высота» (и, соответственно, время выполнения операций) пропорциональна логарифму числа вершин.

При представлении с использованием указателей, помимо ключа *key* и дополнительных данных, также хранятся и указатели *left*, *right*, *p* (левый ребенок, правый ребенок и родитель). Если таковые отсутствуют, то соответствующее поле содержит NIL.

Ключи в двоичном дереве поиска хранятся с соблюдением свойства упорядоченности: пусть *x* – произвольная вершина двоичного дерева поиска. Если вершина *y* находится в левом поддереве вершины *x*, то $key[y] \leq key[x]$. Если *y* находится в правом поддереве *x*, то $key[y] \geq key[x]$.

Процедуру поиска можно представить алгоритмом вида:

```
Tree-Serch(x, k)
1 while x ≠ NIL and k ≠ key[x]
2 do if k < key[x]
3     then x ← left[x]
4     else x ← right[x]
5 return x.
```

Приведём пример представления упрощенной онтологии некоторого домена предметной области «Операционные системы» (рис. 8.11) в виде бинарного дерева и реализации процедуры выборки определенной ветви данной онтологии.

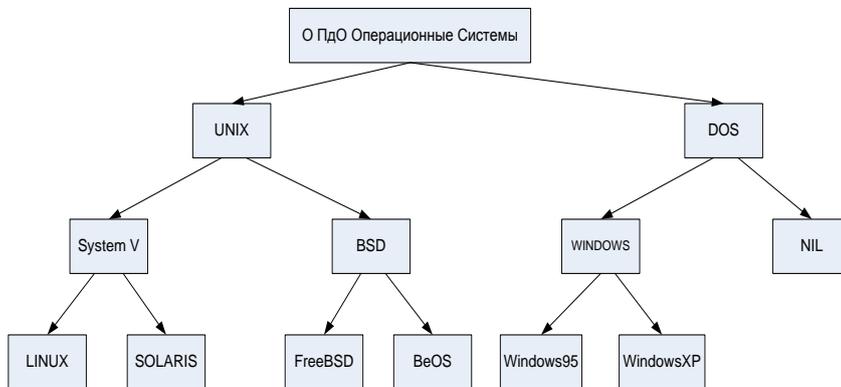


Рис. 8.11. Онтология домена предметной области «Операционные системы»

Составим таблицу в СУБД *mysql* для хранения элементов онтологии:

```
mysql> create table base (
-> id int,
-> name_hub varchar(15),
-> parent varchar(15),
-> child_1 varchar(15),
-> child_2 varchar(15),
-> key numb int);
```

Заполним таблицу согласно О ПдО ОС, после чего она примет вид:

```
mysql> select * from base;
```

id	name_hub	parent	child_1	child_2	keynumb
1	ОПО ОС	NULL	UNIX	DOS	10
2	UNIX	ОПО ОС	System V	BSD	5
3	DOS	ОПО ОС	Windows	NULL	15
4	System V	UNIX	Linux	Solaris	2

5	BSD	UNIX	FreeBSD	BeOS	7
6	Windows	DOS	Windows95	WindowsNT	12
7	Linux	System V	NULL	NULL	1
8	Solaris	System V	NULL	NULL	3
9	FreeBSD	BSD	NULL	NULL	6
10	BeOS	BSD	NULL	NULL	8
11	Windows 95	Windows	NULL	NULL	11
12	Windows NT	Windows	NULL	NULL	14

Поле *keynumb* позволяет представить данную таблицу в виде бинарного дерева. Выборка отдельной ветви при данной организации является тривиальной операцией, требующей одного прохода по базе данных. К примеру, выделим ветвь с названием узла 'UNIX':

```
mysql> select *
-> from base
-> where keynumb >= 1
-> and keynumb < 10
-> order by id;
```

id	name_hub	parent	child_1	child_2	keynumb
2	UNIX	ОПО ОС	System V	BSD	5
4	System V	UNIX	Linux	Solaris	2
5	BSD	UNIX	FreeBSD	BeOS	7
7	Linux	System V	NULL	NULL	1
8	Solaris	System V	NULL	NULL	3
9	FreeBSD	BSD	NULL	NULL	6
10	BeOS	BSD	NULL	NULL	8

7 rows in set (0.00 sec)

Данную выборку из таблицы можно представить в виде ветви бинарного дерева (рис. 8.12). Расположение и зависимости элементов друг от друга содержатся в поле *keynumb*.

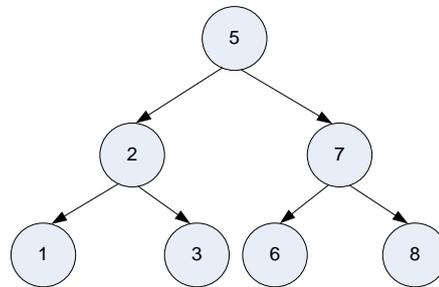


Рис. 8.12. Бинарное дерево для представления ветви "UNIX"

8.4. Примеры разработки некоторых процедур Text Processing

8.4.1. Классификация (рубрикация) текстов из заданной ПдО

Рассмотрим работу системы (рис. 6.5) на примере решения задачи классификации найденных в сети Интернет текстовых документов, алгоритмы которой лежат в основе более сложных задач обработки текстовой информации (в том числе поиска релевантной информации, построения персонифицированных баз знаний и др.). При этом блок прикладного процессинга представлен упомянутой выше ЗнПС.

Формальную постановку задачи классификации представим следующим образом:

– предполагается, что ЗнПС находит в сети некоторое множество текстовых документов $D = \{d_i\} (\forall i = 1 \div m, m - \text{мощность множества})$, представленных для классификации в заданной ПдО;

– всё множество D разбивается на непересекающиеся подмножества классов

$$C = \{c_j\}, j = 1 \div n, \quad \bigcup_{j=1}^n c_j = D, \quad c_j \cap c_k = \{\emptyset\}, (j \neq k).$$

Задачей классификации является определение класса, к которому принадлежит данный документ.

Исходя из архитектурно-структурной организации ОУИС, представленной на рис. 6.7, задана база знаний ПдО (в нашем случае достаточно и онтологии ПдО) и база знаний лексики ЕЯ. При этом заметим, что:

– если множество лексем, описывающих заданную ПдО, не является подмножеством множества лексем ЯОКМ, то необходимо создать лингвистическую онтологию заданной ПдО;

– мы сознательно опускаем описание работы той части алгоритма, в которой реализуется снятие неоднозначностей при анализе текстовых документов;

– в блоке прикладного процессинга заданы алгоритмы построения О ТД для классов C и семантическая память большого объёма для хранения классифицированных документов.

Последовательность шагов решения задачи классификации следующая:

1. Каждый, найденный ЗнПС документ проходит предварительную обработку (анализ на релевантность, создание текстового файла и др.) и передаётся на вход ОУИС, а точнее – на вход грамматического анализатора.

2. В грамматическом анализаторе выполняется морфологический и синтаксический анализ документа, при этом он активно взаимодействует с ЛБД блока базы знаний лексики ЕЯ. В результате работы грамматического анализатора получаем набор синтаксических деревьев предложений документа. Указанный набор передаётся на вход семантического анализатора.

3. Семантический анализатор преобразует синтаксические деревья предложений документа в семантические деревья и в конечном итоге выполняет предварительное объединение семантических фрагментов документа. При этом его блоки активно взаимодействуют с ЯОКМ. Одним из важных результатов работы семантического анализатора является снятие всех неоднозначностей в семантическом представлении документа. Набор семантических фрагментов документа и результат их объединения передаются на вход СИИ.

4. СИИ при решении задачи классификации документов выполняет ограниченные функции, а именно – на основе разрозненных семантических фрагментов документа строит взаимосвязанное множество семантических структур документа.

5. Такое взаимосвязанное множество семантических структур документа в упрощённом виде можно рассматривать как онтологию документа, пригодную для сравнения с онтологиями классов. Другие алгоритмы блока прикладного процессинга реализуют распределение по классам найденных ЗнПС документов и их запоминание в семантической памяти.

8.4.2. Обработка текстов библиографического характера

Текст библиографического характера является хорошо структурированным, в котором практически в явном виде представлены семантические отношения. Знания, извлечённые из такого ЕЯТ, естественным способом интерпретируются в языке высказываний. Продемонстрируем это примером.

Пусть множество предметных констант составляют сотрудники некоторого отдела и статьи, авторами которых они являются. В данном случае высказывания представляют отношения между авторами и написанными ими статьями.

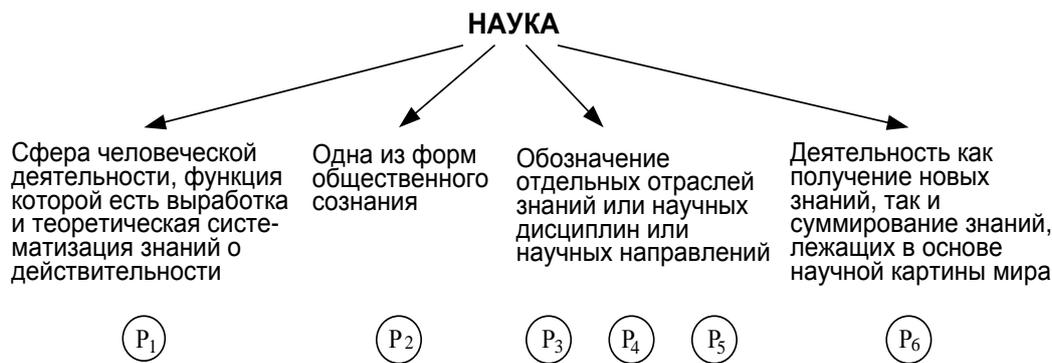
Пусть $c_i, i=1, 2, \dots, n$, означают предметные константы, $s(c_i)$ – семантическое значение предметной константы. Для нашего примера рассмотрим такие предметные константы:

КОНСТАНТЫ	СЕМАНТИЧЕСКОЕ ЗНАЧЕНИЕ
автор1	$s(\text{автор1}) = \text{Палагин}$
автор2	$s(\text{автор2}) = \text{Яковлев}$
автор3	$s(\text{автор3}) = \text{Кривый}$
автор4	$s(\text{автор4}) = \text{Петренко}$
автор5	$s(\text{автор5}) = \text{Опанасенко}$
автор6	$s(\text{автор6}) = \text{Кургаев}$

Предметные константы составляют алфавит объектного языка, а семантическое значение – интерпретацию предметных констант или их сущность. Язык L^0 включает предикатные константы (отношения) $P_j, j = 1, 2, \dots, m$, семантические значения которых являются множествами. Семантическое значение предикатной константы P_j будем обозначать $s(P_j)$. Для нашего случая определим такие предикаты:

ПРЕДИКАТНАЯ КОНСТАНТА	СЕМАНТИЧЕСКОЕ ЗНАЧЕНИЕ
$P_{ав}$	$s(P_{ав}) = \text{множество авторов}$
$P_{ст}$	$s(P_{ст}) = \text{множество статей}$
$P_{ас}$	$s(P_{ас}) = \text{множество пар } (x, y), \text{ где } x - \text{автор,}$ $y - \text{статья, автором которой } - x$
$P_{арс}$	$s(P_{арс}) = \text{множество троек } (a, p, c), \text{ где}$ $a - \text{автор, } p - \text{рецензент, } c - \text{статья}$
$P_{чс}$	$s(P_{чс}) = \text{множество пар } (x, y), \text{ где } x - \text{читатель, } y - \text{статья}$

Обработка значений. Рассмотрим текст, который условно можно назвать структурированным. Это естественно-языковые определения. Ниже даны примеры таких определений, взятые из энциклопедического словаря.

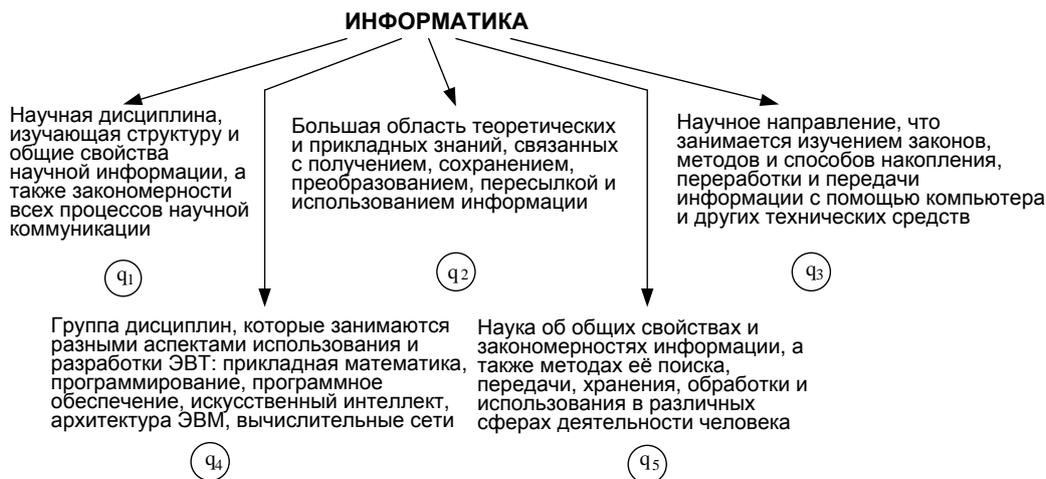


$$K(a) = \{p_1, p_2, p_3, p_4, p_5, p_6\} - \text{класс эквивалентности.}$$

Следующие определения также взяты из энциклопедического словаря.



$$K(c) = \{r_1, r_2, r_3\}$$



$$K(b) = \{q_1, q_2, q_3, q_4, q_5\}$$

Приведенные примеры показывают, что построение классов эквивалентности не представляет особых трудностей. В результате их построения появляются объекты:

$$K(a) = \{p_1, p_2, p_3, p_4, p_5, p_6\}, K(b) = \{q_1, q_2, q_3, q_4, q_5\}, K(c) = \{r_1, r_2, r_3\}.$$

Проблема появляется при вычислении второго отношения, которое определяет отношение подчинения (иерархии) между найденными классами эквивалентности. Но при таком абстрактном представлении классов $K_i(x)$ это отношение

определить невозможно (мало информации). Для этого необходимо знать структурные характеристики элементов из классов $K_i(x)$. Следовательно, естественным способом появляется необходимость в структуре элементов из классов эквивалентности. Например, если возвратиться к вышерассмотренным примерам, то каждый элемент из класса $K(a)$ принимает вид:

$$p_1 = (p_{11}, p_{12}, p_{13}), p_2 = (p_{21}), p_3 = (p_{31}, p_{32}, p_{33}), p_4 = (p_{41}, p_{42}), p_5 = (p_{51}, p_{52}), \\ p_6 = (p_{61}, p_{62}),$$

где P_{11} = «сфера человеческой деятельности», P_{12} = «выработка знаний об объективной действительности», P_{13} = «система знаний о объективной действительности», P_{21} = «форма общественного сознания», P_{31} = «отрасль знаний», P_{32} = «научная дисциплина», P_{33} = «научное направление», P_{41} = «деятельность по получению новых знаний», P_{42} = «суммирование знаний о научной картине мира».

Аналогично структурируются и оставшиеся элементы в классах эквивалентности.

$$r_1 = (r_{11}, r_{12}, r_{13}, \dots), r_2 = (r_{21}, r_{22}, r_{23}, \dots), r_3 = (r_{31}, r_{32}, r_{33}, \dots), \\ q_1 = (q_{11}, q_{12}, q_{13}, \dots), q_2 = (q_{21}, q_{22}, q_{23}, \dots), q_3 = (q_{31}, q_{32}, q_{33}, \dots), \\ q_4 = (q_{41}, q_{42}, q_{43}, \dots) \\ q_5 = (q_{51}, q_{52}, q_{53}, \dots)$$

Из приведенной структуризации вытекает следующая формализация. Если класс эквивалентности относится к объекту a , то его формальное определение выглядит как дизъюнкция элементов, составляющих этот класс. Каждый элемент, входящий в тот или иной класс эквивалентности, описывается соответствующим предикатом, то есть если $K(a) = \{p_1, p_2, p_3, p_4, p_5, p_6\}$, то $p(a) \Leftrightarrow p_1(a) \vee \dots \vee p_6(a)$, где p_i – предикаты, характеризующие элементы из класса $K(a)$, а их дизъюнкция характеризует целый класс понятия a .

Далее, если $q_i \in K(a)$ и $q_i = (q_{i1}, q_{i2}, \dots, q_{ik})$, то элемент p_i (или объект p_i), который характеризуется атрибутами p_{ij} , представляется в виде конъюнкции

$$p_i(a) \Leftrightarrow p_{i1}(a) \wedge \dots \wedge p_{ik}(a),$$

где $p_{ij}(a)$ – предикат, характеризующий отдельный атрибут понятия

a , $i = 1, \dots, l$; $j = 1, \dots, k$. Таким образом, каждый класс $K(a)$ описывается дизъюнктивной формой вида

$$p(a) \Leftrightarrow (p_{11}(a) \wedge \dots \wedge p_{lm_1}(a)) \vee \dots \vee (p_{l1}(a) \wedge \dots \wedge p_{lm_l}(a)).$$

Введенная формализация определяет отношение частичного порядка следующим образом

$$K(a) \leq K(b) \Leftrightarrow (\exists p_i(a))(\exists q_j(b))(q_j(b) \leq p_i(a)),$$

где $q_j(b) \leq p_i(a)$ означает, что $q_j(b)$ входит в виде конъюнктивного члена в $p_i(a)$.

Введенное таким образом отношение частичного порядка естественным образом требует предикатно-реляционного представления объектов из классов эквивалентности и самих этих классов [59]. Для иллюстрации сказанного обратимся к вышеприведенному примеру. Выясним, каким образом определяется тот факт, что классу «НАУКА» подчиняется класс «ИНФОРМАТИКА».

Класс «НАУКА», обозначенный как $K(a)$, описывается формулой:

$$p(a) \Leftrightarrow p_1(a) \vee p_2(a) \vee p_3(a) \vee \dots \vee p_6(a),$$

где $p_1(a) \Leftrightarrow$ СФЕРА-ЧЕЛОВЕЧ-ДЕЯТ(a), $p_2(a) \Leftrightarrow$ ОТРАСЛЬ ЗНАНИЙ(a),

$$p_3(a) \Leftrightarrow$$
 НАУЧНАЯ-ДИСЦИП(a), $p_4(a) \Leftrightarrow$ НАУЧН-НАПРАВЛ(a),

$p_5(a) \Leftrightarrow$ ДЕЯТ-ПОЛУЧ-НОВЫХ-ЗН(a), $p_6(a) \Leftrightarrow$ ДЕЯТ-СУММИР-ЗН-НКТ(a),

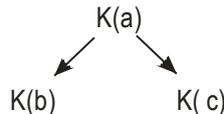
Класс «ИНФОРМАТИКА», обозначенный как $K(b)$, описывается формулой:

$$q(b) \Leftrightarrow q_1(b) \vee q_2(b) \vee q_3(b) \vee \dots \vee q_l(b),$$

где $q_1(b) \Leftrightarrow p_3(a) \wedge$ ОБЩ-СВОЙ-НАУЧ-ИНФ(b) \wedge ЗАКОН-ПРОЦ-НАУЧ-ДЕЯТ(b) \wedge

Используя приведенное выше определение отношения частичного порядка, находим, что $p_3(a)$ входит в определение класса $K(a)$, причём $q_1(b) \vee p_3(b) = p_3(b)$ (на основании закона поглощения), а это означает, что $K(b) \leq K(a)$.

Аналогично определяется подчинённость класса $K(c)$ классу $K(a)$, в результате чего получаем граф



Полученную таким образом иерархию можно изменять или модифицировать путём диалога с пользователем с целью достижения более правильного представления.

Подводя итог всему сказанному, вышеописанный процесс обработки текста определений формализуется следующим образом.

Пусть T – множество текстов определений. По этому множеству T строится множество классов, которое определяется отношением эквивалентности R и составляет фактор-множество D . На полученном таким образом множестве D определяются специальные отношения R_2, \dots, R_k , которые описывают характеристические свойства элементов из D , то есть элементов из классов эквивалентности. Эти отношения представляются в виде предикатов, которые определяют отношения частичного порядка R_l . Это отношение является вторым отношением, по которому строится онтология. Более точно, онтология строится по транзитивному замыканию R_l^* отношения R_l , согласованного с отношением R .

Определение 8.1. Отношения R и R_l^* будем называть согласованными, если $\forall a, b \in D$ имеет место включение $(a, b) \in R^* R_l^*$, где $R^* R_l^*$ суперпозиция R и R_l .

Из этого определения логично вытекает первичная онтология, имеющая вид: $O = (D = T / R, \mathfrak{R} = \{R_1, R_2, \dots, R_k\}, \varphi, A)$, где $\varphi: D \rightarrow T$ – интерпретация, A – множество аксиом, определяемое предикатами, которые описывают характерис-

тические свойства элементов из D , R_2, \dots, R_k – соответствующие им отношения, а R_1 – отношение частичного порядка.

Обработка ЕЯТ относительно темпоральных и объектных отношений

Рассмотрим ещё один пример конкретизации отношений R и R_1 , которые будем называть временными и объектными.

Отношение эквивалентности R определяется конкретным объектом (отсюда и название этого отношения), фигурирующим в тексте T (например, личность, театр, институт, вуз, кафедра и т. д.), а все объекты текста T , которые с ним связаны, составляют класс эквивалентности по этому отношению.

Отношение R_1 определяет связь объектов из классов эквивалентности отношения R или в хронологическом порядке, или генеалогическом, или другом подобном порядке.

Отношения R и R_1 являются источником построения новых отношений. Например, если R_1 описывает временные зависимости, то можно определить отношение, связывающее объекты, относящиеся к данному конкретному моменту времени (год, месяц, день и т. д.). Объясним это на примере.

Пусть $D = \{a, b, c, d\}$ – объекты, фигурирующие в обрабатываемом тексте. Тогда $D/R = \{K(a), K(b), K(c), K(d)\}$ состоит из классов, элементами которых являются моменты времени t_{xi} , связанные с объектом $x \in D$, $i = 1, \dots, j_x$. Допустим, что нас интересуют объекты, фигурирующие в момент времени t . Тогда момент времени t определяет отношение R_t , которое включает объекты, связанные с этим моментом времени. Арность этого отношения определяется мощностью множества D . Если $D = \{K(a), \dots, K(b)\}$, то $R_t = \{(a', \dots, b') \mid a' \in K(a) \wedge \dots \wedge b' \in K(b)\}$.

Далее, на отношениях R_t определяется отношение линейного порядка:

$R_t \leq R_{t'} \Leftrightarrow t \leq t'$. Пользуясь этим отношением порядка, можно ввести такое понятие.

Определение 8.2. *Элементарным временным сценарием* для временного интервала $[t_1, t_k]$ называется цепь $R_{t_1} \leq R_{t_2} \leq \dots \leq R_{t_k}$, где R_{t_i} – отношения, определённые выше.

Это определение можно модифицировать в зависимости от семантики отношений R и R_1 . Действительно, цепь отношений $R_{t_1} \leq R_{t_2} \leq \dots \leq R_{t_k}$ описывает хронологию определённых событий во времени, связанных с конкретными людьми, объектами и т. д. Такого типа цепь действительно может служить основой некоторого реального сценария.

8.4.3. Знание-ориентированная поисковая система и релевантный поиск документов в сети Интернет

Постоянно возрастающий поток оригинальных электронных документов в среде Интернет требует разработки новых подходов к решению проблемы поиска

информации, релевантной запросу пользователя, исходя из его первичной системы знаний в заинтересованной ПдО и адекватного отображения объекта его интересов в структурированные семантические модели [142]. Одним из главных подходов является эффективный механизм поиска информации на основе онтологических принципов, которые связывают базовые концепты концептуальными отношениями и более сложным путём организованные постоянные конструкции для запросов и ответов с “хорошо” обозначенной семантикой, а также семантической маркировки ресурсов [3, 10].

При этом важной составляющей является подсистема анализа и понимания естественно-языковых текстов, опирающаяся на языково-онтологическую картину мира, что в совокупности с программно-аппаратными средствами составляет языково-онтологическую информационную систему [27].

Такая система, вместе с эффективными механизмами взаимодействия субъектов в информационной среде, может служить инструментарием для разработки интеллектуальных информационных систем в различных предметных областях, в частности в знание-ориентированных поисковых системах.

Современные машины поиска используют алгоритмы, которые базируются на ключевых словах. При этом полученные индексы связывают информацию с терминами, учитывая только актуальный для данного запроса лексический или синтаксический контекст. Большинство современных классических поисковых систем используют механизм автоматической индексации данных и соответственно хранение данных в виде реляционной СУБД. Поэтому и алгоритмы поиска сводятся к поиску информации в простых базах данных. Такая структура поисковых систем не отвечает современным требованиям к хранению и обработке данных и нуждается в разработке новых моделей анализа информации и перехода к использованию семантики ПдО при поиске релевантной информации с помощью ЗнПС.

Также, быстрое развитие украинского сегмента Интернет и корпоративных сетей с большими потоками электронных документов требует разработки более эффективных средств поиска и структурирования информации, к тому же с ориентацией на украинско-язычные документы. Ненасыщенность данного сегмента программными продуктами сегодня компенсируется преимущественно поисковыми системами зарубежного производства, которые, как правило, обрабатывают украинско-язычный запрос поверхностно, или предлагают пользователю переформулировать запросы на другом языке.

На основе репрезентативного анализа опыта построения информационно-поисковых систем предложена общая архитектура ЗнПС, приведенная на рис. 8.13. Её основным назначением является эффективная работа с ЕЯТ при поиске и классификации документов.

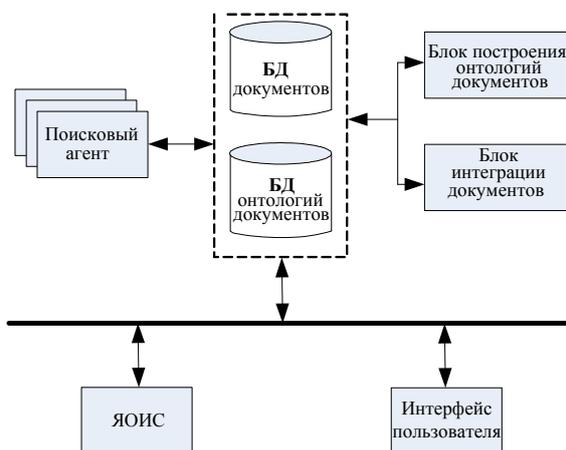


Рис. 8.13. Общая архитектура ЗнПС

Кроме традиционного поиска документов, ЗнПС совместно с ЯОИС, позволяет:

- значительно повысить релевантность найденных документов по запросу пользователя за счёт снятия разного рода неоднозначностей и формального отображения структуры документа на онтологическую структуру ЯОКМ;
- учитывать “фоновые” знания документа, и выполнять поиск документов по родственным тематикам, который даст возможность расширить поисковый запрос без участия пользователя системы с сохранением высокой релевантности ответа ЗнПС;
- использовать ЗнПС как корпоративную поисковую систему.

Поиск информации в сети выполняют поисковые агенты, которые взаимодействуют между собой и с другими блоками системы. Информация в Интернет перестаёт быть статической, она дополняется динамической составляющей, скриптами и служебной информацией, которая нуждается в дополнительном анализе в процессе загрузки для дальнейшего семантического анализа документов.

