

# Онтологии и онтологическое моделирование: основные понятия

Материалы курса  
*«Моделирование процессов и систем»*

*Лядова Л.Н.*

# Онтология: ???

Слово «онтология» имеет два значения:

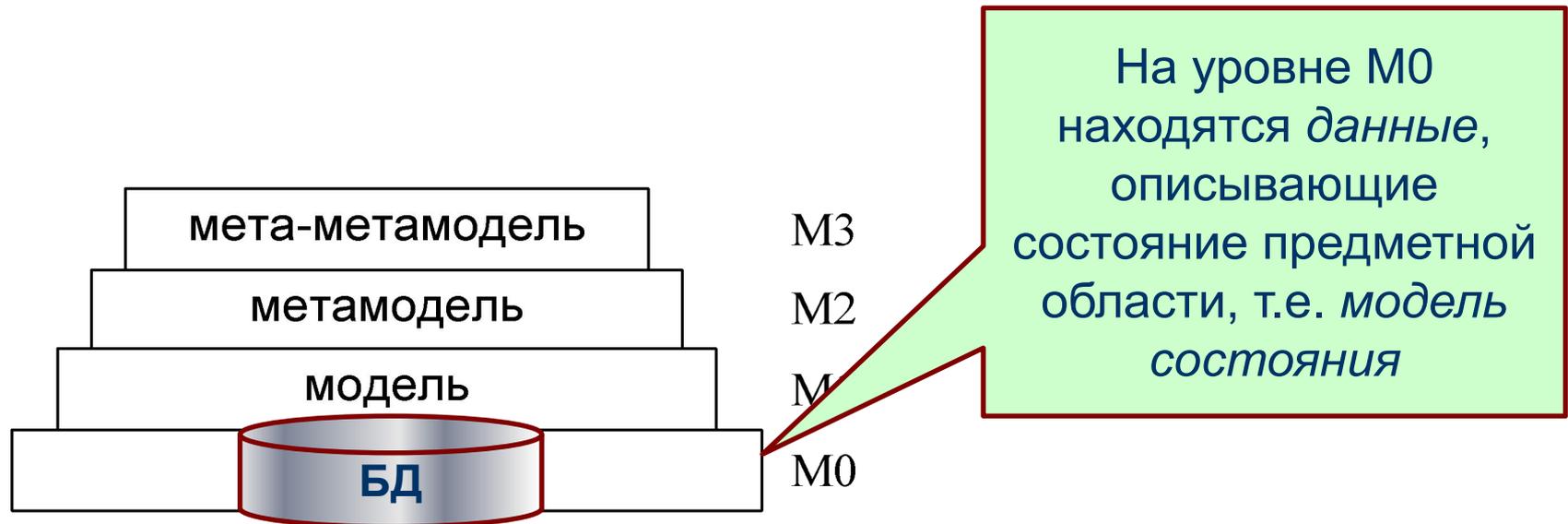
**Онтология** – *философская дисциплина*, которая изучает наиболее общие характеристики бытия и сущностей.

**Онтология** – артефакт, структура, описывающая значения элементов некоторой системы.

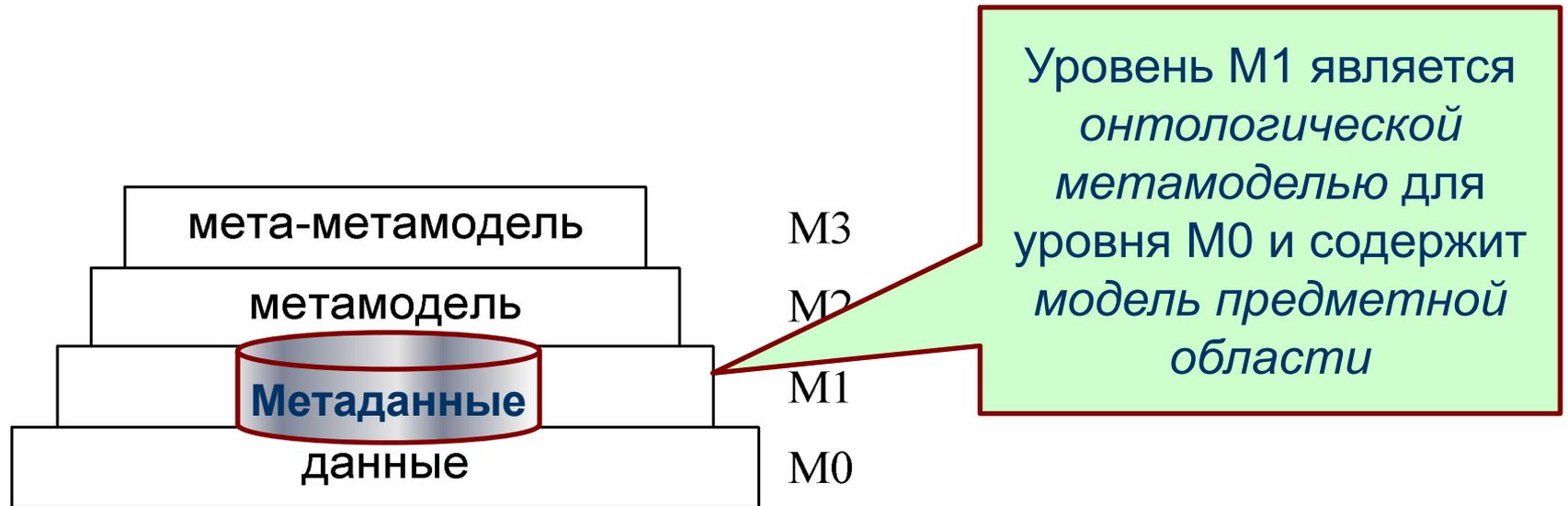
# Моделирование и метамоделирование: классическая четырехуровневая иерархия моделей



# Моделирование и метамоделирование: классическая четырехуровневая иерархия моделей



# Моделирование и метамоделирование: классическая четырехуровневая иерархия моделей



# Моделирование и метамоделирование: классическая четырехуровневая иерархия моделей



Уровень M2 определяет лингвистическую метамодель для уровней M1 и M0. Другими словами, на уровне M2 находится модель языка моделирования, с которым работают аналитики, разработчики, CASE-средства и пр.

# Моделирование и метамоделирование: классическая четырехуровневая иерархия моделей



Уровень M3 определяет язык, на котором описываются метамодели уровня M2 (обычно описывается «на самом себе»).

# Онтологическое моделирование и метамодели

**Определение.** *Лингвистическая метамодель* – это метамодель, которая описывает предметно-независимый язык моделирования.



**Определение.** *Онтологическая метамодель* – это метамодель, которая описывает предметно-зависимый язык моделирования.



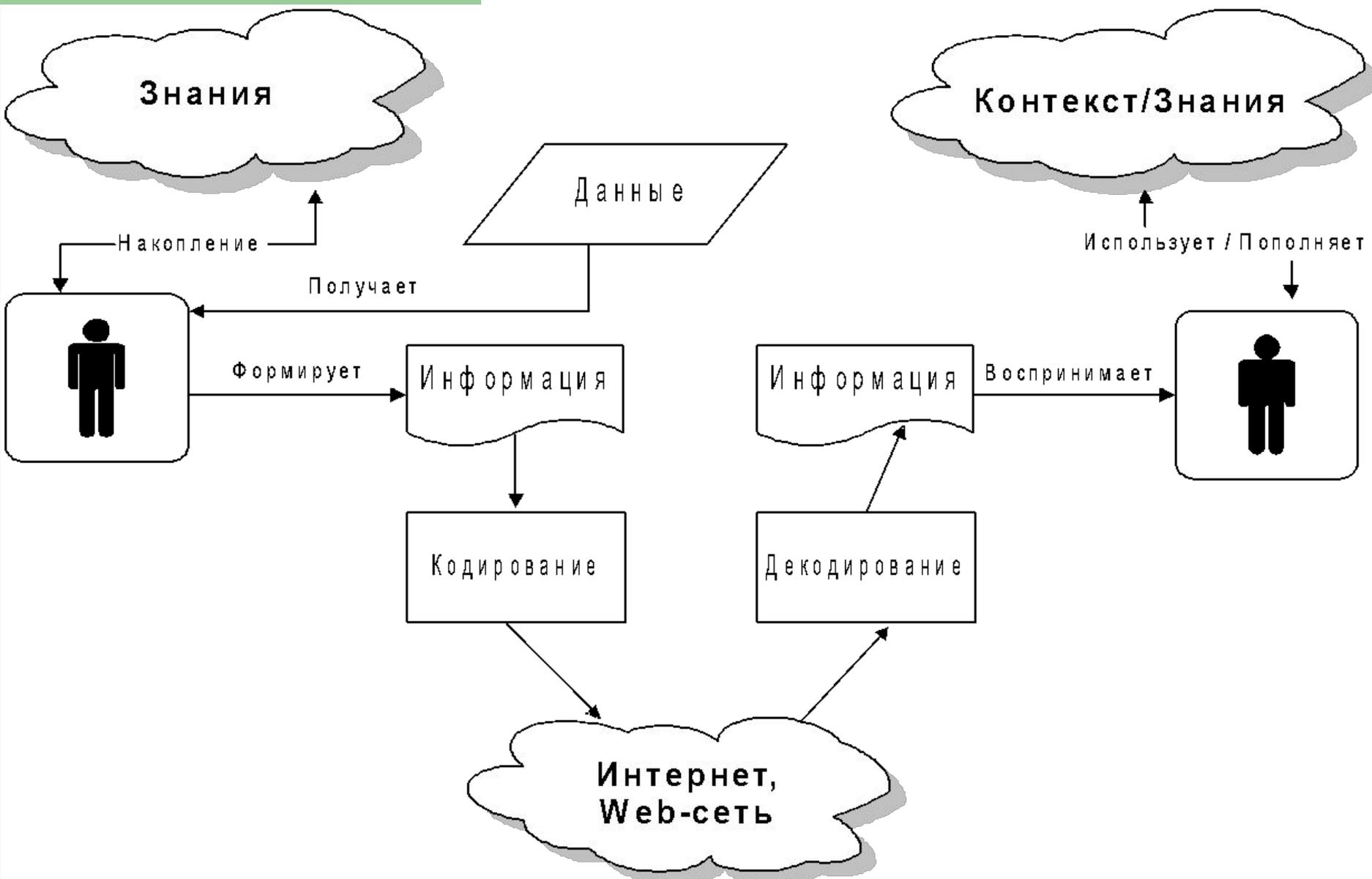
## Понятия онтологии

**Онтология** – это спецификация концептуализации (Н. Грубер).  
(*Концептуализация* – это структура реальности, рассматриваемая независимо от словаря предметной области и конкретной ситуации.)

