

Ontologies and the Semantic Web

Ian Horrocks

<horrocks@cs.man.ac.uk>

Information Management Group

School of Computer Science

University of Manchester

The Semantic Web

Today's Web

BCS HOME | TEXT | CONTACT | SITE MAP | SEARCH | FORUMS | MEMBERS AREA

JOIN BCS | ABOUT | NEWS | AWARDS & EVENTS | GROUPS | PRODUCTS & SERVICES | INFORMATION & ADVICE | CAREERS

You are in: Home /

British Computer Society

Welcome to The British Computer Society

The British Computer Society (BCS) is the industry body for IT professionals, and a Chartered Engineering Institution for Information Technology (IT). With members in over 100 countries around the world, the BCS is the leading professional and learned Society in the field of computers and information systems.

SITE FEATURES

For Members... Browse SFIAPlus™
Explore Browse SFIAPlus™ - the detailed framework for IT knowledge, skills and training. [Browse SFIAPlus™](#)

A Manager's Guide to IT Law
The most relevant legal frameworks, explained in plain English. Includes: IT contracts, employment problems, instructing consultants, intellectual property law and data protection. [Further information about the book](#)

New Review out now
BCS's annual look at the IT industry is out. Security, IT strategy, software, professionalism, networks, storage and more are covered. [Check it out](#)

Latest News
[BCS, Govt and Industry Get to Work on Professionalism in IT Contractors and BCS in New Membership Agreement](#)
[BCS Research Awards Open for Nominations](#)
[IT Architects Offered New BCS Qualification](#)

REGULAR FEATURES

Online Services
I am already registered - [login](#)
I would like to [register](#)
I have [forgotten my password](#)

I would like to:
[Update my contact details](#)
[Renew my membership](#)
[Upgrade my membership](#)

Join the BCS
Why have 16,000 members joined the BCS since the 1st May 2004? Because as IT professionals we recognise the benefits of BCS membership. [Join the BCS today](#)

It's about setting standards. Not standing still.
[BCS Code of Conduct](#)
[BCS Code of Good Practice](#)

Art of IT Competition
The BCS has launched an international competition aimed at all computing students. [View some sample entries and request an entry form](#)

BCS HOME | TEXT | CONTACT | SITE MAP | SEARCH | FORUMS | MEMBERS AREA

JOIN BCS | ABOUT | NEWS | AWARDS & EVENTS | GROUPS | PRODUCTS & SERVICES | INFORMATION & ADVICE | CAREERS

You are in: Home / Products & Services / Professional Development

Professional Development

- Browse SFIAPlus
- Conferences
- Networking
- Professional Experience Award

Individual Continuing Professional Development

Browse SFIAPlus

The BCS promotes the use of the SFIAPlus IT skills, training and development standard which forms the basis of a range of new professional development products and services. The Skills Framework for the Information Age (SFIAPlus) is the high level UK Government backed IT skills competency framework. SFIAPlus combines the SFIAPlus framework of IT skills plus the detailed training and development standard: Industry Structure Model (ISM), developed and promoted for over a decade by the BCS.

Browse SFIAPlus provides BCS Members with free and simple

BCS HOME | TEXT | CONTACT | SITE MAP | SEARCH | FORUMS | MEMBERS AREA

JOIN BCS | ABOUT | NEWS | AWARDS & EVENTS | GROUPS | PRODUCTS & SERVICES | INFORMATION & ADVICE | CAREERS

You are in: Home / Products & Services / Publishing / Books / BCS Review / BCS Review

BCS Review

The British Computer Society Annual Review 2006

Bundling with chapters on today's hot issues within IT, the BCS Review provides analysis of the current concerns and trends within the IT industry and offers practical solutions to combating common problems, such as IT governance and compliance, implementing VoIP and managing the risks of electronic communications.

Each year this unique book brings together the most up-to-date thinking and practical experience of industry professionals on a variety of topics facing the IT community. It presents the very best insights, perspectives and experiences of industry leaders on an annual basis. It's packed full of case studies, best practice guidelines, practical advice and high level issue analysis.