Вообще, выделяют четыре характеристики, общие для всех агентов [26]:

- автономность (во время работы агенты “саморегулируются” и выполняют свои действия согласно своим внутренним распоряжениям и директивам);
- социальность (агенты общаются с другими агентами и людьми с помощью некоторого языка коммуникаций агентов);
- реактивность (агенты “ощущают” аспекты среды и соответственно реагируют на них);
- инициативность (агенты могут брать инициативу на себя при выполнении целенаправленных действий).

Данный агент является автономным поисковым элементом, который взаимодействует с другими поисковыми агентами, серверами DNS и с центральным узлом поисковой системы. В него встроен *синтаксический анализатор* для анализа служебных данных документов, которые могут нести предыдущую семантическую составляющую информации документа. Он служит для анализа динамично обновляемой информации порталов. Агенты активизируются при нахождении информационно наполненного Интернет-портала и работают исключительно с ним. *Блок взаимодействия* предназначен для обмена данными и связи между агентами. Алго-

ритмы взаимодействия всех блоков системы между собой выполняются под управлением *блока управления*.

Архитектура поискового агента приведена на рис. 8.14.

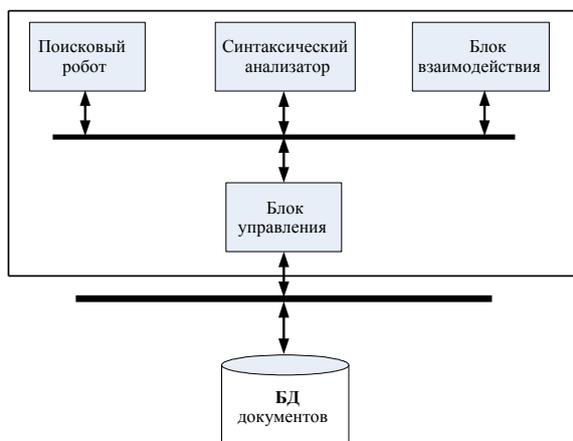


Рис. 8.14. Архитектура поискового агента

В ЗнПС используются две БД – это база данных накопленных документов и база данных их онтологических структур. Первая из них построена по технологии реляционной СУБД и используется для хранения документов, которые подлежат обработке. Перед загрузкой в эту базу документу присваивается идентификационный номер, выделяются все URL из документа, и проводится его архивация. Информация, предварительно обработанная поисковым агентом, прибавляется к данному документу. Одной из задач БД онтологических структур является сохранение “фоновых” знаний о ПдО и общеупотребительной информации. Перспективным направлением разработки архитектуры БД считается создание надстройки над XML-ориентированной базой данных. Выбор данной технологии как базовой объясняется рядом преимуществ, которые она предлагает при развитии распределенных информационных интеллектуальных систем. Во-первых, это даёт возможность значительно улучшить уровень интеграции данных и обеспечить взаимодействие таких систем в рамках Интернет. Во-вторых, эта технология с самого начала ориентирована на работу с электронными документами и на их сохранение, работу со структурированными данными, которые поступают от многих источников [142]. Еще одна важная проблема при разработке БД онтологических структур заключается в том, что при работе с онтологиями документов, формируются не полностью структурированные данные. Примером таких данных из повседневной жизни могут служить записи в медицинской карточке больного. Для одного пациента в ней могут сохраняться данные о прививке, для другого – это полная история болезни, для третьего – показатели веса, артериального давления и другие параметры. Не полностью структурированные данные неэффективно сохранять в реляционной БД, поскольку это приводит к значительному увеличению разнообразных таблиц, а, следовательно, увеличению количества связей между ними и времени поиска информации в такой БД.

Основными блоками системы являются блок построения онтологий документов и блок интеграции онтологий документов. Для построения онтологий ЗнПС взаимодействует с ЯОИС.

Для связи системы с внешней средой используется блок интерфейса пользователя (рис. 8.15). На него возлагаются функции взаимодействия с пользователем, грамматический разбор запроса с помощью лингвистического процессора и его представление в формате внутренней структуры системы, а также вывод результатов пользователю.

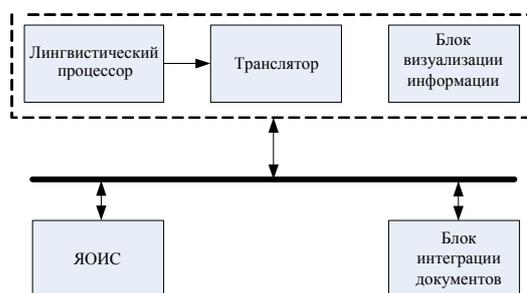


Рис. 8.15. Архитектура интерфейса пользователя

На лингвистический процессор возлагается задача анализа и снятия неоднозначностей входного запроса пользователя. Результаты работы ЛП транслируются во внутреннее представление ЗнПС для дальнейшей обработки информации, выполнение семантического анализа содержания запроса и корректировка дерева зависимостей онтологий, соответственно с запросом пользователя.

ЗАКЛЮЧЕНИЕ

В монографии рассмотрены теоретические основы и основные задачи проектирования ЗОИС с онтолого-управляемой архитектурой для обработки больших объёмов ЕЯ-информации, предложены алгебра и система списочных структур для формального описания процедур анализа и понимания ЕЯТ, извлечения (обнаружения) новых знаний, автоматизированного построения онтологических баз знаний предметных областей и соответствующего инструментального комплекса онтологического назначения. В частности:

- предложены теоретические основы построения знание-ориентированных онтолого-управляемых информационных систем, включая систему формализованных методик и алгоритмов анализа ЕЯ-информации и синтеза предметных знаний;

- модифицирован известный логико-информационный подход к проектированию вычислительных систем с учётом онтологической парадигмы применительно к ЗОИС, работающих с ЕЯО. Он позволяет установить однозначное соответствие между объектами онто-логической модели: предметно-ориентированными процедурами, алгоритмами, фрагментами алгоритмов и их информационно-кодowymi эквивалентами на всех уровнях иерархической системы, какой представляется проектируемая система;

- предложены принципы системно-онтологического анализа предметно-ориентированной области знаний, предложена методика формализованного проектирования онтологии предметной области и связанного с ней проблемного пространства, включающей онтологию объектов, онтологию процессов и онтологию задач. Методика учитывает категориальный уровень знаний и использует в качестве источника знаний лингвистический корпус текстов;

- исследованы проблемы анализа и понимания ЕЯ компьютером, формального представления и обработки знаний, извлечённых из ЕЯТ. Предложена модель интегрированной информационной технологии обработки знаний, содержащихся в естественно-языковых объектах, включающей технологию компьютерной обработки ЕЯ, извлечения и представления знаний и обработки знаний в заданной предметной области, а также основные алгоритмы и соответствующие инструментальные средства её реализации;

- предложена методика проектирования и архитектурно-структурная организация языково-онтологической информационной системы, которая включает грамматический процессор, семантический процессор и базу знаний ЕЯ;

- предложена методика и алгоритмы системной интеграции междисциплинарных научных знаний.

Очевидно, что для решения описанных выше проблем необходимы дальнейшие исследования с разработкой новых формально-логических теорий, методов интеграции технологий Data & Text mining, управления сверхбольшими объёмами хранилищ знаний, а также концепции эффективного управления распределёнными междисциплинарными знаниями.

ЛИТЕРАТУРА

1. *Сагатовский В.Н.* Основы систематизации всеобщих категорий / Сагатовский В.Н. – Издательство Томского университета, Томск, 1973. – 432 с.
2. *Палагин А.В.* Системно-онтологический анализ предметной области / А.В. Палагин, Н.Г. Петренко. – УСиМ, 2009. – № 4. – С. 3–14.
3. *Палагин А.В.* Системная интеграция средств компьютерной техники / А.В. Палагин, Ю.С. Яковлев. – Винница: УНІВЕРСУМ, 2005. – 680 с.
4. *Башмаков А.И.* Интеллектуальные информационные технологии: Учеб. пособие / А.И. Башмаков, И.А. Башмаков. – М.: Изд.-во МГТУ им. Н.Э. Баумана, 2005. – 304 с.
5. *Гаврилова Т.А.* Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский. – СПб.: Питер, 2001. – 384 с.
6. *Guarino N.* Ontologies and Knowledge Bases. Towards a Terminological Clarification / N. Guarino, P. Guaretta. – Amsterdam: IOS Press, 1995. – 7 p.
7. *Gruber T. R.* A translation approach to portable ontology specifications / Gruber T. R. – Knowledge Acquisition, 5 (2), 1993. – PP. 199–220.
8. *Gruber, T.R.* Toward Principles for the Design of Ontologies Used for Knowledge Sharing / Gruber T. R.. – International Journal of Human and Computer Studies, 43(5/6), 1995. – PP. 907–928.
9. *Niles I.* Towards a Standard Upper Ontology / I. Niles, A. Pease // 2nd International Conference on Formal Ontology and Information Systems (FOIS-2001), Welty C., and Smith B., eds., Ogunquit, Maine. – 17–19, October, 2001. – 8pp. – Режим доступа <http://home.earthlink.net/~adampease/professional/FOIS.pdf>. – Дата доступа: 16.12.2008. – Название с экрана.
10. *Палагин А.В.* Архитектура онтолого-управляемых компьютерных систем / Палагин А.В. – Кибернетика и системный анализ, 2006 – №2. – С. 111–124.
11. *Палагин А.В.* Организация и функции "языковой" картины мира в смысловой интерпретации ЕЯ-сообщений / А.В. Палагин. – Information Theories and Application, 2000. – Vol. 7, №4. – С. 155–163.
12. *Чернавский Д.С.* Синергетика и информация: Динамическая теория информации / Д.С. Чернавский – Изд. 2-е, испр., доп. М.: УРСС, 2004. – 288 с. – Режим доступа: <http://ru.wikipedia.org/wiki/BookSources/5354002419>. – Дата доступа: 16.12.2008. – Название с экрана.
13. *Новейший философский словарь.* Синергетика. – Режим доступа: http://slovari.yandex.ru/dict/phil_dict/article/filo/filo-707.htm. – Дата доступа: 17.05.2011. – Название с экрана.
14. *Князева Е.Н.* Основания синергетики. Синергетическое мировидение / Е.Н. Князева, С.П. Курдюмов. – М.: КомКнига, 2005. – 240 с.
15. *Капитонова Ю.В.* Парадигмы и идеи академика В.М. Глушкова / Ю.В. Капитонова, А.А. Летичевский. – Киев, Наукова думка, 2003. – 456 с.
16. *Искусственный интеллект: В 3 кн. Кн. 1.* Системы общения и экспертные системы: [Справочник / Научн. ред. Э.В. Попов]. – М.: Радио и связь, 1990. – 464 с.
17. *Искусственный интеллект: В 3 кн. Кн. 3.* Программные и аппаратные средства: Справочник / [Научн. ред. В.Н. Захаров, В.Ф. Хорошевский]. – М.: Радио и связь, 1990. – 368 с.
18. *Палагин А.В.* К решению основной задачи эмуляции / А.В. Палагин. – УСиМ, 1980. – №3. – С. 24–28.
19. *Микропроцессорные системы обработки информации* / [Палагин А.В., Денисенко Е.Л., Белицкий Р.И., Сигалов В.И.]. – Киев: Наук. Думка, 1993. – 352 с.

20. *Ракитов А.И.* Курс лекций по логике науки / А.И. Ракитов – М.: Изд-во «Высшая школа», 1971. – 176 с.
21. *Кондаков Н.И.* Логический словарь-справочник / Н.И. Кондаков – М.: Наука, 1975. – 720 с.
22. *Гаазе-Рапопорт М.Г.* Структура исследований в области искусственного интеллекта. Толковый словарь по искусственному интеллекту / М.Г. Гаазе-Рапопорт, Д.А. Поспелов. – М.: Радио и связь, 1992. – С. 5–20.
23. *Налимов В.В.* Спонтанность сознания: вероятностная теория смыслов и смысловая архитектура личности / В.В. Налимов – М.: Изд-во «Прометей» МГПИ им. Ленина, 1989. – 288 с.
24. *Хилькевич А.П.* Проблема расширения традиционной силлогистики / А.П. Хилькевич – Минск: Изд. БГУ, 1981. – 115 с.
25. *Рыков В.В.* Управление знаниями / В.В. Рыков – Режим доступа: <http://gukkurc2.narod.ru/part2.doc>. – Дата доступа: 17.05.2011. – Название с экрана.
26. *Sowa J. F.* Knowledge Representation: Logical, Philosophical, and Computational Foundations / J. F. Sowa – Brooks Cole Publishing Co., Pacific Grove, CA, ©2000. – 594 pp.
27. *Палагин О.В.* Модель категоріального рівня мовно-онтологічної картини світу / О.В. Палагин, М.Г. Петренко. – Математичні машини і системи, 2006. – №3. – С. 91–104.
28. *Палагин О.В.* Розбудова абстрактної моделі мовно-онтологічної інформаційної системи / О.В. Палагин, М.Г. Петренко. – Математичні машини і системи, 2007. – №1. – С. 42–50.
29. *Палагин О.В.* Розвиток та порівняльні характеристики логіко-онтологічних формальних теорій / О.В. Палагин, М.Г. Петренко, А.В. Михайлюк. – Математичні машини і системи, 2007. – №2. – С. 3–18.
30. *Палагин А.В.* К вопросу системно-онтологической интеграции знаний предметной области / А.В. Палагин, Н.Г. Петренко. – Математические машины и системы, 2007. – №3,4. – С. 63–75.
31. *Рубашкин В.Ш.* Проблема интерпретации в физической теории / В.Ш. Рубашкин – Сб. «Логика и методология науки». – М.: 1967. – С. 35 – 49.
32. *Соколовская Ж.П.* «Картина мира» в значениях слов / Ж.П. Соколовская – Симферополь: Таврия, 1993. – 197 с.
33. *Кузьмин Е.С.* Система онтологических категорий / Е.С. Кузьмин – Иркутск, 1958. – 183 с.
34. *Свидерский В.И.* Некоторые вопросы диалектики изменения и развития / В.И. Свидерский – М.: Мысль, 1965. – 228 с.
35. *Библер В.С.* О системе категорий диалектической логики / В.С. Библер – Сталинабад, 1958. – 157 с.
36. *Зиновьев А.А.* Основы логической теории научных знаний / А.А. Зиновьев – АН СССР. Ин-т философии. – М.: Наука, 1967. – 202 с.
37. *Апресян Ю.Д.* Идеи и методы современной структурной лингвистики / Ю.Д. Апресян – (Краткий очерк). М.: Просвещение, 1966. – 302 с.
38. *Широков В.А.* Інформаційна теорія лексикографічних систем / В.А. Широков – К.: Довіра, 1998. – 331 с.
39. *Кургаев А.Ф.* Проблемная ориентация архитектуры компьютерных систем / А.Ф. Кургаев – Киев: Сталь, 2008. – 540 с.
40. *Кургаев А.Ф.* Анализ развития идеала структуры научной теории / А.Ф. Кургаев – Кибернетика и вычислительная техника, 2003. – Вып. 139. – С. 50–63.
41. *Андон Ф.И.* Логические модели интеллектуальных информационных систем / Ф.И. Андон, Л.Е. Яшунин, В.И. Резниченко – К.: Наук. Думка, 1999. – 397 с.
42. *Люггер Дж. Ф.* Искусственный интеллект. Стратегии и методы решения сложных проблем / Дж. Ф. Люггер – Пер. с англ. – 4-е изд. – М.: Издательский дом «Вильямс», 2003. – 864 с.
43. *Палагин А.В.* К вопросу проектирования онтолого-управляемой ИС обработки ЕЯО / А.В. Палагин, Н.Г. Петренко // International Book Series “INFORMATION SCIENCE & COMPUTING”, Varna, Bulgaria. – 2008. – № 2 – PP. 160–164.

44. *Логические* методы и формы научного познания / [Павлов В.Т., Руденко К.Ф., Семенов И.С. и др.]. – К.: Вища шк., 1984. – 208 с.
45. *Искусственный интеллект*. – В 3-х кн. Кн.2. Модели и методы: [Справочник / Научн. ред. Д.А. Поспелов]. – М.: Наука, 1990. – 304 с.
46. *Гладун В.П.* Процессы формирования новых знаний / В.П. Гладун – София: СД "Педагог 6", 1994. – 192 с.
47. *Алгебро-логічний підхід до аналізу та обробки текстової інформації* / [Палагін О.В., Кривий С.Л., Петренко М.Г., Бібіков Д.С.]. – Проблеми програмування. Спеціальний випуск. – 7-а міжнарод. наук.-практ. конф. з програмування "УкрПРОГ'2010". – [Україна, Київ], 25-27 травня, 2010 р. – № 2,3. – С. 318–329.
48. *Корпусна лінгвістика* / [Широков В.А., Бугайов О.В., Грязнухіна Т.О. та ін.] – К.: Довіра, 2005. – 471 с.
49. *Палагин А.В.* Знание-ориентированные информационные системы с обработкой естественно-языковых объектов: основы методологии и архитектурно-структурная организация / А.В. Палагин, С.Л. Крытый, Н.Г. Петренко. – УсиМ, 2009. – № 3. – С. 42–55.
50. *Гаврилова Т.А.* Об одном подходе к онтологическому инжинирингу / Т.А. Гаврилова – Новости искусств. Интеллекта, 2005. – 3. – С. 25–30.
51. *Uschold M.* Ontologies: Principles, Methods and Applications / M. Uschold, M. Gruninger. Knowledge Engineering Review 11(2), 1996. – PP. 93–136.
52. *Гаврилова Т.А.* Онтологический инжиниринг / Т.А. Гаврилова. – Режим доступа: http://www.kmtec.ru/publications/library/authors/ontolog_engineering.shtml. – Дата доступа: 17.05.2011. – Название с экрана.
53. *Клини С.* Математическая логика / С. Клини – М.: Мир, 1973. – 480 с.
54. *Куратовский К.* Теория множеств / К. Куратовский, А. Мостовский. – М.: Мир, 1970. – 416 с.
55. *Френкель А.* Основания теории множеств / А. Френкель, И. Бар-Хиллел. – М.: Мир, 1966. – 638 с.
56. *Курош А.Г.* Лекции по общей алгебре / А.Г. Курош – М.: Наука, 1972. – 399 с.
57. *Мальцев А.И.* Алгебраические системы / А.И. Мальцев – М.: Наука, 1970. – 370 с.
58. *Стол Р.* Множества, логика, аксиоматические теории / Р.Стол – М.: Просвещение, 1968. – 230 с.
59. *Кривий С.Л.* Дискретна математика: Вибр. питання: Навч. посіб. для студ. вищ. навч. закл. / С.Л. Кривий – К.: Вид. дім. "Києво-Могилянська академія", 2007. – 572 с.
60. *Логический* подход к искусственному интеллекту: От классической логики к логическому программированию / [Тейз А., Грибомон П., Луи Ж. и др.]; пер. с франц. – М.: Мир, 1990. – 429 с.
61. *Ивлев Ю.В.* Логика: учебник для вузов / Ю.В. Ивлев – М.: «Логос», 1997. – 272 с.
62. *Рубашкин В.Ш.* Представление и анализ смысла в интеллектуальных информационных системах / В.Ш. Рубашкин – М.: Наука, 1989. – 191с.
63. *Логический* подход к искусственному интеллекту: От модальной логики к логике баз данных / [Тейз А., Грибомон П., Юлен Г. и др.]. – М.: Мир, 1998. – 492 с.
64. *Широков В.А.* Феноменологія лексикографічних систем / В.А. Широков. – К.: Наукова думка, 2004. – 327 с.
65. *Лингвистический* процессор для сложных информационных систем / [Апресян Ю.Д., Богуславский И.М., Иомдин Л.Л. и др.]. – М.: Наука, 1992. – 256 с.
66. *К анализу* естественно-языковых объектов / [Палагин А.В., Крытый С.Л., Величко В.Ю., Петренко Н.Г.]. – International Book Series "INFORMATION SCIENCE & COMPUTING", BOOK Intelligent Processing, Varna, Bulgaria. – 2009. – № 9. – PP. 36–43.
67. *Новые материалы: сб. научн. трудов* / – [Научн. ред. Ю.С. Карабасов]. – М.: МИСИС, 2002 – 736 с.
68. *Стяжин И.И.* Становление идей математической логики / И.И. Стяжин. – М.: Наука, 1964. – 304 с.
69. *Гуц А.К.* Математическая логика и теория алгоритмов: учебное пособие / А.К. Гуц. – Омск: Издательство «Наследие». Диалог-Сибирь, 2003. – 108 с.

70. *Новиков Ф.А.* Дискретная математика для программистов / Ф.А. Новиков. – СПб.: Питер, 2000. – 304 с.
71. *Робинсон Дж.* Логическое программирование – прошлое, настоящее и будущее / Дж.Робинсон. – В кн.: Логическое программирование. М.: Мир, 1988.
72. *Чёрч А.* Введение в математическую логику / А. Чёрч. – М.: ИЛ, 1960. – 484 с.
73. *Гаврилов А.В.* Системы искусственного интеллекта: методические указания для студентов заочной формы обучения / А.В. Гаврилов. – АВТФ. – Новосибирск: Новосибирский государственный технический университет, 2004. – 59 с.
74. *Искусственный интеллект: история развития.* – Режим доступа: <http://library.by/shpargalka/belarus/computers/001/com-019.htm>. – Дата доступа: 17.05.2011. – Название с экрана.
75. *Минский М.* Фреймы и представление знаний / М. Минский. – М.: Энергия, 1979. – 150 с.
76. NCITS T2 (1998) Conceptual Graphs, A Presentation Language for Knowledge in Conceptual Schemas, Working draft of proposed American national standard, Document No. X3T2/96-008.
77. NCITS T2 (1998) Knowledge Interchange Format, Working draft of proposed American national standard, document – Режим доступа: <http://logic.stanford.edu/kif/dpans.html>). – Дата доступа: 11.03.2007. – Название с экрана.
78. *Режим доступа:* <http://www-ksl.stanford.edu/htw/dme/thermal-kb-tour/kif-ontology.lisp.html>. – Дата доступа: 12.11.2010.
79. *Логика.* – Режим доступа: <http://ru.wikipedia.org/wiki/логика.htm>. – Дата доступа: 09.04.2009. – Название с экрана.
80. *Теория выбора и принятия решений* / [Макаров И.М., Виноградская Т.М., Рубчинский А.А., Соколов В.Б.]. – М.: Наука, 1982. – 328 с.
81. *Братко И.* Программирование на языке Пролог для искусственного интеллекта: пер. с англ. / И. Братко. – М.: Мир, 1990. – 560 с.
82. *Палагин А.В.* Концептуальные графы и семантические сети в системах обработки естественно-языковой информации / Палагин А.В., Кривый С.Л., Петренко Н.Г. – Математические машины и системы, 2009. – №3. – С. 67–79.
83. *Sowa J. F.* Conceptual Graphs as a universal knowledge representation / Sowa J. F. – In: Semantic Networks in Artificial Intelligence, Spec. Issue of An International Journal Computers & Mathematics with Applications. (Ed. F. Lehmann), Vol.23, Number 2–5, 1992. – Part 1, –PP. 75–95.
84. *Леонтьева Н.Н.* Семантический словарь РУСЛАН как инструментарий компьютерного понимания / Н. Леонтьева, С. Семенова – М.: МГТИИ, 2003. – С. 41–46.
85. *Леонтьева Н.Н.* К теории автоматического понимания естественных текстов. Часть 1 // Моделирование системы "мягкого понимания" текста: информационно-лингвистическая модель / Н.Н. Леонтьева – М., МГУ, 2000. – 43 с.
86. *Леонтьева Н.Н.* К теории автоматического понимания естественных текстов. Часть 2 // Семантические словари: состав, структура, методика создания / Н.Н. Леонтьева – М.: МГУ, 2001. – 41 с.
87. *Толковый словарь по искусственному интеллекту* / [авт.-сост. Аверкин А.Н., Гаазе-Рапопорт М.Г., Поспелов Д.А.] – М.: Радио и связь, 1992. – 256 с.
88. *Object-oriented modeling and design* / J. Rumbaugh, M. Blaha, W. Premerlani et al. // Englewood Cliffs, 1991. – New Jersey: Prentice Hall. – 180 p.
89. *Natalya F. Noy.* Ontology Development 101: A Guide to Creating Your First Ontology' / F. Noy Natalya, L. McGuinness Deborah // Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March, 2001. – Режим доступа: http://protege.stanford.edu/publications/ontology_development/ontology101.html. – Дата доступа: 22.02.2006. – Название с экрана.
90. *Guarino N.* Formal Ontology and Information Systems / N. Guarino In N. Guarino (ed.) Formal Ontology and Information Systems / FOIS'98, 6–8 June 1998, Trento, Italy: – IOS Press, Amsterdam, 1998. – PP. 3–15.

91. *Онтологии и тезаурусы: модели, инструменты, приложения* / [Добров Б.В., Иванов В.В., Лукашевич Н.В., Соловьев В.Д.]. – Интернет-университет информационных технологий – ИНТУИТ.ру, БИНОМ. Лаборатория знаний, 2009 – 176 с.
92. *Методы и средства автоматизированного проектирования прикладной онтологии* / [Добров Б.В., Лукашевич Н.В., Невзорова О.А., Федун Б.Е.]. – Изв. РАН. Теория и системы управления. М., 2004. – № 2. – С. 58–68.
93. *Невзорова О.А.* Онтолингвистические системы: методологические основы построения / О.А. Невзорова. – Научная сессия МИФИ-2007: сб. научн. трудов. Интеллектуальные системы и технологии. – М., 2007. – Том 3. – С. 84–85.
94. *Ontologies for NLP: Introduction.* – Режим доступа: <http://crl.nmsn.edu/Research/Projects/mik-ro/htmls/asis.paper-htmls.node1.html>. – Дата доступа: 17.03.2007. – Название с экрана.
95. *The Mikrokosmos Mashine Translation Project.* – Режим доступа: <http://crl.nmsn.edu/Research/Projects/mikro/htmls/asis.paper-htmls.node4.html>. – Дата доступа: 17.03.2007. – Название с экрана.
96. *Философско-методологические основания системных исследований: системный анализ и системное моделирование: сб. научн. трудов / ВНИИ системных исследований.* – М.: Наука, 1983. – 324 с.
97. *Згуровский М.З.* Системный анализ: проблемы, методология, приложения / М.З. Згуровский, Н.Д. Панкратова. – К.: Наук. Думка, 2005. – 744 с.
98. *Методический комплекс по дисциплине “Современные проблемы науки” / Методология анализа предметных знаний.* – Режим доступа: http://old.ulstu.ru/people/SOSNIN/umk/Modern_Scientific_Problems/beloborodov/item_znan.htm. – Дата доступа: 11.10.2007. – Название с экрана.
99. *Палагін О.* Міждисциплінарні наукові дослідження: оптимізація системно-інформаційної підтримки / О. Палагін, О. Кургаєв. – Вісник НАН України, 2009. – № 3. – С. 14–25.
100. *WordNet: An Electronic Lexical Database* / Fellbaum, Christiane, ed. MIT Press, Cambridge, MA, 1998. – 422 p.
101. *Miller G. A.* "WordNet: a lexical database for English" / G. A. Miller. – Communications of the ACM 38: 11, 1995. – PP. 39–41. – Режим доступа: <http://www.cogsci.princeton.edu/~wn/>. – Дата доступа: 25.05.2007. – Название с экрана.
102. *Широков В.А.* Элементы лексикографии / В.А. Широков. – К.: Довіра, 2005. – 304 с.
103. *Режим доступа:* <http://www.isi.edu/naturallanguage/projects/ontologies.html>. – Дата доступа: 15.03.2010.
104. *Philpot A.* The Omega Ontology / A. Philpot, E. Hovy, P. Pantel. – Information Sciences Institute of University of Southern California, 2005. – 8pp. – Режим доступа: <http://omega.isi.edu/doc/>, <http://www.isi.edu/div3/div3/pubs/papers/philpot/2005omega.pdf>. – Дата доступа: 17.11.2007. – Название с экрана.
105. *OpenCyc Documentation.* – Режим доступа: <http://www.opencyc.org/doc>. – Дата доступа: 17.11.2005. – Название с экрана.
106. *WonderWeb Deliverable D18: Ontology Library (final)* / [Masolo C., Borgo S., Gangemi A., Guarino N., Oltramari A.]; Laboratory For Applied Ontology – ISTC–CNR, 2003. – 349pp. – Режим доступа: <http://www.loa-cnr.it/Papers/D18.pdf>. – Дата доступа: 27.06.2006. – Название с экрана.
107. *Giuglea A.* Knowledge Discovering using FrameNet, VerbNet and PropBank / A. Giuglea, A. Moschitti. – 2004. – 6pp. – Режим доступа: <http://olp.dfki.de/pkdd04/giuglea-final.pdf>. – Дата доступа: 17.12.2006. – Название с экрана.
108. *Kingsbury P.* PropBank: the Next Level of the TreeBank / P. Kingsbury, M. Palmer. – University of Pennsylvania, Department of Computer and Information Science – 2003. – 12pp. – Режим доступа: http://w3.msi.vxu.se/~rics/TLT2003/doc/kingsbury_palmer.pdf. – Дата доступа: 17.01.2008. – Название с экрана.

