Тема 3.

Методы представления знаний

Онтологический подход

Термин «Онтология». Общие сведения

Происходит из греческих слов «*онтос*» - сущее и «*логос*» - наука.

Изначально «онтология» - философский термин, в общем подразумевающий науку об объектах окружающего мира и их взаимосвязях.

Основной вопрос онтологии «что существует?»

Термин был предложен Рудольфом Гоклениусом в его «Философском словаре».

Формально **онтология** состоит из понятий, организованных в *таксономию*, их описания и правил вывода.

Таксономия (греч. «таксис» - порядок, «номос» - закон) — термин также имеющий отношение к онтологии, поскольку представляет собой науку о принципах классификации и систематизации.

Математически таксономией является древообразная структура классификаций определённого набора объектов с увеличивающейся степенью детализации и специфичности.

Часто термины «таксономия» и «систематика» рассматриваются как синонимы.

Не путать с **а**нтологией (сборник лучших произведений)

Онтологический подход представления знаний

Термин «онтология» в ИИ употребляется в контексте с такими понятиями как концептуализация, знания, модели знаний, системы, основанные на знаниях

Под концептуализацией понимается процесс перехода от представления проблемной области на естественном языке к точной спецификации этого описания на некотором формальном языке, ориентированном на компьютерное представление.

Концептуализация также трактуется как результат подобного процесса, т.е. описание множества понятий (концептов) предметной области, знаний о них и связях между ними.

Проще говоря, онтология — это формально представленные на базе концептуализации знания о предметной области.

Самым распространённым является определение, согласно которому **онтология есть точная (выраженная формальными средствами) спецификация концептуализаци**

С этой точки зрения каждая БЗ, система, оснванная на знаниях, или программный агент явно или неявно базируются на некоторой концептуализации.

Множества понятий и отношений между ними отражаются в словаре. Такми образом, считается что основу онтологии составляют множества представленных в ней терминов.

В простейшем случае онтология описывает только иерархию концептов, связанных отношениями категоризации.

В более сложных случаях в нее также включаются аксиомы для выражения других отношений между концептами и организации их интерпретации.

Независимо от вида онтологии она должна включать **словарь терминов** и некоторые спецификации их значений, что позволяет ограничивать возможные интерпретации терминов и отражать взаимосвязь понятий предметной области.

При таком подходе онтология похожа на инвестное понятие **тезауруса**.

Выделяют 6 интерпретаций понятия «онтология»:

- неформальная концептуальная схема
- формальный взгляд на семантику
- спецификация концептуализации
- представление концептуальной схемы через логическую теорию
- словарь, используемый логической теорией
- метауроневая спецификация логической теории

Задачи, решаемые с помощью онтологий:

1. создание и использование БЗ

2. Организация эффективного поиска в БД

3. создание систем, реализующих механизмы рассуждений

4. Организация поиска по смыслу в текствовой информации

5. Семантический поиск в Интернете

6. Представление смысла в метаданных об информационных ресурсах

7. построение и использование баз общих знаний для различных интеллектуальных систем

8. Обеспечение общей терминологии для множества специалистов и совместно используемых приложений

9. Многократное применение БЗ и информационных массивов, предоставляющих сведения о технических системах на различных стадиях их жизненного цикла.

1. Создание и использование БЗ

Онтологии позволяют формировать модели предметной области, интегрируя декларативные описания и определения.

Выделяют следующие основные требования экспертов в прикладных областях к средствам построения онтологий:

1. близость языка, которым опрерирует система, основанная на знаниях, к языку специалиста предметной области;

2. возможность использования введенных знаний для решения большинства предметных задач, а не формирование БЗ заново каждый раз для постановки и решения новой задачи;

3. открытость языка, т.е. возможность включения в него новых языковых конструкций, которые появляются в данной предметной области;

Требования со стороны разработчиков и программистов:

1. универсальность языка, т.е. возможность представления знаний разного типа независмо от предметной областию;

2. унифицированность языковых конструкций, обеспечивающая возможность разным ИИС обмениваться знаниями;

3. наличие эффективных алгоритмов разбора языковых выражений;

Существует несколько **языков представления зняний**.