The BCS Annual Review 2006 is available in the following formats:

BCS Review 2006

- 2006
- 2005
- 2004
- 2003
- 2002
- 2001
- 2000
- 1999

Publisher: Again in association with the BCS
Pub. Date: 1st October 2005
ISBN: 1-90340-18-3
Format: Paperback, 210mm x 287mm
Price: £20.00 members
£25.00 non-members

BCS BCS Student Membership

The Art of IT

Thank you for your interest in the student competition. We have posted a selection of entries on these pages.

About the Art of IT Competition

The BCS is encouraging students of computing to use a blank postcard provided by the BCS to show their feelings on the subject of IT - a statement or an illustration, serious or frivolous! The judges are looking for creativity and originality, with entries invited from all drawing capabilities.

The competition is open to both UK and international students and is seeking entries for the categories of **Most Humorous, Thought Provoking and Most Original**.

For each category winner there will be a digital camera up for grabs and a top prize of a digital camcorder for the overall winner.

To request a blank competition postcard, please contact Sarah Moore on smoore@bcs.ac.uk

- Distributed hypertext/hypermedia
- Information accessed via (keyword based) search and browse
- Browser tools render information for **human consumption**

What is the Semantic Web?

- Web was “invented” by **Tim Berners-Lee** (amongst others), a physicist working at CERN
- His vision of the Web was much more ambitious than the reality of the existing (syntactic) Web:



“... a set of **connected applications** ... forming a **consistent logical web of data** ...”

“... an extension of the current web in which information is given **well-defined meaning**, better enabling computers and people to work in cooperation ...”

- This vision of the Web has become known as the **Semantic Web**

Hard Work using “Syntactic Web”

Find images of Peter Patel-Schneider, Frank van Harmelen and Alan Rector...



Rev. Alan M. Gates, Associate Rector of the Church of the Holy Spirit, Lake Forest, Illinois

Impossible (?) using “Syntactic Web”

- Complex queries involving **background knowledge**
 - Find information about “animals that use sonar but are neither bats nor dolphins” , **e.g., Barn Owl**
- Locating information **ies**
 - Travel enquiries
 - Prices of goods and services
 - Results of human activities
- Finding and using “**v**”
 - Given a **DNA sequence**, determine the **proteins** they can produce, and the **biological processes** they control
- Delegating complex **nts”**
 - Book me a holiday **where warm, not too far away, English** and where they spend their holidays



What is the Problem?

Consider a typical web page:

The screenshot shows the BCS website homepage. At the top is a blue navigation bar with the BCS logo and links for HOME, TEXT, CONTACT, SITE MAP, SEARCH, FORUMS, and MEMBERS AREA. Below this is a secondary navigation bar with links for JOIN BCS, ABOUT, NEWS, AWARDS & EVENTS, GROUPS, PRODUCTS & SERVICES, INFORMATION & ADVICE, and CAREERS. The main content area is titled "Welcome to The British Computer Society" and includes a sidebar with a menu (Join BCS, About, News, Awards & Events, Groups, Products & Services, Information & Advice, Careers, Contact) and several featured articles. Red arrows point to specific elements: one points to the main heading, another to a link in the "For Members... Browse SFIPlus" section, a third to a link in the "A Manager's Guide to IT Law" section, a fourth to a link in the "Join the BCS" section, and a fifth to a link in the "Art of IT Competition" section.

- Markup consists of:
 - rendering information (e.g., font size and colour)
 - Hyper-links to related content
- Semantic content is accessible to humans, but not (easily) to computers...

What is the (Proposed) Solution?

