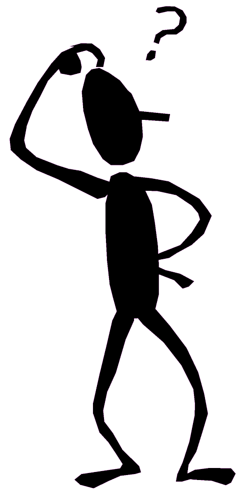
Scalable Ontology Systems

Ian Horrocks
<ian.horrocks@comlab.ox.ac.uk>
Information Systems Group
Oxford University Computing Laboratory

What is an Ontology?

What is an Ontology?



"A specification of a conceptualization" [Gruber]

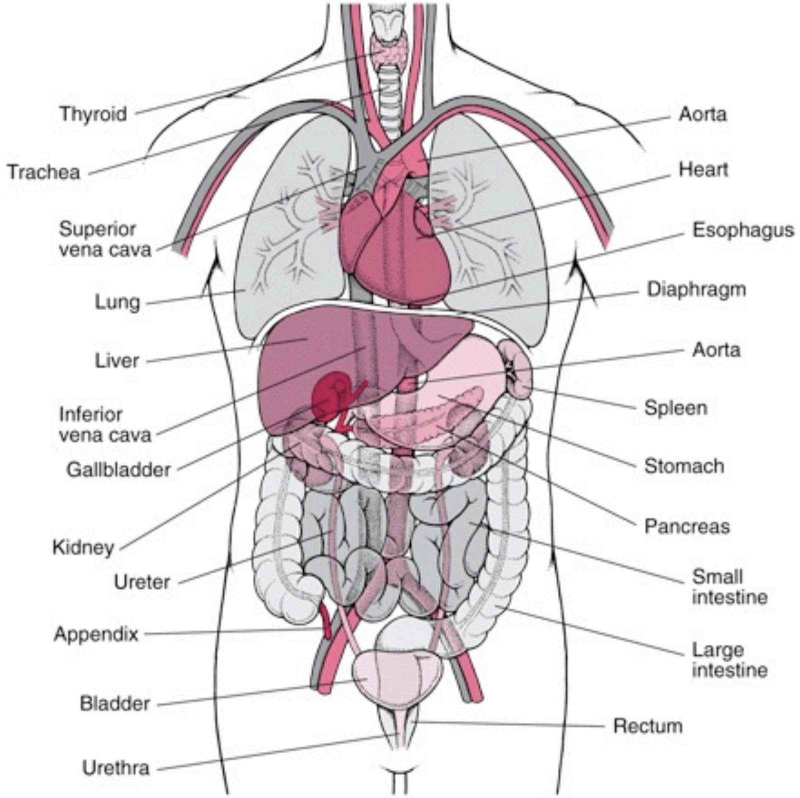
What is an Ontology?

A model of (some aspect of) the world

What is an Ontology?

A model of (some aspect of) the world

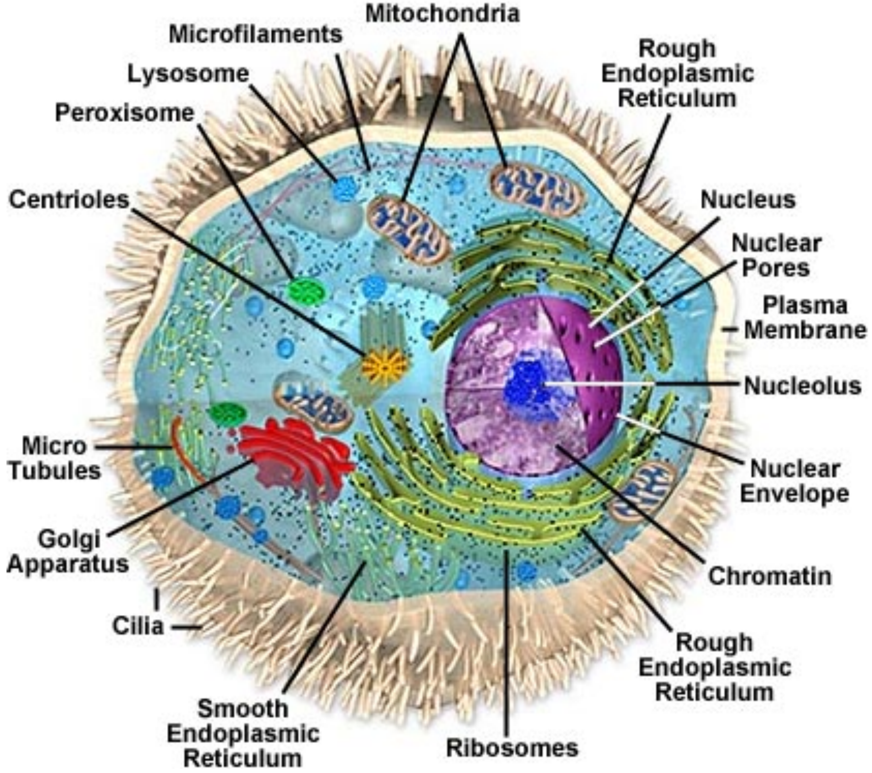
- Introduces **vocabulary** relevant to domain, e.g.:
  - Anatomy



What is an Ontology?

A model of (some aspect of) the world

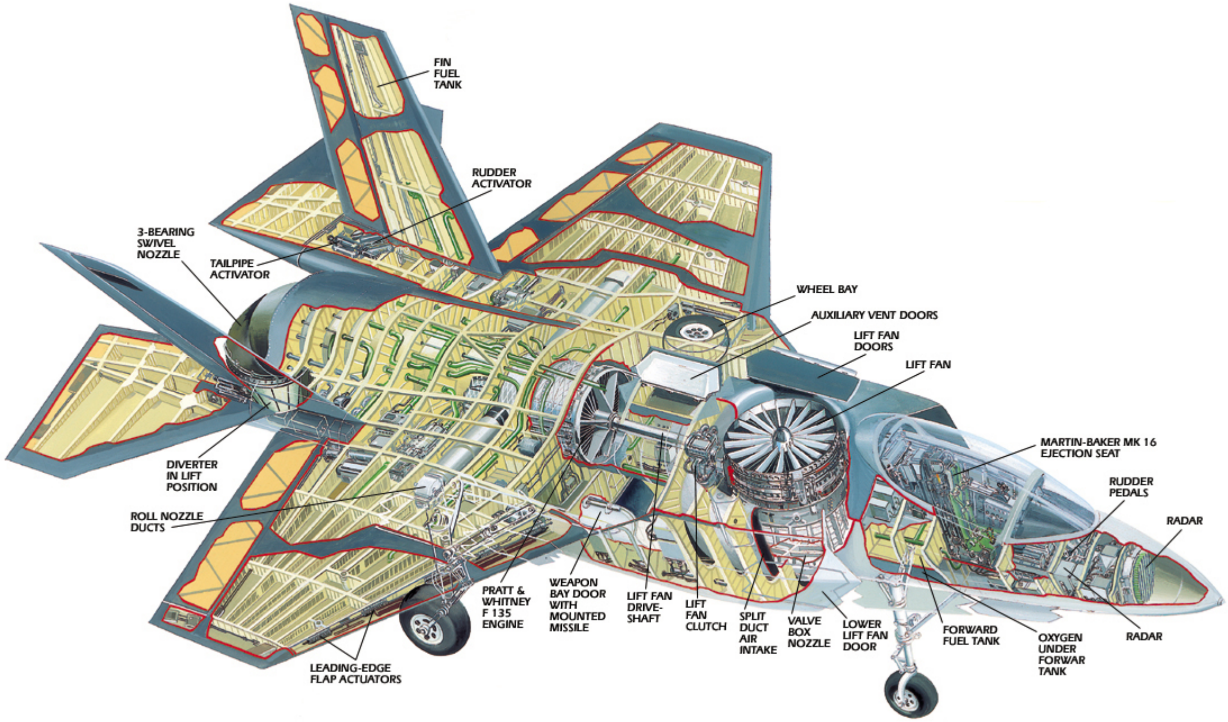
- Introduces **vocabulary** relevant to domain, e.g.:
  - Anatomy
  - Cellular biology



What is an Ontology?