**Онтология** – это структура данных с заданными в ней символами, позволяющими представлять концептуализации для обработки компьютерными программами (Э. Хов).

**Онтология** – это спецификация некоторой предметной области, которая включает в себя словарь терминов (понятий) предметной области и множество связей между ними, которые описывают, как эти термины соотносятся между собой в конкретной предметной области.

**Онтология** – это БЗ специального типа, которые могут «читаться» и пониматься, отчуждаться от разработчика и/или физически разделяться между пользователями (Гаврилова Т.А., Хорошевский Ф.В., 2001).



## Понятие онтологии: *определение концептуализации*

*Онтология* – это формальное, явное, точное определение (спецификация) совместно используемой концептуализации.

*Концептуализацией* именуется абстрактное (упрощённое) представление мира, которое формируется для некоторых целей.

*Концептуализация* – это структура реальности, рассматриваемая независимо от словаря предметной области и конкретной ситуации.



## Понятие онтологии: определение концептуализации

*Онтология* – это формальное, явное, точное определение (спецификация) совместно используемой концептуализации.

*Концептуализацией* именуется абстрактное (упрощённое) представление мира, которое формируется для некоторых целей.

*Например, если мы рассматриваем простую предметную область, описывающую размещение объектов на карте, то концептуализацией является «контурная карта», показывающая возможное положение объектов, а не конкретное расположение конкретных объектов в текущий момент времени, или «генеральный план», но не его конкретную реализацию.*

## Понятие онтологии: детализация

**Онтология** – это *формальное, явное, точное определение (спецификация) совместно используемой концептуализации.*

**Концептуализацией** именуется абстрактное упрощенное представление мира, которое формируется для некоторых целей.

Онтология является *точным определением* (спецификацией) потому, что она представляет концептуализацию в конкретной форме.

Она является *явной*, потому что все используемые в ней ограничения явно определены.

Слово *формальная* означает, что онтология должна пониматься машиной – должна быть представлена в форме, пригодной для компьютерной обработки.

«*Совместно используемая*» указывает на то, что онтология содержит *согласованные знания.*

## Понятие онтологии: детализация

Неформально, онтология представляет собой некоторое *описание взгляда на мир применительно к конкретной области интересов предметной области.*

Это описание состоит из терминов и правил использования этих терминов, ограничивающих их значения в рамках конкретной области.

Таким образом, онтология представляет собой *модель предметной области.*

На формальном уровне, онтология – это система, состоящая из набора понятий и набора утверждений об этих понятиях, на основе которых можно строить классы, объекты, отношения, функции и теории предметных областей.

## Понятие онтологии: *компоненты онтологии*

Основными *компонентами* онтологии являются:

- *Классы*, или *понятия* – «символы» онтологии.
- *Отношения* – связи между понятиями.
- *Функции* – особый вид отношений, в котором элемент отношения однозначно определяется предшествующими элементами.
- *Аксиомы* представляют тип взаимодействия (связи) между понятиями предметной области, правила, ограничения – логические утверждения, заданные для классов, понятий.
- *Примеры* (конкретные экземпляры классов).

## Понятие онтологии: детализация

Онтология соединяет «человеческое» и компьютерное понимание символов.

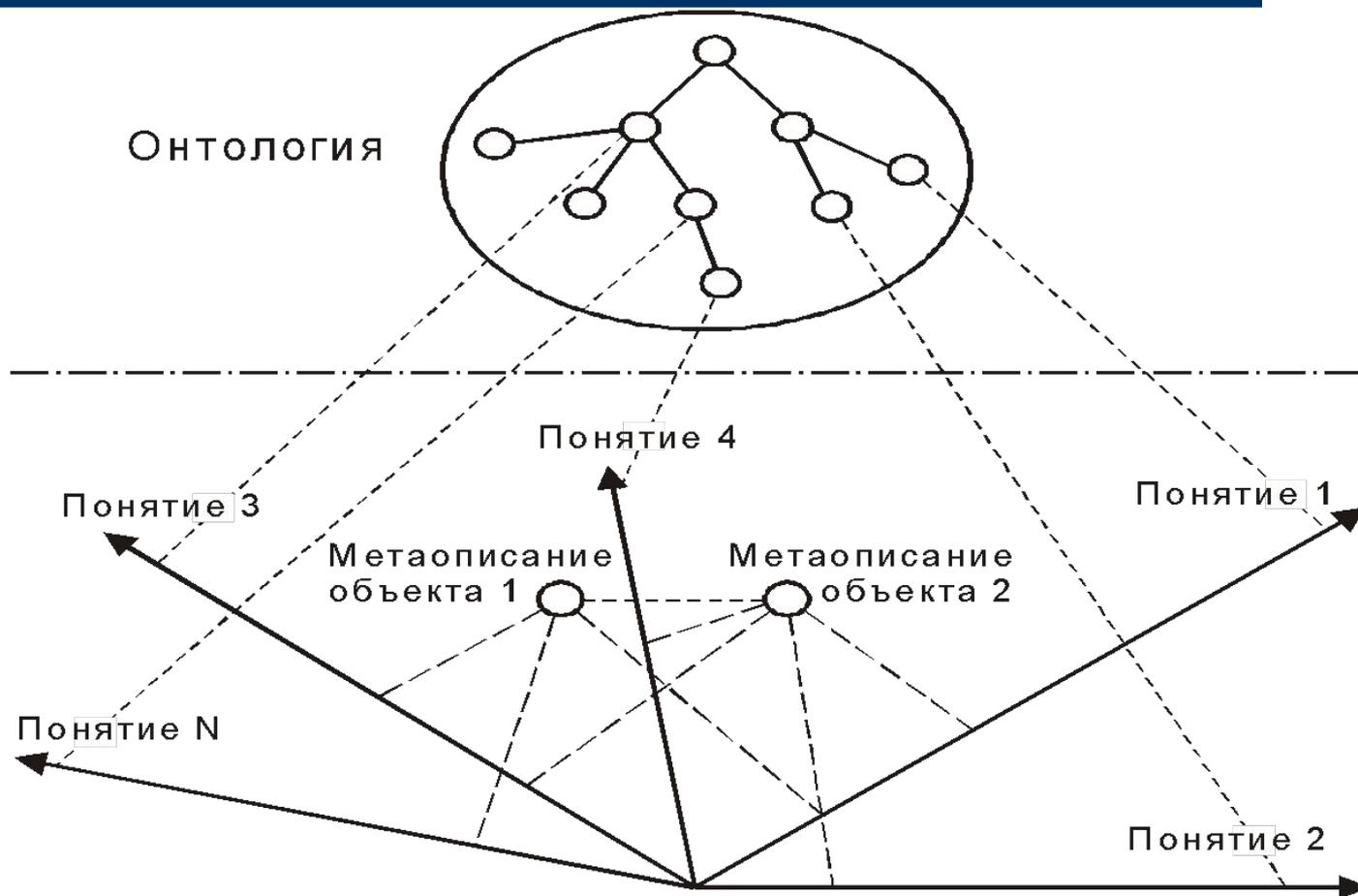
Эти символы, также называемые *терминами* (точными определениями понятий), могут интерпретироваться как людьми, так и машинами.

Термин понятен для человека, так как это слово, написанное на естественном языке.

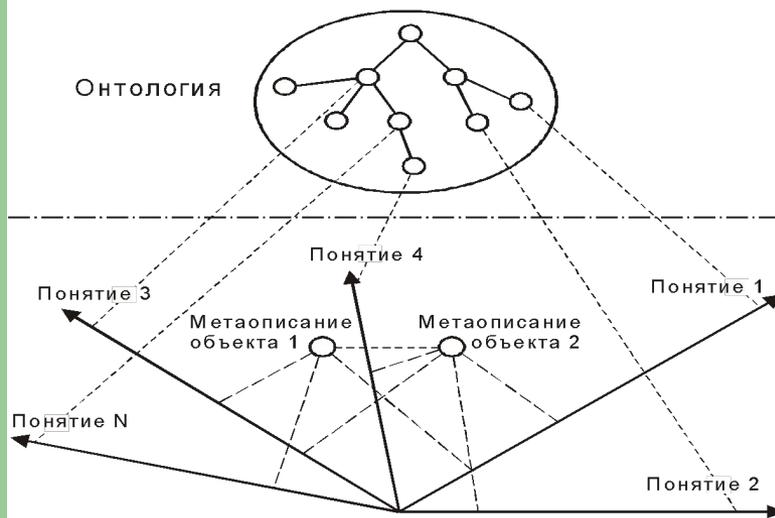
Понятны человеку и связи между терминами, например: «суперпонятие – подпонятие» (род – вид), обычно обозначаемые как *is-a* (являться). Эта связь обозначает тот факт, что одно понятие (субпонятие) является более общим, чем другое (подпонятие).

В качестве примера возьмем такое понятие, как компьютер, которое является менее общим, чем понятие машина (это может быть и автомобиль, трактор, танк и т.д.).

# Схема интеллектуального пространства



# Схема интеллектуального пространства



Здесь:

- В качестве *системы координат* используется онтология предметной области.
- Описания объектов, содержащие знания, задаются в виде их *метаописаний*, составленных из основных понятий онтологии.
- В качестве меры близости объектов (метрики) используется семантическая близость их метаописаний.

## Семантическая сеть

*Семантическая сеть* означает «смысловая» сеть, а, собственно, семантика – это наука, устанавливающая отношения между символами и объектами, которые они обозначают. Иначе говоря, семантика – это наука, определяющая смысл знаков [Люггер Д.Ф., 2003].

По своей структуре семантическая сеть – это ориентированный граф, вершины которого – понятия, а дуги – отношения между ними. Характерной особенностью семантических сетей является наличие трех типов отношений:

- отношение класс – элемент класса;
- отношение свойство – значение свойства;
- отношение фрагмент – элемент класса.