К их числу относится получивший достаточное распространение *Knowledge Interchange Format (KIF)*, предназначенный для межмашинного обмена знаниями и взаимодействия программ.

2. Организация эффективного поиска в БД

Использование онтологий позволяет точнее **интерпретировать смысл терминов**, фигурирующих в запросах, а также **дополнять или расширять запрос** понятиями, которые связаны с терминами запроса.

В современных поисковых машинах Интернет-онтологии испльзуются для уточнения смысла запросов путем «фильтрации» их содержания, что способствует уменьшению информационного шума.

Для этого применяются т.н. профили информационных интересов пользователей и процедуры семантического пересечения запроса или информации, приготовленной к выдаче, с этими профилями.

3. Создание систем, реализующих механизмы рассуждений (ЭС, системы управления, интеллектуальные роботы, и др.)

Обязательным компонентом таких систем является **блок объяснения решения**.

Как при принятии решения, так и при объяснении должна учитываться семантика как отдельных терминов, так и составленных из них высказываений и их композиций.

Достижению данной цели способствует использованеи онтологий

4. Организация поиска по смыслу в текствовой информации

Для организации **поиска по смыслу** в текствовой информации необходимы методы извлечения семантики из текстовых документов и запросов и споставления получаемых семантических представлений.

Новыми задачами, связанными с **извлеченем знаний из текста**, являются

1. Формирование сообщений на заданную тему;
2. Извлечение новых фактов по интересующей теме;
3. Реализация виртуального собеседника;

5. Семантический поиск в Интернет

Онтологии позволяют формировать **информационные профили узлов сети** и на этапе предварительного отбора подходящих для поиска узлов отсеивать нерелевантные узлы.

6. Представление смысла в метаданных об информационных ресурсах

Современные языки представления метаданных, как правило, строятся на базе языка *XML* и модели *RDF* . В

рамках данной задачи онтологии применяются при **формировании пространств имен**, словарей и квалификаторов для обеспечения их единообразных интерпретаций

В модели *RDF* используется ОО **сисема классов**. Базовая модель включает три типа объектов:

1. ресурс, идентифицируемый URI
2. свойство описывающее ресурс (ссылка на другой ресурс или описание ресурса)
3. утвреждение, состоящее из ресурса-субъекта, свойства (ресурса-объекта) и предиката, связывающего их и представляющего значения свойства

Методогия управления знанияниями при использовании онтологического подхода позволяет решать заадчи **каталогизации и классификации информационных ресурсов**.

7. Построение и использование баз общих знаний для различных интеллектуальных систем

Человек в процессе рассуждений использует не только знания предметной области, но и знания более высокой степени общности.

К таким знаниям относят **описания свойств** пространства, времени, личности и т.п. **Знания верхнего уровня** позволяют доопределить модели конкретных предметных ситуаций с учетом взглядов и роли человека.

Эти знания представляются в онтологиях верхнего уровня (**общих онтологиях, онтологиях общих знаний**)

Степень общности отражаемых знаний служит основанием для выделения трех уровней онтологий:

1. Общих онтологий
2. Предметных онтологий
3. Онтологий задач

8. Обеспечение общей терминологии для множества специалистов и совместно используемых приложений

Онтологический подход упрощает решение проблемы взаимодействия специалистов различного уровня и различной предметной подготовки при решении различных задач.

9. Многократное применение БЗ и информационных массивов, предоставляющих сведения о технических системах на различных стадиях их жизненного цикла.

В данном случае онтология описывает техническую систему на разных этапах ее жизненного цикла и выступает в качестве базиса для построения и трансформации информационных моделей, БД и документов.

Наличие онитологии обеспечивает их согласование и адекватное понимание.