A model of (some aspect of) the world

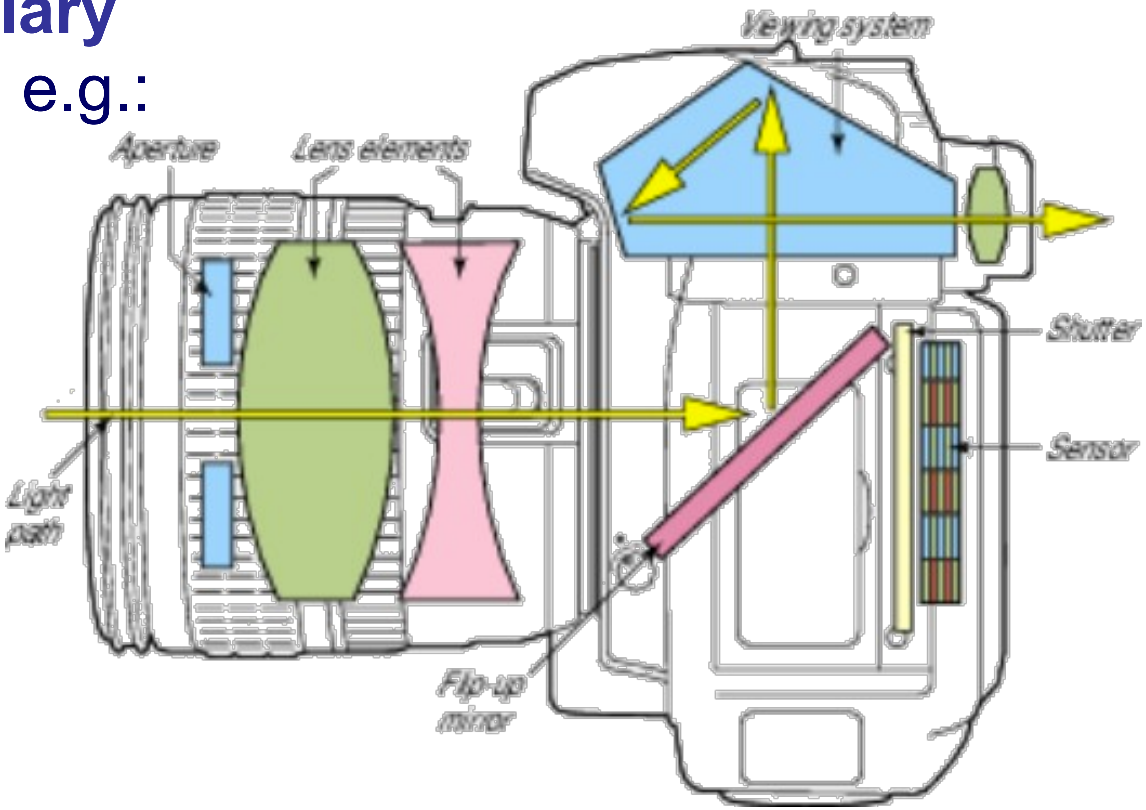
- Introduces **vocabulary** relevant to domain, e.g.:
  - Anatomy
  - Cellular biology
  - Aerospace



What is an Ontology?

A model of (some aspect of) the world

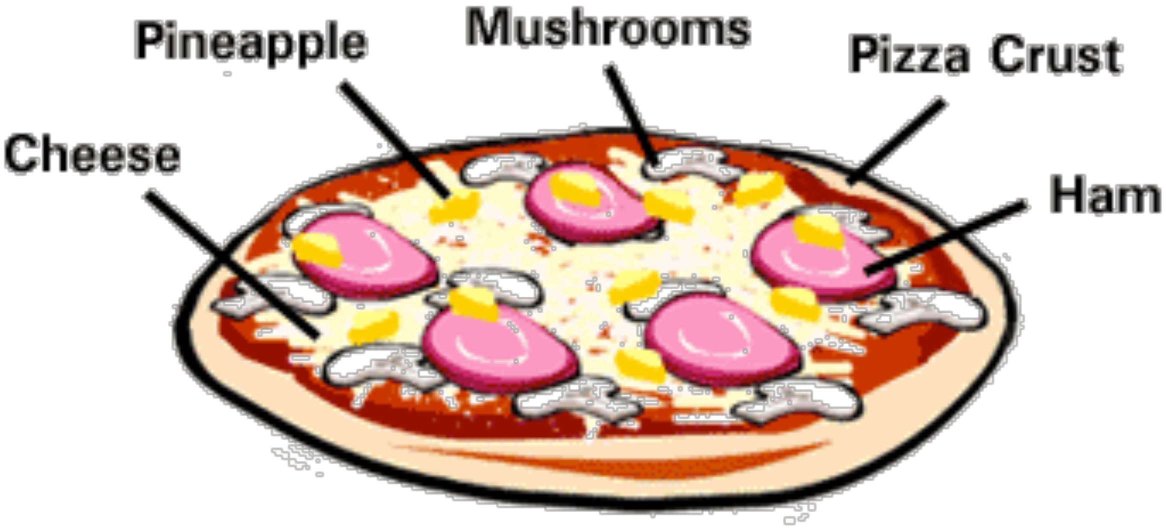
- Introduces **vocabulary** relevant to domain, e.g.:
  - Anatomy
  - Cellular biology
  - Aerospace
  - Photography



What is an Ontology?

A model of (some aspect of) the world

- Introduces **vocabulary** relevant to domain, e.g.:
  - Anatomy
  - Cellular biology
  - Aerospace
  - Photography
  - Pizzas
  - …

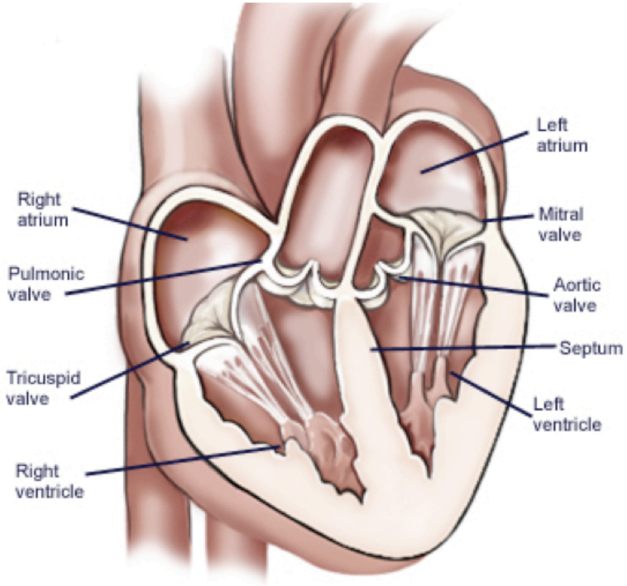


What is an Ontology?

A model of (some aspect of) the world

- Introduces **vocabulary** relevant to domain
- Specifies **meaning** (semantics) of terms

Heart is a muscular organ that is part of the circulatory system



What is an Ontology?

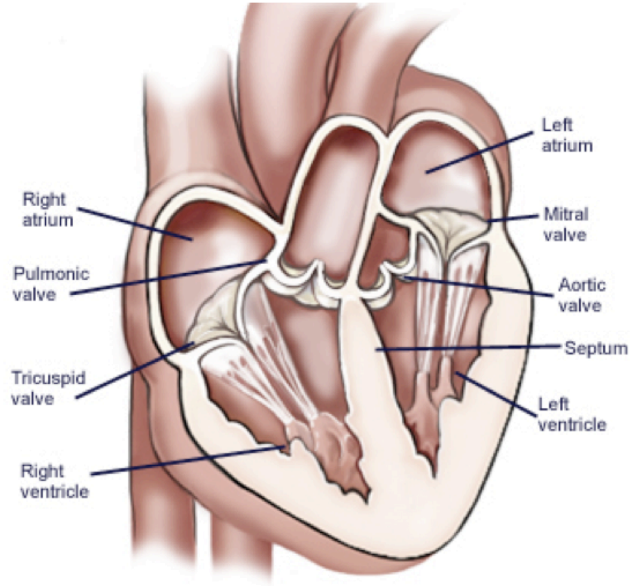
A model of (some aspect of) the world

- Introduces **vocabulary** relevant to domain
- Specifies **meaning** (semantics) of terms

  Heart is a muscular organ that is part of the circulatory system

- **Formalised** using suitable logic

$$\forall x.[\mathsf{Heart}(x) \rightarrow \mathsf{MuscularOrgan}(x) \wedge \exists y.[\mathsf{isPartOf}(x,y) \wedge \mathsf{CirculatorySystem}(y)]]$$



Description Logics (DLs)

- Fragments of **first order logic** designed for KR
- Useful computational properties
  - **Decidable** (essential)
  - Low complexity (desirable)
- Succinct and **variable free syntax**

$$\forall x.[\mathsf{Heart}(x) \rightarrow \mathsf{MuscularOrgan}(x) \wedge \exists y.[\mathsf{isPartOf}(x,y) \wedge \mathsf{CirculatorySystem}(y)]]$$

$$\mathsf{Heart} \sqsubseteq \mathsf{MuscularOrgan} \sqcap \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem}$$

Description Logics (DLs)

DL **Knowledge Base** (KB) consists of two parts:

– Ontology (aka **TBox**) axioms define terminology (schema)

$$\begin{aligned} \mathsf{Heart} &\sqsubseteq \mathsf{MuscularOrgan} \sqcap \\ &\quad \exists \mathsf{isPartOf}.\mathsf{CirculatorySystem} \\ \mathsf{HeartDisease} &\equiv \mathsf{Disease} \sqcap \\ &\quad \exists \mathsf{affects}.\mathsf{Heart} \\ \mathsf{VascularDisease} &\equiv \mathsf{Disease} \sqcap \\ &\quad \exists \mathsf{affects}.(\exists \mathsf{isPartOf}.\mathsf{CirculatorySystem}) \end{aligned}$$

– Ground facts (aka **ABox**) use the terminology (data)

$$\begin{aligned} \mathsf{John} : \mathsf{Patient} &\sqcap \\ &\exists \mathsf{suffersFrom}.\mathsf{HeartDisease} \end{aligned}$$

Ontology Applications

THE SEMANTIC WEB

What is the Semantic Web?

What is the Semantic Web?

- According to **TBL** circa 1998:

  "... a **consistent logical web of data** ..." in which
  "... information is given **well-defined meaning** …"

What is the Semantic Web?

- According to **TBL** circa 1998:

  "... a **consistent logical web of data** ..." in which
  "... information is given **well-defined meaning** …"

- By now has evolved into:

  "a platform for distributed applications and sharing (linking) data"

What is the Semantic Web?

- According to **TBL** circa 1998:

  "... a **consistent logical web of data** ..." in which
  "... information is given **well-defined meaning** …"

- By now has evolved into:

  "a platform for distributed applications and sharing (linking) data"

  – **RDF** provides uniform syntactic structure for data

  – **Ontologies** provide machine readable schemas

What is the Semantic Web?