# Формальная модель онтологии (Meadche A., Zacharias V.)

В общем виде формальная модель онтологии может быть описана следующим кортежем [Meadche A., Zacharias V., 2002]:

$$O = \{L, C, F, G, H, R, A\}, \text{ где}$$

- $L = LC \cup LR$  – *словарь онтологии*, содержащий набор лексических единиц (знаков) для понятий  $LC$  и набор знаков для отношений  $LR$ ;
- $C$  – набор *понятий онтологии*, причем для каждого понятия  $c \in C$  в онтологии существует по крайней мере одно утверждение;
- $F$  и  $G$  – *функции ссылок* такие, что  $F: FLC \rightarrow 2C$  и  $G: FLR \rightarrow 2R$ , то есть  $F$  и  $G$  связывают наборы лексических единиц  $\{L_j\} \subset L$  с наборами понятий и отношений, на которые они соответственно ссылаются в данной онтологии. При этом одна лексическая единица может ссылаться на несколько понятий или отношений и одно понятие или отношение может ссылаться на несколько лексических единиц. Инверсиями функций ссылок являются  $F^{-1}$  и  $G^{-1}$ ;
- $H$  – фиксирует *таксономический характер отношений* (связей), при котором понятия онтологии связаны нереклексивными, ациклическими, транзитивными отношениями  $H \subset C \times C$ . Выражение  $H(C_1, C_2)$  означает, что понятие  $C_1$  является подпонятием  $C_2$ ;
- $R$  – обозначает *бинарный характер отношений* между понятиями онтологии, фиксирующие пары области применения (domain)/области значений (range), то есть пары  $(D, R)$  с  $D, R \in C$ ;
- $A$  – *набор аксиом* онтологии.

# Формальная модель онтологии (Гаврилова, Хорошевский)

*Онтология* – это тройка

$$O = \langle X, K, \Phi \rangle,$$

где

- $X$  - конечное **множество концептов** (понятий, терминов) предметной области, которую представляет онтология  $O$ ;
- $K$  - конечное **множество отношений между концептами** (понятиями, терминами) заданной предметной области;
- $\Phi$  - конечное **множество функций интерпретации** (аксиоматизация), заданных на концептах и/или отношениях онтологии  $O$ .

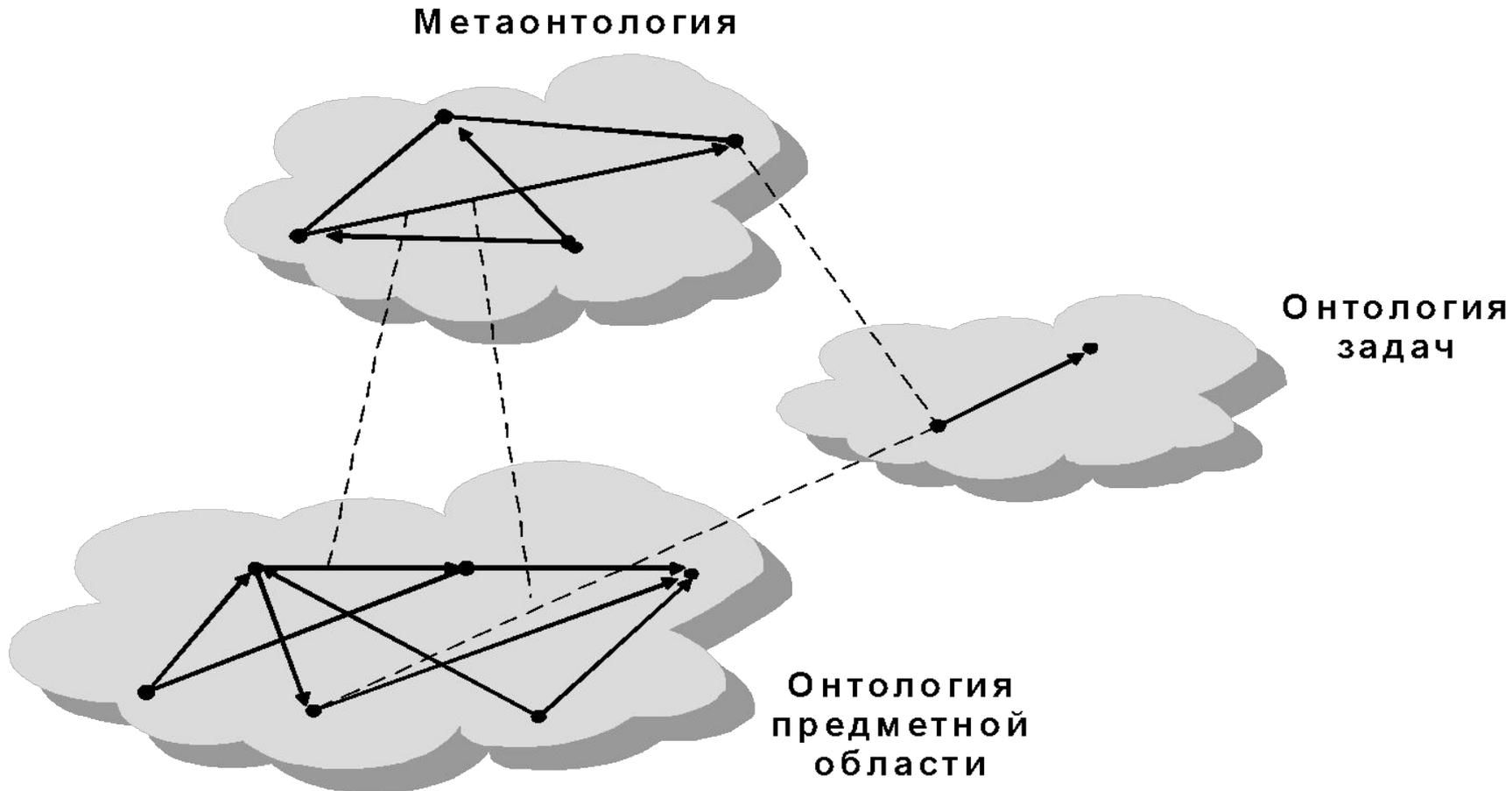
# Классификация онтологий по уровню универсальности

Онтологии **верхнего уровня**, или *метаонтологии*, описывают общие понятия, независимо от задач конкретного домена. Примером такой онтологии может служить WordNet [Fellbaum С., 1998]. Масштаб WordNet весьма обширный – весь английский язык с описанием каждого термина, его синонимов и гипер/гипо (более/менее) общих терминов и отношений между ними. В то же время уровень детальности в WordNet очень низкий, имеются лишь описания на естественном языке терминов, которые не могут быть поняты машиной, и зафиксированы только самые простые отношения между ними.

Онтологии **предметных областей** описывают относительно общие понятия для общих задач. В какой-то мере она относится к онтологиям верхнего уровня, так как ее можно использовать во множестве предприятий различных предметных областей.

Онтологии **приложений** (задач) описывают понятия, зависящие как от домена, так и от решаемой задачи.

# Взаимосвязь между различными онтологиями формальной модели онтологической системы



# Классификация онтологий по степени формальности. «Спектр онтологий»

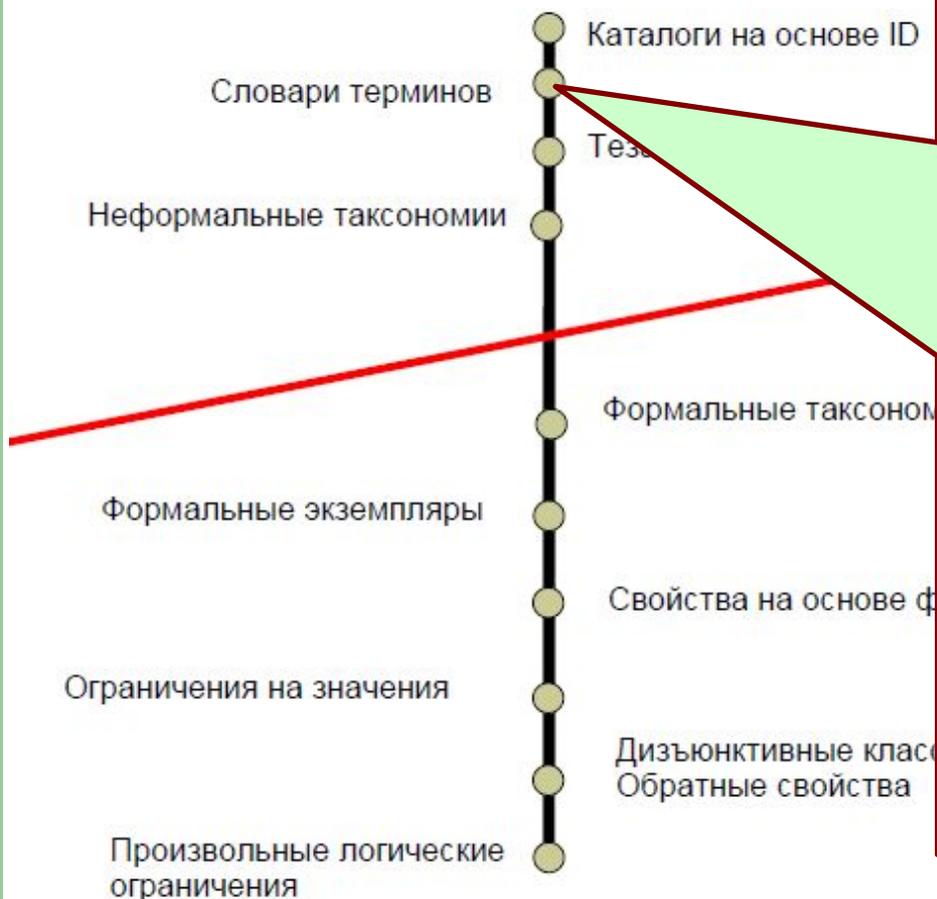
Онтологии могут быть представлены *как спектр* в зависимости от деталей реализации:



# Классификация онтологий по степени формальности. «Спектр онтологий»



# Классификация онтологий по степени формальности. «Спектр онтологий»

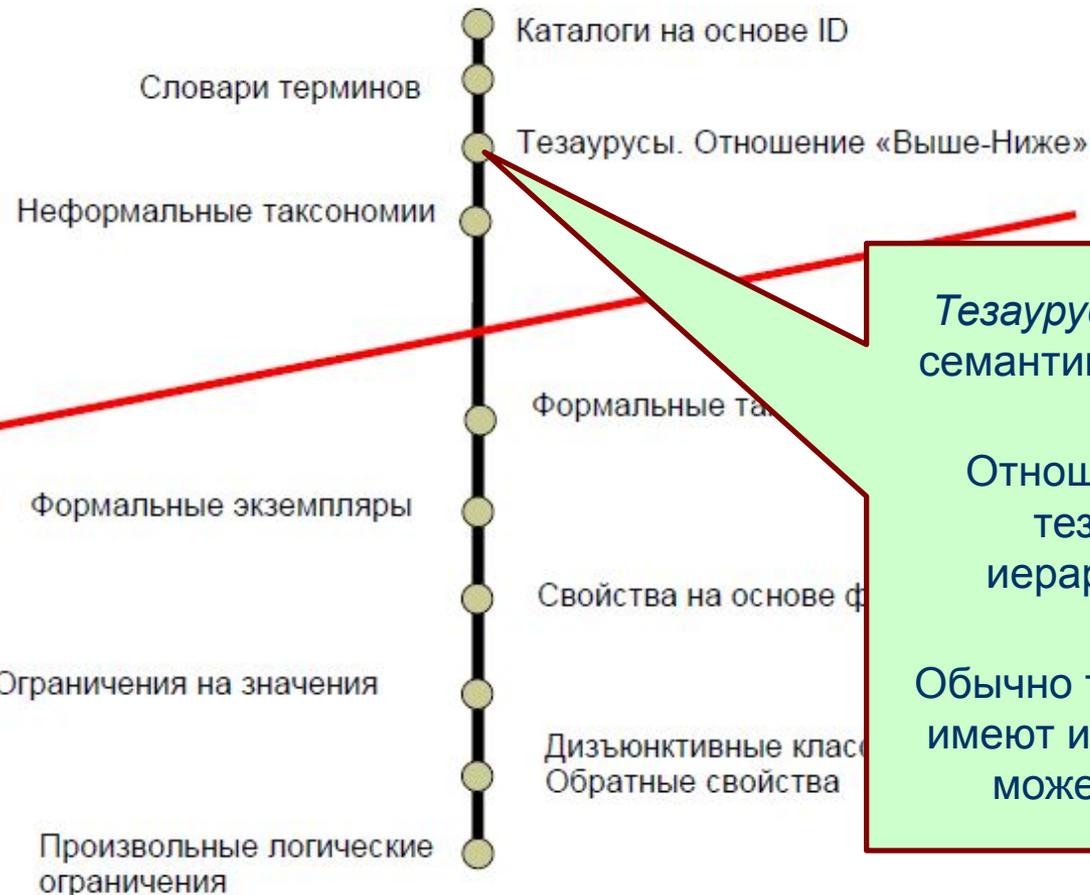


*Глоссарий* – список терминов с их значениями.

Значения описываются в виде комментариев на естественном языке. Это даёт больше информации, поскольку люди могут прочесть такой комментарий и понять смысл термина.

Интерпретации терминов могут быть многозначными. Глоссарии непригодны для автоматической обработки программными агентами, но можно, как и для контролируемых словарей, присвоить терминам ID.

# Классификация онтологий по степени формальности. «Спектр онтологий»

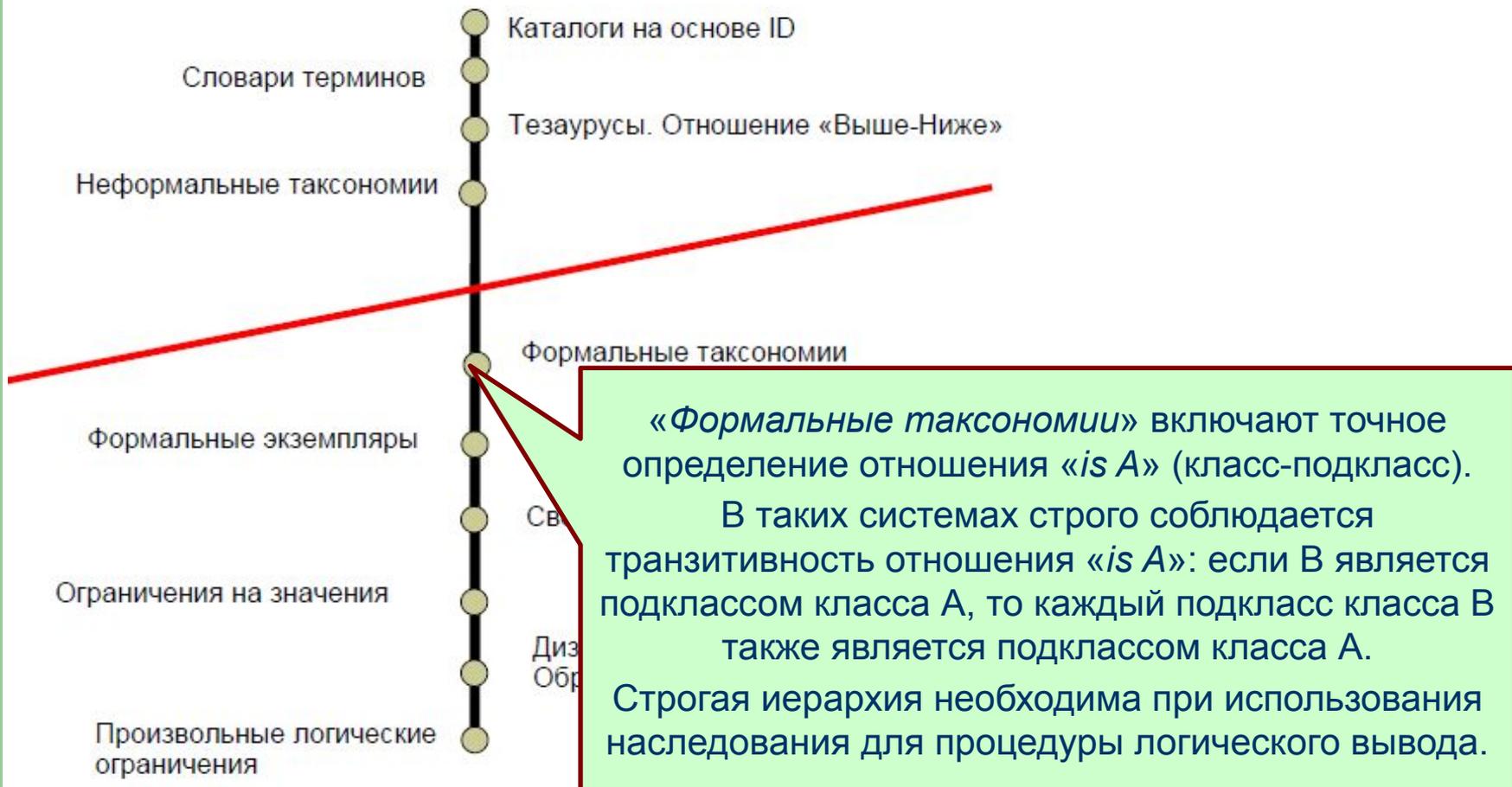


*Тезаурусы* несут дополнительную семантику, определяя связи между терминами.

Отношения свойственные для тезаурусов: синонимия, иерархическое отношение и ассоциация.

Обычно тезаурусы в явном виде не имеют иерархии терминов, но она может быть восстановлена.

# Классификация онтологий по степени формальности. «Спектр онтологий»



# Классификация онтологий по степени формальности. «Спектр онтологий»



Иерархии терминов, определяющие *общие понятия обобщения и уточнения*.  
Явная иерархия, не соответствовала в точности формальным свойствам иерархического отношения (*isA*). Здесь не выполняется важное свойство отношения *isA* – транзитивность.

# Классификация онтологий по степени формальности. «Спектр онтологий»



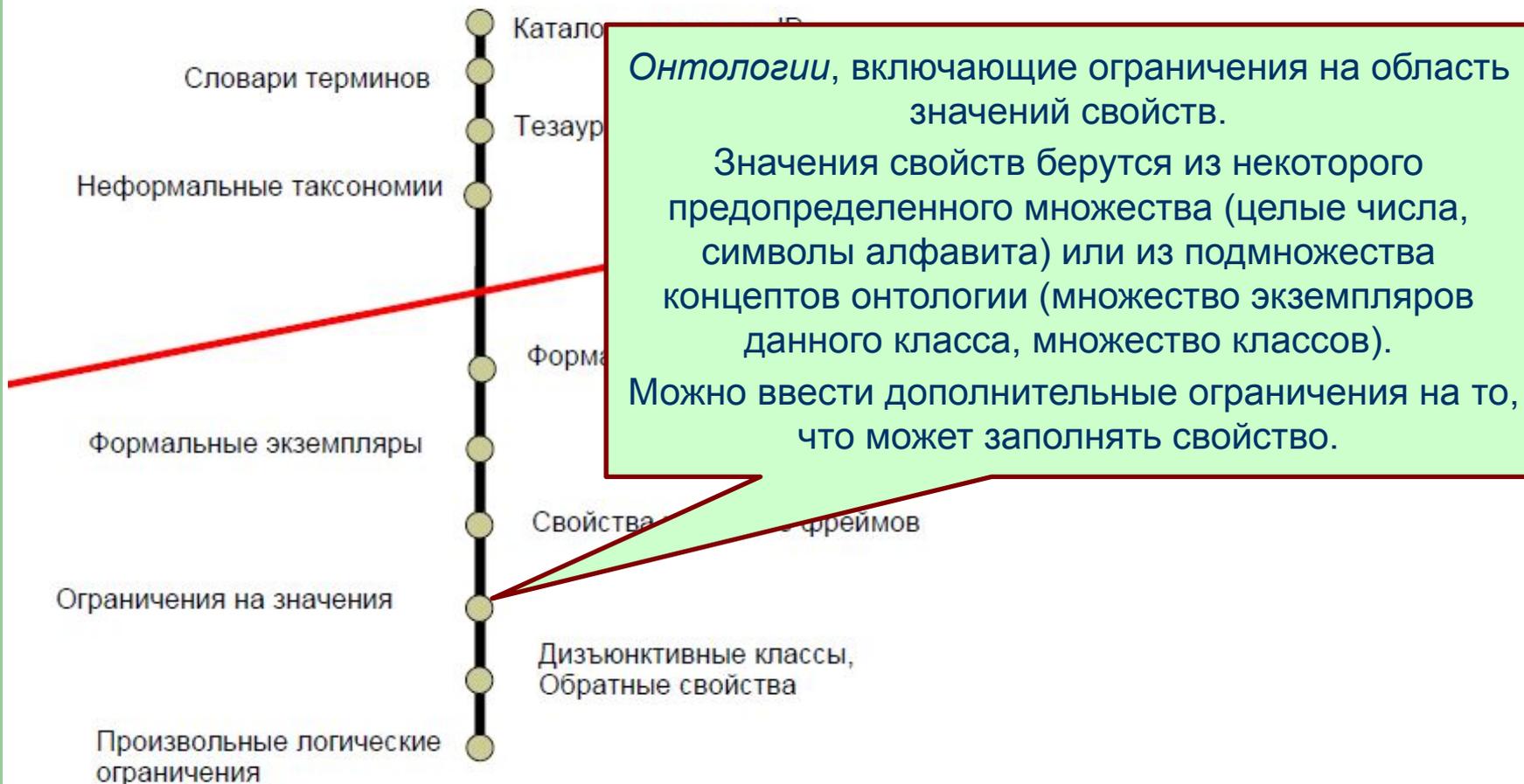
# Классификация онтологий по степени формальности. «Спектр онтологий»



Классы (*фреймы*) могут иметь информацию о *свойствах (слотах)*.