- Add semantic annotations to web resources



Dr. Alan Rector, Professor of Computer Science, University of Manchester



Rev. Alan Gates, Associate Rector of the Church of the Holy Spirit, Lake Forest, Illinois

Giving Semantics to Annotations

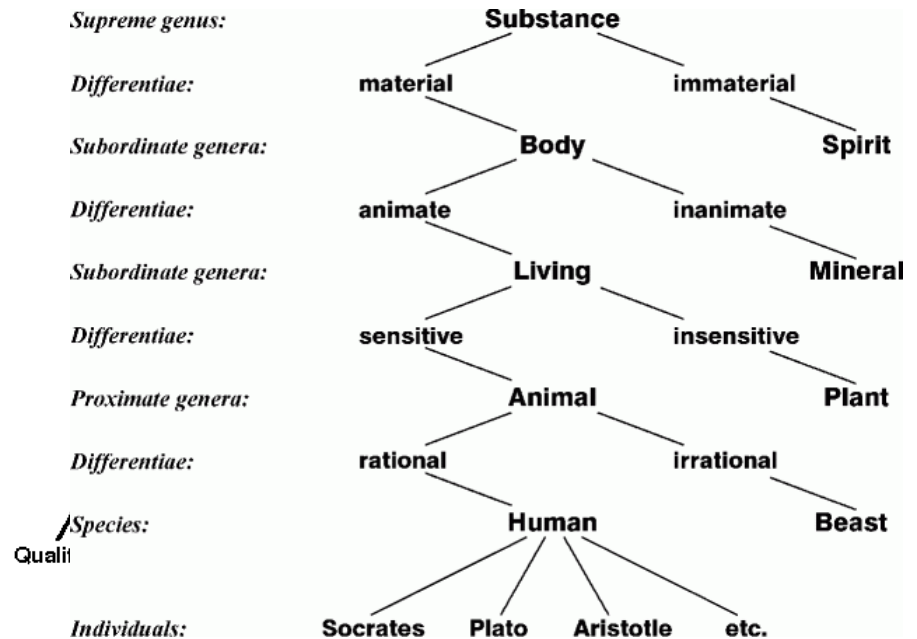
- External agreement on annotations
 - Agree on meaning
 - E.g., Dublin
 - Limited flexibility
 - Limited number of annotations
- Use Ontologies
 - Agree on labels
 - New terms
 - Meaning
 - Can combine/relate terms in multiple ontologies
 - Meanings of labels
 - Existing ones
 - Fully specified



Ontologies

Ontology: Origins and History

- In Philosophy, fundamental branch of metaphysics
 - Studies “being” or “existence” and their **basic categories**
 - Aims to find out what **entities** and **types of entities** exist



Ontology in Information Science

- An ontology is an engineering artefact consisting of:
 - A **vocabulary** used to describe (a particular view of) some domain
 - An **explicit specification** of the **intended meaning** of the vocabulary.
 - Often includes classification based information
 - Constraints capturing **background knowledge** about the domain
- Ideally, an ontology should:
 - Capture a **shared understanding** of a domain of interest
 - Provide a **formal** and **machine manipulable** model

Example Ontology (Protégé)

pizza Protégé 3.1 (file:/Users/horrocks/Work/latex/teaching/Dresden2005/Ontologies/pizza.pprj, OWL Files (.owl or .rdf))

owlClasses Properties Forms Individuals Metadata OWLViz

SUBCLASS RELATIONSHIP CLASS EDITOR

For Project: pizza For Class: RealltalianPizza (instance of owl:Class)

Asserted Hierarchy

- owl:Thing
 - DomainConcept
 - Country
 - IceCream
 - Pizza
 - CheesyPizza
 - InterestingPizza
 - MeatyPizza
 - NamedPizza
 - NonvegetarianPizza
 - RealltalianPizza**
 - SpicyPizza
 - SpicyPizzaEquivalent
 - VegetarianPizza
 - VegetarianPizzaEquivalent
 - VegetarianPizzaEquivalent
 - PizzaBase
 - PizzaTopping
 - ValuePartition

Name

RealltalianPizza

rdfs:comment (en)

describe the members – that all RealltalianPizzas must only have ThinAndCrispy bases.