- According to **TBL** circa 1998:

  "... a **consistent logical web of data** ..." in which
  "... information is given **well-defined meaning** …"

- By now has evolved into:

  "a platform for distributed applications and sharing (linking) data"

  – **RDF** provides uniform syntactic structure for data
  – **Ontologies** provide machine readable schemas

- **A wide ranging research effort:**
  – aimed at extracting "useful information" from web content
  – with KR (in particular ontologies) playing a key role

Web Ontology Languages

- RDF extended to **RDFS**, a primitive ontology language
  - classes and properties; sub/super-classes (and properties); range and domain (of properties)
- But RDFS **lacks** important **features**, e.g.:
  - existence/cardinality constraints; transitive/inverse properties; localised range and domain constraints, …
- And RDF(S) has “higher order flavour” with no (later **non-standard**) **formal semantics**
  - difficult to understand
  - difficult to provide reasoning support

From RDFS to OWL

- **OIL** language developed in **On-To-Knowledge** project



From RDFS to OWL

- **OIL** language developed in **On-To-Knowledge** project
- **DAML-ONT** language later developed in **DAML** program
- Efforts soon merged to produce **DAML+OIL**
  - Further development carried out by “Joint EU/US Committee”



From RDFS to OWL

- **OIL** language developed in **On-To-Knowledge** project
- **DAML-ONT** language later developed in **DAML** program
- Efforts soon merged to produce **DAML+OIL**
  - Further development carried out by “Joint EU/US Committee”
- DAML+OIL submitted to W3C as basis for standardisation
- **WebOnt** Working Group formed
  - WebOnt developed OWL language based on DAML+OIL
  - OWL became a W3C recommendation
  - “Web-friendly” syntax for $\mathcal{SHOIN}$

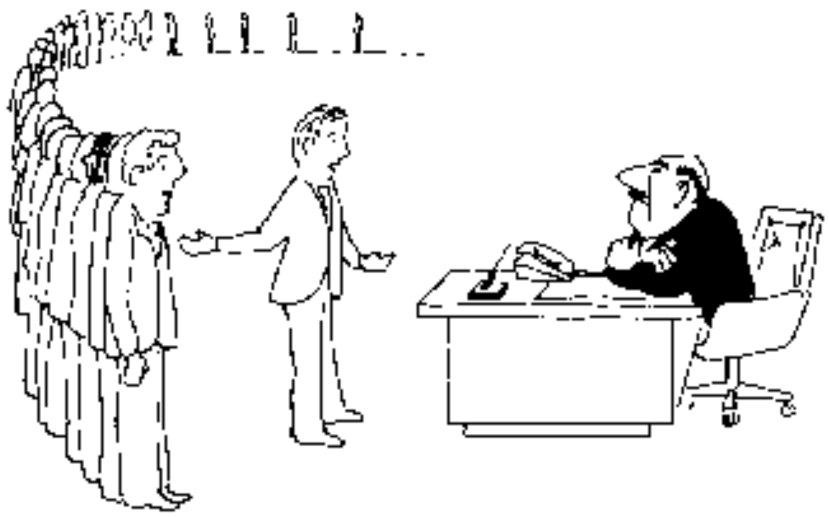
Why (Description) Logic?

- OWL exploits results of 20+ years of DL research
  - Well defined (model theoretic) **semantics**

Constructor	DL Syntax	Example	FOL Syntax
intersectionOf	$C_1 \sqcap \ldots \sqcap C_n$	Human $\sqcap$ Male	$C_1(x) \wedge \ldots \wedge C_n(x)$
unionOf	$C_1 \sqcup \ldots \sqcup C_n$	Doctor $\sqcup$ Lawyer	$C_1(x) \vee \ldots \vee C_n(x)$
complementOf	$\neg C$	$\neg$Male	$\neg C(x)$
oneOf	$\{x_1\} \sqcup \ldots \sqcup \{x_n\}$	{john} $\sqcup$ {mary}	$x = x_1 \vee \ldots \vee x = x_n$
allValuesFrom	$\forall P.C$	$\forall$hasChild.Doctor	$\forall y.P(x,y) \rightarrow C(y)$
someValuesFrom	$\exists P.C$	$\exists$hasChild.Lawyer	$\exists y.P(x,y) \wedge C(y)$
maxCardinality	$\leqslant nP$	$\leqslant$1hasChild	$\exists^{\leqslant n} y.P(x,y)$
minCardinality	$\geqslant nP$	$\geqslant$2hasChild	$\exists^{\geqslant n} y.P(x,y)$

Why (Description) Logic?

- OWL exploits results of 20+ years of DL research
  - Well defined (model theoretic) **semantics**
  - **Formal properties** well understood (complexity, decidability)



I can't find an efficient algorithm, but neither can all these famous people.

[Garey & Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, 1979.]

Why (Description) Logic?

- OWL exploits results of 20+ years of DL research
  - Well defined (model theoretic) **semantics**
  - **Formal properties** well understood (complexity, decidability)
  - Known **reasoning algorithms**

$\sqcap$-rule	if 1. $(C_1 \sqcap C_2) \in \mathcal{L}(v)$, v is not indirectly blocked, and 2. $\{C_1, C_2\} \not\subseteq \mathcal{L}(v)$ then $\mathcal{L}(v) \longrightarrow \mathcal{L}(v) \cup \{C_1, C_2\}$.
$\sqcup$-rule	if 1. $(C_1 \sqcup C_2) \in \mathcal{L}(v)$, v is not indirectly blocked, and 2. $\{C_1, C_2\} \cap \mathcal{L}(v) = \emptyset$ then $\mathcal{L}(v) \longrightarrow \mathcal{L}(v) \cup \{E\}$ for some $E \in \{C_1, C_2\}$
$\exists$-rule	if 1. $\exists r.C \in \mathcal{L}(v_1)$, v_1 is not blocked, and 2. v_1 has no safe r-neighbour v_2 with $C \in \mathcal{L}(v_1)$, then create a new node v_2 and an edge $\langle v_1, v_2 \rangle$ with $\mathcal{L}(v_2) = \{C\}$ and $\mathcal{L}(\langle v_1, v_2 \rangle) = \{r\}$.
$\forall$-rule	if 1. $\forall r.C \in \mathcal{L}(v_1)$, v_1 is not indirectly blocked, and 2. there is an r-neighbour v_2 of v_1 with $C \notin \mathcal{L}(v_2)$ then $\mathcal{L}(v_2) \longrightarrow \mathcal{L}(v_2) \cup \{C\}$.
$\forall_+$-rule	if 1. $\forall r.C \in \mathcal{L}(v_1)$, v_1 is not indirectly blocked, and 2. there is some role r' with Trans(r') and $r' \mathrel{\underline{\circledast}} r$ 3. there is an r'-neighbour v_2 of v_1 with $\forall r'.C \notin \mathcal{L}(v_2)$ then $\mathcal{L}(v_2) \longrightarrow \mathcal{L}(v_2) \cup \{\forall r'.C\}$.
choose-rule	if 1. $\leqslant n\, r.C \in \mathcal{L}(v_1)$, v_1 is not indirectly blocked, and 2. there is an r-neighbour v_2 of v_1 with $\{C, \dot{\neg} C\} \cap \mathcal{L}(v_2) = \emptyset$ then $\mathcal{L}(v_2) \longrightarrow \mathcal{L}(v_2) \cup \{E\}$ for some $E \in \{C, \dot{\neg} C\}$.
$\geqslant$-rule	if 1. $\geqslant n\, r.C \in \mathcal{L}(v)$, v is not blocked, and 2. there are not n safe r-neighbours $v_1, \ldots, v_n$ of v with $C \in \mathcal{L}(v_i)$ and $v_i \neq v_j$ for $1 \leqslant i < j \leqslant n$

Why (Description) Logic?

- OWL exploits results of 20+ years of DL research
  - Well defined (model theoretic) **semantics**
  - **Formal properties** well understood (complexity, decidability)
  - Known **reasoning algorithms**
  - **Scalability** demonstrated by **implemented systems**