Свойства бывают особенно полезными, когда они определены на верхних уровнях иерархии и наследуются подклассами.

# Классификация онтологий по степени формальности. «Спектр онтологий»

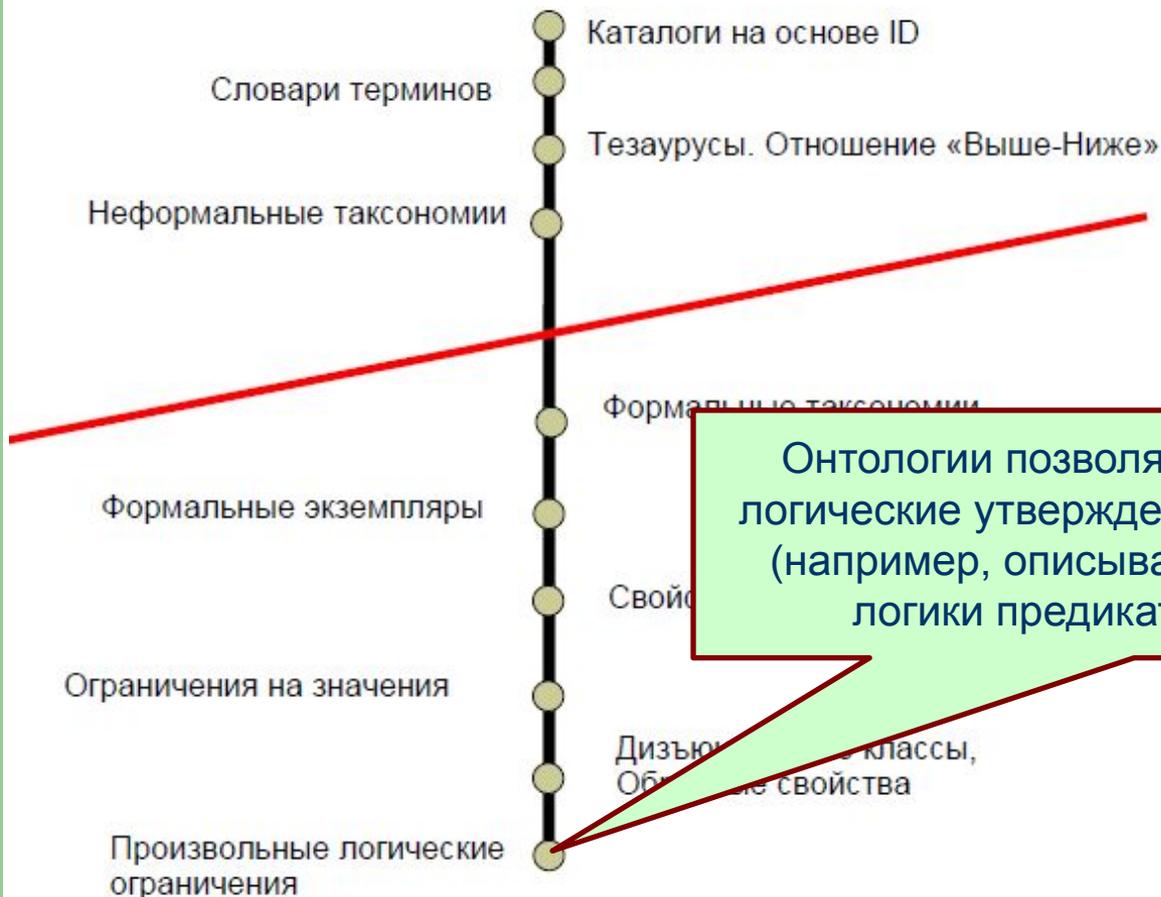


# Классификация онтологий по степени формальности. «Спектр онтологий»



Онтологии позволяют объявлять два и более классов *дизъюнктивными* (непересекающимися). Это означает, что у данных классов не существует общих экземпляров.

# Классификация онтологий по степени формальности. «Спектр онтологий»



Онтологии позволяют делать произвольные логические утверждения о концептах – *аксиомы* (например, описывать утверждения на языке логики предикатов первого порядка).

## «Спектр» онтологий

### ОНТОЛОГИИ



# Классификация онтологий по цели создания



# Классификация онтологий по цели создания



# Классификация онтологий по цели создания



Их назначение в создании единой «правильной онтологии», фиксирующей знания, общие для всех предметных областей и многократном использовании данной онтологии.

Существует несколько серьезных проектов:  
SUMO, Sowa's Ontology, Cyc.

# Классификация онтологий по цели создания



# Классификация онтологий по цели создания



Назначение этих онтологий в том, чтобы описать концептуальную модель конкретной задачи или приложения. Они содержат наиболее специфичную информацию.