Annotations

| Property | Value | Lang |
|--------------|----------------------|------|
| rdfs:comment | This defined cl...en | |
| rdfs:label | PizzaltalianaRealpt | |

Asserted Inferred

Asserted Conditions

- Pizza
- hasCountryOfOrigin \ni Italy
- \forall hasBase ThinAndCrispyBase
- \exists hasBase PizzaBase [from Pizza]

Properties

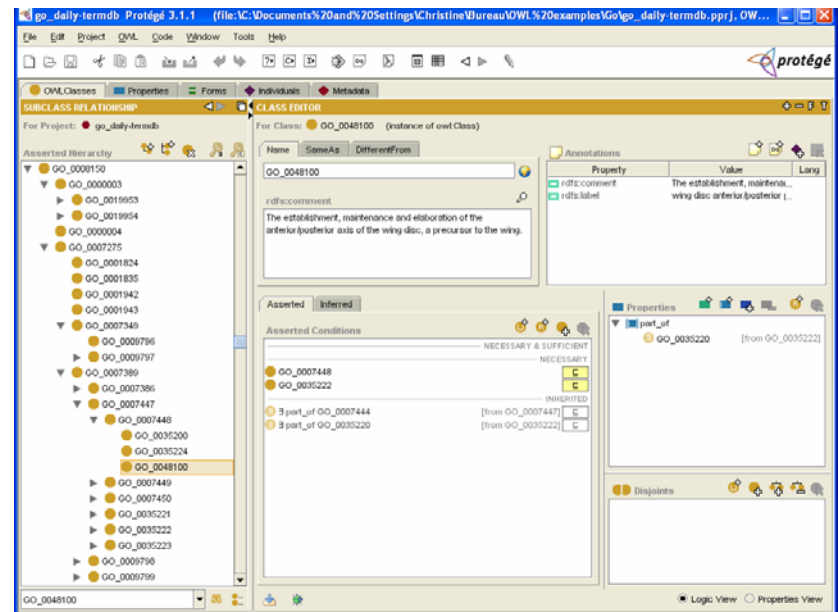
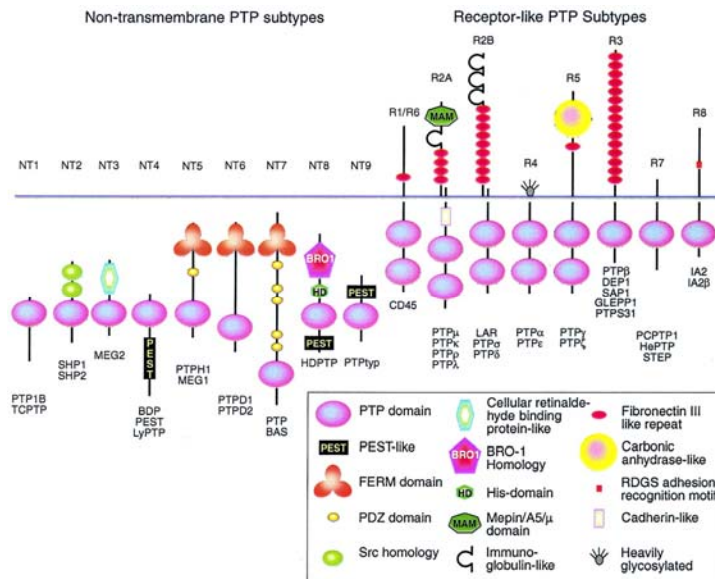
- hasBase (single PizzaBa
 - ThinAndCrispyBase
 - PizzaBase [from Pizza]
- hasCountryOfOrigin
 - Italy
- hasTopping (multiple P