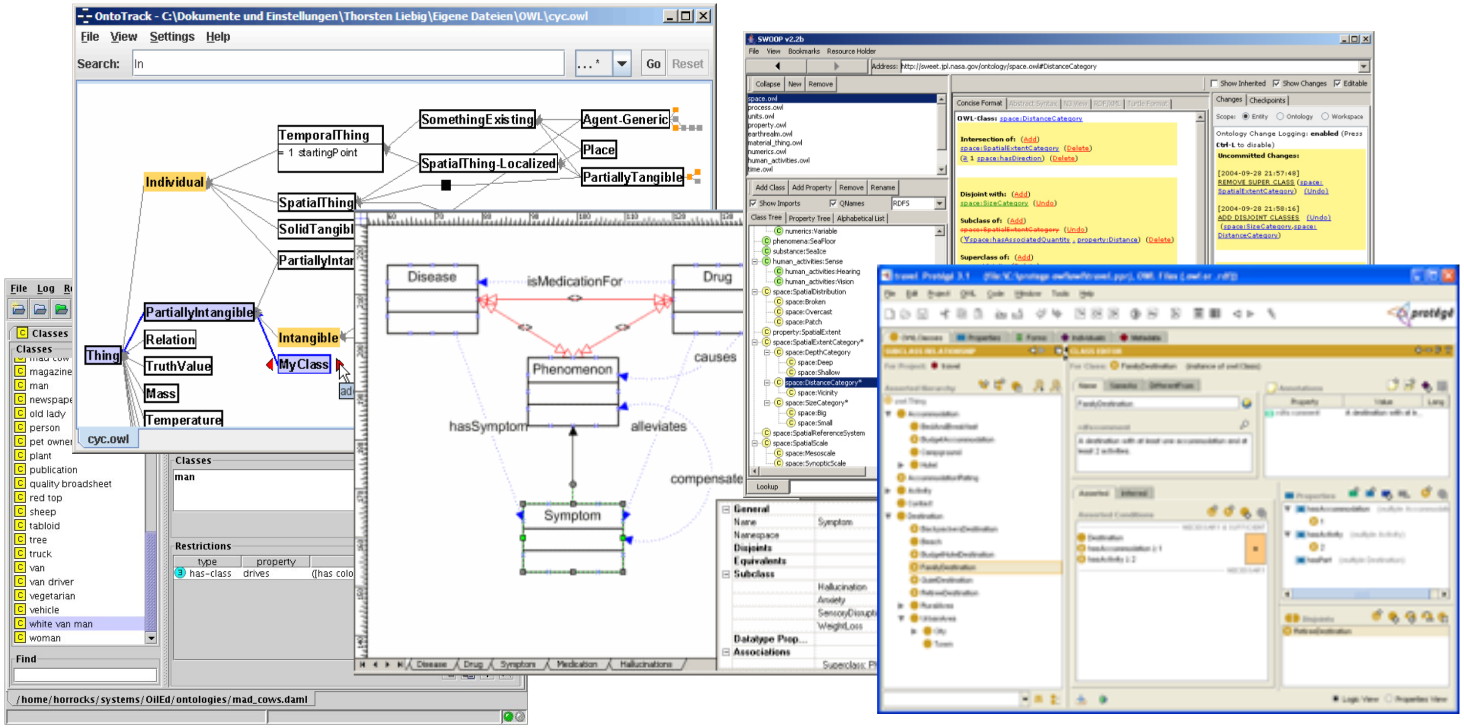
Tools, Tools, Tools

Major benefit of OWL has been huge increase in range and sophistication of tools and infrastructure:

Tools, Tools, Tools

Major benefit of OWL has been huge increase in range and sophistication of tools and infrastructure:

- Editors/development environments



Tools, Tools, Tools

Major benefit of OWL has been huge increase in range and sophistication of tools and infrastructure:

- Editors/development environments
- Reasoners

Hermit

FaCT++

Racer

MINDSWAP Pellet

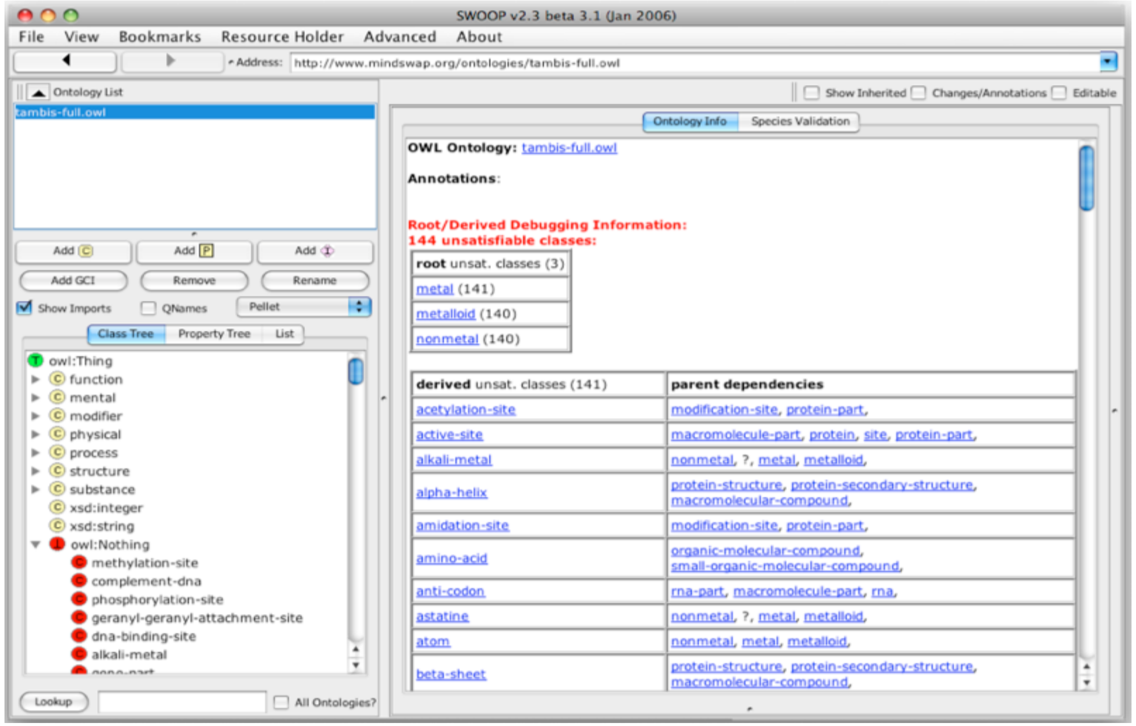
KAON2

CEL

Tools, Tools, Tools

Major benefit of OWL has been huge increase in range and sophistication of tools and infrastructure:

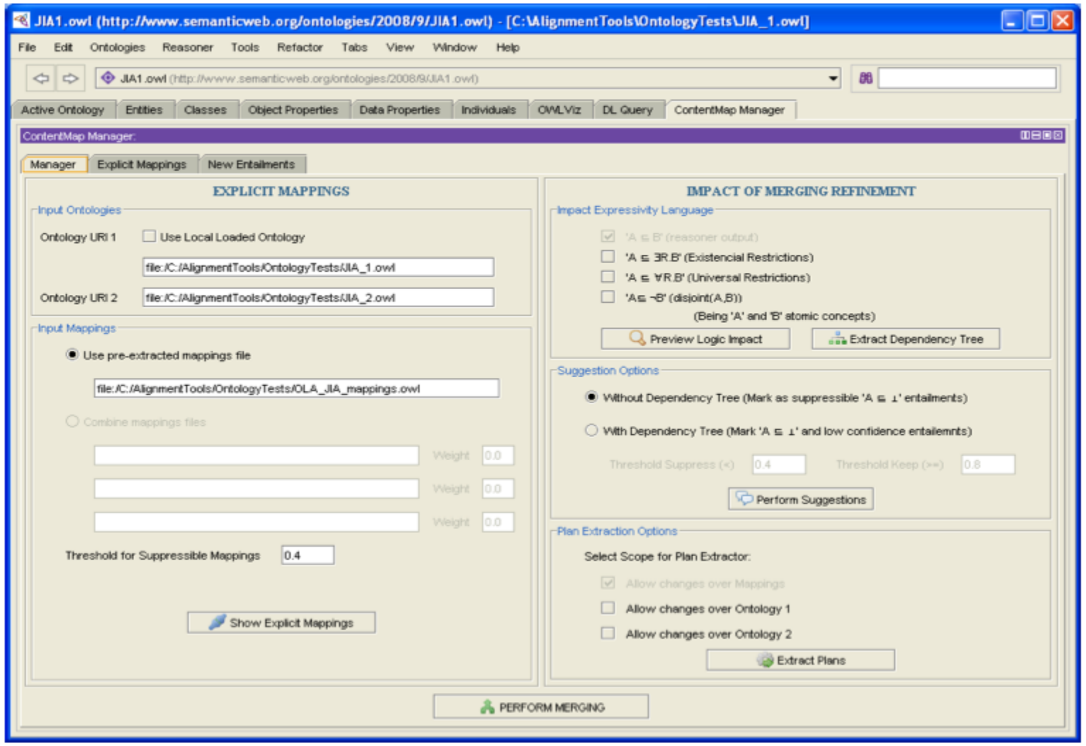
- Editors/development environments
- Reasoners
- Explanation, justification and pinpointing



Tools, Tools, Tools

Major benefit of OWL has been huge increase in range and sophistication of tools and infrastructure:

- Editors/development environments
- Reasoners
- Explanation, justification and pinpointing
- Integration and modularisation



Tools, Tools, Tools

Major benefit of OWL has been huge increase in range and sophistication of tools and infrastructure:

- Editors/development environments
- Reasoners
- Explanation, justification and pinpointing
- Integration and modularisation
- APIs, in particular the [OWL API]

Revision 1403 - (download) (annotate)
Fri Dec 18 17:14:37 2009 UTC (4 months, 2 weeks ago) by matthewhorridge
File size: 4711 byte(s)