# Классификация онтологий по содержанию

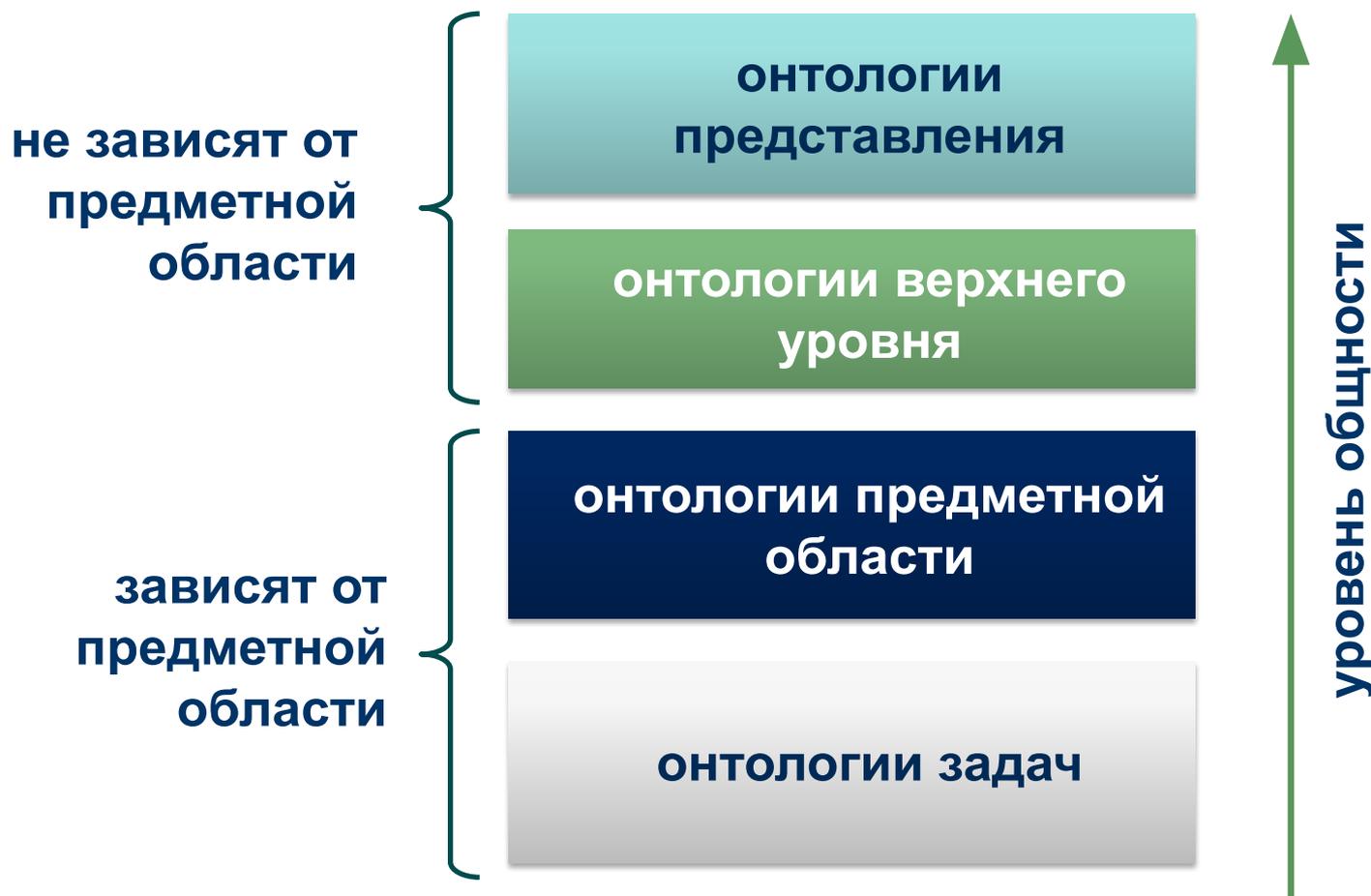


# Классификация онтологий по содержанию

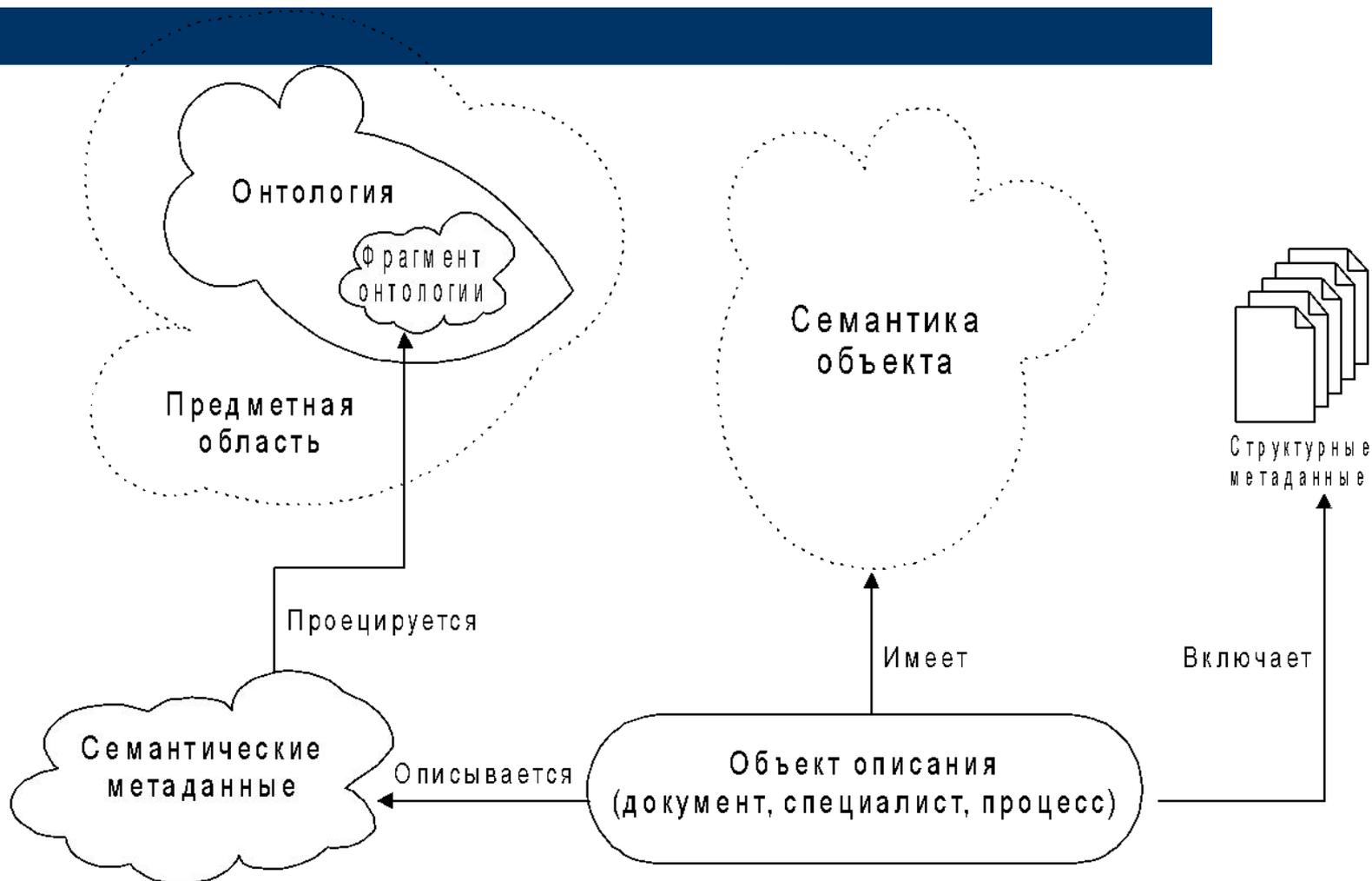


Здесь упор делается на реальное содержимое онтологии, а не на абстрактную цель, преследуемую авторами.

# Классификация онтологий по уровню общности



# Отражение смысла объекта в семантических метаописаниях



## Значение метаданных

- **Системные метаданные** предназначены для функционирования информационных систем, СУБД и систем управления знаниями. Они включают имена файлов и баз, даты их создания, тип и формат, размер файла и вид носителя и т.п.
- **Структурные метаданные** содержат, как правило, справочную информацию об объектах. Это могут быть наименование, статус, структурная принадлежность, профиль и тому подобное, то есть описания, использующиеся при идентификации и категоризации объектов в тех или иных целях.
- **Семантические метаописания** – особый вид описаний, включающий концептуальное (аннотированное) изложение содержания и смысла информации об объекте.

# Структура онтологии

Описание онтологии включает:

- Понятия.
- Отношения.
- Аксиомы.
- Экземпляры.

# Структура онтологии

Понятия

Отношения

Аксиомы

Экземпляры

Концептуализации класса всех представителей  
некой сущности или явления

Программное  
обеспечение

Системное ПО

Прикладное ПО

Операционная  
система

# Структура онтологии

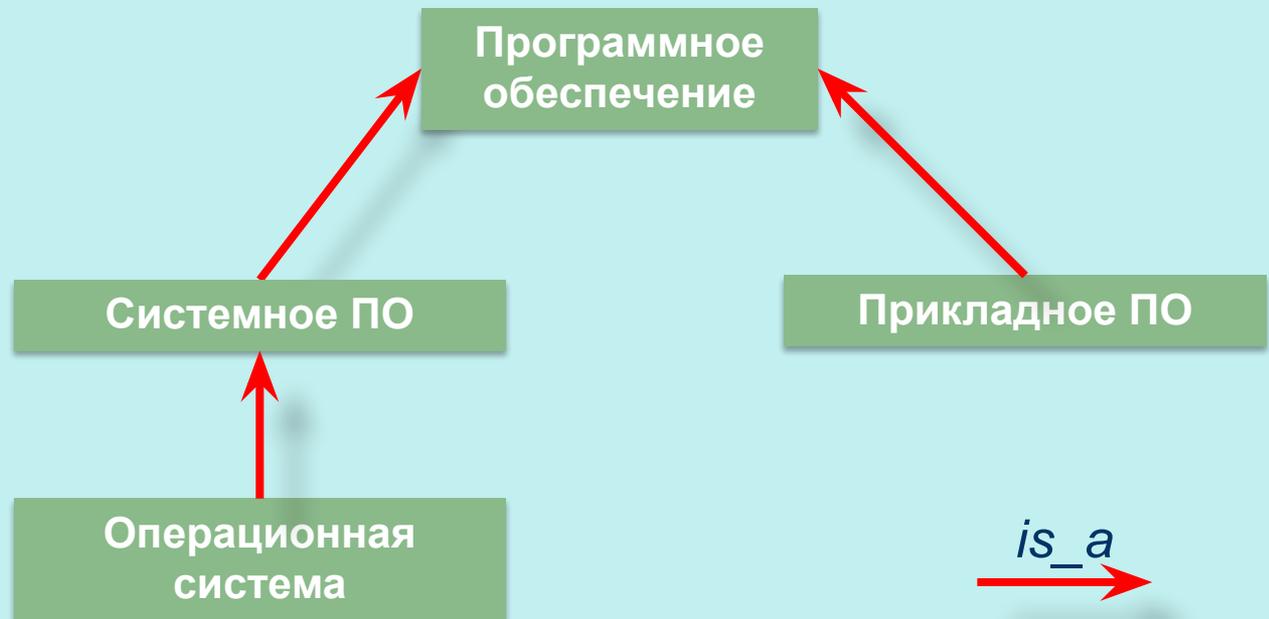
Понятия

**Отношения**

Аксиомы

Экземпляры

Концептуализации класса всех представителей  
некой сущности или явления



# Структура онтологии

Понятия

Отношения

**Аксиомы**

Экземпляры

Концептуализации класса всех представителей  
некой сущности или явления

Программное

A1. К системному ПО относятся  
программы общего назначения.  
A2. Программы, не относящиеся к  
системному ПО, относятся к  
прикладному ПО.

с

ю

Операционная  
система

*is\_a*

# Структура онтологии

Понятия  
Отношения  
Аксиомы  
Экземпляры

Концептуализации класса всех представителей  
некой сущности или явления



# Использование онтологий

**Преимущества** при использовании онтологий в определённых предметных областях могут быть реализованы:

- в *вычислительном плане* (для сокращения времени решения задач);
- в *экономическом плане* (для сокращения затрат на разработку и сопровождение программного обеспечения, обработку данных).

# Использование онтологий

**Задачи**, решаемые с помощью онтологий:

- Информационный поиск.
- Интеграция гетерогенных источников данных.
- SemanticWeb.
- Моделирование предметных областей при разработке информационных систем (ИС).

# Использование онтологий

**Задачи**, решаемые с помощью онтологий:

- ***SemanticWeb***.
- Информационный поиск.
- Интеграция гетерогенных источников данных.
- Моделирование предметных областей при разработке информационных систем (ИС).

Наиболее крупной «площадкой» для экспериментов с семантическими технологиями в настоящее время является *проект Semantic Web*.

# Использование онтологий: *Semantic Web*

Идея *Семантической Сети* (Semantic Web) впервые была провозглашена в 2001 году Тимом Бернерсом-Ли (создателем World Wide Web).

Суть ее состоит в *автоматизации «интеллектуальных» задач* обработки значения (в семантическом смысле) тех или иных ресурсов, имеющихся в Сети.

Обработкой и обменом информации должны заниматься не люди, а специальные *интеллектуальные агенты* (программы, размещённые в Сети).

Для того, чтобы взаимодействовать между собой, агенты должны иметь *общее (разделяемое всеми) формальное представление значения для любого ресурса*. Именно для цели представления общей, явной и формальной спецификации значения в Semantic Web используются онтологии.

# Использование онтологий: *Semantic Web*

В 1997 году консорциум W3C определил спецификацию *RDF* (*Resource Description Framework*).

RDF предоставляет простой, но мощный язык описания ресурсов, основанный на *триплетях* (*triple-based*) "Субъект-Предикат-Объект" и спецификации URI.

В 1999 году RDF получает статус *рекомендации*, что ведёт к улучшению функциональности и обеспечения *интероперабельности* (т.е. возможности обмениваться данными в Сети несмотря на их разнородность).

Концептуально RDF даёт минимальный уровень для представления знаний в Сети.

# Использование онтологий: *Semantic Web*

*Спецификация RDF* опирается на ранние стандарты, лежащие в основе *Web*:

- *Unicode* служит для представления символов алфавитов различных языков.
- *URI* используется для определения уникальных идентификаторов ресурсов.
- *XML* и *XML Schema* служат для структурирования и обмена информацией и для хранения RDF (*XML синтаксис RDF*).

# Использование онтологий: *Semantic Web*

Кроме RDF был разработан *язык описания структурированных словарей для RDF* – язык *RDF Schema (RDFS)*.

Он предоставляет *минимальный набор средств для спецификации онтологий*.

RDFS получил *статус рекомендации W3C* в 2004 году.

Препятствием для *Semantic Web* было то, что документов, написанных на языке *RDF/RDFS*, было относительно мало.

В период с 2001 по 2004 годы шла интенсивная работа по созданию программных средств для обработки и автоматической генерации *RDF-документов*.

# Использование онтологий: *Semantic Web*

В 2004 году представлен язык *GRDDL (Gleaning Resource Descriptions form Dialects of Languages)*.

Его назначение состоит в предоставлении средств для извлечения *RDF-триплетов* из *XML* и *XHTML* данных (в особенности это относится к документам, автоматически генерируемым из закрытых баз данных).

В области создания библиотек классов и построения логических выводов над *RDF-графами* была создана библиотека *Jena Framework*, в области создания модулей расширения для браузеров - *Simile* для *Firefox*.

В области создания визуальных сред редактирования большое число редакторов онтологий стали поддерживать *RDF*.