109. *Fillmore C.J.* Frame semantics and the nature of language / C.J. Fillmore. – Annals of the New York Academy of Sciences, Volume 280, 1976. – PP. 20–32.
110. *Baker C.F.* The Berkeley FrameNet Project / C.F. Baker, C.J. Fillmore, J.B. Lowe //COLING–ACL, Montreal, Canada, 1998. – бр. – Режим доступа: <http://framenet.icsi.berkeley.edu/~framenet/papers/acl98.pdf>. – Дата доступа: 11.05.2006. – Название с экрана.
111. *FrameNet II* FrameGrapher. – Режим доступа: <http://framenet.icsi.berkeley.edu/FrameGrapher>. – Дата доступа: 17.05.2011. – Название с экрана.
112. *Standart* Upper Ontology Knowledge Interchange Format. – Режим доступа: <http://suo.ieee.org/suo-kif.html>. – Дата доступа: 17.05.2011. – Название с экрана.
113. *The Mikrokosmos* Machine Translation Project. – Режим доступа: <http://crl.nmsu.edu/Research/Projects/mikro/htmls/asis.paper-htmls/node1.html>. – Дата доступа: 17.05.2011. – Название с экрана.
114. *The Mikrokosmos* Ontology. – Режим доступа: <http://crl.nmsu.edu/Research/Projects/mikro/htmls/asis.paper-htmls/node4.html>. – Дата доступа: 17.03.2007. – Название с экрана.
115. *Маторин С.И.* Системологическое исследование структуры системы категорий / Маторин С.И. – НТИ. Сер.2, 1997. – №3. – С. 3–7.
116. *Палагін О.В.* Архітектурно-онтологічні принципи розбудови інтелектуальних інформаційних систем / О.В. Палагін, М.Г. Петренко. – Математичні машини і системи, 2006. – №4. – С.15–20.
117. *Экспертные системы для персональных компьютеров: методы, средства, реализации:* Справочное пособие / [Крисевич В.С., Кузьмич Л.А., Шиф А.М. и др.]; – Минск: Выш. шк., 1990. – 197 с.
118. *Guarino N.* Some Ontological Principles for Disigning Upper Level Lexical Resources / N. Guarino. –First International Conference on Language Resources and Evaluation. – Granada, Spain, 1998. – 28–30 May. – 8pp. – Режим доступа: <http://www.ladseb.pd.cnr.it/infor/ontology/ontology.html>. – Дата доступа: 17.05.2006. – Название с экрана.
119. *Niles I.* Linking Lexicons and Ontologies: Mapping WordNet to the suggested Upper Merged Ontology / I. Niles, A. Pease. – In proceedings of the 2003 International Conference on Information and Knowledge Engineering (IKE2003). – Las-Vegas, Nevada. – 2003, June 23–26. – 6pp. – Режим доступа: <http://home.earthlink.net/~adampease/professional/Niles-IKE.pdf>. – Дата доступа: 17.05.2008. – Название с экрана.
120. *Scheffczyk I.* Linking FrameNet to the Suggested Upper Merged Ontology / I. Scheffczyk, A. Pease, M. Ellsworth. – 2006 – 9pp. – Режим доступа: <http://adampease.org/Articulate/publications/FOIS2006.pdf>. – Дата доступа: 27.02.2009. – Название с экрана.
121. *Chow I.C.* Integration of Linguas Resources for Verb Classification: FrameNet, WordNet and SUMO / I.C. Chow, J.J. Webster //Fifth Mexican International Conference on Artificial Intelligence (MICAI'06). – 2006. – PP. 262–268. – Режим доступа: <http://csdl2.computer.org/persagen/DLAbsToc.jsp?resourcePth=/dl/proceedings/micai/&toc=comp/proceedings/micai/2006/2722/00/272>. – Дата доступа: 17.11.2008. – Название с экрана.
122. *Марка Д. А.* Методология структурного анализа и проектирования / Д.А. Марка, К. МакГоуэн. – М.: "МетаТехнология", 1993. – 239 с.
123. *Методология функционального моделирования IDEF0.* РД IDEF0 – 2000 // Госстандарт России. – Москва. – 2000. – 75 с. – Режим доступа: www.nsu.ru/smk/files/idef.pdf. – Дата доступа: 17.05.2006. – Название с экрана.
124. *IDEF5* Method Report. – Armstrong Laboratory AL/HRGA Wright-Patterson Air Force Base, Ohio 45433. – 187p. – Режим доступа: www.idef.com/pdf/Idef5.pdf. – Дата доступа: 17.05.2008. – Название с экрана.

125. *Gomez-Perez A.* Ontological engineering: With Examples from the Areas of Knowledge Management, E-commerce and the Semantic Web / A. Gomez-Perez, O. M. Fernandez-Lopez, O Corcho (2004). – Springer, 2004. – 403 с.
126. *Клещев А.С.* Отношения между онтологиями предметных областей. Ч.1. / А. Клещев, И. Артемьева. – Информационный анализ, Выпуск 1, 2002. – С. 4–9.
127. *Андреев А.М.* Особенности проектирования модели и онтологии предметной области для поиска противоречий в правовых электронных библиотеках / А.М. Андреев, Д.В. Березкин, К.В. Симаков. – Режим доступа: www.inteltec.ru/publish/articles/textan/RCDL2004.shtml. – Дата доступа: 12.12.2006. – Название с экрана.
128. *Войшвилло Е.К.* Понятие / Е.К. Войшвилло. – М.: МГУ, 1967. – 286 с.
129. *SUO*, (2001), The IEEE Standard Upper Ontology web site – Режим доступа: <http://suo.ieee.org>. – Дата доступа: 17.05.2006. – Название с экрана.
130. *Гуревич И.Б.* Тезаурус и онтология предметной области “Анализ изображений” / И.Б. Гуревич, Ю.О. Трусова // Всероссийская конф. с междунар. участием “Знания – Онтологии – Теории” (ЗОНТ–09). – Новосибирск: Институт математики им. С.Л. Соболева СО РАН, 2009. – 10 с. – Режим доступа: <http://www.math.nsc.ru/conference/zont09/reports/95Gurevich-Trusova.pdf>. – Дата доступа: 17.12.2010. – Название с экрана.
131. *Martins A.F.* Models for Representing Task Ontologies / A.F. Martins, R.A. Falbo. – Режим доступа: http://nemo.inf.ufes.br/files/models_for_representing_task_ontologies_2008.pdf. – Дата доступа: 17.02.2010. – Название с экрана.
132. *Tarski A.* The semantic conception of truth. *Philosophy and phenomenological Research* / Tarski A., 1944. – V.4. – pp. 241–375.
133. *Tarski A.* *Logique, Semantique and Metamathematique (1923-1944)* / Tarski A. – Colin. – Paris, 1972. – pp. 158–269.
134. *Devidson D.* *Proceedings of Philosophical Logic* / D. Devidson. – Reidel. – Dordrecht. – 1969.
135. *Montague R.* *Universal grammars. Theoria. Formal Phylisophy: Selected Papers of R. Montague* / Montague R., – Yale University Press, 1974. – Vol. 36. – pp. 222–246.
136. *Демьянков В.З.* Основы теории интерпретации и её приложения в вычислительной лингвистике / В.З. Демьянков. – М.: Изд.-во Моск. ун-та, 1985. – 76 с.
137. *Тестелец Я.Г.* Введение в общий синтаксис / Я.Г. Тестелец. – М., РГГУ, 2001. – 798 с.
138. *Штерн І.Б.* Вибрані топіки та лексикон сучасної лінгвістики. Енциклопедичний словник / І.Б. Штерн. – Київ: АртЕк, 1998. – 336 с.
139. *Замаруева И.В.* Об одном подходе к компьютерному моделированию процесса понимания естественно-языковых текстов / И.В. Замаруева // VI Междунар. конф. “ЗНАНИЕ–ДиАЛОГ–РЕШЕНИЕ”, KDS-97, 15–20 сент. 1997г., Крым, Украина: материалы конф. – Ялта – С. 241–248.
140. *Гладун В.П.* Конспектирование естественно-языковых текстов / В.П. Гладун, В.Ю. Величко // XI-th International Conference “Knowledge-Dialogue-Solution”(KDS’2005). – Varna, Bulgaria, 2005. – vol.2 – pp.344–347.
141. *Гладун В.П.* Планирование решений / Гладун В.П. – Киев: Наукова думка, 1987. – 168с.
142. *Севрук О.О.* Знання-орієнтована пошукова система на основі мовно-онтологічної картини світу / О.О. Севрук, М.Г. Петренко // XIII міжнар. конф. “Автоматика–2006”, 2007. – Вінниця. – С. 462–465.
143. *Петренко М. Г.* Особливості розробки знання-орієнтованого лінгвістичного процесора / Петренко М. Г. – Комп’ютерні засоби, мережі та системи, 2006. – №5. – С.18–22.
144. *Попов Э.В.* Общение с ЭВМ на естественном языке / Э.В. Попов. – М.: Наука, 1982. – 360 с.
145. *Найханова Л.В.* Основные аспекты построения онтологий верхнего уровня и предметной области / Л.В. Найханова. – Интернет–порталы: содержание и технологии, ФГУ ГНИИ ИТТ “Информатика”. – М.: Просвещение, 2005. – С. 452–479.

146. *Палагин А.В.* Реконфигурируемые вычислительные системы / А.В. Палагин, В.Н. Опанасенко. – К.: Просвіта, 2006. – 280 с.
147. *Палагин А.В.* Информационная модель обработки естественно-языковых текстов / А.В. Палагин, В.П. Гладун, Н.Г. Петренко и др. // “Knowledge-Dialogue-Solution” KDS’2007: XI-th Intern. conf. – Varna, Bulgaria, 2007. – vol.2. – PP. 525–527.
148. *Режим* доступа: <http://www.w3.org/2001/sw/BestPractices/SE/ODA/>. – Дата доступа: 17.07.2005.
149. *Поляков В.Н.* Использование технологий, ориентированных на лексическое значение, в задачах поиска и классификации. – Режим доступа: <http://virtualcjlabs.cs.msu.su/html/polyak.html>. – Дата доступа: 17.04.2008. – Название с экрана.
150. *Палагин А.В.* К проектированию онтолого-управляемой информационной системы с обработкой естественно-языковых объектов / А.В. Палагин, Н.Г. Петренко. – Математические машины и системы, 2008. – №2. – С.14-23.
151. *ЭВМ пятого поколения: Концепции, проблемы, перспективы:* [ред. Т. Мотоока; пер. с англ.; предисл. Е.П. Велихова]. – М.: Финансы и статистика, 1984. – 110 с.
152. *Капитонова Ю.В.* Математическая теория проектирования вычислительных систем / Ю.В. Капитонова, А.А. Летичевский. – М.: Наука. Гл. ред. физ.-мат. лит., 1988. – 296с.
153. *Палагин А.В.* К вопросу проектирования онтолого-управляемой ИС обработки ЕЯО / А.В.Палагин, Н.Г.Петренко // International Book Series “INFORMATION SCIENCE & COMPUTING”, Varna, Bulgaria, 2008. – № 2. – pp. 160–164.
154. *Овдей О.М.* Обзор инструментов инженерии онтологий / О.М. Овдей, Г.Ю. Проскудина. – Российский научный электронный журнал «Электронные библиотеки», 2004. – т. 7. – Вып. 4, ISSN 1562-5419. – Режим доступа: <http://www.elbib.ru/index.phtml?page=elbib/rus/journal/2004/part4/op>. – Дата доступа: 19.10.2006. – Название с экрана.
155. *Noy N.* SMART: Automated Support for Ontology Merging and Alignment / N. Noy, M. Musen. – Stanford Medical Informatics, Stanford Univ. – 1999. – 24 p.
156. *Режим* доступа: <http://ais-portal.ru/2009/03>. – Дата доступа: 17.05.2010.
157. *Software Tools for Ontology Design and Maintenance* / [Calvanese D., Cuenca B. Grau, Franconi E. и др.] – 57 p. – Режим доступа: <http://citeseer.its.psu.edu/viewdoc/download?doi=10.1.1.86.7367&rep=rep1&type=pdf>. – Дата доступа: 17.02.2011. – Название с экрана.
158. *Филатов В.А.* Разработка высокоэффективных средств создания и обработки онтологических баз знаний / Филатов В.А., Щербак С.С., Хайрова А.А. – Системи обробки інформації, випуск 8 (66), 2007. – С. 120–124. – Режим доступа: www.nbu.gov.ua/portal/natural/soi/2007_8/Filatov.pdf. – Дата доступа: 17.11.2010. – Название с экрана.
159. *Артемьева И.Л.* Интеллектуальная система, основанная на многоуровневой онтологии химии / И.Л. Артемьева, Н.В. Рештаненко. – Программные продукты и системы, 2008. – № 1. – С. 84-87.
160. *Невзорова О.А.* Инструментальная система визуального проектирования онтологий “OntoEditor+” в лингвистических приложениях / Невзорова О.А. – Вестник КГТУ им. А.Н.Туполева, 2006. – № 3. – С. 56–60.
161. *Интегральная технология разрешения омонимии в системе анализа текстовых документов “LoTA”* / Невзорова О.А., Невзоров В.Н., Зинькина Ю.В., Пяткин Н.В. // Междунар. конф. “Диалог2007”. – М.: ИПИ РАН, 2007. – С. 422–427.
162. *Noy N.* The PROMPT Suite: Interactive Tools for Ontology Merging and Mapping / N. Noy, M. Musen. – International Journal of Human-Computer Studies, Vol.59, №6, 2003. – pp. 983–1024.
163. *Режим* доступа: http://protege.stanford.edu/download/prerelease/full/InstData/Windows/VM/install_protege.exe. – Дата доступа: 19.12.2009.

164. *Про один підхід до аналізу та розуміння природномовних об'єктів* / [Палагін О.В., Світла С.Ю., Петренко М.Г., Величко В.Ю.]. – Комп'ютерні засоби, мережі та системи, 2008, №7. – С. 128–137.
165. *Петренко М.Г.* Розробка методів та засобів онтолого-лінгвістичного аналізу природномовних об'єктів / М.Г. Петренко, О.В. Палагін, В.Ю. Величко, С.Л. Кривий. – Київ: Ін-т кібернетики НАН України, 2009. –38 с. – (Препринт / НАН України, Ін-т кібернетики ім. В. М. Глушкова; 2009-2).
166. *Величко Віталій.* Автоматизированное создание тезауруса терминов предметной области для локальных поисковых систем / Віталій Величко, Павел Волошин, Светлана Світла // “Knowledge – Dialogue – Solution” International Book Series “INFORMATION SCIENCE & COMPUTING”, Supplements to the International Journal “INFORMATION TECHNOLOGIES & KNOWLEDGE” № 14. – Vol. 3, 2009 – FOI ITHEA Sofia, Bulgaria. – 2009. – pp. 24–31.
167. *Палагін А.В.* Концептуализация и проблематика онтологий // А.В. Палагін, С.П. Риппа, А.А. Саченко. – Искусственный интеллект, 2008. – №3. – С. 374–379.
168. *Знание-ориентированные* информационные системы с обработкой естественно-языковых объектов: онтологический подход / [Палагін А.В., Кривий С.Л., Петренко Н.Г., Величко В.Ю.]. – УСиМ, 2010. – № 4, № 5. – С.3–14.
169. *Грушвицкий Р.И.* Проектирование систем на микросхемах программируемой логики / Р.И. Грушвицкий, А.Х. Мурсаев, Е.П. Угрюмов. – СПб.: БХВ-Петербург. – 2002. – 608 с.
170. *Зотов В.Ю.* Проектирование цифровых устройств на основе ПЛИС фирмы Xilinx в САПР WebPACk ISE / В.Ю. Зотов. – М.: – Горячая линия – Телеком, 2003. – 624с.
171. *Зотов В.Ю.* Проектирование встраиваемых микропроцессорных систем на основе ПЛИС фирмы Xilinx / В.Ю. Зотов. – М.: – Горячая линия – Телеком, 2006. – 520с.
172. *П'ятимовний тлумачний словник з інформатики* / [авт.–уклад. Іваницький Р.В., Кияк Т.Р.]. – К., 1995. – 372 с.
173. *Комп'ютерні науки та комп'ютерна інженерія: тлумачний російсько-українсько-англійський словник.* / [авт.–уклад. Бардачов Ю.М., Костін В.О., Ходаков В.Є.]. – Х: Олді-плюс, 2006. – 1028 с.
174. *Англо-український тлумачний словник з обчислювальної техніки, інтернету і програмування* / [авт.–уклад. Пройдаков Е.М., Теплицький Л.А.]. – [2-ге вид.]. – К.: Софт-Прес, 2006. – 768 с.
175. *Энциклопедия кибернетики (в двух томах)* / – Главная редакция Украинской советской энциклопедии (отв. научн. ред. В.М. Глушков), – Киев, 1974. – Т. 1,2. – 1233 с.
176. *Знание-ориентированные* информационные системы с обработкой естественно-языковых объектов: онтологический подход / [Палагін А.В., Кривий С.Л., Петренко Н.Г., Величко В.Ю.]. – УСиМ, 2010. – № 4, 5. – С. 3–14.
177. *Кургаев А.Ф.* Особенности проектирования процессора на современной элементной базе / А.Ф. Кургаев, Н.Г. Петренко. – УСиМ, 2003. – № 5. – С.16–18.
178. *Петренко Н.Г.* Об одном подходе к переводу интерпретатора информационных структур в ПЛИС-реализацию / Н.Г. Петренко, А.А Софиюк. – УСиМ, 2003. – №6. – С. 48–56.
179. *Бухарестская декларация (общеевропейская конференция в Бухаресте по подготовке всемирной встречи на высшем уровне по вопросам информационного общества).* – Режим доступа: www.itu.int/dms_pub/itus/md/03/wsispc2/doc/S03WSISPC2DOC0005!!MSW-R.doc. – Дата доступа: 17.08.2010. – Название с экрана.
180. *Семинар «Электронное образование (e-learning) в сфере высоких технологий» в МЭСИ / Электронное издание «Наука и технологии в России», 17 ноября 2009.* – Режим доступа: <http://gusnanonet.ru/news/29170/>. – Дата доступа: 22.11.2010. – Название с экрана.
181. *How E-learning Works / L. A. Obringer.* – Режим доступа: <http://communication.howstuff-works.com/elearning.htm>. – Дата доступа: 30.05.2010. – Название с экрана.

182. *E-learning 2.0* by Stephen Downes, National Research Council of Canada October 17, 2005. – Режим доступа: <http://www.elearnmag.org/subpage.cfm?article=29-1§ion=articles>. – Дата доступа: 17.05.2008. – Название с экрана.
183. Федорук П.І. Адаптивна система дистанційного навчання та контролю знань на базі інтелектуальних Інтернет-технологій / П.І. Федорук. – Ів.-Фр.: – Прикарп. нац. ун.-т, 2008. – 326 с.
184. *Moodle*. – Режим доступа: <http://www.moodle.org>. – Дата доступа: 26.05.2011. – Название с экрана.
185. *Ontologies and the Semantic Web for E-learning* / Sampson, D. G., Lytras, M. D., Wagner, G., & Diaz, P. *Educational Technology & Society*, 7 (4). – 2004. – pp. 26–28. – Режим доступа: <http://www.miltiadislytras.net/8.pdf>. – Дата доступа: 21.03.2006. – Название с экрана.
186. *К вопросу автоматизированного построения онтологии предметной дисциплины для электронных курсов обучения II* / [Палагин А.В., Петренко Н.Г., Тихонов Ю.Л., Величко В.Ю.]. – Вісник східноукраїнського університету ім. В. Даля. – 2010. – №4 (150). – С. 171–178.
187. Палагин А.В. Об одном подходе к формализованному представлению онтологии текстового документа / А.В. Палагин, Н.Г. Петренко, Севрук А.О. – Комп'ютерні засоби, мережі та системи, 2007. – №6. – С. 14–20.

ОПИСАНИЕ ОНТОЛОГИЙ *MIKROKOSMOS* И *SUMO*

А.1. Описание онтологии *MIKROKOSMOS*

Онтология *Mikrokosmos* разработана в Computing Research Laboratory (CRL) University of New Mexico State, ориентирована (в первую очередь) на лексику испанского языка и содержит лексикон из 7 тыс. слов, поддержанных в онтологии более чем 5 тыс. концептов. Многими исследователями рассматривается как онтология верхнего уровня (ОВУ).

Она определяется как модель мира и основные знания о мире, иерархически структурированные как направленный онтограф или дерево со многими связями. Все знания разделены на две (внутри связанные) базы знаний. Первая БЗ относится к онтологии и содержит знания о концептах. Вторая база знаний, названная ономастиком, идентифицирует экземпляры тех концептов, которые включены в тело общих, но доменно-зависимых знаний.

На самом вершине онтологии категории разделены на *Object* (Объект), *Event* (Событие) и *Property* (Свойство) и организованы в простую таксономию (*IS-A* иерархия). В реальной онтологии каждый узел может иметь и другие связи, структурирующие его с другими узлами в более сложную онтологию, например *IS-PART-OF*, *IS-AN-OCCUPANT-OF*, *MANUFACTURED-BY* и др. Эти связи (отношения) представлены как слоты на концептах, или графически, как помеченные связи между концептами. Например, концепт *EAT* может иметь слоты, такие как *AGENT* и *THEME* (описывающие того, кто ест и съеденные продукты).

В определениях отношений существуют ограничения на размещение концептов в слотах для конкретных прикладных областей; эти ограничения также являются концептами из онтологии. Ограничения на разрешенные заполнители различных слотов называются семантическими фасетами слота, в то время как заполнители сами по себе являются фасетами значения. Аксиоматика онтологии представлена 36 аксиомами, записанными формулами исчисления предикатов первого порядка, определяющими правильность и непротиворечивость элементов онтологии. На рисунке А.1 а) – в) приведены несколько уровней ОВУ *Mikrokosmos*.