Модель онтологии

Определение онтологии как формального представления предметной области, построенного на базе концептуализации, предполагает выделение её трёх взаимосвязных компонентов:

- таксономии терминов,
- описаний смысла терминов,
- а также правил их использования и обработки.

Таком образом, модель онтологии задает тройка:

$$O = (X, R, \Phi)$$

где *X* – конечное множество концептов (понятий, терминов) предметной области, которуб представляет онтология,

R – конечное множество отношений между ними

Ф — конечное множество функций интерпретации, заданных на концептах и(или) отношениях

Если

$$R = \varnothing$$

и

$$\Phi = \varnothing$$

То онтология вырождается в модель простого словаря:

$$O = (X, \varnothing, \varnothing) = V$$

Другой крайний случай имеет место, когда

$$R = \varnothing$$

а

$$\Phi \neq \varnothing$$

В этом случае множество терминов *X* разбивается на два класса

$$X = X_1 \cup X_2, \quad X_1 \cap X_2 = \varnothing$$

где

X_1 - множество интепретируемых терминов,

X_2 - множество интерпретирующих терминов.

Таким образом, смысл термина x интерпретируется каждый раз при обращении к нему

Методики построения онтологий и требования к средствам их спецификации

В настоящее время известен только один стандарт, регламентирующий процесс разработки онтологий и связанных с этим исследований: *IDEF5*.

Несмотря на это существует множество предложений по методикам разработки онтологий, в рамках которых оыбчно выделяют следующие основные задачи:

1. Анализ целей создания и области применения создаваемой онтологии.

2. Построение онтологии

    2.1 Сбор и фиксация знаний о предметной области

    2.2 Кодирование

2.1 Сбор и фиксация знаний о предметной области включает:

- определение основных понятий и их взаимоотношений в выбранной ПО
- Создание точных непротиворечивых определений для каждого основного понятия и отношения
- Определение терминов, которые связаны с основными понятиями и отношениями.
- Согласование перечисленных компонентов онтологии

2.2 Кодирование включает:

- разбиение совокупности основных терминов, используемых в онтологии, на классы
- выбор или разработку специального языка для представления знаний.
- формирование концептуализации в рамках выбранного языка представления знаний

Стандарт онтологического исследования *IDEF5* подготовлен компанией *Knowledge Base Systems, Inc*. в качестве проекта национального стандарта США.

Процесс построения онтологии в рамках *IDEF5* состоит из пяти основных этапов.

1. Изучение и систематизация начальных условий. Этот этап устанавливает основные цели и контекст разработки онтологии, а также распределяет роли членов проекта

2. Сбор и накопление данных для построения онтологии.

3. Анализ и группировка собранных данных для облегчения согласования терминологии.

4. Нчальное развитие онтологии. На этом этапе формируется предварительная онтология на основе систематизированных данных.

5. Уточнение и утверждение онтологии (заключительный этап).

Для поддержки процесса построения онтологий в *IDEF5* определены специальные онтологические языки:

- схематический язык (*Schematic Language – SL*)
- язык доработок и уточнений (*Elaboration Language - EL*)

язык SL является наглядным графическим языком.

Обеспечивает формирование начального представления онтологии.

Он включает в себя разнообразные типы диаграмм и схем, служащих для визуального представления основной онтологической информации.

Язык EL является структурированным текстовым языком, позволяющим детализиорвать элементы онтологии.

В стандарте *IDEF5* предусмотрены **четыре основных вида схем** для представления онтологической информации в наглядной графической форме

1. Диаграммы классификации
Служат средством логической систематизации знаний, наколенных при изучении системы. Существуют два типа таких диаграмм:

- диаграмма строгой классификации (*Description Subsumption – DS*) и
- диаграмма естественной или видовой классификации (*Natural Kind Classification – NKC*)

2. Композиционные схемы
Composition Schematics — служат для графического представления **состава классов онтологии**. В частности, с помощью них можно наглядно отобразить сосотав объектов, относящихся к тому или иному классу.