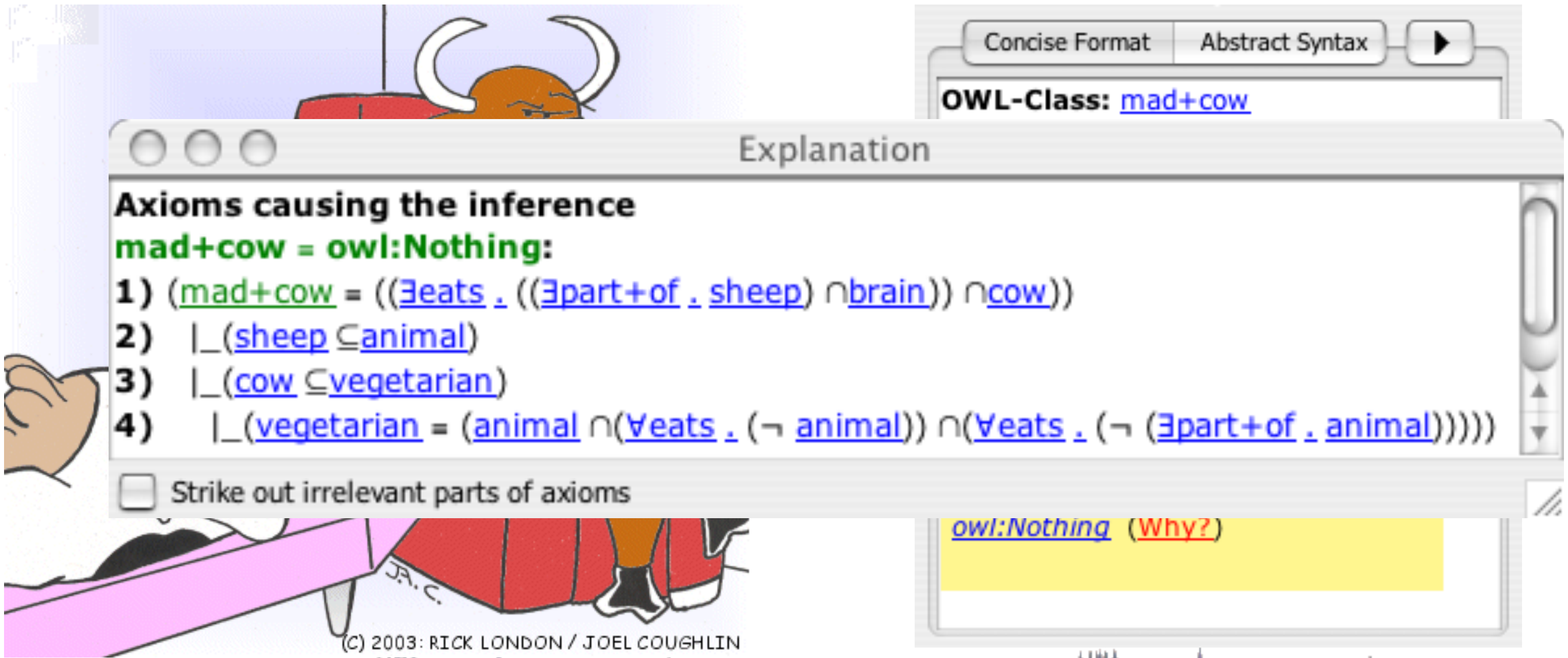
```
package org.coode.owlapi.examples;

import org.semanticweb.owlapi.apibinding.OWLManager;
import org.semanticweb.owlapi.model.*;
import org.semanticweb.owlapi.util.DefaultPrefixManager;
/*
 * Copyright (C) 2009, University of Manchester
 *
 * Modifications to the initial code base are copyright of their
 * respective authors, or their employers as appropriate.  Authorship
 * of the modifications may be determined from the ChangeLog placed at
 * the end of this file.
 *
 * This library is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 2.1 of the License, or (at your option) any later version.

 * This library is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
```

Why Ontology Reasoning?

- Developing and maintaining quality ontologies is ***hard***

Why Ontology Reasoning?

- Developing and maintaining quality ontologies is ***hard***
- Reasoners allow domain experts to check if, e.g.:
  – classes are consistent (no “obvious” errors)

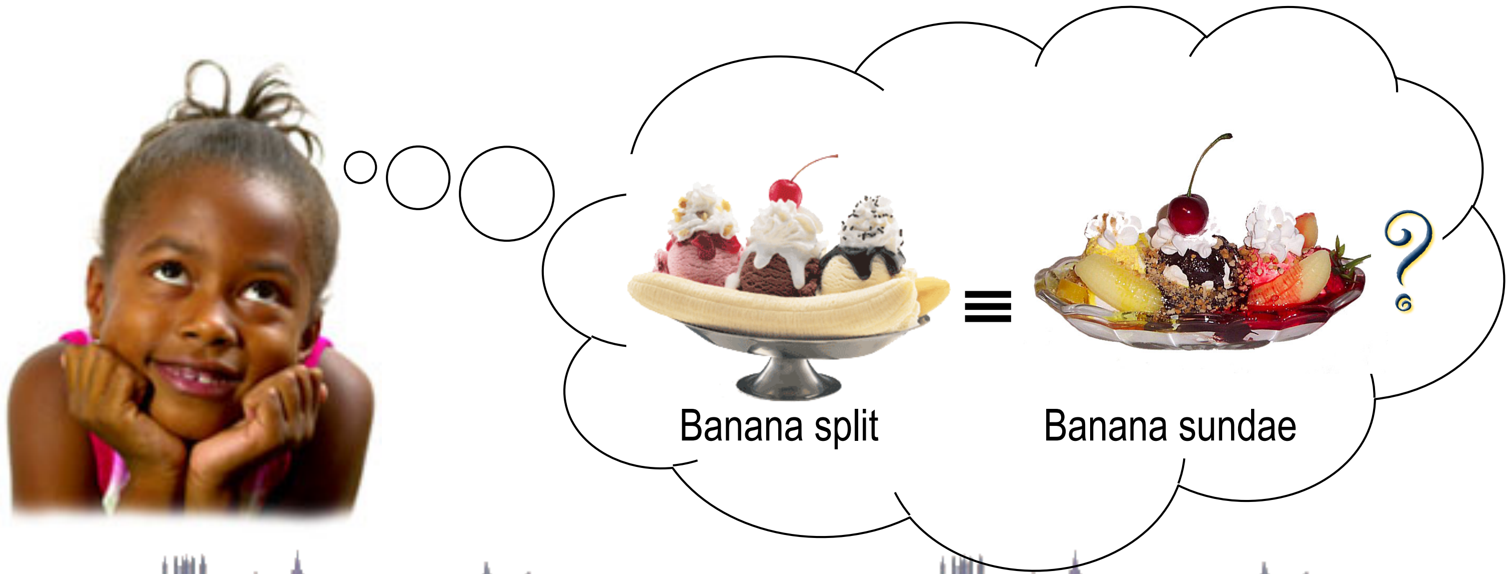


Why Ontology Reasoning?

- Developing and maintaining quality ontologies is ***hard***
- Reasoners allow domain experts to check if, e.g.:
  - classes are consistent (no “obvious” errors)
  - expected subsumptions hold (consistent with intuitions)

Why Ontology Reasoning?

- Developing and maintaining quality ontologies is ***hard***
- Reasoners allow domain experts to check if, e.g.:
  - classes are consistent (no “obvious” errors)
  - expected subsumptions hold (consistent with intuitions)
  - unexpected equivalences hold (unintended synonyms)



Ontology Applications

- OWL ontologies being **deployed** in increasing number and range of applications
  - eScience, eCommerce, geography, engineering, defence, …
  - major impact in healthcare and life sciences
- Now a mainstream technology supported by, e.g., **Oracle 11g**
  - Increasing impact in business applications

Ontology Applications

Designing and Building Business Ontologies

An Intensive 4-DAY SEMINAR with Workshops and Demonstrations, on Semantically Enabling the Enterprise led by Dave McComb and Simon Robe

---

Seminar Objectives

Participants will:

- Gain an understanding of what an ontology is and what it can be used for.
- Understand how representing information in an ontology goes beyond a conceptual model or a simple taxonomy
- Understand the difference between frame based/ declarative classes and description logic based/ derivable classes.
- Understand the difference between open world and closed world models.
- Understand the basic principles for designing Ontologies for corporate applications.