А.2. Описание онтологии *SUMO*

Начиная с 2000 года, исследователи из различных областей знаний, таких как искусственный интеллект, информатика, лингвистика, библиотечное дело и др. объединились под эгидой IEEE комитета в рабочую группу по созданию стандарта онтологии верхнего уровня (SUO WG). Под онтологией они понимают формальную, эксплицитно заданную спецификацию совместно используемой концептуализации. Она является абстрактной моделью некоторой совокупности явлений в мире, эксплицитно представленных как концепты, отношения и ограничения, которые являются машинно-читаемыми и встроенными в концептуальные знания некоторой области знаний. Позже проект был одобрен и рабочая группа получила

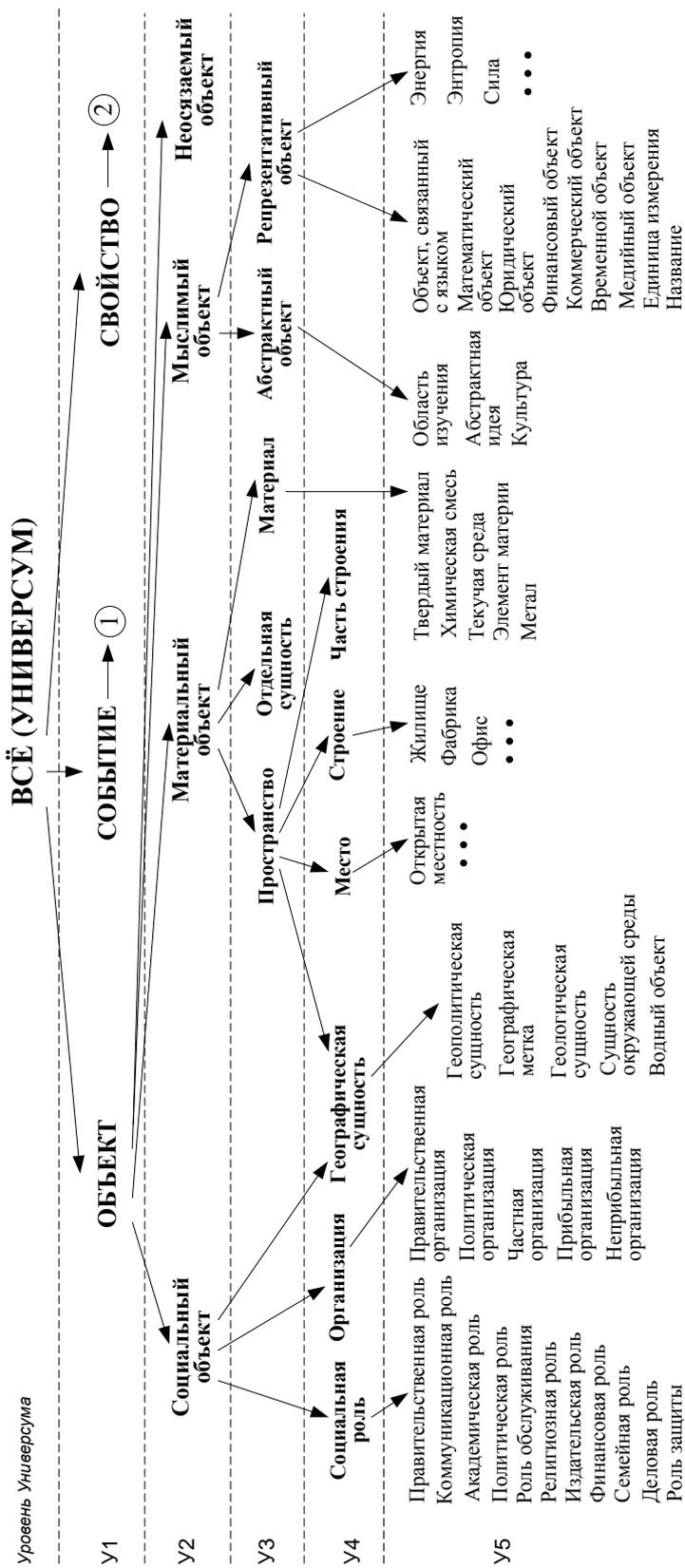


Рис. А.1 а. Смешанная онтология верхнего уровня "Mikrokosmos"

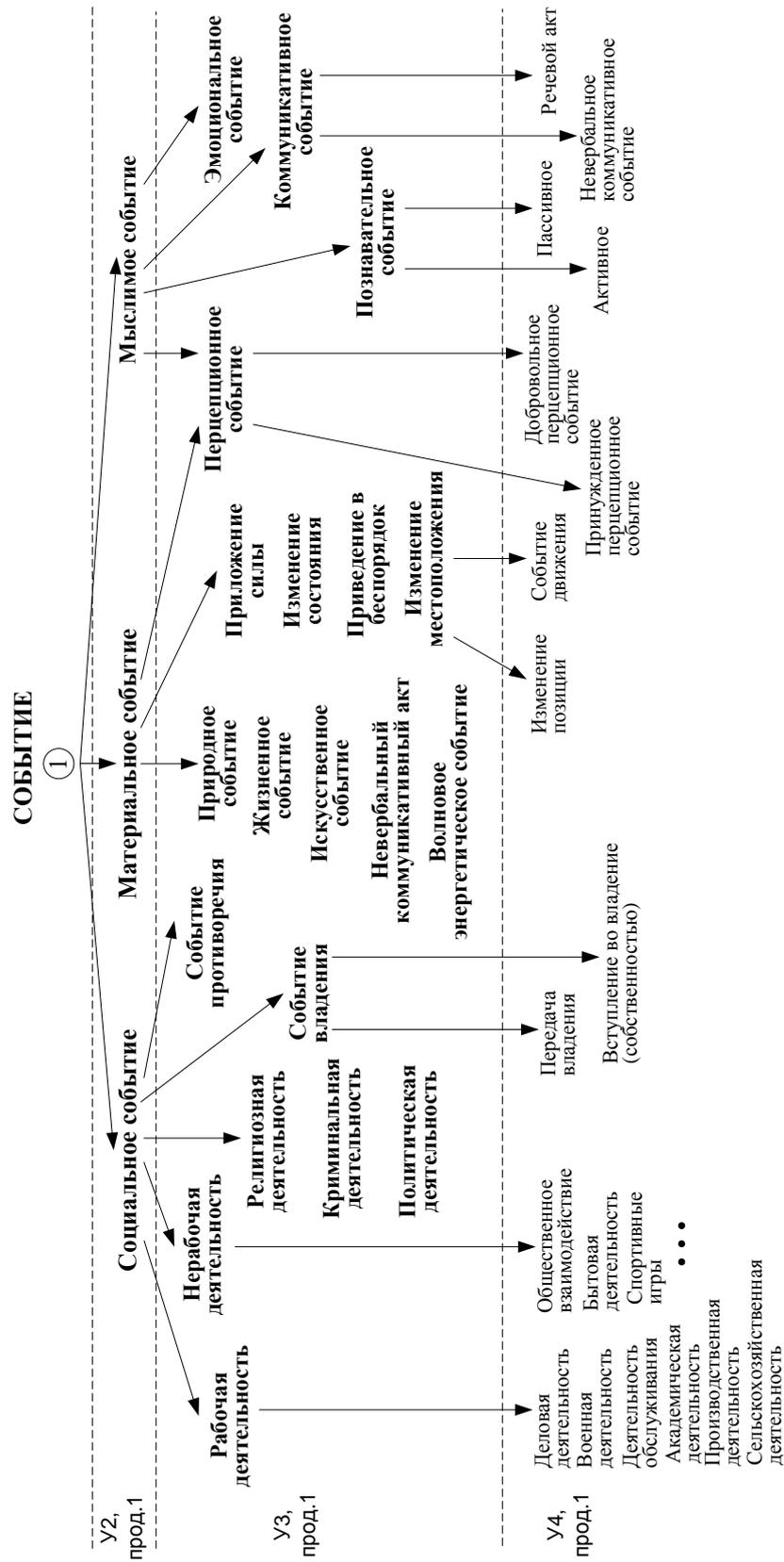


Рис. А.1 б. Смешанная онтология верхнего уровня "Mikrokosmos"

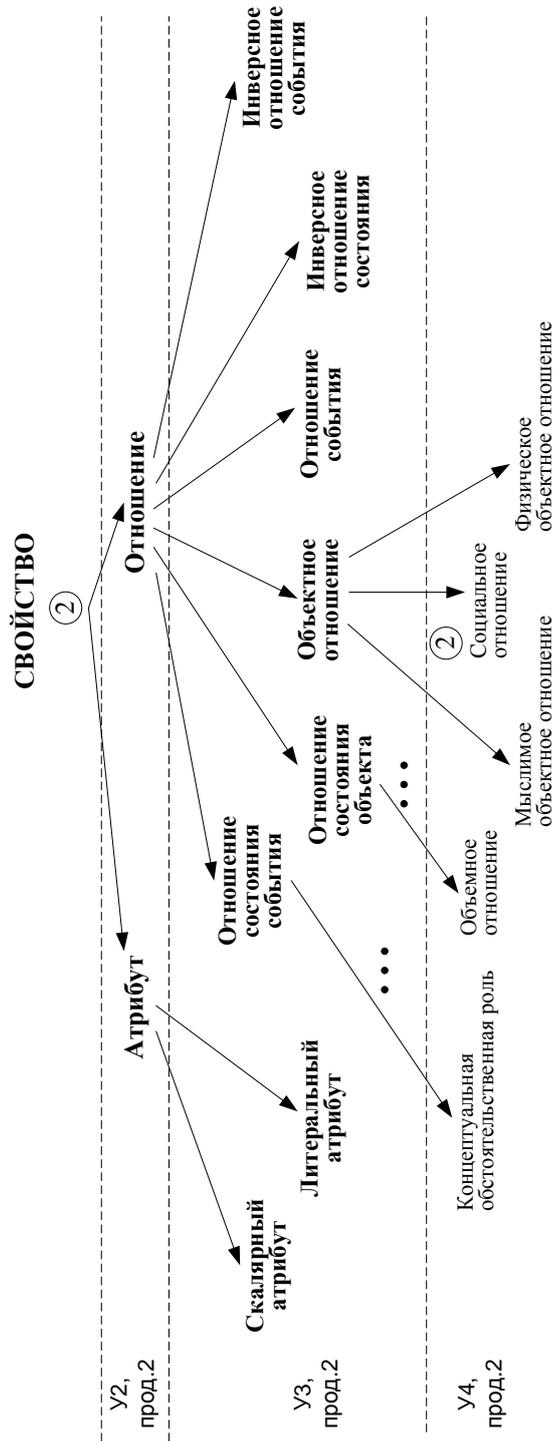


Рис. А.1 в. Смешанная онтология верхнего уровня "Mikrokosmos"

шифр P1600.1. Объем онтологии должен составлять от 1000 до 2500 терминов и около 10 аксиоматизированных утверждений для каждого термина.

Основными целями SUO являются:

- проектирование новых баз знаний и баз данных. Разработчики могут определять новые знания и новые элементы данных в терминах общей онтологии, и таким образом получить функциональную совместимость с другими аналогичными системами;
- многократное использование и интегрирование совместимых баз данных. Элементы данных из существующих систем могут только однократно быть отображены в общей онтологии;
- интегрирование проблемно-ориентированных онтологий. Такие онтологии (совместимые с SUO) смогут взаимодействовать между собой на основе совместно используемых терминов и определений (в таком смысле некоторой аналогией является итальянский проект по созданию библиотеки онтологий WonderWeb Foundational Ontologies Library и её первого модуля DOLCE).

Рабочий проект SUO был создан на базе корпорации Teknowledge Corporation и получил название Suggested Upper Merged Ontology (SUMO). SUMO объединяет общедоступные, широко известные онтологические контенты (онтологии на серверах Ontolingua и Institute of Cognitive Sciences and Technology (ICST, Italy), онтологию верхнего уровня Дж. Соуы и некоторые другие). Языком представления знаний в SUMO был выбран KIF-язык, который после доработок получил название SUO-KIF. Он предложен как самостоятельная единица стандарта. Рабочая группа с начала разработки проекта уже выпустила многочисленные версии SUMO (в настоящее время доступна версия 1.75). SUMO включает в себя две части – онтологию верхнего уровня (переработанные онтологии Дж. Соуы и Рассела-Норвига) и онтологию среднего уровня (реструктурированные остальные онтологии). После завершения создания каждой части в отдельности, они были объединены в единую концептуальную структуру. На рис. А.2 а) – б) приведено несколько уровней онтологии SUMO.

Основными концептами SUMO, как и во многих онтологиях верхнего уровня, являются “Сущность” и её категории – “Материальное” и “Абстрактное”. Первая категория включает всё, что имеет положение в пространстве-времени, а вторая – всё остальное (а точнее только то, что существует в сознании). “Материальное” делится на “Объект” и “Процесс”, что соответствует подходу, реализованному в DOLCE. Непосредственно под концептом “Объект” находятся два непересекающихся понятия: “Объект_восприятия” и “Совокупность”. Первое обозначает любой объект, все части которого непосредственно или косвенно связаны друг с другом. Концепт “Объект_восприятия” разделен на два концепта: “Вещество” и “Корпускулярный_объект”. “Вещество” характеризуется тем, что все его части (вплоть до самого низкого уровня деления) имеют такие же свойства, как и целое. Такие субстанции, как вода и глина, могут быть подклассами концепта “Вещество”, так же как и поверхности и географические области. Ниже на диалекте SUO-KIF языка KIF записаны формальные аксиомы, определяющие различие между концептами “Вещество” и “Корпускулярный_объект”.

A1. (\Rightarrow)

(and

(subclass-of ?ОБЪЕКТПУРЕ Вещество)

(instance-of ?ОБЪЕКТ ?ОБЪЕКТПУРЕ)

(part-of ?PART ?ОБЪЕКТ))

(instance-of ?PART ?ОБЪЕКТПУРЕ))

A2. (equal КорпускулярныйОбъект (ComplementFn Вещество)).

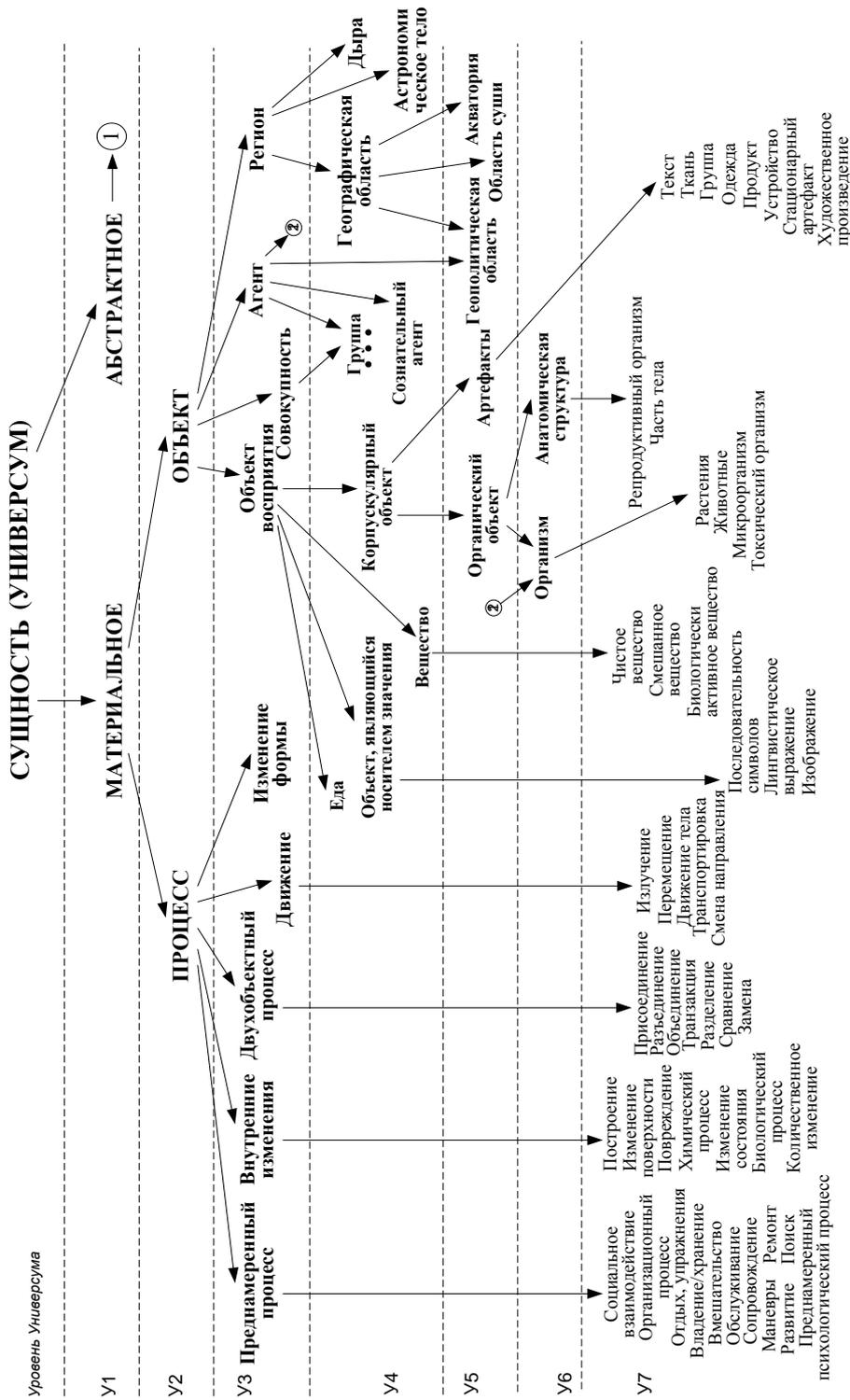


Рис.А.2 а. Онтология верхнего уровня "SUMO"

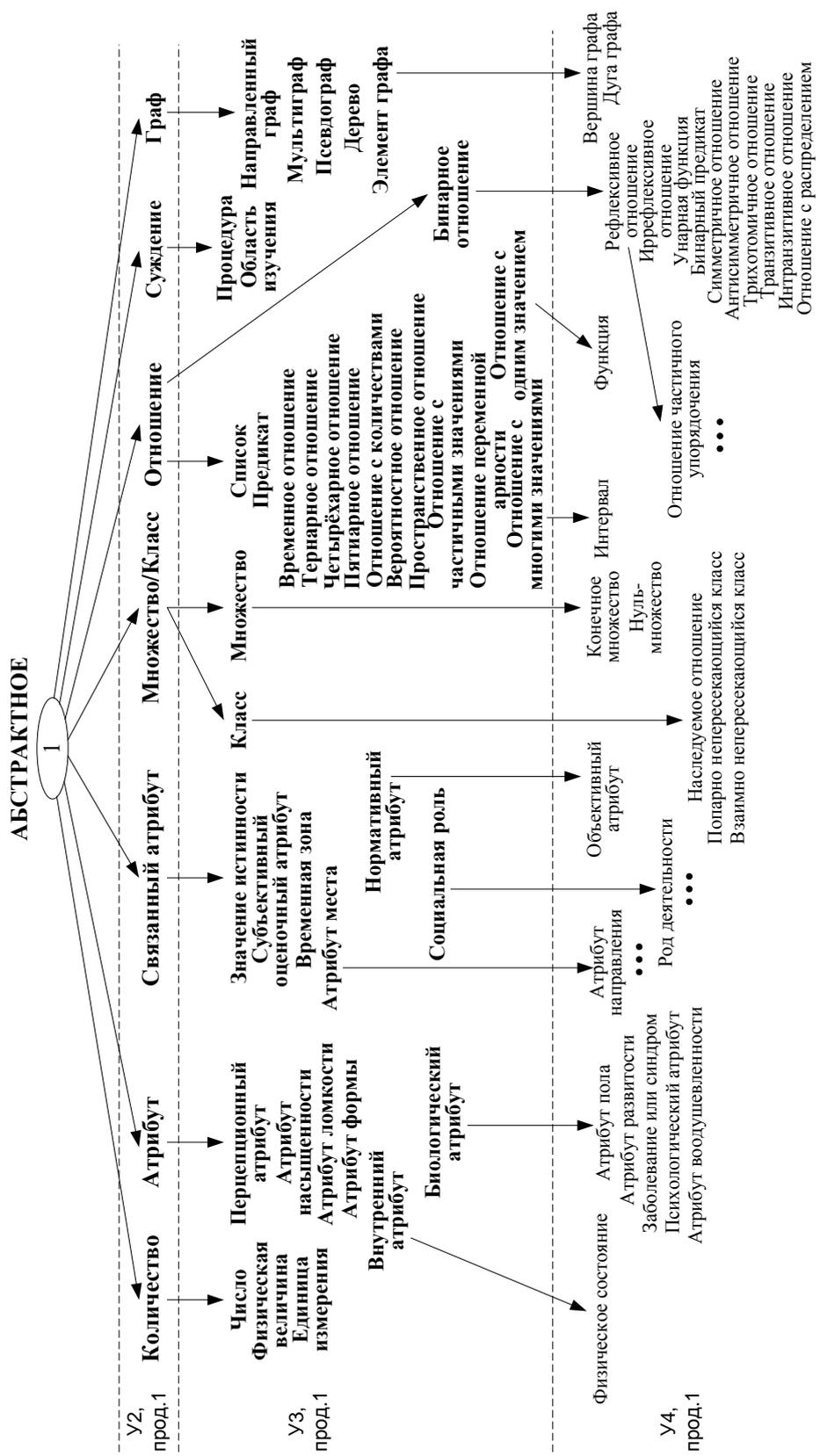


Рис. А.2.6. Онтология верхнего уровня "SUMO"

Аксиома A1 формализует утверждение “Если (PART) является частью объекта (OBJECT), являющегося, в свою очередь, экземпляром некоторого подкласса (OBJECTTYPE) класса “Вещество”, то эта часть (PART), так же как и OBJECT, является экземпляром класса OBJECTTYPE”. Аксиома A2 постулирует факт, что классы “КорпускулярныйОбъект” и “Вещество” являются взаимодополняющими.

“Совокупность” в SUMO отделена от “Вещество”. “Совокупность” строится из несвязанных частей с использованием отношения “ЧЛЕН-СОВОКУПНОСТЬ” между частями и соответствующей им совокупностью. Здесь, так же как в OpenCus, проводится разграничение понятий “Совокупность”, “Класс” и “Множество”. Предикат “быть членом совокупности” отличен от предикатов “быть экземпляром класса” и “быть элементом множества”, относящих объекты к понятиям “Класс” или “Множество”, которым они соответствуют. В отличие от “Классов” и “Множеств”, “Совокупности” занимают некоторое положение в пространстве-времени (они не абстрактны, как в OpenCus, а материальны), члены могут добавляться и удаляться из совокупности, не меняя её идентичности. Возвращаясь к концептам уровня “Материальное”–“Абстрактное”, рассмотрим ветвь “Абстрактное”. Категория “Абстрактное” разделяется на “Множество/Класс”, “Суждение”, “Количество”, “Атрибут”, “СвязанныйАтрибут”, “Математическое_Отношение” и “Граф”. “Множество” – обычное понятие теории множеств, включает понятие “Класс”, который, в свою очередь, имеет подкласс “Отношение”. Понятие “Класс” понимается как множество со свойством или пересечением свойств, которые определяют принадлежность к “Классу”, “Математическое_Отношение” есть “Класс” упорядоченных пар. “Математическое_Отношение” по смыслу ближе к “Классу”, чем к “Множеству”. “Математическое_Отношение” ограничено только теми упорядоченными парами, которые описывают его содержимое.

Концепт “Суждение” соответствует понятию семантического или информационного содержимого. Однако SUMO не накладывает никаких ограничений на это содержимое. Это более общее понятие, чем используемое в большинстве онтологий; почти невозможно принципиально разделить абстрактное содержимое, выраженное одним предложением, и абстрактное содержимое, выраженное многочисленными речевыми единицами.

Понятие “Атрибут” включает все количества, свойства и т. д., которые не представимы как “Объекты”. Например, вместо того чтобы делить класс “Животные” на “ЖивотныеЖенскогоПола” и “ЖивотныеМужскогоПола”, создаются экземпляры “Женский” и “Мужской” класса “БиологическийАтрибут”, который является подклассом “Атрибут”.

Наконец, “Количество” (или “Величина”) разделяется на “Число” и “ФизическаяВеличина” и “ЕдиницаИзмерения”. Первое понимается как независимая от системы измерения величина, второе – как составная величина, состоящая из “Числа” и конкретной единицы измерения, а в третьем понятии собраны все известные единицы измерения.

Аксиомы ограничивают интерпретацию концептов и предоставляют основу для систем автоматизированного рассуждения, которые могут обрабатывать базы знаний, соответствующие по своей структуре онтологии SUMO. Пример аксиомы: “Если C является экземпляром процесса горения, то существуют выделение тепла H и излучение света L такие, что оба они – H и L – являются подпроцессами C ”. Более сложные предложения говорят, что процессы выделения тепла и излучения света сопутствуют каждому процессу горения. Аксиомы кодируются в SUMO на формальном логическом языке SUO-KIF.

Помимо онтологии верхнего и среднего уровней разработчики SUMO предоставляют пользователям доменно-зависимые онтологии, такие как Communications, Countries and Regions, distributed computing, Economy, Finance, engineering component, Geography, Government, People, Transportations, World Airports и другие.

МЕТОДИКА ПРОЕКТИРОВАНИЯ ОНТОЛОГИИ ПРЕДМЕТНОЙ ОБЛАСТИ

Введение

Под *компьютерной онтологией* предметной области понимается известная тройка:

$$O = \langle X, R, F \rangle,$$

где $X = \{ X_1, X_2, \dots, X_i, \dots, X_n \}$, $i = \overline{1, n}$, $n = \text{Card } X$ – конечное множество концептов (понятий) заданной ПдО;

$R = \{ R_1, R_2, \dots, R_k, \dots, R_r_m \}$, $R: X_1 \times X_2 \times \dots \times X_n$, $k = \overline{1, m}$, $m = \text{Card } R$, – конечное множество семантически значимых отношений между концептами ПдО. Они определяют тип взаимодействия между понятиями. В общем случае, отношения делят на *общезначимые* (из которых выделяют, как правило, отношения частичного порядка) и *конкретные отношения* заданной ПдО;

$F: X \times R$ – конечное множество функций интерпретации, заданных на концептах и/или отношениях. Частным случаем задания множества функций интерпретации F является глоссарий, составленный для множества понятий X . Определение понятия X_i , в общем случае, включает подмножество понятий $\{x_{i-1}\}$, через которые определяется X_i ; отношение R_k , связывающее X_i с $\{x_{i-1}\}$; и множество атрибутов (признаков), присущих X_i .

Компьютерная онтология является (формальным) выражением концептуальных знаний о предметной области и по своей значимости сопоставима с базой знаний интеллектуальной информационной системы, а её построение является специфической формой человеческого мышления. Оно (мышление) в процессе познания оперирует, в том числе, суждениями, утверждениями, понятиями и отношениями между ними. А последние являются фундаментом, основой для построения составной части научной теории – онтологической базы знаний в заданной предметной области. При этом такие знания описываются в декларативной форме.

В простом случае *методика проектирования* онтологии ПдО включает три этапа проектирования.

1. Предварительный анализ заданной ПдО.

2. Построение (вручную) онтографа ПдО. Причем, под онтографом понимается двудольный граф, вершинами которого являются понятия ПдО, а дугами – связи между ними. Двудольный граф – это однонаправленный ориентированный граф, в одну вершину которого может входить и выходить несколько дуг.

3. Графическое (визуальное) проектирование онтографа ПдО и составление формализованного описания онтологии ПдО.

Анализ предметной области

Этап (системного) анализа предметной области заключается в:

– составлении систематизированного представления знаний о ПдО, понимании сути происходящих в ней процессов, правил и ограничений;

– многократном абстрагировании при описании и спецификации знаний заданной ПдО, в результате которого из всего многообразия характеристик и свойств понятий предметной области выделяются наиболее существенные, релевантные множеству конкретных задач пользователей;

– составлении и документировании глоссария терминов (понятий) ПдО. В случае отсутствия в глоссарии определения для некоторого понятия (т. е. отсутствия *F*) разработчик онтологии ПдО сам определяет функцию интерпретации для этого понятия в соответствии со своими профессиональными знаниями.

Основными принципами при системном анализе ПдО, в частности, являются: принцип конечной цели; принцип единства; принцип связности; принцип иерархии; принцип развития (интеграция с другими фрагментами онтологии ПдО или «родственными» онтологиями). На этом этапе также формируются требования к разрабатываемой онтологии ПдО.

Построение компонент онтологического описания ПдО

Напомним некоторые известные определения, непосредственно относящиеся к построению множеств концептуальной модели ПдО или её онтологии.

Понятие есть целостная совокупность суждений, в которых что-либо утверждается об отличительных признаках исследуемой сущности, ядром которой являются суждения (или утверждения) о наиболее общих и в то же время существенных признаках этой сущности.

Каждое понятие характеризуется объёмом и содержанием. Объём и содержание понятия – две взаимосвязанные стороны понятия. **Объём** – класс обобщённых в понятии предметов, **содержание** – совокупность (обычно существенных) признаков, по которым произведено обобщение и выделение предметов в данном понятии.

Каждому понятию соответствует одно или несколько имен.

Все понятия (или концепты) делятся на ряд классов (по семантической зависимости):

- В зависимости от отображения вида или рода предметов – на *видовые* и *родовые* понятия.
- В зависимости от отображения части или целого предметов – на *понятия-части* и *понятия-целые*.
- В зависимости от количества отображаемых предметов – на *единичные* и *общие* понятия.
- В зависимости от отображения предмета или свойства, абстрагированного от предмета, – на *конкретные* понятия и *абстрактные* понятия.

Онтология ПдО – это концептуальная модель реального мира и её понятия должны отражать эту реальность.

Построение фрагмента онтологии верхнего уровня

Построение ОВУ для любой онтологии ПдО является важным этапом в общем алгоритме проектирования:

- во-первых, обычно онтологии ПдО строятся фрагментарно. Связывание в дальнейшем нескольких таких фрагментов в общую онтологию осуществляется через понятия ОВУ;

- во-вторых, понятия О ПдО, связанные с понятиями ОВУ, являются их подклассами, и поэтому, наследуют признаки понятия-класса (конечно, если они связаны между собой отношением частичного порядка), например, отношение «род-вид» – отношение частичного порядка, а отношение «целое-часть» – нет.

Проектирование ОВУ может быть выполнено следующим образом.

- Список понятий, входящих в ОВУ, может быть составлен из понятий, входящих в определения для понятий верхнего уровня О ПдО (если определение составлено согласно способу «определение понятия через понятия верхнего уровня»). Понятно, что такой список будет не полным.
- Пополнение списка осуществляется на основе профессиональных знаний разработчика О ПдО.
- Для каждого понятия, вошедшего в список понятий ОВУ, составляется глоссарий (множество определений), причём, чем больше включено в глоссарий определений для одного понятия, тем точнее может быть построен онтограф ПдО.
- Связывание дугами вершины понятий ОВУ и О ПдО. Причём, в определениях этих понятий понятия, с которыми выполняется связывание, должны быть указаны в явном виде.