3. Схемы взаимосвязей
Relation Schematics – позволяют оразработчикам визуализировать и изучать **связи между различными классами** объектов системы. В некоторых случаях эти схемы используются для представления зависимостей между взаимосвязями классов

4. Диаграмма состояния объекта
Object State Schematics – позволяет описать **процесс изменения состояния объекта**. С объектом могут произойти два типа изменений: он может поменять либо своё состояние, либо класс.

Стандарт *IDEF5* отражает методологию, с помощью которой можно наглядно и эффективно разработывать онтологии.

На сегодняшний день существуют единичные программные средства, поддерживающие *IDEF5*, кроме того, данный стандарт охватывает не все этапы создания онтологий.

Существует еще одна методология построения онтологий:

Ontology Design Environment (инструментальая среда проектирования онтологий).

Ontology Design Environment включет подсистемы управления проектом и поддержки разработки.

Первая подсистема обеспечивает решение задач планирования, контроля за ходом выполнения проекта и управления качеством.

Вторая ориентирована на на задачи приобретения знаний, их оценки, интеграции, документирования и управления конфигурациями.

Процесс разработки онтологии включает 4 стадии:

1 Спецификаця
2 Концептуализация
3. Формализация
4. Реализация

Спецификации подлежат цели создания онтологии, ее предполагаемое применение и потенциальные пользователи.

Концептуализация обеспечивает структурирование предметных знаний в камках эксплицитной модели.

На этапе реализации создается вычислительная модель, соответствующая онтологии и выражаемая на одном из языков представления знаний.

Концептуализация квлючает в себя два этапа, в рамках которых решаются задачи:

1. Построение глоссария терминов
2. Построение классификационных деревьев концептов.

Вторая задача начинает решаться тогда, когда объем глоссария по мнению экспертов достигает существенного объёма.

Затем для каждого классификационного дерева формируется словарь концептов и совокупность таблиц, описывающих бинарные отношения между концептами, экземпляры, атрибуты экземпляров и классов, логические аксиомы, константы и формулы.

Программные решения для создания ИИС на основе онтологического подхода разрабатывает комания *Ontoprise GmbH*.

Пакет *Smarter Knowledge Suite* состоит из следующих компонент:

1. Редактор онтологий *OntoEdit*
2. Машину вывода для онтологий *OntoBroker*
3. Интеллектуальную поисковую систему *SemanticMiner*
4. Систему *OntoOffice*, являющуюся интеллектуальной поисковой надстройкой над пакетом MS Office.

Структура онтологии, которой оперирует OntoEdit:

Ядро онтологии — иерархия концептов (абстрактных понятий ПО).

Иерархические отношения: тип «род-вид» (механизмы наследования)

Прочие иерархические отношения описываются бинарными отношениями.

Концептам приписываются атрибуты, которые рассматриваются как отношения определенного типа.

Онтология содержит аксиомы в виде правил, выраженных на основе отношений:

ЕСЛИ *X* <является отцом> *Y* И *Y* <имеет пол> <мужской>, ТО *Y* <является сыном> *X*

Система OntoBroker

Реализует машину дедуктивного вывода и интерфейс выполнения запросов, оперирующие с онтологией.

В основе функционирования — анализ и интерпретация аксиом.

Механизмы вывода не зависят от последовательности применения правил и последовательности утверждений внутри правил.

Реализована на базе Java:

- Может применяться в качестве автономного приложения
- Сервера выполнения запросов
- Библиотеки для средств дедуктивного вывода

Интеллектуальная ИПС SemanticMiner:

Построена в рамках архитектуры клиент-сервер.

Функции системы:

- Извлечение знаний из различных источников (текстов, БД, web-страниц, и т. д.)
- Формирование БЗ и ее визуализация
- Автоматическая кластеризация документов
- Поиск в БЗ и навигация по ней

Наиболее известные онтологические проекты

Среди проектов онтологий верхнего уровня существенные результаты достигнуты в рамках «***Cyc***» корпорации *Cycorp* (Техас, США)

Проект направлен на формирование информационно-логической основы для создания и функционирования систем, реализующих механизм рассуждений.