# Использование онтологий: *Semantic Web*

В 2004 году статус рекомендации получил язык *OWL (Web Ontology Language)*. Он имеет 3 диалекта (3 множества структурных единиц), используемых в зависимости от требуемой выразительной мощности. OWL фактически является *надстройкой* над *RDF/RDFS* и поддерживает *эффективное представление онтологий в терминах классов и свойств, обеспечение простых логических проверок целостности онтологии и связывание онтологий друг с другом* (импорт внешних определений).

Многие формализмы описания знаний могут быть отображены на формализм OWL (два из его диалектов (OWL Lite и OWL DL) соответствуют двум дескриптивным логикам, имеющим разную выразительную силу).

Большое число создаваемых в настоящее время онтологий кодируются на OWL.

# Использование онтологий: *Semantic Web*

В 2005 году началась работа над форматом обмена правилами – *RIF (Rule Interchange Format)*.

Его назначение – соединить в одном стандарте несколько формализмов для описания правил, по которым может осуществляться нетривиальный логический вывод: логику клауз Хорна, логики высших порядков, продукционные модели и т.п.

*Язык SPARQL (язык запросов к RDF-хранилищам)* в январе 2008 г. приобрёл статус официальной рекомендации Консорциума W3C. Синтаксически он очень похож на SQL.

Язык SPARQL уже широко используется разработчиками информационных систем.

# Использование онтологий

**Задачи**, решаемые с помощью онтологий:

- SemanticWeb.
- **Информационный поиск**.
- Интеграция гетерогенных источников данных.
- Моделирование предметных областей при разработке информационных систем (ИС).

Второй по масштабу задачей можно условно считать область *информационного поиска* (**Information Retrieval**).

# Использование онтологий: *Информационный поиск*

Примерами задач в области информационного поиска являются:

- собственно информационный поиск документов по запросу пользователя;
- *автоматическая рубрикация документов* по заранее заданному рубрикатору (классификация);
- *автоматическая кластеризация документов* – разбиение на кластеры близких по смыслу документов;
- *разработка вопросно-ответных систем* – поиск точного фрагмента текста, отвечающего на вопрос пользователя, а не целого документа;
- *автоматическое аннотирование* (составление аннотации) документа и многие другие.

# Использование онтологий: Информационный поиск

В современных поисковых системах тексты автоматически индексируются по набору слов, составляющих эти тексты.

Такое представление текстов как простого набора слов ("*bag of words*") имеет большое количество недостатков, затрудняющих поиск релевантных текстов, например:

- *избыточность*: в пословном индексе используются слова-синонимы, выражающие одни и те же понятия;
- *слова текста считаются независимыми друг от друга*, что не соответствует свойствам связного текста;
- *многозначность слов*: поскольку многозначные слова могут рассматриваться как дизъюнкция двух или более понятий, выражающих различные значения многозначного слова, то маловероятно, что все элементы этой дизъюнкции интересуют пользователя.

# Использование онтологий: *Информационный поиск*

Этих недостатков лишено так называемое *концептуальное (семантическое) индексирование*, то есть такое индексирование, когда *текст индексируется не по словам, а по понятиям*, которые обсуждаются в данном тексте.

При такой технологии:

- все *синонимы* сведены к одному и тому же понятию;
- *многозначные слова* отнесены к разным понятиям;
- *связи между понятиями и соответствующими словами* описаны и могут быть использованы при анализе текста.

Для того чтобы попытаться реализовать схему автоматического концептуального индексирования и концептуального поиска, необходимо иметь ресурс, описывающий систему понятий данной предметной области, то есть *онтологию в данной предметной области*.

# Использование онтологий: Информационный поиск

Использование онтологий для информационного поиска в реальных широких предметных областях имеет ряд особенностей:

- онтология должна быть очень *большой величины*;
- понятия онтологии должны иметь аккуратно установленные *связи с языковыми единицами* – терминами предметной области;
- онтология реальной предметной области *не может быть полной*, поэтому методы информационного поиска на основе онтологий должны сочетаться с методами информационного поиска на основе пословных методов в едином поисковом механизме;
- задача информационного поиска предполагает *использование онтологий для анализа свободных неограниченных связных текстов*, для которых не существует хорошо развитых методов автоматической обработки.

# Использование онтологий

**Задачи**, решаемые с помощью онтологий:

- SemanticWeb.
- Информационный поиск.
- **Интеграция гетерогенных источников данных.**
- Моделирование предметных областей при разработке информационных систем (ИС).

*Интеграция данных из разнородных источников (баз данных, баз знаний)* – одна из важнейших задач в современных условиях

# Использование онтологий: *Интеграция гетерогенных данных*

Здесь рассматриваются три важные задачи, возникающие при управлении данными:

- выражение концептуальной модели предметной области (онтологии) для конкретного источника данных,
- интеграция нескольких источников,
- выражение и выполнение запросов.

Для каждой из задач существует подход с использованием *DL* (описательной логики).

# Использование онтологий: *Интеграция гетерогенных данных*

Источниками данных могут быть базы данных (БД) и базы знаний (БЗ).

Под *базой данных* понимается совокупность (коллекция) согласованных взаимосвязанных данных, которые имеют некоторое «скрытое (внутри) значение», описывают состояние некоторой системы, объектов реального мира.

БД используются для описания некоторой предметной области с целью хранения, обработки и доступа к необходимой информации о ней.

Базы данных содержат (и способны обрабатывать) большие массивы относительно простой информации (при этом доступ возможен только к этим явно введённым данным).

# Использование онтологий: *Интеграция гетерогенных данных*

В базах знаний (БЗ) обычно хранится меньший объем информации, но они имеют более сложную структуру, что позволяет использовать возможности логического вывода и получать такие утверждения, которые не были в явном виде введены.

К любой БЗ могут быть применимы 3 операции:

- *определить (define),*
- *сказать (tell) (т.е. сделать утверждение) и*
- *спросить (ask).*

Каждая из операций может использовать один или более собственных языков, например язык описания схем и ограничений, язык обновления (для новых утверждений), язык запросов и язык описания результата.

# Разработка онтологий

*Методологическое ядро*, пригодное, прежде всего для построения доменных онтологий, сформировано на основе опыта разработчиков.

Практика разработки онтологий предусматривает следующую *последовательность конструктивных действий*:

- *Отбор концептов*, значимых для предметной/проблемной области.
- *Категоризация терминов*.
- *Дальнейшая внутрикатегорная систематизация* – построение таксономии.
- *Установление нетаксономических отношений*

# Разработка онтологий

Краткое содержание этих этапов:

- *Спецификация* – определение функциональности онтологии.
- *Концептуализация* – структурирование (как можно более тщательное) доменной терминологии на содержательном уровне.
- *Формализация* «преобразует концептуальную модель в формальную или полувывчислимую (semi-computable) модель».
- *Реализация (implementation)* – строит вычислимую модель на языке представления онтологий.

Собственно содержательный этап *структурирования терминосистемы* – концептуализация.

# Разработка онтологий: *Концептуализация*

Для этого этапа (*conceptual modeling*) определяется следующая последовательность действий:

1. *Построение словаря терминов*, включая сюда также естественно-языковые определения терминов, синонимы и акронимы.
2. *Построение таксономии для концептов*. Отмечается необходимость одновременно фиксировать отношение несовместимости (*disjoint*).
3. *Построение ad hoc* (т.е., экспертно) *графа бинарных отношений между концептами* (необходимо точное определения условий заполнения аргументов отношений (*domains and ranges*)).
4. Построение "словаря концептов" с привязкой к концептам *атрибутов и отношений*.

# Разработка онтологий: Концептуализация (продолжение)

5. *Детализация описания бинарных отношений* (п. 3): предусматривается дополнительное указание числа возможных значений (cardinality – 1:1, 1:N), логических характеристик отношения (симметричность, транзитивность) и наличие конверсива.
6. *Описание атрибутов*: область применимости (domain); тип данных для значения; единица измерения, точность представления, область определения ("от...до...") (если атрибут числовой); минимальное и максимальное число возможных значений – (0,1), (1,1), (0,N), (1,N).
7. *Описание специфичных для области знаний констант*.
8. *Описание экземпляров*: отнесение к классу и указание значений релевантных для класса атрибутов.

## Разработка онтологий: *Концептуализация (продолжение)*

Есть ещё 2 пункта, которые, по существу, подразумевают переход к следующему этапу – *этапу формализации* и предусматривают формулировку аксиом и правил, так или иначе ограничивающих допустимые описания экземпляров.

# Разработка онтологий: Формализация

*Общие рекомендации по формализации терминосистем:*

- *Не отдельные термины, а терминосистемы.*  
Формализация должна быть одинаковой для всего ряда однотипных языковых явлений.
- Опора на хорошо выстроенную онтологию верхнего уровня.

*Действия по формализации частных терминосистем:*

- *Жёсткая поддержка DisjointUnion* – дисциплины построения иерархической системы классов – разбиение каждого из исходных классов (Partition) с явным указанием оснований деления.
- Выявление базовых признаков и других информационно значимых элементов путём *содержательного анализа словарных определений.*

# Редакторы онтологий

*Редактор онтологий* - инструментальное программное средство, созданное специально для проектирования, редактирования и анализа онтологий.