Построение множества X считается наиболее важным моментом при разработке онтологии ПдО. Оно должно быть обязательно не пустым.

Для хорошо проработанных предметных областей за основу множества элементов $\{x_i\}$ может быть взято содержимое различных толковых словарей. В противном случае следует составить полный список терминов, в котором указать (причём пересечение объёмов и содержаний понятий в таком предварительном списке не существенно):

- чем является каждый термин – понятием-классом предметов или конкретным понятием;
- указать для каждого термина возможные существенные отношения с другими терминами из списка;
- описать возможные существенные свойства понятий.

Следующим шагом является уточнение и определение окончательного списка классов-понятий, имена которых будут входить в разрабатываемую онтологию и являться вершинами онтографа. Также следует принять единые правила присваивания имён понятиям и их свойствам.

Следующим шагом является упорядочивание списка понятий по некоторому типу отношения «выше-ниже» на основе профессиональных знаний разработчика О ПдО и, возможно, следует повторить некоторые фрагменты процесса анализа ПдО (с привязкой к составленному списку понятий), выполненные на предварительном этапе.

В результате должен быть получен полный список существенных для заданной ПдО (и предполагаемых приложений) понятий и их машинно-интерпретируемые формулировки.

Построение множества R также основано на результатах этапа предварительного анализа ПдО. По сути, требуется установить для каждого элемента $x_i \in X$ семантическое отношение R_k с элементом $x_j \in X$, $x_i R_k x_j$, $i, j = \overline{1, n}$, $i \neq j$, $k = \overline{1, m}$. Другими словами, необходимо построить множество дуг, связывающих вершины *направленного онтографа*. В качестве вершин онтографа выступает множество понятий ПдО. Вершиной (или вершинами) онтографа (без учета ОВУ)

является родовое понятие, которое не имеет надкласса, а нижний уровень представляют конкретные понятия, т. е. не имеющие видовых понятий в заданной ПдО.

На практике множество R сначала представляют некоторым обобщённым отношением “*выше – ниже*”. Известно несколько подходов для разработки иерархии классов: процесс *нисходящей* разработки, процесс *восходящей* разработки и *комбинированный* процесс разработки. Последний наиболее часто используется разработчиками, так как он является более естественным, сначала оперируют понятиями среднего уровня, к которым наиболее часто обращаются разработчики. Затем эти понятия обобщаются и ограничиваются.

При связывании двух и более вершин онтографа (взятых поочередно, начиная с первых записей составленного на предыдущем этапе списка) следует извлечь информацию из соответствующих определений понятий о конкретных семантических отношениях R_k . В случае отсутствия такой информации или её неполноты – отношение формируется на основе профессиональных знаний разработчика О ПдО.

В заключение данного подэтапа следует соотнести разработанные классы и их иерархии результатам предварительного анализа ПдО. В частности, уточняются зависимости для конкретных пар (x_i, x_j) . В процессе соотнесения (и построения иерархии) следует учитывать, что:

- прямые подклассы в иерархии должны располагаться на одном уровне обобщения;
- класс может быть подклассом нескольких классов, и тогда он может наследовать свойства от всех этих классов;
- если класс имеет только один прямой подкласс, то, возможно, при моделировании допущена ошибка или онтология неполная;
- если у данного класса есть более дюжины (иногда говорят о числе 7) подклассов, то, возможно, необходимы дополнительные промежуточные классы;
- в онтологии не должно быть больше классов, чем это необходимо для предполагаемых приложений.

Следует помнить также, что не существует единственно правильной иерархии классов.

Описанное построение онтографа является специальным видом классификации понятий ПдО – *онтологической классификацией*.

Построение множества F

Для данной методики построение функций интерпретации заключается в составлении глоссария терминов ПдО, которые являются вершинами онтографа ПдО. Такой глоссарий составляется на этапе предварительного анализа ПдО, а на последующих этапах уточняется и дополняется. Причём, на этапе составления онтографа ПдО – учитывается информация (из определений понятий) о понятиях и отношениях между ними, а на этапе формирования формализованного описания – информация о существенных признаках, характеризующих определяемое понятие.

Графическое (визуальное) проектирование онтографа ПдО

На основе построенных множеств кортежа выполняется синтез концептуальной модели ПдО, например, с помощью известного инструментального средства Protege (ИСП) и получить формальное описание разработанной онтологии на одном из языков описания онтологий (например, OWL), а также графическое представление онтографа.

ИСП поддерживает ручной ввод элементов множеств X и R , в результате чего на экране получим визуальное представление онтографа ПдО. Кроме того, признаки, взятые из определений понятий, заполняются в соответствующие слоты.

В заключение в ИСП можно автоматически сформировать формализованное описание O ПдО.

Блок-схема алгоритма проектирования онтологии ПдО представлена на рис. Б.1.

Ниже приведены *два примера* проектирования онтологии ПдО в соответствии с описанной выше методикой.

В первом примере выполнено построение фрагмента онтологии из ПдО “Вычислительная техника”. Из словарей по информатике и вычислительной технике выбраны следующие понятия: “Оперативная память”, “Вычислительная машина”, “Устройство ввода-вывода”, “Аналоговая вычислительная машина”, “Микропроцессор”, “Центральный процессор”, “Цифровая вычислительная машина”, “Центральный процессор на основе микропроцессора фирмы AMD”, “Информационные шины”, “Управляющие шины”, “Центральный процессор на основе микропроцессора фирмы Intel”, “Теоретический базис”, “Архитектура вычислительных систем”, “Программирование”, “Компьютерные сети”, “Проектирование средств вычислительной техники”, “Теория автоматов”, “Разработчик средств вычислительной техники”, “Software (программное обеспечение, ПО)”, “Hardware (аппаратные средства, АС)”, “Архитектура АС вычислительной системы”, “Единая система стандартов”, “Проектирование вычислительной системы”, “АС вычислительной системы”, “Архитектура вычислительной системы типа SISD”, “Архитектура вычислительной системы типа MISD”, “Архитектура вычислительной системы типа SIMD”, “Архитектура вычислительной системы типа MIMD”, “Архитектура вычислительной системы”, “Программное обеспечение вычислительной системы”. При этом понятием ОВУ будет понятие “Информатика” и “Вычислительная техника”.

Определения выбранных понятий приведены (в том числе) в приложении В.

Далее выполним ранжирование списка терминов по обобщённому отношению “*выше-ниже*”.

1. “Информатика”.
2. “Вычислительная техника”, “Теоретический базис”.
3. “Единая система стандартов”, “Hardware”, “Software”, “Разработчик средств ВТ”, “Архитектура вычислительных систем”, “Программирование”, “Компьютерные сети”, “Проектирование средств вычислительной техники”, “Теория автоматов”.
4. “Вычислительная система (ВС)”.
5. “Проектирование вычислительной системы”, “АС вычислительной системы”, “ПО вычислительной системы”, “Архитектура ВС”.
6. “Вычислительная машина”, “Архитектура АС вычислительной системы”.
7. “Цифровая вычислительная машина”, “Аналоговая вычислительная машина”, “Архитектура вычислительной системы типа SISD”, “Архитектура вычислительной системы типа MISD”, “Архитектура вычислительной системы типа SIMD”, “Архитектура вычислительной системы типа MIMD”.
8. “Центральный процессор”, “Устройство ввода-вывода”.
9. “Центральный процессор на основе микропроцессора фирмы AMD”, “Центральный процессор на основе микропроцессора фирмы Intel”.
10. “Оперативная память”, “Микропроцессор”.

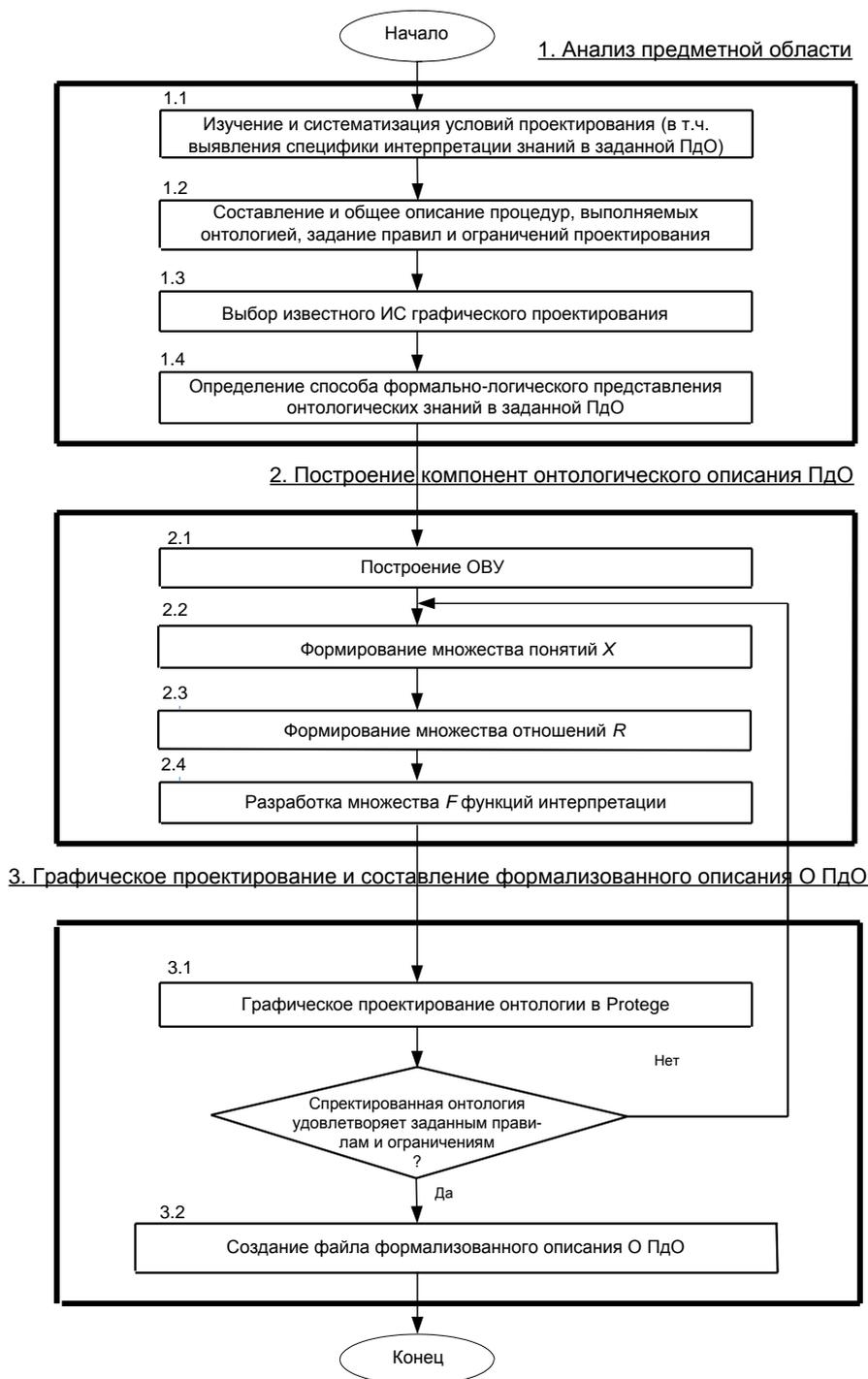


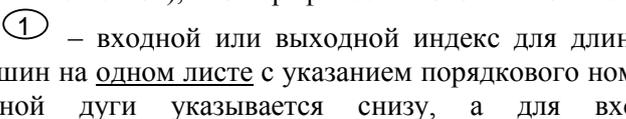
Рис. Б.1. Блок-схема алгоритма проектирования онтологии ПДО

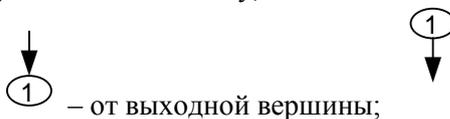
Множество отношений состоит из элементов – {категорное_отношение, участник, множество-элемент, регламентировать, быть_характеристикой, род-вид, целое-часть, разработать, содержаться в}.

На рис. Б.2 приведен онтограф фрагмента ПДО “Вычислительная техника”.

РЕКОМЕНДАЦИИ к дизайну представления структуры онтографов

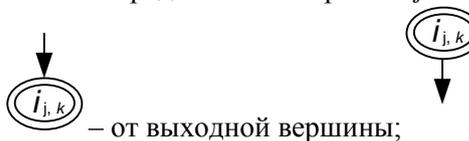
1. Онтограф ПдО имеет одну корневую вершину, денотатом которой является заданная ПдО. Как правило, корневая вершина располагается на первом уровне иерархии. Однако это не исключает наличие нулевого уровня, на котором расположены понятия домена предметных областей (онтологии верхнего уровня). Особенно полезна такая структуризация, если понятия связаны между собой отношениями частичного порядка.
2. Все вершины онтографа должны располагаться строго по уровням иерархии (вершина-предок располагается на один уровень ниже от вершины-родителя).
3. Вершины онтографа для понятий-объектов и понятий-процессов должны располагаться в противоположных сторонах листа и на одном уровне для связанных объектов и процессов.
4. Дуги, связывающие между собой две вершины, направлены от вершины-родителя к вершине-предку. Для каждой дуги в явном виде указывается (прямо или косвенно) семантическое отношение, связывающее между собой два понятия.
5. Для проектируемой онтологии ПдО все понятия (вершины онтографа) имеют различное семантическое значение. Наиболее значимые вершины в онтографе должны быть выделены. Это может быть форма вершины, шрифт и цвет имени вершины и т. п.
6. Выходящие из вершины дуги, обозначающие одно и то же отношение, представляют на онтографе пучком.
7. Предпочтительно онтограф располагать на одном листе А4. Если требуется связать между собой две вершины, расположенные в отдалённых местах листа (длинные связи), то графически это выполняется следующим образом:





8. Если онтограф не помещается на одном листе А4, то его следует разбить на несколько листов. При этом разбивка по вертикали предпочтительна, так как при этом отдельные ветви онтографа располагаются на одном листе. Графически связи между листами выполняются следующим образом:

а) $\textcircled{i_{j,k}}$ – индекс, обозначающий начало (из вершины выхода) или окончание (во входную вершину) дуги (связь между двумя вершинами), выходящей с k -го листа, j -го уровня и i -ым порядковым номером на j -ом уровне;



- б) индекс проставляется сверху от входной дуги и внизу от выходной дуги;
 в) внизу листа перечисляются все связи с вершинами на других листах.

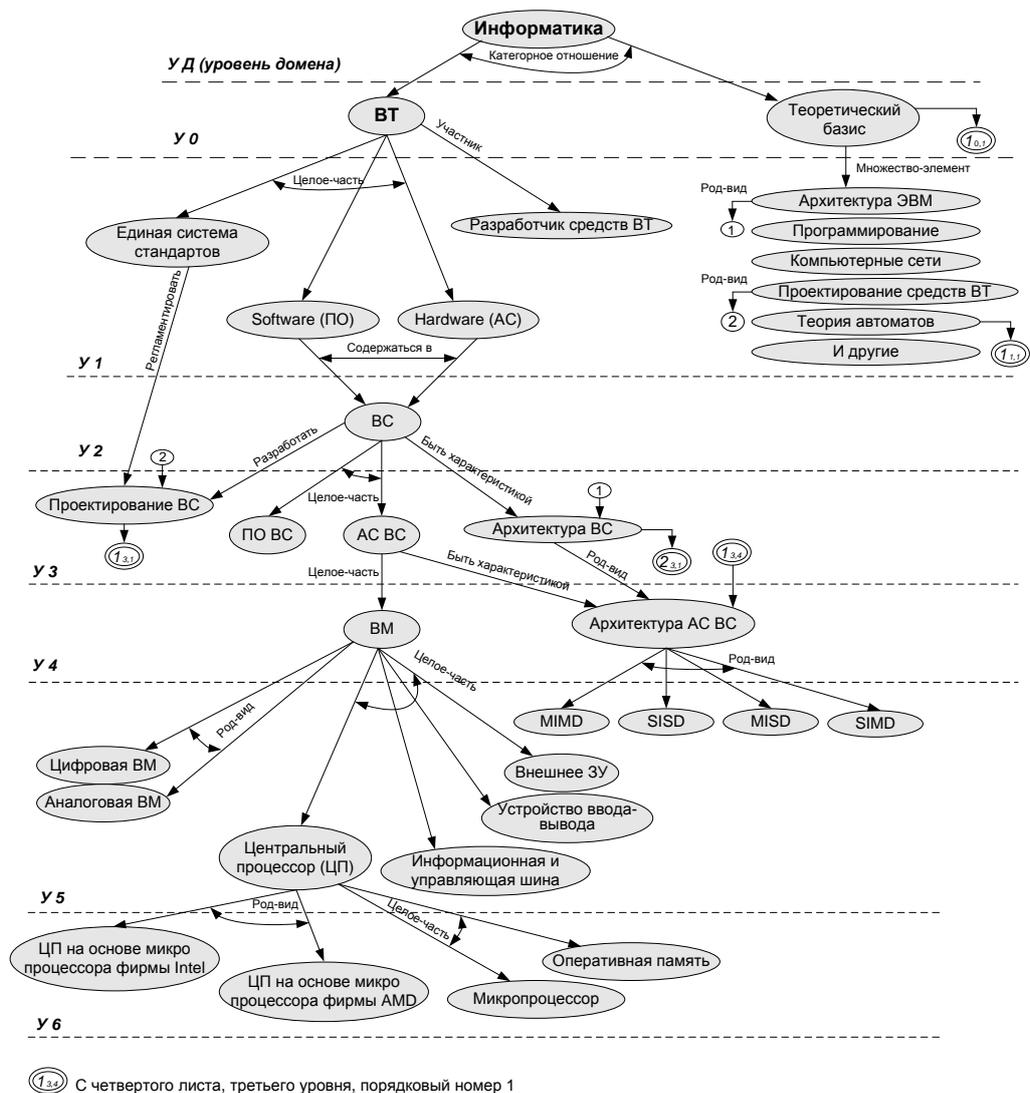


Рис. Б.2. Онтограф фрагмента ПдО “Вычислительная техника”

9. На онтографе могут быть указаны условные связи между вершинами, дающие дополнительные пояснения онтологии ПдО. Они проводятся в виде дуг с прерывистыми линиями. В явном виде указанные пояснения приводятся в описаниях соответствующих функций интерпретации.

10. При построении онтографа следует всегда помнить о минимизации его компонент.

Во втором примере выполнена разработка онтологии домена “Информатика”, онтограф которой приведен на рис. Б.3.

**КРАТКИЙ ТОЛКОВЫЙ СЛОВАРЬ
ПО ОНТОЛОГИЧЕСКОМУ ИНЖИНИРИНГУ**

Ниже представлен краткий толковый словарь, составленный по материалам библиографии (в частности энциклопедии и толковым словарям), статьям русскоязычной и англоязычной Википедии и некоторых международных стандартов, а также его онтограф (рис. В.1). Особенностью этого толкового словаря является его онтологическая направленность или онтологический характер. В (4.2) множества функций интерпретации F и аксиом A строятся на основе общеизвестных толкований понятий X , т. е. взятых из энциклопедий, толковых словарей, тезаурусов и лингвистического корпуса текстов из заданной ПдО, для которой строится компьютерная онтология. Другими словами, здесь не преследуется цель оригинальности толкований (определений) понятий, описывающих заданную ПдО, хотя это и не исключается.

Абдукция – правдоподобный *вывод* от частного к частному.

Абстракция – процесс отсечения единичного, случайного или несущественного для последующих шагов работы. Абстракция всегда имеет место при *представлении данных и знаний* о внешнем мире в *интеллектуальных системах*. Абстракция используется при *обобщении знаний*, проведении рассуждений и планировании целесообразной деятельности. Абстракция является средством образования *понятий*.

Автомат – абстрактная машина, преобразующая последовательности входных символов в последовательности выходных символов. В зависимости от числа внутренних состояний памяти автомата различаются конечные автоматы и бесконечные автоматы; в зависимости от однозначности или неоднозначности формирования выходных последовательностей – детерминированные автоматы и недетерминированные автоматы; в зависимости от особенностей структуры – магазинные автоматы, стековые автоматы и клеточные автоматы.

Автомат детерминированный – автомат, у которого в любой такт работы набор входных символов и внутреннее состояние однозначно определяет набор выходных символов и внутреннее состояние детерминированного автомата в последующем такте работы.

Автомат конечный – автомат, работа которого определяется двумя функциями:

$$y(t+1) = F_1(x(t), y(t)),$$

$$z(t) = F_2(x(t), y(t)).$$

Первая функция задаёт смену состояний автомата в дискретные такты времени t и называется функцией переходов; вторая – выходные сигналы автомата и называется функцией выхода; x , y , и z – множества двоичных векторов фиксированной длины, т. е. конечные множества.

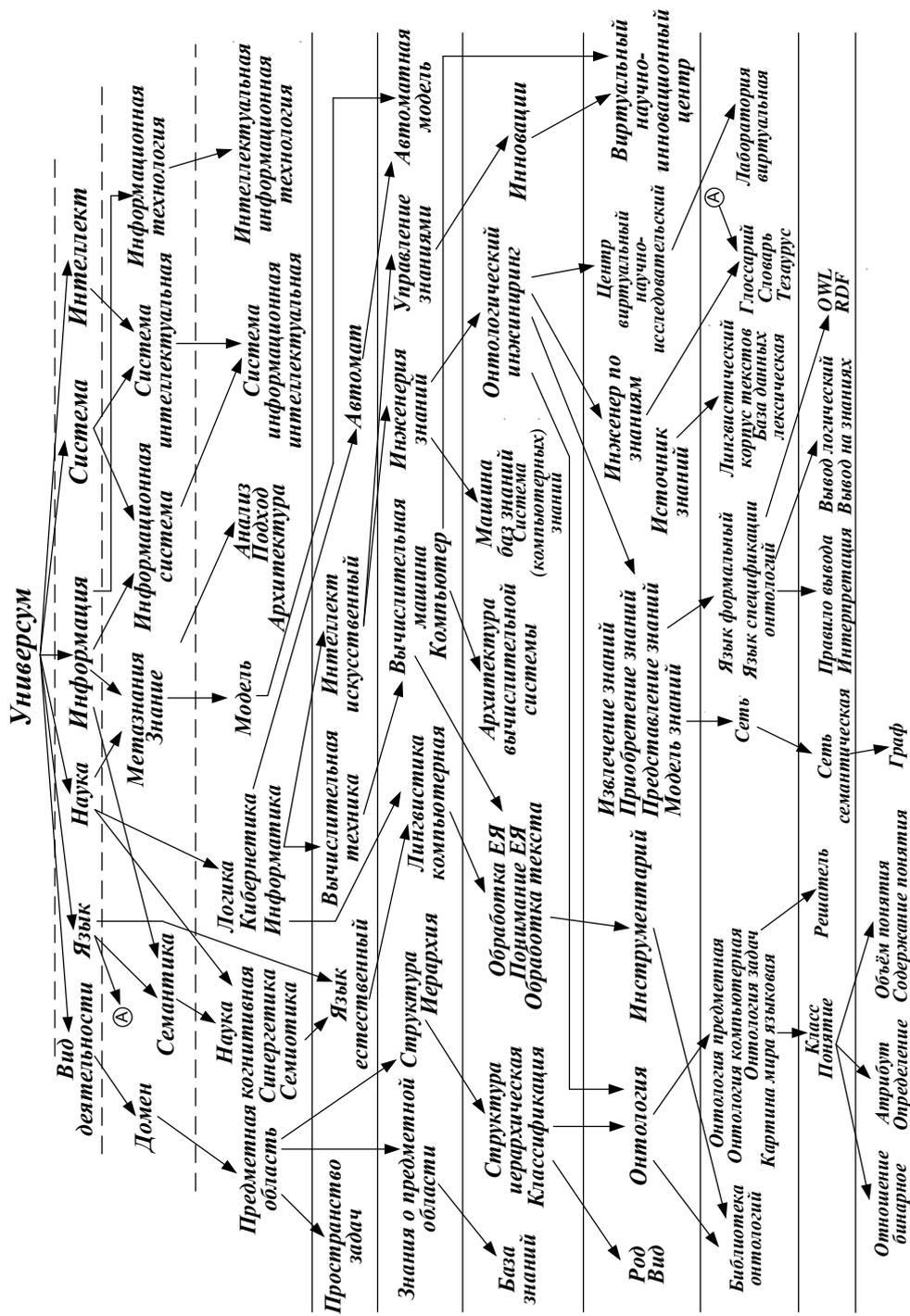


Рис. В.1.1. Онтограф избранных топиков по онтологическому инжинирингу

Автомат Мура (Ми́ли) – конечный автомат, в котором выходной символ на каждой стадии зависит только от текущего состояния и не зависит от входного символа. В противоположность ему в автомате Мили выходной сигнал зависит не только от текущего состояния, но и от входного сигнала.

Автомат недетерминированный – автомат, у которого в некоторые такты работы набор входных символов и внутреннее состояние задают альтернативный выбор набора выходных символов и / или внутреннего состояния недетерминированного автомата в последующем такте работы. Частный случай недетерминированного автомата являются вероятностный автомат и стохастический автомат.

Автомат стохастический – автомат, у которого вместо функций переходов и выходов в общем случае задаются распределения вероятностей дискретного типа. Стохастический автомат часто используется для описания процесса адаптации к среде, в которой он функционирует.

Автомат структурный – автомат, заданный множеством элементов и схемой их соединения.

Автоматная модель – описывает проект в виде *граф-схемы алгоритма* функционирования объекта проектирования.

Автоматизация – разработка и применение комплекса *методов, приборов, устройств и систем*, направленных на уменьшение роли прямого участия человека в материальных, энергетических и информационных процессах.

Автоматизация научных исследований – использование *компьютера* для выполнения научно-исследовательских работ: разработки математических моделей, проведения исследований с использованием моделей, обработки полученных данных.

Автоматизированный – термин применяется к *технологии* или *процессу*, в котором часть работы выполняет *компьютер*.

Автоматический – термин применяется к процессу или к оборудованию, которые в определенных условиях работают без вмешательства человека.

Агент – аппаратная или программная сущность, способная действовать в интересах достижения целей, поставленных перед ним владельцем и / или пользователем.

Аксиома – утверждение, которое априорно считается истинным.

Актант – термин “актант” связан с определённой синтаксической концепцией, согласно которой элементарные высказывания расчленяются на функции (субъект, объект, предикат) и предикат рассматривается как ядро высказывания.

Актор – специальным образом организованная процедура, характерная для *объектно-ориентированного* стиля *программирования*. Особенность актора состоит в том, что он самостоятельно включается в работу (активизируется), когда выполняются условия активизации. После окончания работы актор передаёт полученные им результаты другим акторам. Использование актора обеспечивает децентрализованное и максимальное параллельное решение задач.

Актуализация – процесс, обеспечивающий постоянное внесение текущих изменений в состояние системы, *базы данных* или *базы знаний*.

Анализ – способ, вид исследования при котором реальный или мыслимый объект расчленяется на составляющие части (элементы) и изучаются эти элементы и связи между ними.

Архитектура – концепция взаимосвязей элементов сложной структуры. Включает компоненты логической, физической и программной структур.

Архитектура аппаратных средств – наиболее обобщённый способ классификации архитектур аппаратных средств компьютера базируется на понятиях потока команд I и потока данных D в вычислительной структуре. При этом различают одинарный поток S и множественный поток M . Соответственно этому подходу можно определить четыре класса структур аппаратных средств ЭВМ – *SISD*, *MISD*, *SIMD*, *MIMD*.