Онтология «***Cyc***» базируется на ядре, включающем миллион утверждений, веденных в БЗ вручную.

Проект с открытым исходным кодом «*OpenCyc*» содержит описания 6 000 концептов и свыше 60 000 аксиом, отражающих непротиворечивые знания о мире.

Модель GUM (Generalized Upper Model)

Онтология верхнего уровня.

Цель проекта — поддержка обработки естественного языка (EN, DE, IT).

Модель развивает идеи онтологии Penman Upper Model.

Уровень абстракции онтологии GUM лежит лежит между лексическими и концептуальными знаниями.

Онтологии предметного уровня

Модель ***TOVE*** (Toronto Virtual Enterprise).

Цели проекта:

- Создание средств представления общей терминологии для ПО, приложения которой могут совместно использоваться и адекватно пониматься каждым членом сообщества,
- Построение точных непротиворечивых определений терминов на основе логики первого порядка,
- Обеспечение возможностей описания семантики с помощью множества аксиом, которые автоматически позволяют получать ответы на вопросы о ПО.

Проект *KACTUS*

Цель проекта: создание методологии многократного применения знаний о технических системах на протяжении их жизненного цикла.

- Производственные и технологические методы,
- Методы инженерии знаний на базе создания онтологической и вычислительной основы для многократного использования полученных знаний разными приложениями.

Основным формализмом в KACTUS является язык CML (Conceptual Modeling Language)

В рамках проекта создан комплекс инструментальных средств для:

- Просмотра,
- Редактирования,
- Управления онтологиями

Проект *(KA)*^{2}

Knowledge Annotation Iniative of the Knowledge Community — аннотация знаний сообществом приобретения знаний

Проект направлен на развитие технологий интеллектуального поиска в Internet и автоматического извлечения знаний из Web-ресурсов.

3 направления исследований:

- Онтологический инжиниринг,
- Автоматическое аннотирование web-страниц,
- Построение и выполнение запросов, относящихся к информации представленной на web-страницах, и вывод ответов с использованием онтологических знаний

В рамках ***(KA)²*** предполагается разработать достаточно общую систему отнологий с помощью средств Ontolingua.

На сегодняшний день построено 8 онтологий:

- Онтологии организации,
- Проекта,
- Персоны,
- направления исследований,
- Публикации,
- События
- Исследовательского продукта,
- Исследовательской группы

Web-страницы с экземплярами онтологий аннотируются с помощью нового типа HTML-тега «**ONTO**», содержимое которого обрабатывается системой **Ontocrawler**.

Система Ontocrawler создается в рамках проекта **Ontobroker *(KA)*^{2}**

Ontobroker включает 3 основные подсистемы:

- Интерфейс построения запросов (query interface),
- Машину вывода ответов (inference engine),
- Машину доступа к Internet-ресурсам, служащую для извлечения знаний из Internet и их накопления.

Система **Ontolingua**, разработанная в Knowledge System Laboratory (KSL) (Стенфордский университет), представляет собой инструментальное средство для совместного формирования, просмотра, редактирования и использования онтологий в среде WWW.

Реализует принципы ОО подхода и является расширением языка KIF.

В рамках проекта ***SHOE*** (Simple HTML Ontology Extensions) создаются средства аннотирования web-страниц, позволяющие вносить в них семантическое содержание, доступное для обработки интеллектуальными агентами.

Основным компонентом SHOE является онтология, представляющая сведения о некоторой ПО.

Примеры использования онтологий для расширения поисковых запросов

Пример1. Запрос: «*Какие существуют фазовые состояния?*»

Для расширения запроса система использует отношение род-вид (**R1**), экземпляры которого представлены в онтологии:

R1(«*фазовое состояние*», «***твёрдое вещество***»);

R1(«*фазовое состояние*», «***жидкость***»);

R1(«*фазовое состояние*», «***газ***»);

R1(«*фазовое состояние*», «***плазма***»);

ответом на запрос служит множество терминов :
{«***твёрдое вещество***», «***жидкость***», «***газ***», «***плазма***»}

Пример 2.