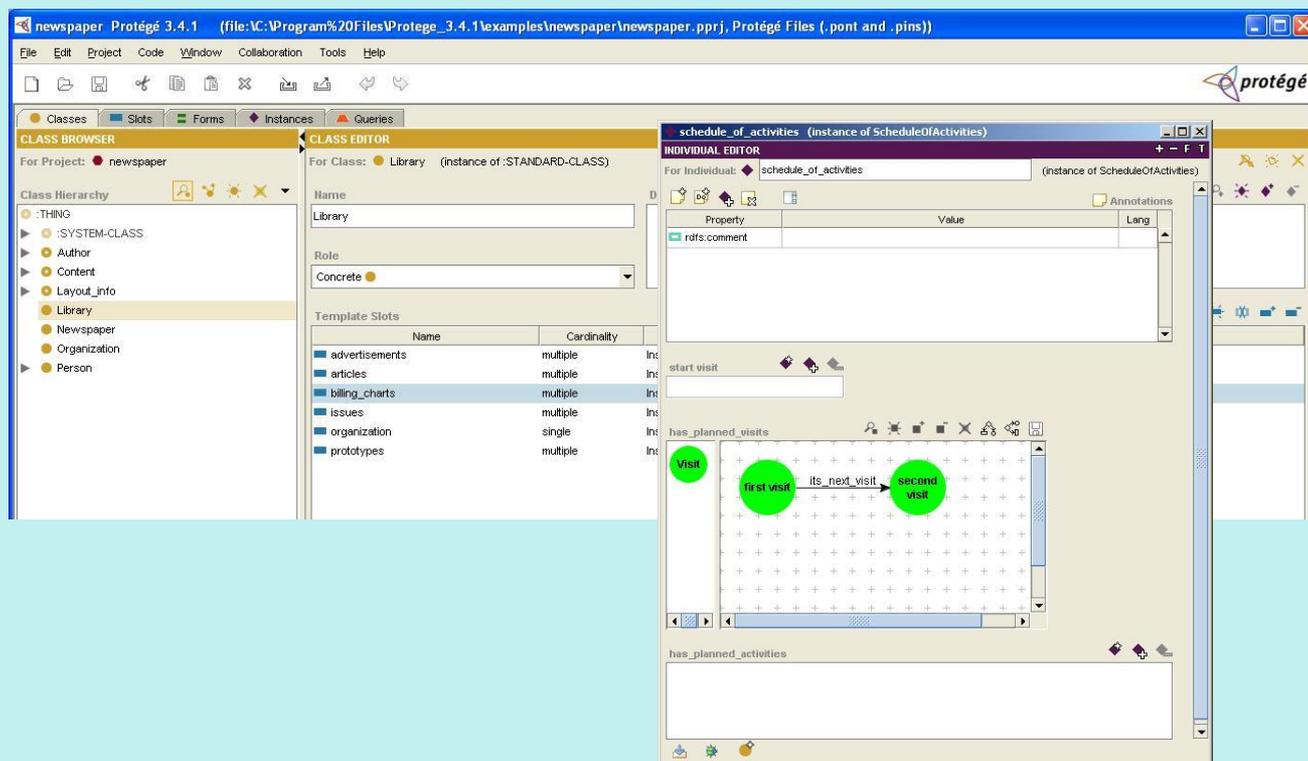
Основные характеристики редакторов:

- *Поддерживаемые формализмы* (теоретический базис, лежащий в основе способа представления онтологических знаний).
- *Формат представления* (языки представления онтологий).

# Редакторы онтологий: примеры

Protégé  
OntoStudio

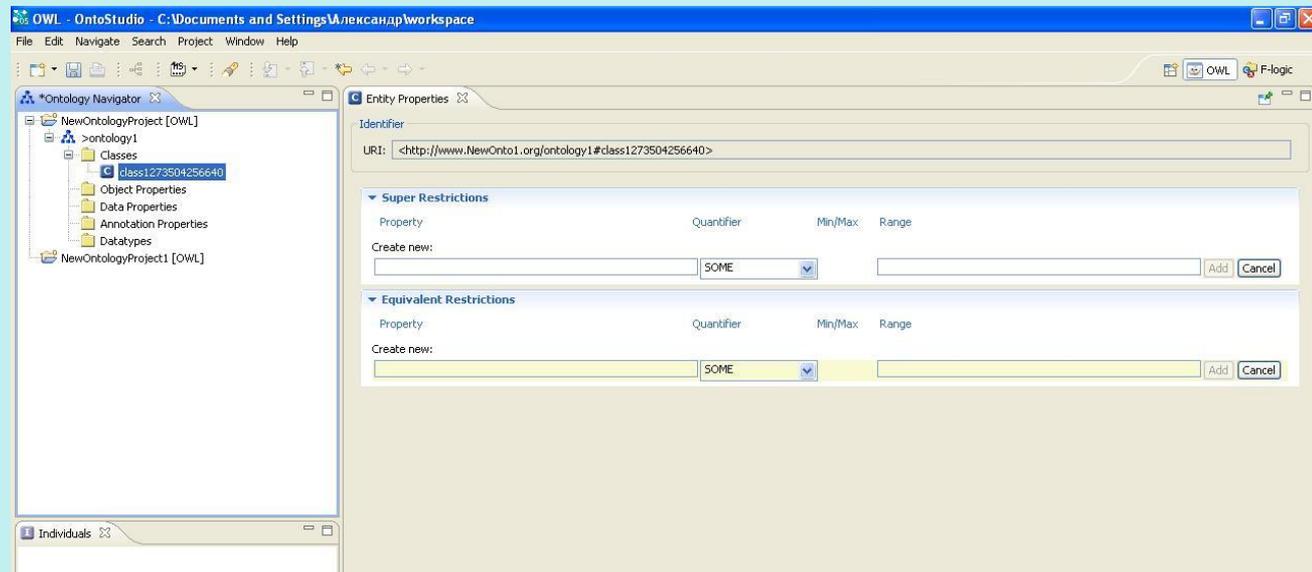
Свободный, открытый редактор онтологий и фреймворк для построения баз знаний.



# Редакторы онтологий: примеры

Protégé  
OntoStudio

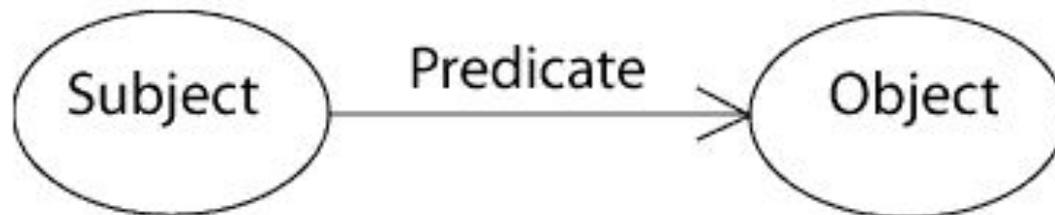
Инструментальное средство для моделирования онтологий и администрирования проектов, в основе которых лежат онтологии.



# Языки описания онтологий. Resource Description Framework (RDF)

Использует представление вида:  
*субъект, предикат, объект.*

Утверждения сохраняются в виде  
«троек информации», триплетов.



**В ящике лежит коробок спичек**

Объект1	Объект2	Вид связи
ящик	коробок спичек	лежит в

# Языки описания онтологий. Ontologies Web Language (OWL)

Основные элементы языка:

- Классы.
- Свойства.
- Индивиды (экземпляры классов или свойств).

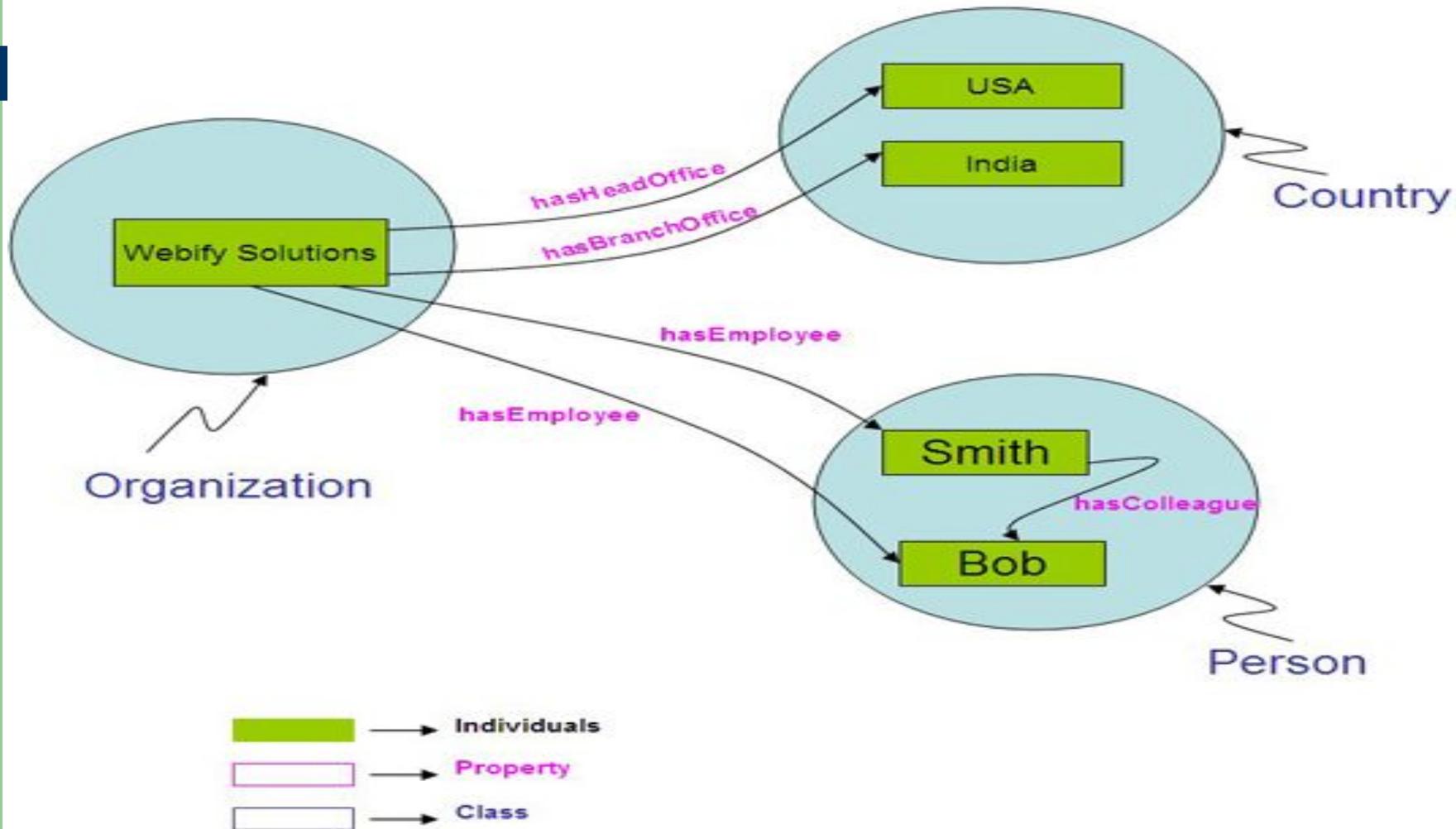
OWL напоминает объектно-ориентированные языки:

- можно конструировать свои классы,
- можно определять экземпляры классов,
- возможно множественное наследование.

Возможна разработка ограничений, например:

- обязательно относится к классу,
- минимальное/максимальное значение и т.д.

# OWL: Пример



## Рекомендованные источники:

1. *Добров Б., Иванов В., Лукашевич Н., Соловьев В.* Онтологии и тезаурусы: модели, инструменты, приложения [Электронный ресурс]  
[URL: <http://www.intuit.ru/studies/courses/1078/270/info>]  
Опубликован: 05.11.2008. ISBN: 978-5-9963-0007-5.