Disjoints

Logic View Properties View

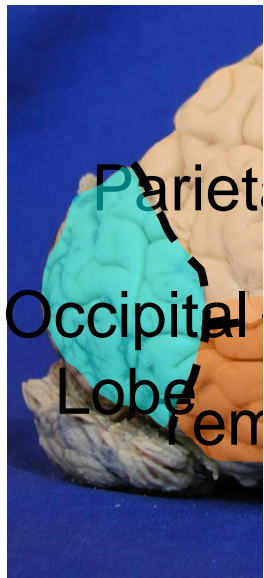
Applications of Ontologies

- e-Science, e.g., Bioinformatics
 - Open Biomedical Ontologies Consortium (GO, MGED)
 - Used e.g., for “in silico” investigations relating theory and data
 - E.g., relating data on phosphatases to (model of) biological knowledge



Applications of Ontologies

- Medicine
 - Building/maintaining terminologies such as Snomed, NCI & Galen



Asserted Inferred

Asserted Conditions

NECESSARY & SUFFICIENT

- Gyrus
- (IsMAEConnectedTo = 1 postCentralGyrus) ⊔ (IsMAEContiguousTo = 1 postCentralGyrus)
- ∀ IsMAEBoundedBy (centralSulcus ⊔ preCentralGyrus)
- IsMAEBoundedBy = 1 centralSulcus
- IsMAEBoundedBy = 1 preCentralSulcus

NECESSARY

- ∀ hasDirectAnatomicalPart (superiorParsPreCentralGyrus ⊔ inferiorParsPreCentralGyrus)
- hasDirectAnatomicalPart = 1 superiorParsPreCentralGyrus
- hasDirectAnatomicalPart = 1 inferiorParsPreCentralGyrus
- Left-preCentralGyrus ⊔ Right-preCentralGyrus

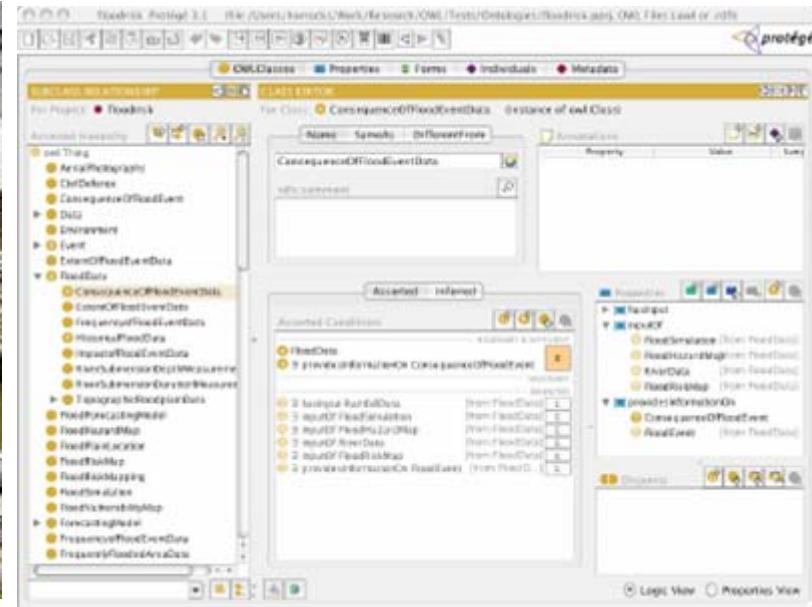
INHERITED

- angularGyrus ⊔ anteriorOrbitalGyrus ⊔ cingulateGyrus ⊔ cuneus ⊔ dentateGyrus ⊔ fronto

properties View

Applications of Ontologies

- Organising complex and semi-structured information
 - UN-FAO, NASA, Ordnance Survey, General Motors, Lockheed Martin, ...