Исходный запрос: «*электрическая ёмкость*».

Для расширения запроса система использует отношение синонимии **Rs**, отражённое в онтологии:

Rs(«*элетрическая ёмкость*», «***электрический конденсатор***»);

Расширенный запрос имеет вид: «*элетрическая ёмкость*»***И***«*электрический конденсатор*»;

Пример3.

Исходный запрос: «*электромагнитная индукция*».

Для расширения запроса система использует отношение ассоциации **Ra**, отражённое в онтологии:

Ra(«*электромагнитная индукция*», «***Фарадея-Максвелла — Ленца закон***»);

Расширенный запрос имеет вид: «*элетрическая ёмкость*»***И***«*Фарадея-Максвелла — Ленца закон*»;

Рекомендации по описанию онтологий

В литературе по искусственному интеллекту содержится много определений понятия онтологии.

В данном случае под онтологией будем понимать формальное явное описание

- **понятий** в рассматриваемой предметной области (**классов** - иногда их называют понятиями),
- **свойств** каждого понятия, описывающих различные свойства и атрибуты понятия (**слотов** - иногда их называют ролями или свойствами),
- **ограничений**, наложенных на слоты (иногда их называют ограничениями ролей).

Онтология вместе с набором индивидуальных экземпляров классов образует базу знаний.

На практике разработка онтологии включает:

- определение классов в онтологии;
- расположение классов в таксономическую иерархию (подкласс – надкласс);
- определение слотов и описание допускаемых значений этих слотов;
- заполнение значений слотов экземпляров.

После этого можно создать базу знаний, определив отдельные экземпляры этих классов, введя в определенный слот значение и дополнительные ограничения для слота.

Методология инженерии знаний

Не существует единственного «правильного» способа или методологии разработки онтологий.

Обсудим общие моменты, которые нужно учитывать, и рассмотрим один из возможных способов разработки онтологии.

Описывается итеративный подход к разработке онтологии:

- начинается с первого чернового просмотра онтологии.
- Затем проверяется и уточняется получаемая онтология и добавляются детали.

Попутно обсуждаются решения, касающиеся моделирования, которые должен принять разработчик, а также «за» и «против» и результаты принятия различных решений.

Во-первых, выделим некоторые фундаментальные правила разработки онтологии, к которым будем неоднократно обращаться. Эти правила могут показаться довольно категоричными. Тем не менее, во многих случаях они могут помочь принять проектные решения.

1) Не существует единственного правильного способа моделирования предметной области – всегда существуют жизнеспособные альтернативы. Лучшее решение почти всегда зависит от предполагаемого приложения и ожидаемых расширений.

2) Разработка онтологии – это обязательно итеративный процесс.

3) Понятия в онтологии должны быть близки к объектам (физическим или логическим) и отношениям в интересующей вас предметной области. Скорее всего, это существительные (объекты) или глаголы (отношения) в предложениях, которые описывают вашу предметную область.

Шаг 1. Определение области и масштаба онтологии

Начало разработки онтологии предлагается с определения ее области и масштаба. То есть, с ответа на несколько основных вопросов:

1. Какую область будет охватывать онтология?

2. Для чего мы собираемся использовать онтологию?

3. На какие типы вопросов должна давать ответы информация в онтологии?

4. Кто будет использовать и поддерживать онтологию?

Ответы на эти вопросы могут измениться во время процесса проектирования онтологии, но в любой заданный момент времени они помогают ограничить масштаб модели.

Шаг 2. Рассмотрение вариантов повторного использования существующих онтологий

Повторное использование существующих онтологий может быть необходимым, если проектируемой системе нужно взаимодействовать с другими приложениями, которые уже вошли в отдельные онтологии или контролируемые словари.