Архитектура вычислительной системы – совокупность обобщённых сведений о структуре основных блоков вычислительной системы, их функционировании и информационно-управляющих связях между ними.

Архитектура компьютера – 1) логическая структура и связанные с ней функциональные характеристики компьютера, в том числе взаимосвязи между компонентами хардвера и софтвера; 2) совокупность обобщённых сведений о структуре основных блоков компьютера, их функционировании и информационно-управляющих связях между ними.

Атрибут (Attribute) – 1) отдельная специфическая характеристика объекта данных; 2) признаки, характеристики или свойства объекта в системе, которыми можно управлять, чтобы изменить его вид или состояние; 3) уникальное имя, приписываемое домену значений некоторой информационной единицы.

База данных лексическая – компьютерная структура для хранения и использования корпуса лексем с лексикологической информацией.

База знаний (Knowledge base) (БЗ) – 1) семантическая модель, предназначенная для представления в компьютере знаний, накопленных в определённой предметной области; 2) совокупность программных средств, обеспечивающих поиск, хранение, преобразование и запись в памяти компьютера сложно структурированных информационных единиц (знаний); логическая структура для систематизированного хранения знаний об окружающем мире, о некоторой предметной области.

База знаний замкнутая – БЗ, содержимое которой в процессе функционирования не изменяется. Логический вывод в такой базе эквивалентен выводу в формальной системе и обладает свойством монотонности, т. е. ранее выведенные утверждения остаются верными на весь период функционирования замкнутой БЗ.

База знаний интенциональная – БЗ, в которой описаны общие закономерности, характерные для некоторой проблемной области, а также способы постановки и решения задач в этой области.

База знаний открытая – БЗ, позволяющая в процессе её функционирования изменять (пополнять содержимое базы и убирать знания из базы). Свойство открытости приводит к тому, что вывод в такой базе является немонотонным, т. е. истинность выведенных в ней утверждений может меняться в процессе работы системы с такой базой.

Библиотека – особым образом организованный файл, содержащий элементы программы (процедуры, подпрограммы, макроопределения), доступные по имени и могут быть использованы с целью присоединения к некоторой программе.

Библиотека онтологий – набор *онтологий* многоразового использования, вмещающий домен (область) и задачи онтологий.

Блок-диаграмма – диаграмма системы, в которой основные части или функции представляются в виде блоков, соединённых линиями, которые изображают взаимосвязи между блоками.

Блок-схема – графическое представление *процесса* или пошагового решения проблемы путём использования соответствующим образом надписанных плоских геометрических фигур, соединённых линиями, указывающими направления потоков; предназначена для проектирования или документирования процесса или программы.

Величина – элемент *данных*, определённый либо своим именем, либо значением, либо и тем и другим.

Вид – элемент (подразделение) *классификации системы*, который является подчинённым по отношению к более общему подразделению – *роду*.

Вид деятельности – государственная, экономическая, социальная, культурная и т.д. деятельность.

Внедрение – процесс передачи разработанного проекта в производство, обучение персонала.

ВНИЦ (виртуальный научно-инновационный центр) – организационно-технологическая система, предназначенная для обеспечения эффективного управления процессами развития *научных исследований*, высоких технологий и национальной экономики в целом.

Вывод логический – в *экспертных системах* и искусственном интеллекте – процесс получения новых фактов (заключений) с помощью фактов, которые уже есть, и правил вывода.

Вывод на знаниях – вывод, использующий в качестве посылок выражения, хранящиеся в *БЗ*. Вывод на знаниях может быть *достоверным*, если эти выражения являются достоверными или правдоподобными (или снабжены оценками правдоподобия). Как правило, процедуры вывода на знаниях включают поиск необходимых *знаний* для вывода.

Вывод немонотонный – вывод, при котором нарушается свойство монотонности при выводе.

Вывод нечеткий – вывод, при котором используются *нечеткие кванторы* или значения *функций принадлежности*. При нечётких кванторах правила вывода определяют тот квантор, который следует приписать результату при заданных значениях кванторов посылок.

Высказывание (Assertion) – *логическое выражение*, относительно которого всегда можно утверждать, что оно либо истинно, либо ложно.

Высказывание атомарное – высказывание, структура которого далее не расчленима. В формальных системах атомарному высказыванию соответствуют базовые

элементы. *Интерпретация* истинности атомарного высказывания задаётся для *формальной системы* извне и на этой основе определяется интерпретация всех правильно построенных формул формальной системы.

Выражение логическое – выражение, в котором операнды – объекты, над которыми выполняются *логические операции*.

Вычислительная машина – комплекс *устройств* и *программных средств* для автоматической обработки данных.

Вычислительная техника – 1) область *науки* и техники, объединяющая средства автоматизации математических вычислений и обработки информации в различных областях человеческой деятельности; 2) дисциплина, изучающая *вычислительные машины*, принципы их построения и использования. Включает исследование таких аспектов, как: программирование, информационные структуры, *программное обеспечение*, *языки программирования*, компиляторы и операционные системы, *аппаратные средства*, их разработка и тестирование, *архитектура вычислительных систем*, сети ЭВМ и сопряжение вычислительных машин: системный анализ, теория информации, систем и вычислений, прикладная математика, вычислительные методы, методы машинной графики и приложения.

Гипотеза – предположение или система взаимосвязанных предположений о причине какого-либо явления, вероятность которых ещё не доказана.

Глоссарий (*glossary*) – толковый словарь.

Граф – конечное множество *вершин*, соединённых *рёбрами*.

Граф онтологический – двудольный граф, вершинами которого являются *понятия предметной области*, а дугами – *отношения* между ними. Двудольный граф – *односторонний ориентированный граф*, в одну вершину которого может входить и выходить несколько дуг.

Граф ориентированный – *граф*, в котором ни одна пара вершин не соединена симметричной парой дуг.

Граф связный – *граф*, в котором существует связь между любыми двумя вершинами.

Данные – 1) отдельные факты, характеризующие объекты, процессы и явления *предметной области*, а также их свойства; 2) представление информации в некотором формализованном виде, пригодном для *передачи*, *интерпретации*, *обработки* и *хранения*; 3) факты, понятия или команды, представленные в формализованном виде, что позволяет осуществлять передачу, интерпретацию, обработку и хранение.

Данные входные – *данные*, поступающие в систему обработки информации (или в любую её часть) для *хранения* или *обработки*.

Данные выходные – *данные*, которые *система обработки информации* (или любая её часть) передаёт во внешнее окружение этой системы (или части системы).

Действие – единица процесса деятельности, активность, направленная на достижение определенной цели. Действие может быть внутренним, направленным на преобразование информации внутри *интеллектуальной системы*, или внешним, направленным во внешнюю среду (сообщение пользователю, движение манипулятора, автономного робота и т. п.).

Декомпозиция задач – разбиение задачи на подзадачи с последующим разбиением этих подзадач до получения базовых (элементарных) задач, для которых заранее известно решение. Декомпозиция задач применяется в интеллектуальных системах при *анализе* сложных объектов и планировании поведения в пространстве задач. В более общем смысле декомпозиция задач позволяет определить размерность решаемой задачи.

Денотат – реальный объект, процесс, явление, ситуация и т. п. в физическом мире, для которого есть специальное именуемое выражение в некотором языке. Это именуемое выражение является десигнатом для данного денотата.

Дистанционное обучение – обучение с использованием компьютерной сети, за счёт чего достигается некоторый эффект совместной работы ученика и учителя в тех случаях, когда они территориально разобщены.

Домен – некоторая обобщённая предметная область, состоящая из родственных (видовых) предметных областей. Например, математика, информатика, медицина и т. п.

Дуга – ориентированное сообщение между двумя вершинами графа; стрелка, которая соединяет одну вершину графа с другой.

Знание (Knowledge) – 1) вид информации, которая хранится в БЗ и представляет формальное описание определённой предметной области; 2) информация о процессах их интерпретации и закономерностях применения которой к данным порождает новую информацию, закономерности предметной области (принципы, связи, законы) получают в результате профессионального опыта и позволяют ставить и решать задачи в этой области; 3) совокупность понятий и представлений об объективной действительности, их внутренне взаимосвязанных систем (суждений, положений, концепций, теорий и т. д.), выработанных обществом в процессе познания и преобразования мира; 4) организованное, интегрированное собрание фактов и обобщений, представленных в формализованном виде.

Знания декларативные – знания, которые записаны в памяти интеллектуальной системы и представляют информацию о свойствах предметной области, фактах, имеющих в ней место, и т. п. По содержанию и форме представления декларативные знания противопоставляются процедурным знаниям.

Знания о предметной области – совокупность сведений о предметной области, хранящихся в БЗ интеллектуальной системы. В знания о предметной области входят факты, относящиеся к предметной области, закономерности, характерные для неё, гипотезы о возможных связях между явлениями, процессами и фактами, процедуры для решения типовых задач в данной проблемной области. В процессе функционирования интеллектуальной системы знания о предметной области могут пополняться. Знания о предметной области используют при поиске решений задач, возникающих в экспертных и других интеллектуальных системах.

Знания о языке – комплекс знаний о порождении (синтез) и понимании (анализ) правильных языковых конструкций, противопоставляется знаниям о мире.

Знания прагматические – 1) знания о способах решения задач в заданной предметной области; 2) в ЕЯ знания о прагматической компоненте текстов.

Знания процедурные – знания, хранящиеся в памяти интеллектуальной системы в виде описаний процедур. В виде процедурных знаний обычно описываются инфор-

мация о предметной области, характеризующая способы решения задач в этой области, а также различные инструкции, методики и т. п. По форме представления процедурным знаниям противопоставляются *декларативные знания*.

Знания эвристические – знания, накапливаемые *интеллектуальной системой* в процессе её функционирования, а также знания, заложенные в ней априорно, но не имеющие статуса абсолютной истинности в данной проблемной области. Часто эвристические знания связаны с отражением в БЗ человеческого (неформального) опыта решения задач.

Знание энциклопедическое – наиболее ёмкий и информативный тип толкового знания; одна из классических толковых информационных структур, которая с максимальной полнотой описывает конкретную *предметную область* путём детального толкования реалий и их внутренних связей.

Знания экспертные – знания, которыми располагает *специалист* в некоторой предметной области.

Значение логическое – значение, которое могут приобретать *логические переменные*: "истина" и "ложь".

Значение по умолчанию – значение переменной, которое ей автоматически приписывается, если её значение не задается.

Идентификация знаний – определение характеристик *знаний*, необходимых для решения задачи.

Иерархия – структура объектов, при которой строго выражены их уровни. Объекты нижнего уровня «подчиняются» объектам верхнего уровня, если между ними есть связи. Иерархия может быть представлена *деревом*.

Иерархия наследования – в *представлении знаний* – иерархическая организация единиц представления, при которой в случае отсутствия информации о некотором свойстве *видового понятия* или экземпляра, используется описание этого свойства для *родового понятия*.

Извлечение знаний – 1) получение наиболее полного из возможных представлений о предметной области и связанных с ней проблем; 2) процедура взаимодействия *эксперта* с источником знаний, в результате которой становятся явными процесс суждений специалистов при принятии решений и структура их представления о предметной области; 3) получение *информации* о предметной области от специалистов и выражение её на языке представления знаний. Извлечение знаний используется при построении *экспертной системы* или БЗ; 4) совокупность коммутативных, текстологических и других методов, используемых для воссоздания *модели* предметной области, необходимой для принятия решений пользователем (*инженером по знаниям*).

Извлечение информации – действия, *методы* и *процедуры*, направленные на получение информации о заданном предмете из общего *массива данных*.

Инженер по знаниям (*когнитолог, инженер-интерпретатор, аналитик, Knowledge engineer*) – 1) специалист, основной задачей которого является *проектирование баз знаний* и наполнение их знаниями о проблемной области. В процессе этой деятельности инженер по знаниям выбирает *форму представления знаний*, удобную для данной проблемной области, организует *приобретение знаний* из различных источников (официальные документы, учебники, монографии и т. п.), а

также в результате общения с экспертами-специалистами в данной проблемной области; 2) специалист в области *искусственного интеллекта*, выступающий в роли промежуточного буфера между *экспертом* и *БЗ*.

Инженерия знаний – раздел *искусственного интеллекта*, в рамках которого решаются проблемы, связанные с *извлечением знаний, приобретением знаний, представлением знаний и манипулированием знаниями*. Инженерия знаний служит основой для создания *экспертных систем* и других *интеллектуальных систем*.

Инновация – высшая форма деятельности по своей сложности и ценности. Инновация получается в результате деятельности, направленной на *использование знаний* для генерации идей, улучшающих процессы, продукты и услуги.

Инструментарий – набор *программных средств*, вспомогательный относительно главной задачи.

Инструментальные средства редактирования онтологий на определённом языке онтологий – используются для объединения двух онтологий с целью создания одной новой (Protégé с плагином PROMPT, Chimera, OntoMerge и др.).

Интеллект – способность человека мыслить, предусматривать возможные последствия своей и чужой деятельности, находить оптимальные варианты, создавать *новые знания*.

Интеллект искусственный (Artificial intelligence (AI)) – 1) одно из направлений *информатики*, целью которого является разработка *компьютерных систем*, способных выполнять функции, которые традиционно считаются интеллектуальными, – *понимание языка, логический вывод*, использование накопленных знаний, *обучение, планирование действий* и т.д.; 2) одно из направлений *информатики*, целью которого является разработка аппаратно-программных средств, позволяющих пользователю – непрограммисту ставить и решать интеллектуальные задачи в своей предметной области, общаясь с *компьютером* на языке, близком к естественному; 3) научное направление, в рамках которого ставятся и решаются задачи аппаратного или программного *моделирования* тех видов человеческой деятельности, которые традиционно считаются интеллектуальными и традиционно считаются прерогативой человека; 4) раздел *информатики*, занимающийся вопросами имитации мышления человека с помощью компьютера, в частности, рассуждения, обучения и самосовершенствования.

Интеллектуальная информационная технология (ИИТ) – ИИТ представляет собой комплекс *программных, лингвистических и логико-математических средств* для поддержки деятельности человека с помощью *интеллектуальных информационных систем*.

Интервью – способ работы с *экспертом* при *приобретении знаний*, когда инженер по знаниям выступает в роли интервьюера.

Интероперабельность – способность системы взаимодействовать с другими системами без каких-либо ограничений доступа и реализации.

Интерпретация – в *искусственном интеллекте* – установление связи между двумя системами описаний, что позволяет понимать одну систему с позиций другой.

Интерфейс естественно-языковый – совокупность *программных и аппаратных средств*, обеспечивающих общение интеллектуальной системы с пользователем на ограниченном рамках проблемной области ЕЯ. В состав естественно-языкового

интерфейса входят словари, отражающие словарный состав и лексику языка, а также *лингвистический процессор*, осуществляющий анализ текстов (морфологический, синтаксический, семантический и прагматический) и синтез ответов пользователю.

Информатика (*Computer science, CS*) – 1) научная дисциплина, изучающая структуру и общие свойства *информации*, а также закономерности всех процессов информационного взаимодействия; 2) *наука*, которая занимается изучением законов, методов и способов накопления, переработки и передачи, защиты и использования информации во всех сферах человеческой деятельности с помощью компьютеров; 3) группа дисциплин, которые занимаются разными аспектами разработки и использования компьютеров и *компьютерных систем*: прикладная математика, программирование, программное обеспечение, искусственный интеллект, архитектура ЭВМ, вычислительные сети; 6) наука о целесообразной обработке информации, рассматриваемой как представление знаний и сообщений в технических, экономических и социальных областях.

Информационная система – система, обеспечивающая ввод, хранение, поиск, обработку и вывод *данных* по запросам или регулярно необходимых данных.

Информационная технология (ИТ) – комплекс взаимосвязанных научных, технологических, инженерных и организационных методов и средств, обеспечивающих информационное взаимодействие с людьми и техническими объектами для решения производственных, научных и социально-экономических проблем с применением *компьютерных систем*.

Информация – любой *факт, понятие* или *значение*, полученные из данных, а также контекст, выбранный из знаний или ассоциированный с ними.

Источник знаний – *текст* (инструкция, монография, фотография, кинолента и др.) наблюдения или специалист-носитель необходимой информации. От источника знаний поступает информация, которая превращается в знания, фиксируется в памяти интеллектуальной системы.

Картина мира языковая – способ отображения реальности в сознании человека, который заключается в восприятии этой реальности через пирамиду языковых и культурологических особенностей индивидуума (языкового коллектива).

Кибернетика – наука об общих закономерностях управления в живых организмах, машинах и обществе.

Класс – элемент системы *классификации*, который принимается за основной и представляет собой совокупность предметов, понятий и т. д., объединённых по одним или нескольким признакам.

Классификация – система распределения объектов (предметов, явлений, процессов, понятий) по *классам* в соответствии с определёнными признаками.

Классификация иерархическая – классификация, в которой каждое подразделение высшего порядка состоит из подразделений низшего порядка, которые не пересекаются между собой.

Компьютер (ЭВМ) – программно-управляемое электронное устройство обработки *информации* на основе арифметических и логических операций.

Контекст – завершённый в смысловом отношении отрывок *текста*, в рамках которого можно правильно установить значение принадлежащих ему слов или словосочетаний.

Концепция – совокупность взаимосвязанных представлений, взглядов на какую-то *проблему*.

Лаборатория виртуальная – *информационно-технологическая система*, обеспечивающая взаимодействие научных коллективов, а также информационных, инструментальных, индустриальных и других материальных ресурсов и процессов в территориально распределённой среде.

Лаборатория виртуальная учебная – позволяет моделировать поведение объектов реального мира в *виртуальной компьютерной образовательной среде* и помогает обучаемым овладевать новыми знаниями и умениями в научно-естественных дисциплинах.

Лингвистика компьютерная – дисциплина на границе между *лингвистикой* и *искусственным интеллектом*, занимается изучением формальных свойств *естественных языков* с помощью *компьютера* и моделированием процессов *анализа, синтеза* и *понимания* естественно-языковых текстов.

Литера – любая *константа, переменная* или её отрицание.

Лингвистический корпус текстов – *коллекция* текстов по заданной тематике (из некоторой предметно-проблемной области), которые являются доступными в *электронной* форме.

Логика – *наука* о правильных способах *рассуждений*. В классическом варианте состоит из учения о понятиях, учения о суждениях и учения об умозаклучениях. В течение долгого времени с логикой связывалось учение Аристотеля о силлогистических умозаклучениях. *Силлогистика* была первой дедуктивной системой, возникшей в науке. В основе логики лежит понятие аксиоматической системы. Сила чистой логики, отвлекающейся от семантики предметной области, состоит в общности её методов и положений. Важно отметить, что логика есть наука о мышлении в понятиях, а не о познании мира посредством мышления о понятиях. Это показывает, что в *интеллектуальных системах* чисто логические решатели задач не могут исчерпать весь запас средств, необходимых для воссоздания интеллектуальной деятельности. На основе логики в конце XIX в. начала создаваться *математическая логика*, в основе которой лежит теоретико-множественные категории и понятие *формальной системы*.

Логика первого порядка – *формальная система*, в которой кванторы общности и существования могут связывать только индивидуальные переменные, но не могут связывать символы предикатов или иные функциональные символы.

Машина баз данных – блок управления *базой данных* в информационных системах. Специализированный либо универсальный *процессор* с собственной памятью, выполняющий обработку запросов.

Машина баз знаний – блок управления *базой знаний* в машине пятого поколения. Специализированный процессор (система процессоров), который выполняет обработку запросов и формирование ответов в некоторой *предметной области* на основе использования совокупностей *фактов* и *знаний* о предметной области, представленных в виде правил.

Машина вывода – программа, управляющая перебором правил.

Машина параллельного вывода – специализированный процессор (система процессоров), реализующий параллельно основные операции, характерные для вывода на знаниях.

Метаданные – данные, являющиеся описанием других данных.

Метазнания – часть базы знаний, которая описывает её структуру или содержит знания о системе, использующей эту базу.

Метаязык – язык, используемый для описания других языков.

Метод – 1) алгоритмическая процедура, набор алгоритмических процедур или алгоритмическая схема, обладающие следующими свойствами: областью применимости – перечнем задач, которые данный метод позволяет решить, входными и выходными данными, набором подзадач, решение которых необходимо для реализации данного метода (составной метод), или некоторым оператором (простым или составным) (простой метод); 2) систематизированная совокупность шагов, действий, которые необходимо предпринять, чтобы решить определённую задачу или достичь определённой цели.

Метод интервью – в инженерии знаний приём, с помощью которого добываются знания у экспертов-профессионалов. Инженер по знаниям выступает в роли репортера, берущего интервью. Он задаёт вопросы, цель которых уточнить сведения, сообщаемые экспертом относительно предметной области, в которой работает эксперт. Существуют специальные приёмы, которые входят в стандартный метод интервью, делающий беседу целенаправленной и эффективной.

Методология анализа и проектирования сложных систем в произвольной предметной области (SADT) – методология породила семейство методик (и соответствующих стандартов) IDEF (Integrated DEFINITION), ориентированных на разработку моделей ПДО и акцентирующих внимание на каком-то конкретном аспекте проектирования. В частности:

- методика IDEF0 рекомендована для содержательного анализа и функционального проектирования сложных систем управления, в том числе и программного обеспечения (в последнем случае трудно определить её преимущества по сравнению с методологией ООП). Описание объектов и процессов в методике выполняется в виде иерархической совокупности диаграмм с лаконичным описанием функций. Блоки на диаграммах выражают функции, поэтому их названия – глаголы или отглагольные существительные;
- методика IDEF1х предназначена для информационного моделирования, основана на концепции “сущность-связь”. Обычно отправным пунктом для разработки информационной модели является IDEF0-модель;
- методика IDEF3 описывает поведенческие аспекты конкретных приложений, рассматривает последовательность выполнения и причинно-следственные связи между ситуациями и событиями для структурного представления знаний о ПДО. Если IDEF0 связана с функциональными аспектами и отвечает на вопросы “Что делает система?”, то в IDEF3 детализируются IDEF0-функции. Эта модель отвечает на вопросы “Как система это делает?”;
- методика IDEF5 предназначена для онтологического анализа ПДО, анализа основных терминов и понятий, используемых для описания объектов и процессов, границ использования, а также взаимосвязей между ними. Служит для эф-

фективного исследования и документирования: словаря терминов, используемых при описании характеристик объектов и процессов, имеющих отношение к рассматриваемой ПдО, точных и однозначных определений всех терминов и классификации логических взаимосвязей между ними.

Для каждой из методик семейства IDEF разработаны этапы и стадии построения модели ПдО, языки и диаграммы представления результатов. Анализ получаемых в результате применения методик IDEF моделей ПдО и их описаний с точки зрения формализации и компьютерной обработки показал, что, по сути, они (функциональные модели и описания) представляют собой содержательное техническое задание на проектирование онтологической системы, включающей онтологию объектов (сущностей), онтологию процессов и онтологию задач.

Механизм вывода – совокупность *правил вывода* и стратегии управления выводом (применения этих правил). Частным случаем механизма вывода может быть произвольное применение правил вывода, как это делается в *логических исчислениях*.

Механизм наследования – приём, используемый в БЗ. Заключается в том, что на множестве информационных единиц вводятся *классифицирующие отношения* типа "класс-элемент", "род-вид" и т. п. При этом информация, относящаяся ко всем элементам класса или ко всем видам рода, содержится соответственно в описании класса или рода, а подчинённые им информационные единицы наследуют эту информацию, когда это необходимо.

Моделирование информационное – связано с исследованием *моделей*, в которых отсутствует непосредственное подобие физических процессов, происходящих в моделях, реальным процессам. В этом случае стремятся отобразить лишь некоторую функцию, рассматривают реальный объект как «*черный ящик*», имеющий ряд входов и выходов, и моделируют некоторые связи между выходами и входами.

Модель – 1) материальный *объект*, система математических зависимостей или *программа*, имитирующая структуру или функционирование исследуемого объекта; 2) математическое, натурное или иное представление устройства, системы или процесса, используемое для обучения, анализа, управления или планирования.

Модель знаний – описание знаний в БЗ. Известны четыре типа модели знаний: *логические*, в основе которых лежит формально-логическая модель; *сетевые*, в основе которых лежит семантическая сеть; *фреймовые*, основанные на фреймах; *продукционные*, основанные на продукциях. Каждая такая модель знаний определяет *форму представления знаний*.

Модель информационная – отражает отношения между элементами системы в виде структур *данных* и информационных потоков (состав и взаимосвязи).

Модель концептуальная (Conceptual model) – 1) система основных *понятий* и правил комбинирования *классов* понятий, которые не зависят от языков, и является содержательной структурой некоторой предметной области; 2) способ представления понятий или объектов предметной области; 3) единая система определений смысла терминов теории в их взаимной связи и зависимости.

Модель Крипке – одна из моделей *логической семантики*, используемая в искусственном интеллекте. В основе модели Крипке лежит представление о множестве возможных миров, каждый из которых задаётся формальной системой. Переход из одного возможного мира в другой в рамках модели Крипке осуществляется с помощью специального отношения, свойства которого могут варьироваться.

Модель лингвистическая – 1) модель, относящаяся к фиксации тех или иных *знаний о ЕЯ*; 2) описание объекта в терминах лингвистических переменных и рассуждений о них.

Модель логико-лингвистическая – модель, основанная на расширении *формальной системы*, в рамках которой вводятся процедуры изменения всех или части элементов формальной системы в зависимости от решаемых задач. Логико-лингвистическая модель часто используется как способ задания *модели Крипке*.

Модель логическая – модель *представления знаний*, в основе которой лежит *формальная система*.

Модель мира – способ отображения в памяти *интеллектуальной системы* знаний о внешней среде.

Модель обучения – модель, лежащая в основе *процесса обучения* человека или технического устройства. Различают два типа модели обучения – *дескриптивный* и *нормативный*. Дескриптивный тип модели обучения образуется на основе описания процесса деятельности, которому человек или система должны обучаться. Наиболее известный способ основан на процедуре обучения на примерах. Нормативный тип модели обучения задается заранее. Часто обучение нормативного типа называют обучением с учителем.

Модель открытая – модель, в которую в процессе функционирования *интеллектуальной системы* можно добавить новые факты и закономерности.

Модель поведенческая (событийная) – описывает *информационные процессы* (динамику функционирования), в ней фигурируют такие категории, как состояние системы, событие, переход из одного состояния в другое, условия перехода, последовательность событий.

Модель пользователя – совокупность знаний об особенностях работы пользователя с системой, его намерениях, целях и требованиях. Хранится в памяти интеллектуальной системы. Модель пользователя помогает системе организовать эффективный диалог с пользователем, создаёт ему психологический комфорт.

Модель продукционная – или модель, основанная на *правилах*, позволяет представить знания в виде предложений типа «**Если** (условие), **то** (действие)».

Модель сетевая – модель представления знаний, в основе которой лежит *семантическая сеть*.

Модель сознания – в *искусственном интеллекте* – совокупность *процедур* и *декларативных описаний*, с помощью которых в интеллектуальных системах имитируется та часть сознательной деятельности человека, которая поддается *вербализации*. В психологии термин "*сознание*" трактуется более широко. В него включаются, например, способность субъекта, обладающего сознанием, к самонаблюдению (самосознанию), рефлексии и активности.

Модель формальная – *формальное описание* на некотором языке структуры объекта.