Tuition Fee: $2,450

Healthcare and Life Sciences

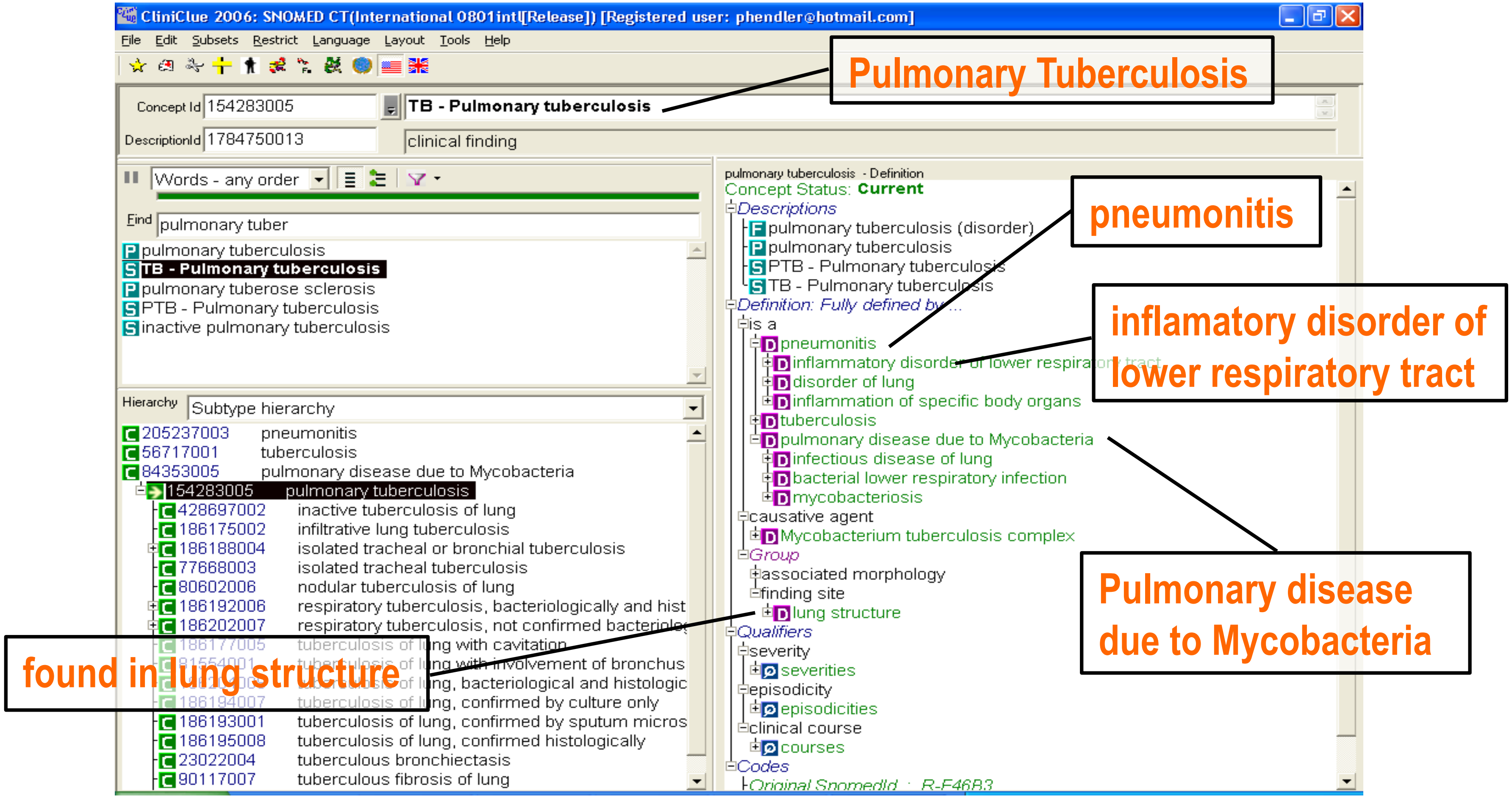
- **OBO foundry** includes more than 100 biological and biomedical ontologies
- **Siemens** "actively building OWL based clinical solutions"
- OWL tools used to find and repair critical errors in ontology used at **Columbia Presbyterian**
- **SNOMED-CT** (Clinical Terms) ontology
  - used in healthcare systems of more than 15 countries, including Australia, Canada, Denmark, Spain, Sweden and the UK
  - also used by major US providers, e.g., Kaiser Permanente
  - ontology provides common vocabulary for recording clinical data

Case Study: SNOMED

It's **BIG** – over **400,000 concepts**

Case Study: SNOMED

It's **BIG** – over **400,000 concepts**



Case Study: SNOMED

- **Kaiser Permanente** extending SNOMED to express, e.g.:
  - *non-viral pneumonia* (negation)
  - *infectious pneumonia* is caused by a *virus* or a *bacterium* (disjunction)
  - *double pneumonia* occurs in two *lungs* (cardinalities)
- This is easy in **SNOMED-OWL**
  - but reasoner failed to find expected subsumptions, e.g., that *bacterial pneumonia* is a kind of *non-viral pneumonia*
- Ontology highly **under-constrained**: need to add disjointness axioms (at least)
  - *virus* and *bacterium* must be disjoint

Case Study: SNOMED

- Adding disjointness led to **surprising results**
  - many classes become inconsistent, e.g., *percutanious embolization of hepatic artery using fluoroscopy guidance*
- Cause of **inconsistencies** identified as class *groin*
  - *groin* asserted to be subclass of both *abdomen* and *leg*
  - *abdomen and leg* are disjoint
  - modelling of *groin* (and other similar “junction” regions) identified as incorrect

Case Study: SNOMED

- Correct modelling of groin is quite complex, e.g.:
  - groin has a part that is part of the abdomen, and has a part that is part of the leg (***inverse properties***)

    $$\mathsf{Groin} \sqsubseteq \exists \mathsf{hasPart}.(\exists \mathsf{isPartOf}.\mathsf{Abdomen}))$$
    $$\mathsf{Groin} \sqsubseteq \exists \mathsf{hasPart}.(\exists \mathsf{isPartOf}.\mathsf{Leg})$$
    $$\mathsf{hasPart} \equiv \mathsf{isPartOf}^{-}$$

  - all parts of the groin are part of the abdomen or the leg (**disjunction**)

    $$\mathsf{Groin} \sqsubseteq \forall \mathsf{hasPart}.(\exists \mathsf{isPartOf}.(\mathsf{Abdomen} \sqcup \mathsf{Leg}))$$

  - ...

Case Study: SNOMED

What we learned:

- Ontology engineering is **error prone**
  - errors of omission (e.g., disjointness) and commission (e.g., modelling of groin)
- **Expressive features** of OWL are sometimes needed
- Sophisticated tool support is **essential**
  - handling ontologies of this size is challenging
  - domain experts (and logicians!) often need help to understand the (root) cause of both inconsistencies and non-subsumptions
  - surprising and unexplained (non-) inferences are frustrating for users and may cause them to lose faith in the reasoner

What About Scalability?

- Tools only **useful in practice** if they can deal with large ontologies and/or large data sets
- Unfortunately, many ontology languages are highly intractable
  - OWL 2 satisfiability is **2NExpTime-complete** w.r.t. schema
  - and **NP-Hard** w.r.t. data (upper bound open)
- Problem addressed in practice by
  - Algorithms that work well in **typical cases**
  - Highly **optimised implementations**
  - Use of **tractable fragments**

Reasoning Algorithms

Most OWL reasoners based on (hyper-) **tableau**

- Reasoning tasks reducible to (un)**satisfiability**
  - E.g., $\mathcal{O} \models \mathsf{HeartDisease} \sqsubseteq \mathsf{VascularDisease}$ iff $\mathcal{O} \cup \{x : (\mathsf{HeartDisease} \sqcap \neg \mathsf{VascularDisease})\}$ is *not* satisfiable
- Algorithm tries to **construct** (abstraction of) a **model**
- Success trivially proves **non-subsumption**
  - we have constructed a counter-model
- Model search designed such that failure proves non-existence of model, and hence **subsumption**

Highly Optimised Implementations

- Lazy unfolding
- Simplification and rewriting
- Search optimisations
- Caching
- Optimised blocking
- Heuristics
- Fast semi-decision procedures
- Algebraic methods
- Nominal absorption
- Individual reuse
- ...

Computationally **sub-optimal**, but **highly effective** in practice

Problem Solved?

Implementation of ExpTime algorithms is futile!

Problem Solved?

Identify (class of) problematic ontologies

Problem Solved?

Identify (class of) problematic ontologies

Implement/ Optimise

Problem Solved?

Identify (class of) problematic ontologies

Implement/ Optimise

Deploy in applications

Problem Solved?

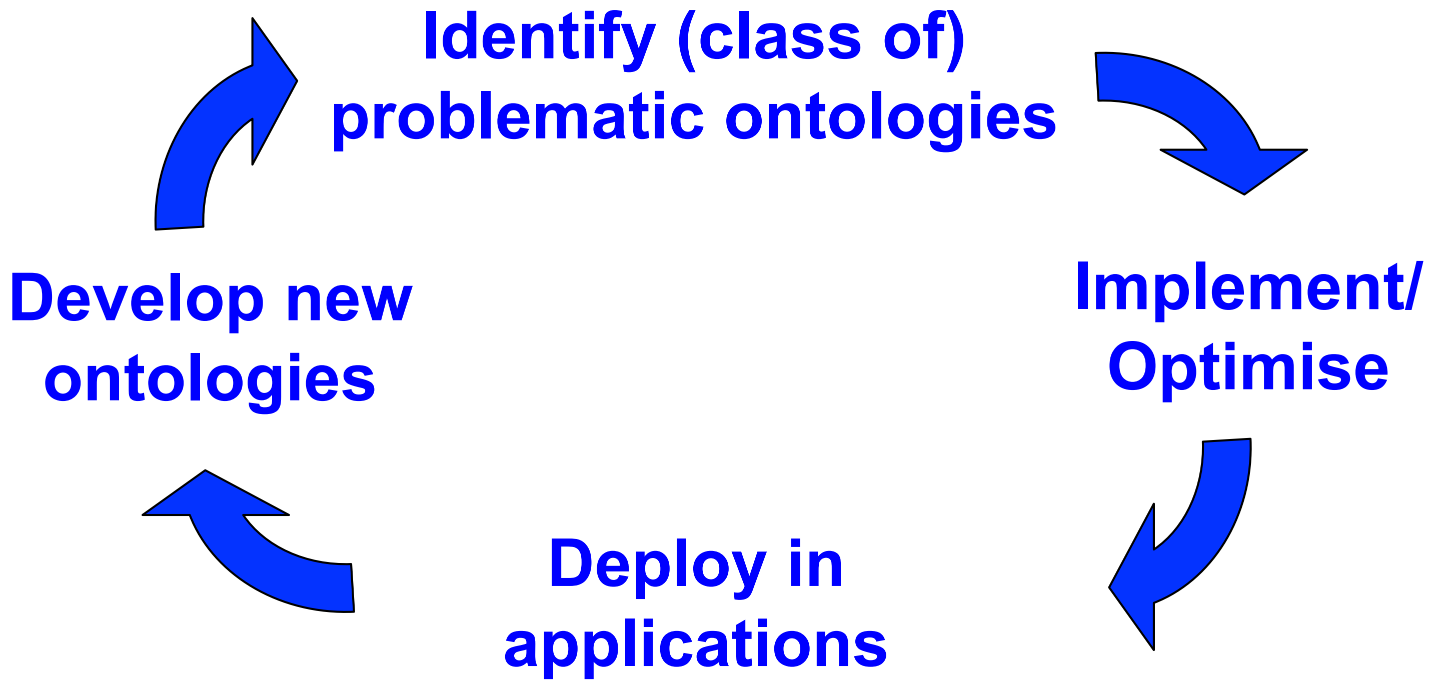
Identify (class of) problematic ontologies