Многие онтологии уже доступны в электронном виде и могут быть импортированы в используемую среду проектирования онтологии.

Формализм онтологии часто не имеет значения, т.к. многие системы представления знаний могут импортировать и экспортировать онтологии.

Даже если система представления знаний не может работать напрямую с отдельным формализмом, задача перевода онтологии из одного формализма в другой обычно не является сложной.

Шаг 3. Перечисление важных терминов в онтологии

Полезно составить список всех терминов, о которых мы хотели бы сказать что-либо или которые хотели бы объяснить пользователю.

1. Какие термины мы бы хотели рассмотреть?
2. Какие свойства имеют эти термины?
3. Что бы мы хотели сказать об этих терминах?

В начале важно получить полный список терминов, не беспокоясь о пересечении понятий, которые они представляют, об отношениях между терминами, о возможных свойствах понятий или о том, чем являются понятия – классами или слотами.

Следующие два шага – разработка иерархии классов и определение свойств понятий (слотов) – тесно переплетены.

Обычно в иерархии даётся несколько формулировок понятий и затем описываются свойства этих понятий и т.д. Также эти два шага – самые важные шаги в процессе проектирования онтологии.

Шаг 4. Определение классов и иерархии классов

Существует несколько возможных подходов для разработки иерархии классов.

- Процесс нисходящей разработки начинается с определения самых общих понятий предметной области с последующей конкретизацией понятий.

- Процесс восходящей разработки начинается с определения самых конкретных классов, листьев иерархии, с последующей группировкой этих классов в более общие понятия.

- Процесс комбинированной разработки – это сочетание нисходящего и восходящего подходов: Сначала определяются более заметные понятия, а затем они соответствующим образом обобщаются и ограничиваются.

Ни один из этих трех методов не лучше других по своей сути. Выбор подхода в большой степени зависит от личного взгляда на предметную область. Если разработчик склонен к рассмотрению предметной области сверху вниз, то ему, возможно, больше подойдет нисходящий метод.

Шаг 5. Определение свойств классов – слотов

Классы сами по себе не предоставляют достаточно информации для ответа на вопросы проверки компетентности из Шага 1.

После определения некоторого количества классов требуется описать внутреннюю структуру понятий.

Выбраны классы из списка терминов, который был создан на Шаге3. Большинство оставшихся терминов, вероятно, будут свойствами этих классов.

Для каждого свойства из списка требуется определить, какой класс оно описывает. Эти свойства станут слотами, привязанными к классам.

Вообще, в онтологии слотами могут стать несколько типов свойств объектов:

- «внутренние» свойства;

- «внешние» свойства;

- части, если объект имеет структуру; они могут быть как физическими, так и абстрактными «частями»;

- отношения с другими индивидными концептами; это отношения между отдельными членами класса и другими элементами

Шаг 6. Определение фацетов (ограничений) слотов

Слоты могут иметь различные фацеты, которые описывают тип значения, разрешенные значения, число значений (мощность) и другие свойства значений, которые может принимать слот.

Несколько общих фацетов:

- **Мощность слота**: определяет, сколько значений может иметь слот

- **Тип значения слота:** какие типы значений можно ввести в слот
  - Строка
  - Число
  - Булевы слоты
  - Нумерованные слоты (список конкретных значений)
  - Слоты-экземпляры (отношения между индивидными концептами)
  -
- **Домен слота и диапазон значений слота:** Разрешенные классы для слотов типа **Экземпляр** часто называют диапазоном значений слота. Некоторые системы позволяют ограничить диапазон значений слота, если слот привязан к определенному классу. Классы, к которым слот привязан, или классы, свойство которых слот описывает, называются **доменом слота**.

Шаг 7. Создание экземпляров

Последний шаг – это создание отдельных экземпляров классов в иерархии. Для определения отдельного экземпляра класса требуется

1) выбрать класс,

2) создать отдельный экземпляр этого класса и

3) ввести значения слотов.