Модель функциональная – описывает совокупность выполняемых системой функций, характеризует *морфологию* системы (её построение) – состав функциональных подсистем, их взаимосвязи.

Модель эмпирическая – система операциональных определений смысла базовых терминов *концептуальной модели* на всём множестве предметов *онтологической картины мира*.

Модель языка – в лингвистике – формализованное представление знаний о языке. Как правило, включает *морфологический, синтаксический, семантический и прагматический* компоненты, которые также могут разделяться на более дробные компоненты.

Монотонность при выводе – свойство, характерное для вывода в *замкнутой формальной системе* и в *закрытой БЗ* и состоящее в том, что ранее выведенные утверждения не теряют истинности при расширении множества посылок для вывода.

Наследование (inheritance) – свойство, используемое в *базах данных и знаний* и заключающееся в том, что если две информационные единицы соединены между собой отношениями типа "*род-вид*" или "*класс-элемент*", то информация, общая для всех видов, входящих в *род*, или для всех элементов, входящих в *класс*, содержится в информационной единице более высокого уровня и при необходимости наследуется единицей более низкого уровня. Наследование позволяет ликвидировать дублирование в *хранении информации* в базах данных и знаний.

Наука – 1) сфера человеческой деятельности, функцией которой является выработка и теоретическая *систематизация знаний* о действительности (*научная картина мира*); 2) одна из форм общественного сознания; 3) обозначение отдельных отраслей знаний, научных дисциплин или научных направлений.

Наука когнитивная – 1) интегральная междисциплинарная область исследований, которая занимается созданием *моделей сознания*, связанных с *переработкой знаний, принятием решений, пониманием ЕЯ, логическим выводом, аргументацией* и другими видами когнитивной деятельности; 2) комплекс научных дисциплин (когнитивная психология, теория аргументации и др.), объединённых единым *предметом исследования* – отражением в познавательных структурах человека окружающей его действительности и исследованием механизмов рассуждения об этой действительности.

Немонотонность при выводе – свойство, характерное для вывода в *открытой формальной системе* и в *открытой БЗ* и состоящее в том, что ранее выведенные утверждения могут перестать быть выводимыми при появлении новых фактов.

Неопределённость лингвистическая – неопределённость, возникающая из-за *неоднозначности* словесных выражений. При описании *качественных знаний* приходится применять специальные приёмы для устранения лингвистической неопределённости.

Неполнота – свойство описания предметной области, заключающееся в том, что это описание не может быть преобразовано в *формальную систему*. При работе с неполной информацией используются правдоподобные рассуждения или рассуждения по умолчанию.

Обеспечение программное (Software) – 1) общий термин для обозначения "*неосязаемых*", в отличие от физических частей *вычислительной системы*; 2) является одним из видов обеспечения вычислительной системы, наряду с техническим (аппаратным), математическим, информационным, лингвистическим, организационным и методическим обеспечением.

Область предметная, слабо структурированная – предметная область, *концептуальная модель* которой не может быть погружена в формальную систему или совокупность формальных систем. Большинство предметных областей, с которыми приходится иметь дело в интеллектуальных системах, являются плохо структурированными.

Область предметная, хорошо структурированная – предметная область, *концептуальная модель* которой может быть погружена в *формальную систему*.

Обобщение знаний – совокупность приёмов и методов, позволяющих в БЗ вводить *новые знания*, получаемые из имеющихся за счёт *кластеризации*, введения *гиперсобытий* и *гипотез*.

Оболочка – *инструментальное средство* для проектирования и создания экспертных систем или других интеллектуальных информационных систем. В состав оболочки входят средства проектирования БЗ с различными формами *представления знаний* и выбора режима работы решателя задач. Для конкретной предметной области *инженер по знаниям* определяет нужное представление знаний и стратегию решения задач, а затем создаёт конкретную экспертную или другую *интеллектуальную информационную систему*.

Обоснование – одна из функций *интеллектуальной системы*, заключающаяся в доказательстве или проверке того, что полученное системой решение не противоречит знаниям, которые хранятся в памяти системы. Таким образом, обоснование является относительным. При изменении содержимого *БЗ* или *БД* обоснование может либо сохранить свою силу, либо стать неверным. Обычно обоснование тесно связано с *объяснением*. К обоснованию близко понятие *аргументации*.

Обработка естественного языка (Natural language processing) – совокупность процессов *анализа* текстов на ЕЯ, их *понимания* и *синтеза* текстов. В процессе анализа в наиболее развитых системах обработки естественного языка сообщений происходит морфологический, синтаксический и семантический анализ текста, в результате чего выявляется глубинная структура текста, которая переводится во внутреннее представление, используемое в базе знаний интеллектуальной системы. Соотнесение этой структуры с теми знаниями, которые хранятся в системе, позволяет понять смысл исходного текста. При синтезе текстов сначала формируется семантическая структура текста, которая затем наполняется лингвистическими единицами с учетом синтаксиса и морфологии выбранного ЕЯ. С обработкой ЕЯ связано решение задач *машинного перевода*, *автоматического реферирования*, *извлечения знаний*, *общения с пользователем* на ограниченном профессиональном ЕЯ и т. п.

Обработка текста – операции *обработки данных* над текстом, такие как *ввод*, *редактирование* текста, *сортировка*, *извлечение*, *хранение*, показ на дисплее и печать.

Образец – фрагмент знаний, по которому осуществляется *поиск по образцу*, или эталон, по которому происходит *классификация* изображений, ситуаций, правил и т. п.

Обучение – усвоение знаний, умений и навыков путём получения и восприятия информации от учителя или обработки наблюдаемой информации с последующим построением на основе этих наблюдений новых общих правил и *закономерностей*.

Обе формы обучения используются в интеллектуальных системах для *приобретения* новых знаний.

Обучение на примерах – вид обучения, при котором индивиду или интеллектуальной системе предъявляется набор положительных и отрицательных примеров, связанных с какой-либо заранее неизвестной закономерностью. В интеллектуальных системах вырабатываются решающие правила, с помощью которых происходит разделение множества примеров на положительные и отрицательные. Качество разделения, как правило, проверяется экзаменационной выборкой примеров. Если качество разделения на экзаменационной выборке оказывается удовлетворительным, то выработанные решающие правила принимаются системой как окончательные. Если экзамен оказался неудовлетворительным, то экзаменационная выборка добавляется к обучающей и строятся новые решающие правила. После этого процесс экзамена повторяется.

Обучение машинное – обучение с помощью *компьютера*; применение компьютера в учебном процессе.

Объём понятия – *класс* обобщённых в понятии предметов.

Объект информационный – идентифицируемый *объект реального мира*, некоторое понятие или процесс, относящиеся к предметной области, о которой хранятся описательные данные.

Объектно-ориентированное программирование (ООП) – *программирование*, в основу которого положено понятие объекта. *Объект* – это элемент программирования, объединяющий в себе как данные, так и действия над ними. ООП позволяет упростить программирование, сделать его более естественным; характеризуется тремя основными свойствами: *инкапсуляцией, наследованием и полиморфизмом*.

Объяснение – одна из функций *интеллектуальной системы*. Объяснение предоставляет пользователю информацию о том, каким путём интеллектуальная система достигла выданное пользователю решение. В отличие от *обоснования* объяснение опирается лишь на тот маршрут, который сохранился в памяти системы от процесса поиска решения. Используя этот маршрут, *интеллектуальная система* формирует пользователю объяснение на профессиональном ЕЯ, позволяющее ему представить все принципиальные шаги решения.

Ограничение целостности – *ограничения*, налагаемые на совокупность информационных единиц, хранящихся в *БД* и *БЗ*. Эти ограничения должны выполняться в любых состояниях, которые определяются текущим содержимым *БЗ* и *БД*.

Онтология – 1) *явная спецификация* концептуализации. Формально онтология состоит из терминов, организованных в *таксономию*, их определений и *атрибутов*, а также связанных с ними *аксиом* и *правил вывода*; 2) *БЗ* специального типа, которая может «*читаться*» и пониматься, отчуждаться от разработчика и / или физически разделяться пользователями; 3) формализованное представление основных понятий *предметной области* и связей между ними.

Онтология лексическая – лексико-семантические структуры, в которых установлены *семантические связи* между словами и словосочетаниями.

Онтология компьютерная:

– *иерархическая структура* конечного множества понятий, описывающих заданную *предметную область*;

- структура представляет собой *онтограф*, вершинами которого являются понятия, а дугами – семантические отношения между ними;
- понятия и отношения интерпретируются в соответствии с общезначимыми функциями интерпретации, взятыми из электронных источников знаний заданной ПДО;
- определение понятий и отношений выполняется с помощью аксиом и ограниченной области действия;
- формально онтограф описывается на одном из *языков описания онтологий*;
- функции интерпретации и аксиомы описаны в некоторой подходящей *формальной теории*.

Онтология предметная – содержит понятия, описывающие конкретную предметную область, отношения, семантически значимые для данной предметной области, и множество *интерпретаций* этих понятий и отношений (*декларативных* и *процедурных*).

Онтология задач – в качестве понятий содержит *типы* решаемых *задач*, а отношения этой онтологии, как правило, специфицируют *декомпозицию задач* на подзадачи.

Онтологический инжиниринг – новое направление в методологии разработки *интеллектуальных систем*, основанное на формализованных методах построения *онтологических описаний* предметных областей и их использовании.

Оправдание – одна из функций *интеллектуальной системы*. С помощью оправдания некоторое решение системы обосновывается не путём логических рассуждений или обращения к имеющимся в системе знаниям, а путём обращения к имеющейся в системе ценностной структуре. Оправдание убеждает в том, что данное решение не противоречит этой ценностной структуре.

Определение – способ фиксации и толкования значения (семантики) слова, возможное средство его терминологизации, в более строгом, математическом понимании – *логическая операция*, которая раскрывает суть понятия.

Определение интенциональное – определение объекта посредством толкования.

Определение рекурсивное – определение, которое ссылается на определяемый объект.

Определение экстенциональное – определение понятия через прямое перечисление *денотатов*, объектов, что составляют это понятие.

Отношение бинарное – отношение между двумя *множествами*.

Отношение виртуальное – отношение, в явной форме не присутствующее в *БЗ*, но определяемое из тех отношений, которые имеются в базе.

Отношение временное – отношение, с помощью которого описываются связи факторов, *событий* и *явлений* во времени. Примерами временного отношения могут служить "*быть раньше*", "*одновременно*", "*заканчиваться одновременно*" и т. п. Временные отношения используются во *временных логиках*, *логиках действий* и других *псевдофизических логиках*.

Отношение действия – отношение, с помощью которого описываются действия, происходящие в реальном мире. Примерами отношения действия могут служить: "*двигаться к*", "*приблизиться*", "*воздействовать*" и т. п. Отношения действия используются в логиках действий, являющихся видом псевдофизических логик.

Отношение интенциональное – отношение на множестве утверждений, относящихся к интенциональной составляющей знаний о *проблемной области*.

Отношение каузальное – отношение, с помощью которого описываются различные виды *причинно-следственных связей* между объектами.

Отношение моделирования нечёткое – отношение, задаваемое специальной *коммутативной диаграммой*, связывающей элементы и операции одного множества с элементами и операциями другого множества с помощью *операторов принадлежности*.

Отношение пространственное – отношение, с помощью которого описываются пространственные взаимосвязи объектов. Примерами пространственных отношений могут служить: "*находиться близко*", "*соприкасаться*", "*быть внутри*" и т. п. Пространственные отношения используются в *пространственных логиках*, действий и других псевдофизических логиках.

Отношение релевантности – связь двух или более информационных единиц, устанавливаемая на основе их *семантической близости*.

Отношение семантическое – отношение, используемое в БЗ, семантика которого задаётся его именем.

Отношение функциональное – отношение, с помощью которого в БЗ задаются связи между информационными единицами. Эти связи определяют *процедуры нахождения* (вычисления) одних единиц через другие.

Отношение экстенциональное – отношение на множестве конкретных фактов, хранящихся в БД.

Парадигма когнитивная – методология исследования и моделирования феномена сознания (процессов, средств, результатов и т. п.) в искусственном интеллекте и в гуманитарных дисциплинах, включающих, по крайней мере, три направления:

- 1) исследование и моделирование механизмов когнитивных актов: логического вывода, аргументации, рассуждений, понимания, принятия решений, экспертизы и т. п.;
- 2) поиск адекватных логик, псевдологик, семиотических моделей для представления знаний;
- 3) отождествление когнитивной деятельности с переработкой знаний, и, как следствие, преобразование знаний в одну из ключевых категорий когнитивной проблематики, а самого процесса извлечения экспертных знаний – в важную и вполне самостоятельную исследовательскую область, которую называют когнитологией или инженерией знаний.

Переменная лингвистическая – переменная, значение которой определяется набором *вербальных* (т. е. словесных) характеристик некоторого свойства.

Переменная пропозициональная – переменная, используемая в *пропозициональной логике*.

Переменная связанная – переменная в логике, стоящая в зоне действий *квантора общности* или *квантора существования*.

Планирование – процесс составления последовательности действий, подзадач, операций, подцелей, поочерёдное выполнение которых должно привести к дости-

жению *целей*, поставленных перед системой. В *интеллектуальных системах* планирование может осуществляться либо в *пространстве задач*, либо в *пространстве состояний*.

Подход – совокупность однородных *методов*.

Понимание естественного языка – 1) раздел *искусственного интеллекта*, который занимается описанием *грамматики* и *семантики ЕЯ*; совокупность моделей и процедур, с помощью которых в интеллектуальных системах происходит сопоставление поступающего текста на ЕЯ с фрагментами знаний из *базы знаний*, а также процедур, позволяющих выводить из имеющихся знаний другие, необходимые для правильной *интерпретации* содержимого введенного текста.

Понятие – 1) элемент *системы знаний*, представляющих собой обобщенную модель некоторого класса элементов мира, отраженного в знаниях, с помощью которой реализуются процессы распознавания и генерации моделей конкретных элементов мира; 2) обозначается именем, присваиваемым классу *сущностей*, объединяемых благодаря общности их признаков структур. В логике понятия являются строго определенными и неизменными образованиями, характеризующимися лишь признаковой структурой, имманентно присущей всем понятиям. В искусственном интеллекте, как и в бытовой практике людей, понятие понимается шире. Для образования понятий в интеллектуальных системах используются различные приемы *обобщения*; 3) целостная совокупность суждений, о которых что-либо утверждается об отличительных признаках исследуемой сущности, ядром которой являются суждения (или утверждения) о наиболее общих и в то же время существенных признаках этой сущности.

Построение базы знаний – *формальное описание* предметной области, в том числе путём получения информации от *профессионалов-экспертов*, выраженное её языком *представления знаний*.

Потомок – вершина *дерева*, расположенная ниже *родительской вершины*.

Правило вывода – правило, с помощью которого в *формальных системах* из множества аксиом порождается правильно построенные формулы, которые интерпретируются как истинные.

Правило синтаксическое – в формальной системе правило, которое определяет способ формирования *синтаксически правильных выражений*. В лингвистике синтаксические правила позволяют отделять синтаксически правильные предложения данного языка от тех, которые таковыми не являются.

Предикат – в исчислении предикатов *специальный знак*, отражающий определённое отношение между конечным множеством *сущностей* – *аргументов*. В обычном варианте исчисления предикатов в качестве значения предиката на множестве означенных аргументов выступают два: *истина* и *ложь*.

Предметная область – 1) конкретная сфера деятельности человека, к которой относится решаемая задача; 2) часть реального мира, которая описывается или моделируется с помощью *БЗ*. Предметная область состоит из информационных объектов, совокупность которых образует объектное ядро; 3) часть *объективной реальности*, существующая независимо как от создателя информационной системы, так и от самой информационной системы.

Представление знаний (*Knowledge representation (KR)*) – 1) совокупность методов и моделей отображения знаний в памяти компьютера; 2) формализация знаний для их ввода в БЗ. На концептуальном уровне представления знаний наиболее распространены модели знаний в виде *логических выражений, семантических сетей, фреймов* и *продукционных систем*. Представление знаний как направление *искусственного интеллекта* традиционно включает также задачи проверки содержимого БЗ на *корректность и полноту*, пополнения знаний за счёт логического вывода на основе имеющихся в БЗ, обобщения знаний и классификация знаний; 3) организация знаний (фактов, правил) в виде *структур данных*. Одна из фундаментальных проблем ИИ.

Представление экстенциональное – представление константных фактов, не содержащих свободных переменных в *БД* или в *БЗ*.

Приобретение знаний (*Knowledge acquisition, Knowledge elicitation*) – совокупность методов и процедур, которые применяет инженер по знаниям при формировании им БЗ. Приобретение знаний предполагает использование источников знаний двух типов: *пассивных* и *активных*. К первым относятся официальные документы, инструкции, печатные издания, кино-фото-документы и многие другие источники, в которых содержатся сведения, важные для описания знаний о предметной области. Ко второму типу источников знаний относятся люди – специалисты в данной предметной области. Инженер по знаниям с помощью специальных психологических методик и инструментальных средств в процессе диалога получает от экспертов необходимые сведения. Все приобретенные знания для ввода в БЗ формализуются в соответствии с требованиями той модели знаний, которая соответствует выбранному проектировщиком системы представлению знаний.

Пространство задач – структура, отражающая *декомпозицию* крупных задач на более мелкие, вплоть до стандартных задач, решение которых предполагается известным. Пространство задач используется в интеллектуальных системах для планирования деятельности и в задачах *автоматического синтеза программ*. Решение нужной задачи ищется как композиция решений стандартных задач.

Пространство семантическое – структура на знаниях, в которой введено понятие "*семантическое расстояние*".

Пространство состояний – совокупность состояний, в которых может находиться *техническая система* или *процесс*. В пространстве состояний может быть задана *метрика*, а также указаны возможные траектории смены состояний под влиянием различных причин. Пространство состояний используется в интеллектуальных системах при автоматическом синтезе программ и при планировании деятельности.

Пространство целевое – совокупность целей с указанием возможных траекторий их достижения. Целевое пространство используется в интеллектуальных системах при планировании деятельности и при автоматическом синтезе программ.

Процесс – 1) последовательность предусмотренных *событий*, которая определяется объектом или явлением и выполняется в заданных условиях; ход событий, происходящих в соответствии с поставленной *целью* или *результата*; 2) упорядоченные и направленные события, определённые их конечной целью или результатом, достигаемым в данных конкретных условиях.

Процессор логического вывода – специализированный процессор (система процессоров), реализующий набор процедур, необходимых для организации логического вывода или извлечения следствий из знаний о некоторой предметной области.

Разработчик средств ВТ – специалист в области вычислительной техники, занимающийся разработкой некоторого вычислительного устройства или класса вычислительных устройств.

Расстояние семантическое – оценка "семантической" близости информационных единиц, хранящихся в памяти интеллектуальной системы или человека. Понятие "семантической" близости неоднозначно. Эксперименты с людьми показывают, что в зависимости от поставленной цели семантическое расстояние может интерпретироваться как ситуативная близость (встреча в некоторых типовых ситуациях), ассоциативная близость (типа "молоток-гвоздь"), таксономическая близость (типа "стол-кровать") и т. д. Семантическое расстояние используется в БЗ для ускорения выбора релевантной информации к данному понятию или ситуации. До настоящего времени нет удовлетворительных моделей, в рамках которых можно было бы измерять семантическое расстояние.

Решатель – система, способная благодаря встроенной в неё общей стратегии нахождения решения (например, путём поиска в пространстве альтернатив или путём логического вывода) находить решения задач. Решатель входит в качестве основного блока в интеллектуальные системы. Иногда этот блок называется полно: решатель задач.

САИД (система активации интеллектуальной деятельности) – продукт многоаспектных действий с привлечением знаний из различных предметных областей, таких как математика, ИИ, философия, нейрофизиология, computer science и др. Ориентирована на поддержку постановки и решения научно-технических проблем (получение новых знаний из уже имеющихся).

Семантема – единица смысла, элементарное значение.

Семантика – 1) правила и условия, определяющие соотношения между элементами языка и их смысловыми значениями, а также интерпретацию содержательного значения синтаксических конструкций языка; 2) один из аспектов семиотики. Рассматривает отношение знаков к обозначаемому (содержание знаков) независимо от адресата; 3) значение отдельных единиц языка; 4) изучение отдельных единиц языка – языковедческая семантика, элементарным объектом изучения которой является единство трёх объектов: означающего, означаемого и денотата. Означающее – внешний элемент (последовательность звуков или знаков), денотат – обозначаемый объект действительности и означаемое – отражение этого объекта в сознании человека.

Семантика глубинная – направление лингвистической семантики и ИИ, занимающееся проблемами представления и использования внеязыковых или экстралингвистических слоёв смысла (знаний) во время семантического анализа текстов.

Семантика когнитивная – когнитивный подход к категоризации понятий, а также анализа организации понятийного пространства в сознании человека, где понятийные структуры рассматриваются как носители базисных знаний о мире, а лексическое значение – как когнитивная категория.

Семантика поверхностная – направление семантических исследований, занимающееся проблемами поверхностно-семантического представления *смысла предложения* и текста в многоуровневых *лингвистических моделях ЕЯ*.

Семантика ситуативная – приписывание некоторым объектам, хранящимся в БЗ, определённых характеристик в зависимости от ситуации, в которой эти объекты наблюдаются или используются. В системах *понимания текстов* на ЕЯ ситуативная семантика связана с приписыванием различных значений *лексемам* в зависимости от того *контекста*, в котором они используются.

Семиотика – наука, изучающая свойства *знаков* и знаковых систем (в основном естественных и *искусственных языков*). Выделяются три основных аспекта исследования: *синтактика*, изучающая внутренние свойства систем знаков безотносительно к интерпретации; *семантика*, рассматривающая отношение знаков к их означаемому, безотносительно к особенностям интерпретатора знаков; *прагматика*, изучающая проблемы *интерпретации знаков*.

Сеть – пятерка $H = \langle A, B, P, P_1, C \rangle$, где A – множество вершин, B – множество имён (весов) вершин; P – множество дуг, соединяющих пары вершин; P_1 – множество отмеченных входных и выходных дуг; C – множество имён (весов) дуг; 2) семантическая структура, состоящая из *сущностей* и их *взаимосвязей*.

Сеть абстрактная семантическая – *ориентированный граф*, вершинами которого выступают слова и словосочетания, а дугами – семантические отношения между словами.

Сеть ассоциативная – *семантическая сеть*, в которой отношения указывают на ассоциативные связи между вершинами, характеризующими объекты, факты и ситуации для описываемой *предметной области*.

Сеть вывода – структура, которая отображает последовательности применения *правил вывода* к исходным посылкам. Из-за неоднозначности выбора правил на каждом шаге возникает множество путей, образующих *сеть вывода*.

Сеть каузальная – *семантическая сеть*, в которой дуги характеризуют отношения, используемые в *каузальной логике*.

Сеть пирамидальная – *ациклический ориентированный граф*, в котором нет вершин, имеющих одну входную дугу.

Сеть семантическая (*Semantic network*) – 1) способ *представления знаний* в виде ориентированного графа, у которого вершины соответствуют понятиям, объектам, действиям, ситуациям или сложным отношениям, а дуги – свойствам или элементарным отношениям; 2) модель *ситуации, среды, предметной области* в виде графа, описывающего связи между объектами, свойствами, состояниями, действиями, модификаторами.

Сеть семантическая интенциональная – семантическая сеть, в которой отражены *интенциональные знания* о предметной области. Эти знания относятся к общим законам области и оперируют не константными фактами, а высказываниями об области, содержащими переменные, которые могут означаться в конкретных ситуациях. При *фреймовом представлении* интенциональной семантической сети соответствует *фрейм-прототип*.

Сеть семантическая экстенциональная – семантическая сеть, в которой отражены *экстенциональные знания* о конкретной ситуации в предметной области. В экс-

тенциональной семантической сети все вершины сети соответствуют конкретным объектам, а связи между ними конкретным связям, которые наблюдаются в описании ситуации. При фреймовом представлении экстенциональной семантической сети соответствует *фрейм-экземпляр*.

Сеть растущая семантическая – семантическая сеть, в которой определены *процессы формирования понятий и связей*.

Силлогизм – специальная форма умозаключения от общего к частному. Силлогизм представляет собой заключение, следующее из двух посылок, содержащих утверждения о соотношении объёмов двух *классов* или о принадлежности некоторого элемента определённому классу. Примером силлогизма может служить следующее умозаключение: все хищники питаются мясом; волк – хищник; волк питается мясом.

Символ – *графическое представление* понятия, которое имеет значение в некотором определённом *контексте*.

Синергетика – междисциплинарное направление научных исследований, задачей которого является изучение *природных явлений и процессов* на основе принципов самоорганизации систем (состоящих из *подсистем*).

Синтаксис (Syntax) – 1) совокупность *правил* некоторого *языка*, определяющих формирование элементов этого языка, другими словами – совокупность правил образования семантически значимых последовательностей символов в данном языке; 2) *способы соединения слов* в словосочетания и простые предложения и простых предложений – в сложные; 3) раздел *языкознания*, изучающий способы соединения слов и предложений, вместе с *морфологией* составляющий грамматику; 4) синтаксис текста – то же, что схема дискурса или неративная (повествовательная) схема; строение текста.

Система – множество *элементов* и *отношений* между ними, рассматриваемых как единое целое.

Система автоматизированного проектирования (САПР) – комплекс средств, предназначенных для помощи *инженеру-проектировщику* в создании *новых технических объектов*. САПР включает большое количество расчетных модулей, базу знаний средств ведения документации и технического черчения.

Система аксиоматическая – *система рассуждений*, в основе которой лежат аксиомы. Если аксиомы являются логическими (т.е. тождественно истинными вне зависимости от предметной области), то аксиоматическая система совпадает с *формальной системой*. Если среди аксиом имеются аксиомы, справедливые только в данной предметной области, то аксиоматическая система применима только в этой области. Часто аксиоматическая система, в которой имеются предметные аксиомы, называют *квазиаксиоматической системой*.

Система вопросно-ответная – совокупность *программ*, позволяющая реализовать вопросно-ответное отношение для пары "*пользователь-ЭВМ*". Вопросно-ответная система обеспечивает поиск релевантных данных и знаний по поступившему в систему вопросу, из которых формируется ответ *пользователю*. Вопросно-ответные системы могут использовать вопросы и ответы, формулируемые на ЕЯ, ограниченном ЕЯ или на специальном формальном языке общения.

Система естественно-языковая – интеллектуальная система, способная воспринимать тексты или речь на ограниченном ЕЯ, понимать их и функционировать в соответствии с информацией, содержащейся в тексте. При необходимости естественно-языковая система может формировать ответные сообщения, связанные с введенным в них текстом.

Система интеллектуальная – техническая или программная система, способная решать задачи, традиционно считающиеся творческими, принадлежащие конкретной предметной области, знания о которой хранятся в памяти интеллектуальной системы. Структура интеллектуальной системы включает три основных блока – БЗ, решатель и интеллектуальный интерфейс.

Система интеллектуальная обучающая – интеллектуальная система для обучения человека какому-либо роду деятельности или используемая в процессе обучения школьников или студентов. Обучающая интеллектуальная система включает естественно-языковой интерфейс, средства графического общения, базу знаний и специальный блок, управляющий процессом подачи материала, анализирующий обучаемого и принимающий решение об оценке степени усвоения.