Implement/ Optimise

Deploy in applications

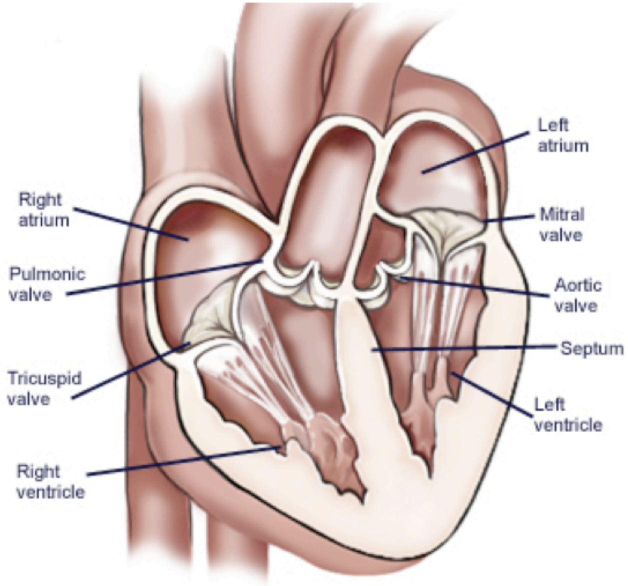
Develop new ontologies

Problem Solved?



Scalability Issues

- Problems with very **large and/or cyclical ontologies**



$$\begin{aligned}
\mathsf{LeftSide} &\sqsubseteq \exists \mathsf{hasComponent}.\mathsf{AorticValve} \\
\mathsf{LeftSide} &\sqsubseteq \exists \mathsf{hasComponent}.\mathsf{MitralValve} \\
\mathsf{AorticValve} &\sqsubseteq \exists \mathsf{hasConnection}.\mathsf{LeftVentircle} \\
\mathsf{MitralValve} &\sqsubseteq \exists \mathsf{hasConnection}.\mathsf{LeftVentircle} \\
\mathsf{LeftVentricle} &\sqsubseteq \exists \mathsf{isDivisionOf}.\mathsf{LeftSide}
\end{aligned}$$

- Ontologies may define 10s/100s of thousands of terms
- Can lead to construction of *very* large models

Scalability Issues

- Problems with **large data sets** (ABoxes)
  - Main reasoning problem is (conjunctive) query answering, e.g., retrieve all patients suffering from vascular disease:
    $Q(x) \leftarrow \mathsf{Patient}(x) \wedge \mathsf{suffersFrom}(x, y) \wedge \mathsf{VascularDisease}(y)$
  - Decidability still open for OWL, although minor restrictions (on cycles in non-distinguished variables) restore decidability
  - Query answering reduced to standard decision problem, e.g., by checking for each individual x if $\mathcal{O} \models Q(x)$
  - Model construction starts with *all* ground facts (data)
- Typical applications may use data sets with **10s/100s of millions** of individuals (or more)

OWL 2

- **New version** of OWL became a rec in October 2009
- **Extends OWL** with a small but useful set of features
  - That are needed in applications
  - For which semantics and reasoning techniques well understood
  - That tool builders are willing and able to support
- Adds **profiles**
  - Language subsets with useful computational properties

New Language Features

Four kinds of new feature:

- **Increased expressive power**
  - qualified cardinality restrictions, e.g.:
    - persons having two friends who are republicans
  - property chains, e.g.:
    - the brother of your parent is your uncle
  - local reflexivity restrictions, e.g.:
    - narcissists love themselves
  - reflexive, irreflexive, and asymmetric properties, e.g.:
    - nothing can be a proper part of itself (irreflexive)
  - disjoint properties, e.g.:
    - you can't be both the parent of and child of the same person
  - keys, e.g.:
    - country + license plate constitute a unique identifier for vehicles

New Language Features

Four kinds of new feature:

- **Extended Datatypes**
  - Much wider range of XSD Datatypes supported, e.g.:

    Integer, string, boolean, real, decimal, float, datatime, …
  - User-defined datatypes using facets, e.g.:

max weight of an airmail letter:
xsd:integer maxInclusive ”20"^^xsd:integer

format of Italian registration plates:
xsd:string xsd:pattern "[A-Z]{2} [0-9]{3}[A-Z]{2}

New Language Features

Four kinds of new feature:

- **Metamodelling and annotations**
  - Restricted form of metamodelling via “punning”, e.g.:

    SnowLeopard subClassOf BigCat (i.e., a class)

    SnowLeopard type EndangeredSpecies (i.e., an individual)
  - Annotations of axioms as well as entities, e.g.:

    SnowLeopard type EndangeredSpecies (“source: WWF”)
  - Even annotations of annotations

New Language Features

Four kinds of new feature:

- **Syntactic sugar**
  - Disjoint unions, e.g.:

    Element is the DisjointUnion of Earth Wind Fire Water

    i.e., Element is equivalent to the union of Earth Wind Fire Water

    Earth Wind Fire Water are pair-wise disjoint
  - Negative assertions, e.g.:

    Mary is not a sister of Ian

    21 is not the age of Ian

Alternative Syntaxes

- Normative exchange syntax is **RDF/XML**

```
<owl:Class rdf:about="#Heart">
    <owl:equivalentClass>
        <owl:Class>
            <owl:intersectionOf rdf:parseType="Collection">
                <rdf:Description rdf:about="#MuscularOrgan"/>
                <owl:Restriction>
                    <owl:onProperty rdf:resource="#isPartOf"/>
                    <owl:someValuesFrom rdf:resource="#CirculatorySystem"/>
                </owl:Restriction>
            </owl:intersectionOf>
        </owl:Class>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="&owl;Thing"/>
</owl:Class>
```

Alternative Syntaxes

- Normative exchange syntax is **RDF/XML**
- **Functional syntax** mainly intended for language spec

```
EquivalentClasses(Heart
   ObjectIntersectionOf(ObjectSomeValuesFrom(isPartOf CirculatorySystem)
      MuscularOrgan))
```

Alternative Syntaxes

- Normative exchange syntax is **RDF/XML**
- **Functional syntax** mainly intended for language spec
- **XML syntax** for interoperability with XML toolchain

```
<EquivalentClasses>
    <Class URI="Heart"/>
    <ObjectIntersectionOf>
        <Class URI="MuscularOrgan"/>
        <ObjectSomeValuesFrom>
            <ObjectProperty URI="isPartOf"/>
            <Class URI="CirculatorySystem"/>
        </ObjectSomeValuesFrom>
    </ObjectIntersectionOf>
</EquivalentClasses>
```

Alternative Syntaxes

- Normative exchange syntax is **RDF/XML**
- **Functional syntax** mainly intended for language spec
- **XML syntax** for interoperability with XML toolchain
- **Manchester syntax** for better readability

```
         Class: Heart
  EquivalentTo: MuscularOrgan
                that isPartOf CirculatorySystem
```

Profiles

- OWL 2 defines three **profiles**:
  - **EL**: polynomial time reasoning for schema and data
  - **QL**: logspace query answering using RDBMs
  - **RL**: polynomial time query answering using rule-extended DBs
- OWL defined only one profile: **OWL Lite**
  - DL research not consulted in design of OWL Lite
  - resulting “fragment” not in fact very Lite (EXPTIME-complete)

OWL 2 EL

- A (near maximal) fragment of OWL 2 such that
  - satisfiability checking is in PTime (**PTime-Complete**)
  - data complexity of query answering also PTime-Complete

OWL 2 EL

- A (near maximal) fragment of OWL 2 such that
  - satisfiability checking is in PTime (**PTime-Complete**)
  - data complexity of query answering also PTime-Complete
- Based on $\mathcal{EL}$ family of description logics

OWL 2 EL

- A (near maximal) fragment of OWL 2 such that
  - satisfiability checking is in PTime (**PTime-Complete**)
  - data complexity of query answering also PTime-Complete
- Based on $\mathcal{EL}$ family of description logics
- Efficient **saturation** based algorithms
  - derive axioms rather than constructing models, e.g.:

$$\frac{A \sqsubseteq B \quad B \sqsubseteq C}{A \sqsubseteq C} \qquad \frac{A \sqsubseteq B \quad A \sqsubseteq C \quad B \sqcap C \sqsubseteq D}{A \sqsubseteq D}$$

$$\frac{A \sqsubseteq \exists R.B \quad B \sqsubseteq C \quad \exists R.C \sqsubseteq D}{A \sqsubseteq D}$$

OWL 2 QL

- A (near maximal) fragment of OWL 2 such that
  - data complexity of conjunctive query answering in **AC^0**
- Based on **DL-Lite** family of description logics
- Efficient **query rewriting** based algorithms
  - ontology axioms used as rewrite rules for query, e.g.:

$$Q(x) \leftarrow \mathsf{treats}(x,y) \wedge \mathsf{Patient}(y)$$
$$Q(x) \leftarrow \mathsf{Doctor}(x)$$
$$Q(x) \leftarrow \mathsf{Consultant}(x)$$