Система интеллектуальная обучающаяся – интеллектуальная система, в которой имеется совокупность средств для пополнения её БЗ. Возможны два способа обучения: с учителем и без учителя. В первом случае обучающаяся интеллектуальная система должна иметь возможность пополнять свою БЗ информацией, сообщаемой ей учителем, корректировать новую информацию с ранее имевшейся и задавать учителю вопросы, когда возникают конфликтные ситуации. При обучении без учителя обучающаяся интеллектуальная система обобщает информацию, которая содержится в её БЗ или наблюдается во внешней среде. С помощью индуктивного вывода и, возможно, рассуждений по аналогии и рассуждений по ассоциации обучающаяся интеллектуальная система формирует гипотезы о новых закономерностях в той предметной области, в которой она работает. Этим новым знаниям система может приписывать оценки правдоподобия.

Система информационная (information system, IS) – система обработки информации, работающая совместно с организационными ресурсами, такими как люди, технические средства и финансовые ресурсы, которые обеспечивают и распределяют информацию.

Система информационная интеллектуальная – объединённая информационным процессом совокупность технических средств и программного обеспечения, которая при взаимосвязи с человеком (коллективом людей) или автономно, может синтезировать цель, вырабатывать решения о действии и находить рациональные способы её достижения.

Система объяснения – часть ЭС, предназначенная для разъяснения пользователю по его просьбе способа, которым получено решение. Система объяснения даёт ответы на вопросы трёх типов. "Как-вопросы" заставляют систему объяснять путь решения. "Почему-вопросы" вызывают у системы объяснения необходимость в пояснении, почему получено именно это решение, а не какое-либо другое. "Что-вопросы" требуют от системы объяснения выдачи всей имеющейся в экспертной системе информации относительно объекта или явления, к которому относится "Что-вопрос". Для ответов на подобные вопросы в системе объяснения имеется набор специальных процедур и рабочая память, в которой в процессе поиска решения формируется информация, необходимая для ответа на вопросы пользователя. С

системой объяснения связана система доверия, увеличивающая объяснительную силу ЭС.

Система (компьютерных) знаний – некоторая конструктивная среда, представленная в базе подходящего *формально-логического языка* и обеспечивающая постановку и решение в ней задач в заданной *предметной области*.

Система, основанная на знаниях (*Knowledge based system*) – интеллектуальная система, функционирование которой определяется совокупностью знаний о проблемной области, в которой она используется.

Система открытая – система, разработчики которой делают общедоступными все необходимые стандарты разработанной системы. Это позволяет другому производителю создать подобную систему, улучшить её характеристики, добавить собственные устройства или программные средства, организовать их взаимодействие.

Система представления знаний – совокупность средств, предназначенных для автоматизированного наполнения БЗ и поддержания их в рабочем состоянии и формального представления.

Система управления базой знаний (*Knowledge Base Management System*) – совокупность *программных и аппаратных средств*, предназначенных для организации в БЗ процедур, связанных с поиском знаний, наполнением базы знаний, её корректировкой и т. д.

Система формальная – модель, лежащая в основе многих математических теорий. Формальная система представляет собой четвёрку: множество базовых элементов, синтаксические правила, аксиомы и правила вывода.

Система фреймов – множество *фреймов*, связанных между собой различными отношениями.

Система экспертная (ЭС) – 1) наиболее распространённый класс ИС, ориентированный на тиражирование опыта высококвалифицированных специалистов в областях, где качество принятия решений традиционно зависит от уровня *экспертизы*, например медицина, юриспруденция, геология, экономика, военное дело и др.; 2) система обработки данных, которая обеспечивает экспертное решение проблем в заданной области применения, строя выводы на основе БЗ, в которой формализован человеческий опыт.

Система экспертная пустая – экспертная система, у которой БЗ пуста. При использовании пустой ЭС в некоторой предметной области пользователю необходимо произвести наполнение БЗ самому с помощью специальных инструкций по заполнению базы либо с привлечением *инженера по знаниям*. Как показал опыт, пустые ЭС могут эффективно использоваться лишь в однотипных предметных областях, на которые ориентированы принятые в ней способы *представления знаний и рассуждений*.

Систематизация (организация) **знаний** – *классификация и категоризация знаний* с целью их последующего целенаправленного извлечения; поддержание целостности данных за счёт реализации соответствующих процессов.

Системный подход – 1) *метод* рассмотрения вопроса, задачи, явления, объекта как части более широкого единого и неразделимого целого – системы взаимосвязанных составных элементов; 2) *направление методологии анализа, синтеза* и исследования, в основе которого лежит рассмотрение сложного объекта как целост-

ного множества элементов в совокупности отношений и связей между ними, то есть рассмотрение объекта как системы; кроме того, системный подход – это некоторый способ организации действий разработчика (проектировщика), который охватывает любой род деятельности, выявляя закономерности и взаимосвязи с целью их более эффективного использования для достижения поставленной цели.

Следствие логическое – факт, полученный в процессе *логического вывода* в *формальной системе*.

Словарь – структура данных или информационная система, обеспечивающая доступ к информации с текстовым именем.

Слот – 1) в *представлении знаний* – составная компонента фрейма, которая характеризует некоторое свойство или связь описываемого фреймом понятия или объекта; 2) основная структурная единица фрейма. Слот представляет собой пару: (*атрибут* (имя слота) – *значение*). В качестве значения могут выступать константные факты, выражения, содержащие переменные, ссылки на другие слоты и т. п. Слот может иметь структуру, элементы которой сами являются слотами. Часто эти более мелкие слоты называют фасетами, ячейками, аспектами и т. п.

Событие – информационная единица, которой в *базе знаний* присваивается интервал времени, в течение которого эта единица существует.

Содержание понятия – совокупность (обычно существенных) *признаков*, по которым произведено *обобщение* и *выделение* предметов в данном понятии.

Сопоставление – в *экспертных системах*, основанных на использовании *семантических сетей*, – операция поиска в *базе знаний* путём сравнения некоторого фрагмента сети, содержащей переменные, с общей семантической сетью.

Сопоставление с образцом – процедура сравнения *фрагмента знаний* или описания с *эталоном-образцом*. Входит в состав процедуры поиска по образцу в базах знаний и используется при распознавании образов.

Спецификация – *описание системы*, полностью определяющее её цель и функциональные возможности.

Средства аппаратные (hardware) – 1) *технические средства* – в отличие от программ, процедур, правил и соответствующей документации; 2) средства, используемые при выполнении программы; любое *физическое устройство*, осуществляющее *обработку данных*.

Средства объяснения – средства *экспертной системы*, позволяющие *пользователю* получить информацию о фактах и правилах, используемых при выводе некоторого утверждения.

Стратегия управления выводом – совокупность *правил*, с помощью которых организуется выбор правил вывода в *формальных системах* или выбор продукции в *системе продукций* при *поиске решения*. Наиболее известные стратегии управления выводом в формальных логических системах – это прямой вывод и обратный вывод. В системах продукций известно много стратегий управления выводом (принцип "*классной доски*", учёт наиболее длинного условия продукции, принцип "*новостки дня*" и т. п.).

Стратификация – упорядочение объектов системы по уровням (*стратам*) с целью создания *иерархического описания* системы.

Структура – отношения между элементами системы.

Структура глубинная – структура, отражающая семантику текста на ЕЯ. В глубинной структуре отсутствуют особенности морфологического или синтаксического строения ЕЯ. Глубинная структура обеспечивает понимание текстов на ЕЯ. От глубинной структуры легко осуществляется переход к представлениям знаний, которые используются в базе знаний.

Структура иерархическая – структура, элементы которой связаны между собой по иерархическому принципу, согласно которому элементы одного уровня подчинены элементам другого, расположенного выше уровня.

Структура когнитивная в ИИ – структура, обеспечивающая интеллектуальное “сознание” и интеллектуальную деятельность систем ИИ. В ИИ в этой функции используют преимущественно модели представления знаний: фреймы, сценарии, планы и т. п. В перспективе предусматривается, кроме знаний, привлекать к когнитивным моделям и некоторые другие когнитивные сущности: мысли, веру, эмоции, интуицию, интенции и т. п., то есть когнитивные единицы нерациональной природы.

Структура поверхностная – структурное представление высказывания или его фрагментов в генеративной грамматике и в многоуровневых языковых моделях типа “Смысл↔Текст”.

Структура таксономическая – иерархическая система понятий, связанная между собой отношением *is-a* («быть элементом класса»).

Структурирование (концептуализация) знаний – разработка неформального описания знаний о предметной области в виде графа, таблицы, диаграммы или текста, которое отражает основные концепции и взаимосвязи между понятиями предметной области.

Суждение – мысль о принадлежности или несущественности признака предмета.

Сущность – любой объект в предметной области.

Сценарий – в искусственном интеллекте – структурное описание действий или процесса с целью анализа или синтеза текстов на естественном языке.

Тезаурус – автоматизированный словарь, отражающий семантические отношения между лексическими единицами дескрипторного информационно-поискового языка и предназначенный для поиска слов по их смысловому содержанию.

Текст (Text) – 1) последовательность слов какого-то языка; 2) данные в форме литералов, символов, слов, фраз, параграфов, высказываний, таблиц или иных литеральных упорядоченных множеств, предназначенных для передачи смысла, толкование которых в значительной мере основано на знаниях читателем некоторого естественного языка или искусственного языка.

Теория – 1) система знаний, которая обобщает опыт общественно-производственной и научной деятельности людей, отражающий основные закономерности развития той или иной области материального мира и психики; 2) форма представления, использования и развития знаний, её наиболее существенные признаки – описание, объяснение, понимание и предсказание истинных сущностей материальных и идеальных объектов, явлений и процессов.

Теория аксиоматическая – логическая теория, в которой все исходные аксиомы являются тождественно истинными, а правила вывода – достоверными.

Теория логическая – теория, в основе которой лежит *формальная система*.

Терм – константа, переменная или выражение вида $f(t_1, t_2, \dots, t_n)$ где f – функциональный символ; t_i – термы.

Тестирование – выявление ошибок в подходе и реализации *прототипа* и разработка *рекомендаций* по доведению системы до промышленного варианта.

Технология информационная (*information technology*, ИТ) – 1) система *методов* и *способов* сбора, накопления, хранения, поиска, обработки и выдачи информации; 2) общий термин, используемый для ссылок на все технологии, связанные с созданием, обработкой, хранением, использованием, пересылкой и управлением информацией.

Технология новая информационная – технология обработки информации и решения задач с помощью информационной системы, опирающаяся на достижения искусственного интеллекта. Основной идеей, используемой в новой информационной технологии, является автоматизация процедуры построения программы, интересующей пользователя, на основании введенного им в систему описания постановки задачи на привычном для него профессиональном языке. Таким образом, новая информационная технология обеспечивает возможность общения с информационной системой *пользователя*, который не является профессиональным программистом. Для того, чтобы была реализована основная идея новой информационной технологии, необходимо, чтобы *информационная система* обладала *интеллектуальным интерфейсом*, *базой знаний* и *решателем*, т. е. была бы интеллектуальной системой. Другой чертой новой информационной технологии является распределённый способ решения задачи, когда пользователи, занятые решением общей задачи, общаются между собой через *сеть* информационных систем, электронную почту и *общую базу знаний*. В сеть входят также *базы данных*, из которых пользователи черпают информацию для решения своей задачи.

Технология семантическая – средства установления взаимосвязей между *терминологией* пользователя и терминологией, используемой в информационной системе.

Универсум – 1) область, на которой определены значения *предиката*. Из этой области берутся значения для означивания переменных, входящих в предикат. Для многоместных предикатов универсум есть *декартово произведение* областей определения для каждого из аргументов предиката; 2) надсистема, вершина *онтологического графа* категориального уровня.

Упорядоченность – определённое размещение *элементов*, которое является результатом упорядочения.

Управление знаниями (*Knowledge management*) – 1) совокупность средств и функций, обеспечивающих *пополнение*, *обновление* и *удаление знаний* из БЗ; 2) под управлением знаниями в общем случае понимается дисциплина, которая обеспечивает интегрированный подход к созданию, сбору, организации, доступу и использованию информационных ресурсов организации. Эти ресурсы включают в себя корпоративные базы данных, текстовую информацию, такую как документы, описывающие правила и процедуры, и, что наиболее важно, неявные знания и опыт

сотрудников организации; 3) совокупность *процессов*, управляющих *созданием, распространением, обработкой и использованием знаний*.

Факт – элемент или группа элементов *базы знаний* или *модели мира* интеллектуальной системы.

Фасет – группа *однородных терминов*, связанных общностью какого-либо признака. Служит средством построения информационно-поисковых языков фасетной структуры.

Формализация – процесс описания объекта исследования в виде, исключающем, по возможности, *неоднозначное* толкование данного объекта.

Формализация знаний – разработка *базы знаний* на языке *представления знаний*, которая соответствует заданной предметной области.

Фрейм – 1) в *искусственном интеллекте* – единица представления знаний, которая описывает понятие или объект. Фрейм состоит из ссылки на суперпонятие (*родовое понятие*) и описания *свойств*, отличающих данный объект от суперпонятия.

Целостность – сохранение информационного содержания и *однозначности интерпретации* данных в компьютере.

Центр виртуальный научно-исследовательский (ЦВНИ) – организационно-технологическая система, предназначенная для обеспечения эффективного *управления* процессами поддержки и развития *научных исследований*, высоких технологий и национальной экономики в целом. ЦВНИ содержит следующие основные технологические компоненты: компьютерные системы (суперкомпьютерные центры), средства обеспечения совместимости различных архитектурных платформ, аудио-, видеоаппаратуру, сенсорные станции, включая роботизированные камеры для пространственного мониторинга; научное оборудование, прежде всего уникальное, с дистанционным доступом к нему всех участников проекта; базы данных и знаний в предметной области и по конкретным проектам, включая результаты научных исследований и экспериментов; общее и специальное программное обеспечение, поддерживающее мультидисциплинарный процессинг в распределённой гетерогенной среде, и режимы объектно-ориентированной *виртуальной реальности*; информационно-телекоммуникационную инфраструктуру, которая учитывает особенности применяемых сетевых архитектур, конфигурацию и характеристики каналов связи.

Центр компьютерный – комплекс *компьютерных средств* (систем), помещения, в котором они располагаются, и обслуживающего персонала, обеспечивающего функционирование аппаратных и программных подсистем и взаимодействие с *пользователями*. Предназначен для оказания услуг по обработке данных.

Шкала Осгуда – шкала, использующая *пары слов-антонимов* (*острый – тупой, хороший – плохой* и т. п.), стоящие на концах шкалы. Средняя позиция шкалы является нейтральной. Кроме того, имеется ещё несколько промежуточных делений, как правило, не обозначаемых никакими словами. Шкала Осгуда используется в психологических экспериментах, связанных с выявлением особенностей индивидуального психосемантического пространства испытуемых.

Эвристика – метод решения задачи, основанный не на строгих математических моделях и алгоритмах, а на соображениях, которые сводятся к "*здравому смыслу*".

Экстенционал – совокупность объектов, обозначаемых определённой языковой единицей.

Язык естественный – 1) человеческая речь, возникшая естественным путём и используемая в общественной практике; 2) язык, правила которого основаны на текущем использовании без их точного предварительного описания.

Язык представления знаний – способ описания *моделей знаний* в системах представления знаний.

Язык представления знаний логический – язык представления знаний, в основе которого лежит *исчисление предикатов первого порядка*. Выражениями логического языка представления знаний являются синтаксически правильные формулы этого исчисления. В виде таких формул записываются все хранимые в системе *декларативные* и *процедурные знания*. Достоинством логического языка представления знаний является их полная формализуемость и наличие для них формальных процедур, позволяющих выполнять вывод и анализ таких характеристик записей, как непротиворечивость, эквивалентность и т. п. Недостатком логического языка представления знаний является плохая наглядность для пользователя.

Язык спецификации онтологий – Ontolingua, CycL, LOOM, OKBC, OCML, Flogic, RDF(S), DAML, OIL, OWL и др.

Язык формальный (*Formal language*) – язык, *синтаксис* и *семантика* которого заданы в явном виде до его использования.

Язык фреймовый – язык представления знаний и манипулирования знаниями, использующий в качестве модели знаний *фреймовые представления*. Наиболее известными фреймовыми языками являются языки FRL и KRL.

ЯОКМ (языково-онтологическая картина мира) – 1) *онтолого-лексикографическая* конструкция, отражающую все возможные отношения (взаимосвязи) между лексемами любого современного языка; 2) открытая, эксплицитно заданная на лексико-смысловом континууме *лексикографическая система*, в которой совокупность понятий формально обоснована и упорядочена в сложную *иерархическую структуру* по основным типам лексико-семантических отношений.

AAAI – American Association for Artificial Intelligence – Американская ассоциация искусственного интеллекта. Некоммерческая организация, занимается популяризацией ИИ, повышением уровня обучения в этой отрасли, проведением ежегодных конференций, издательской деятельностью и т. п. Основана в 1979 году.

DAG – directed acyclic graph – ориентированный ациклический граф.

DARPA – The Defense Advanced Research Projects Agency – агентство перспективных исследований МО США, выполняющее посреднические функции между этим ведомством и научно-исследовательскими организациями.

Data mining – *извлечение знаний* (по данным). Технология анализа хранилищ данных, основанная на методах ИИ и *инструментах поддержки принятия решений*.

Domain model – модель предметной области.

Framework – инфраструктура; каркас, основа; структура, строение.

Machine learning (ML) – 1) Одно из направлений искусственного интеллекта. *Научная дисциплина*, связанная с разработкой алгоритмов индуктивного вывода на

основе эмпирических данных, их классификации, выявлении скрытых закономерностей и *формировании знаний*. Обобщает результаты и идеи, связанные с нейросетевыми вычислениями, эволюционными и генетическими алгоритмами, нечёткими множествами и т. д.; 2) формирование знаний – процесс анализа данных и выявление скрытых закономерностей с использованием специального математического аппарата и программных средств.

Metaknowledge – *метазнания*. В ИИ – часть *базы знаний*, которая описывает её структуру.

OWL 1) (Ontology Web Language) – ориентирован на хранение и отображение большого количества разнородной информации и знаний, а также на поддержку семантики и связей между этими знаниями. Язык имеет три подмножества – OWL Lite, OWL DL (Description Logic) и OWL Full; 2) (Object Windows Library) – библиотека объектов Windows. Входит в Borland C++ 4.5.

RDF (Resource Description Framework) – *технология* (язык программирования). Разработан W3C для описания содержимого сайтов, более глобально – для построения семантической сети Интернет. Основывается на языке XML.

Superclass – суперкласс, *родительский класс*.

Text Mining – *процесс* обнаружения в “сырых” данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретаций знаний, необходимых для принятия решений в различных сферах человеческой деятельности.

W3C (WWW Consortium) – задача *консорциума* – создание и продвижение единых стандартов, которые повышают функциональность WWW.

СОДЕРЖАНИЕ

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ.....	3
ПРЕДИСЛОВИЕ	5
ВВЕДЕНИЕ.....	6
1. ИССЛЕДОВАНИЕ ПРЕДМЕТНО- ОРИЕНТИРОВАННЫХ ЗНАНИЙ.....	13
1.1. Онто-гносеологический анализ предметных знаний	13
1.1.1. <i>Общесистемный аспект исследования “пирамиды” знаний.....</i>	<i>14</i>
1.1.2. <i>Концептуальная парадигма работы со знаниями.....</i>	<i>18</i>
1.1.3. <i>Сущностный анализ понятия “знание”</i>	<i>21</i>
1.1.4. <i>Категориальный уровень знаний.....</i>	<i>25</i>
1.1.5. <i>Источники знаний и проблема формирования новых знаний.....</i>	<i>36</i>
1.2. Область исследования предметно-ориентированных знаний	39
1.3. Общая постановка проблемы	42
1.4. Классификация методов и средств обработки ЕЯО	43
1.5. Онтологический инжиниринг – перспективное направление предметных и междисциплинарных научных исследований	46
2. ЛОГИКО-МАТЕМАТИЧЕСКИЕ ОСНОВЫ	
ОНТОЛОГО-УПРАВЛЯЕМЫХ СИСТЕМ	48
2.1. Множества и отношения	48
2.1.1. <i>Интуитивное понятие множества</i>	<i>48</i>
2.1.2. <i>Операции над множествами и их свойства</i>	<i>51</i>
2.1.3. <i>Декартово произведение множеств. Отношения.....</i>	<i>53</i>
2.1.4. <i>Примеры отношений</i>	<i>55</i>
2.1.5. <i>Матрицы как пример отображения</i>	<i>64</i>
2.2. Частично рекурсивные функции. Тезис Чёрча	66
2.2.1. <i>Алгоритмические проблемы.....</i>	<i>70</i>
2.3. Формальные логические языки	70
2.3.1. <i>Синтаксис и семантика исчисления высказываний</i>	<i>71</i>
2.4. Синтаксис и семантика исчисления предикатов первого порядка.....	74
2.5. Выражение суждений на языке логики предикатов	76
2.6. Пропозициональная модальная логика	77
3. ФОРМЫ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ	82
3.1. Логический аспект представления знаний	82
3.2. Сравнительный анализ формальных логических языков	83
3.3. Формализованное представление ЕЯ-высказываний. Концептуальные графы	95
3.4. Формализованные подходы к извлечению знаний	107
4. КОМПЬЮТЕРНЫЕ ОНТОЛОГИИ ПдО.....	112
4.1. Системно-онтологический анализ ПдО. Понятие компьютерной онтологии.....	112
4.2. Лингвистические онтологии	116
4.2.1. <i>Сравнительный анализ онтологий.....</i>	<i>117</i>
4.2.2. <i>Языково-онтологическая картина мира.....</i>	<i>121</i>
4.3. Системная интеграция лингвистических и предметных онтологий.....	128
4.3.1. <i>Формализованные аспекты интеграции и её оценка.....</i>	<i>129</i>
4.3.2. <i>Практическая интеграция онтологий.....</i>	<i>130</i>

4.4. Методология разработки онтологии ПдО	132
4.4.1. <i>Общий подход к проектированию</i>	132
4.4.2. <i>Онтология объектов ПдО</i>	134
4.4.3. <i>Онтология процессов ПдО</i>	138
4.4.4. <i>Онтология задач проблемного пространства ПдО</i>	139
4.4.5. <i>Алгоритм проектирования онтологий</i>	141
5. АЛГЕБРО-ЛОГИЧЕСКИЕ И ЛИНГВИСТИЧЕСКИЕ	
МОДЕЛИ ОБРАБОТКИ ЕЯТ	144
5.1. <i>Общий подход к проблеме анализа ЕЯТ</i>	144
5.1.1. <i>Методы и модели анализа ЕЯТ</i>	147
5.1.2. <i>Общие принципы анализа и понимания ЕЯТ</i>	150
5.2. <i>Алгебро-логический подход к анализу и обработке ЕЯТ</i>	152
5.2.1. <i>Формальная постановка задачи анализа ЕЯТ</i>	152
5.2.2. <i>Алгебра списочных структур</i>	153
5.2.3. <i>Алгебраическая система списочных структур</i>	156
5.3. <i>Прикладные аспекты лингвистического анализа ЕЯТ</i>	158
5.3.1. <i>Информационная модель этапов лингвистического анализа</i>	158
5.3.2. <i>Графемно-морфологический анализ</i>	159
5.3.3. <i>Синтаксический анализ</i>	160
5.3.4. <i>Семантический анализ</i>	162
6. МЕТОДОЛОГИЧЕСКИЕ ОСНОВЫ ПРОЕКТИРОВАНИЯ ЗОИС С ОБРАБОТ-	
КОЙ ЕЯО	168
6.1. <i>Онтологический подход к проектированию ЗОИС</i>	168
6.2. <i>Онто-логический подход к разработке архитектуры ОУИС</i>	169
6.3. <i>Свойства онтолого-управляемой архитектуры</i>	175
6.4. <i>Языково-онтологическая информационная система</i>	178
6.5. <i>ОУИС с обработкой знаний предметной области</i>	182
6.5.1. <i>Архитектурно-структурная организация ОУИС</i>	184
6.5.2. <i>Обобщённый алгоритм проектирования ОУИС</i>	186
7. ИНСТРУМЕНТАЛЬНЫЙ КОМПЛЕКС ОНТОЛОГИЧЕСКОГО	
НАЗНАЧЕНИЯ	191
7.1. <i>Анализ функциональных возможностей средств построения онтологий</i>	191
7.2. <i>Инструментальный комплекс автоматизированного построения онтологий ПдО</i>	196
7.2.1. <i>Архитектурно-структурная организация</i> <i>и описание компонент ИКОН</i>	197
7.2.2. <i>Алгоритм автоматизированного построения онтологий ПдО</i>	204
7.3. <i>Методология системной интеграции междисциплинарных научных знаний</i>	207
7.4. <i>Перспективы исследования проблем системной интеграции</i> <i>онтологических знаний</i>	211
8. ПРИКЛАДНЫЕ ЗАДАЧИ	215
8.1. <i>Пример автоматизированного построения онтологической базы знаний</i> <i>и онтологии задач ПдО “Проектирование вычислительных устройств</i> <i>и систем на ПЛИС”</i>	215
8.2. <i>Применение онтологического подхода к разработке электронных</i> <i>курсов обучения</i>	231
8.2.1. <i>Проектирование онтологии электронного курса предметной дис-</i> <i>циплины “Базы данных”</i>	231
8.2.2. <i>Разработка ЭК “Общие основы построения знание-</i> <i>ориентированных компьютерных систем” в системе Moodle</i>	236
8.3. <i>Пример формализованного представления онтологии текстового документа</i>	247

8.4. Примеры разработки некоторых процедур Text Processing	251
8.4.1. Классификация (рубрикация) текстов из заданной ПДО	251
8.4.2. Обработка текстов библиографического характера	252
8.4.3. Знание-ориентированная поисковая система и релевантный поиск документов в сети Интернет	256
ЗАКЛЮЧЕНИЕ	261
ЛИТЕРАТУРА	262
Приложение А. ОПИСАНИЕ ОНТОЛОГИЙ МИКРОКОСМОС И SUMO.....	272
Приложение Б. МЕТОДИКА ПРОЕКТИРОВАНИЯ ОНТОЛОГИИ	
ПРЕДМЕТНОЙ ОБЛАСТИ	280
Приложение В. КРАТКИЙ ТОЛКОВЫЙ СЛОВАРЬ ПО ОНТОЛОГИЧЕСКОМУ ИНЖИ-	
НИРИНГУ	289

Наукове видання

ПАЛАГІН Олександр Васильович,
КРИВИЙ Сергій Лук'янович,
ПЕТРЕНКО Микола Григорович

**ОНТОЛОГІЧНІ МЕТОДИ ТА ЗАСОБИ
ОБРОБКИ ПРЕДМЕТНИХ ЗНАНЬ**
(Російською мовою)

В авторській редакції та комп'ютерній верстці авторів

Підписано до друку 21.01.2012.
Формат 70x108 1/16. Папір типогр. Гарнітура Times.
Друк офсетний. Умов. друк. арк. 28,4. Обл. вид. арк. 29,5.
Тираж 300 екз. Вид. № 2708. Замов. № .Ціна договірна.

Видавництво
Східноукраїнського національного університету імені Володимира Даля

Свідоцтво про реєстрацію: серія ДК №1620 від 18.12.2003.

Адреса видавництва: 91034, м. Луганськ, кв. Молодіжний, 20а
Телефон: 8 (0642) 41-34-12. **Факс:** 8 (0642) 41-31-60
E-mail: uni@snu.ua. **http://snu.edu.ua**