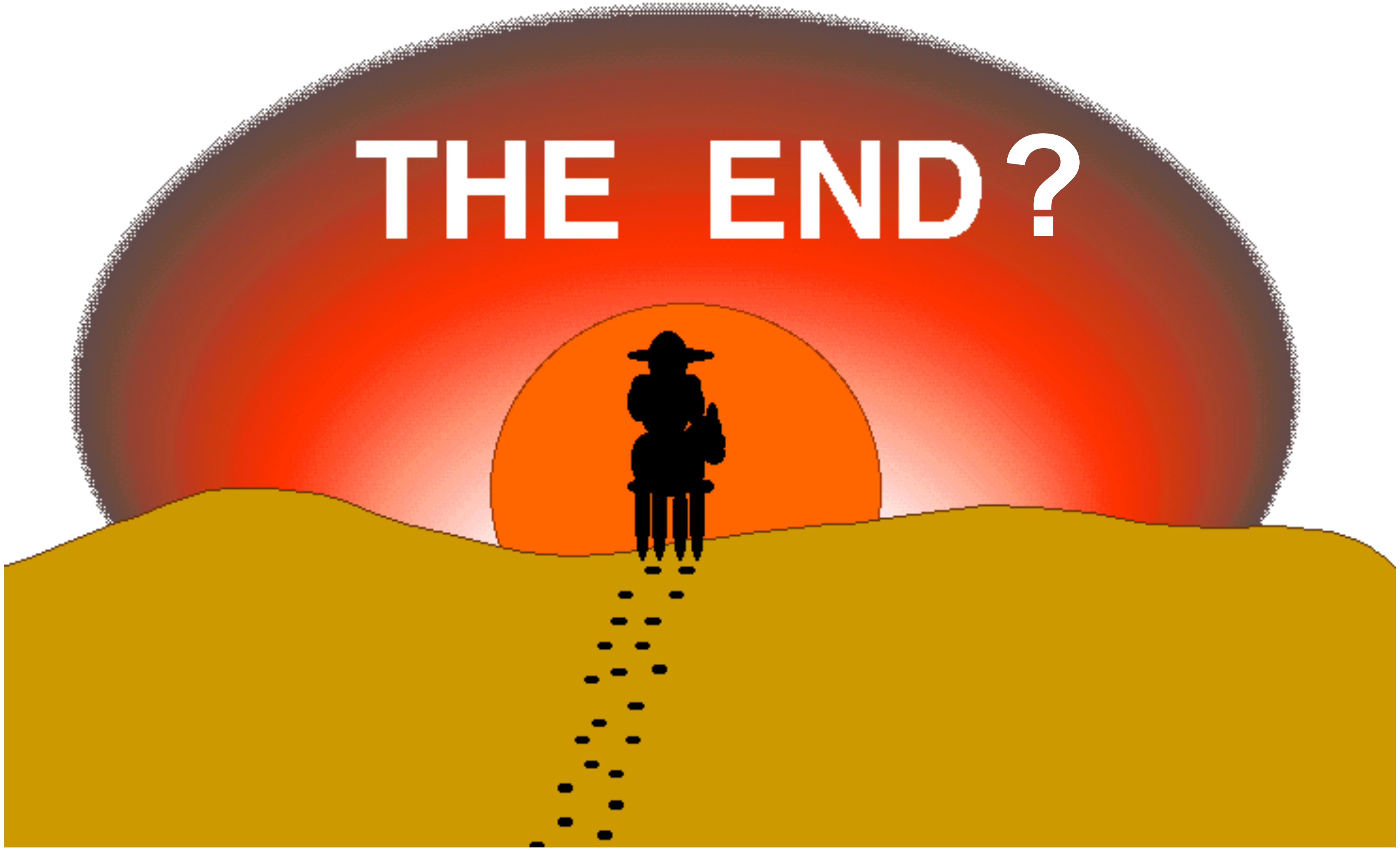
$$\mathsf{Doctor} \sqsubseteq \exists \mathsf{treats}.\mathsf{Patient}$$
$$\mathsf{Consultant} \sqsubseteq \mathsf{Doctor}$$

  - data storage & evaluation of resulting UCQ delegated to RDBMS

$$Q(x) \leftarrow (\mathsf{treats}(x,y) \wedge \mathsf{Patient}(y)) \vee \mathsf{Doctor}(x) \vee \mathsf{Consultant}(x)$$

Profiles as Optimisations

- **EL techniques** as optimisation for OWL classification
  - use saturation algorithm to classify part of ontology
  - use incremental reasoning techniques to add remaining axioms
  - similar optimisation already used to good effect in FaCT++ (can classify extended SNOMED-OWL in 24 minutes)
- **QL techniques** as optimisation for EL query answering
  - in “hybrid” approach, data first extended by partially materialising EL inferences
  - then use modified query rewriting with ontology and extended data


THE END?

Ongoing Research

- Query answering
  - [Kontchakov et al], [Konev et al], [Baader et al], [Glimm et al]
- Diagnosis and repair
  - [Peñaloza et al]
- Reasoning over hidden content
  - [Cuenca Grau et al]
- Probabilistic DLs
  - [Lutz et al]

Ongoing Research

- Optimisation
- Second order DLs
- Temporal DLs
- Fuzzy/rough concepts
- Modularity, alignment and integration
- Integrity constraints
- ...

Ongoing Standardisation Efforts

- Standardised query language
  - SPARQL standard for RDF
  - Currently being extended for OWL, see *http://www.w3.org/TR/sparql11-entailment/*
- RDF
  - Revision currently being considered, see *http://www.w3.org/2009/12/rdf-ws/*


The 9th International Semantic Web Conference
Shanghai International Convention Center, Shanghai, China
Nov 7th -11th, 2010
http://iswc2010.semanticweb.org
ISWC 2010
Semantics for a Better Web!
General Chair
Ian Horrocks
Program Chairs
Peter F. Patel-Schneider
Yue Pan
Local Chair
Yong Yu
Workshop & Tutorial Chairs
Philippe Cudré-Mauroux
Bijan Parsia
Poster & Demo Chairs
Huajun Chen
Axel Polleres
Industry & Semantic Web in Use Chairs
Pascal Hitzler
Peter Mika
Doctoral Consortium Chair
Jeff Pan
Semantic Web Challenge Chairs
Chris Bizer
Diana Maynard
Publicity Chair
Sebastian Rudolph
Metadata Chair
Jie Bao
Proceedings Chair
Birte Glimm
Sponsor Chairs
Kendall Clark
Anand Ranganathan
Local Organization
Dingyi Han
Gui-Rong Xue
Haofen Wang
Lei Zhang
EXPO 2010 SHANGHAI CHINA
Stunning Venue
Great Time
Fascinating City
Excellent Food
SWSA
Semantic Web Science Association
上海交通大学
SHANGHAI JIAO TONG UNIVERSITY
IBM
Research

Thanks To

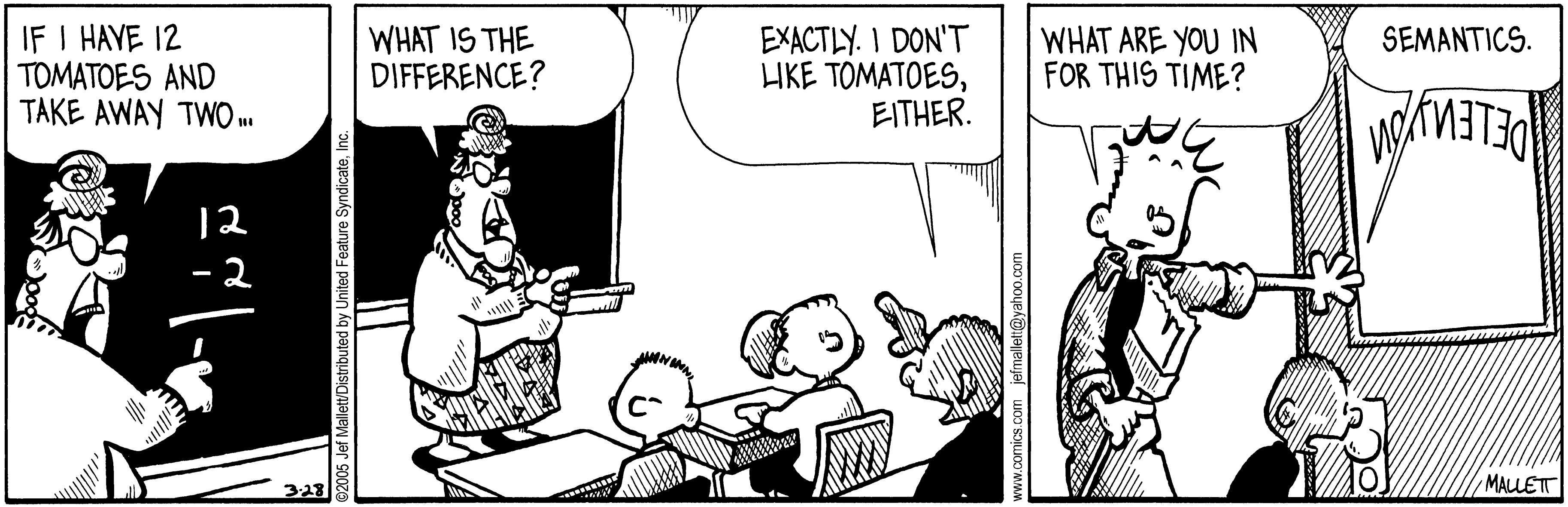
- Boris Motik
- Yevgeny Kazakov
- Héctor Pérez-Urbina
- Rob Shearer
- Bernardo Cuenca Grau
- Birte Glimm






Thank you for listening

Thank you for listening



FRAZZ: © Jeff Mallett/Dist. by United Feature Syndicate, Inc.

Any questions?