Applications of Ontologies

- Military/Government
 - DARPA, NIST, SAIC, Department of Homeland Security, ...
- The **Semantic Web** and so-called **Semantic Grid**



THE
SEMANTIC
WEB

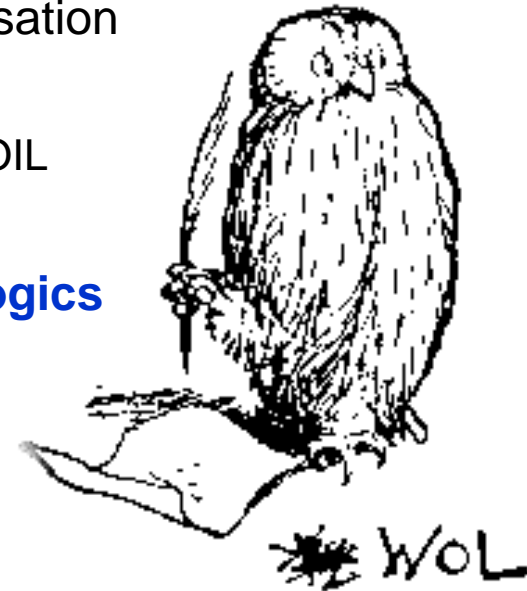
Ontology Languages

Ontology Languages for the Web

- Semantic Web effort led to development of “resource description” language(s)
 - E.g., **RDF**, and later RDF Schema (**RDFS**)
- RDFS is recognisable as an ontology language
 - **Classes** and **properties**
 - **Sub/super-classes** (and properties)
 - **Range** and **domain** (of properties)
- But RDFS **too weak** to describe resources in sufficient detail, e.g.:
 - No **existence/cardinality** constraints
 - No **transitive, inverse or symmetrical** properties
 - No **localised range and domain** constraints
 - ...
- And RDF(S) has “higher order flavour” with non-standard semantics
 - Difficult to provide **reasoning support**

From RDFS to OWL

- Two languages developed to address deficiencies & problems of RDFS:
 - **OIL**: developed by group of (largely) European researchers
 - **DAML-ONT**: developed by group of (largely) US researchers
- Efforts merged to produce **DAML+OIL**
 - Development carried out by “Joint EU/US Committee on Agent Markup Languages”
- DAML+OIL submitted to **W3C** as basis for standardisation
 - Web-Ontology (**WebOnt**) Working Group formed
 - WebOnt developed **OWL** language based on DAML+OIL
 - OWL now a W3C **recommendation** (i.e., a standard)
- OIL, DAML+OIL and OWL based on **Description Logics**
 - OWL is effectively a “Web-friendly” syntax for **SHOIN**



What Are Description Logics?

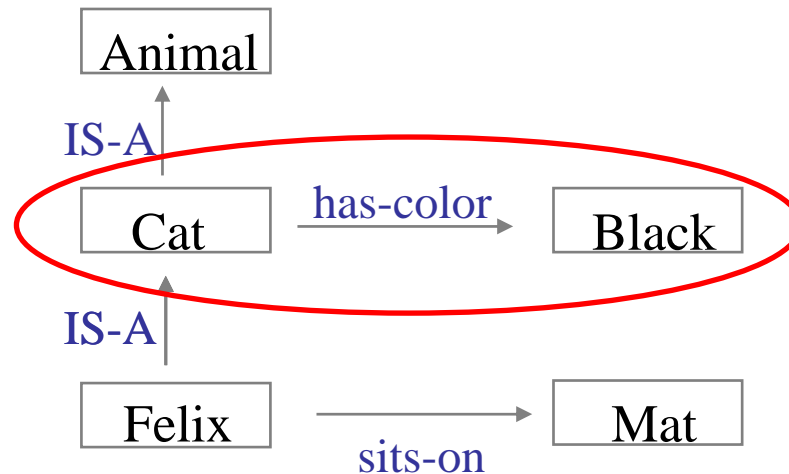
- A family of logic based Knowledge Representation formalisms
 - Descendants of **semantic networks** and **KL-ONE**
 - Describe domain in terms of **concepts** (classes), **roles** (properties, relationships) and **individuals**
 - **Operators** allow for composition of complex concepts
 - **Names** can be given to complex concepts, e.g.:

HappyParent

Parent \sqcap 8hasChild. (Intelligent \sqcap Athletic)

Semantics and Reasoning

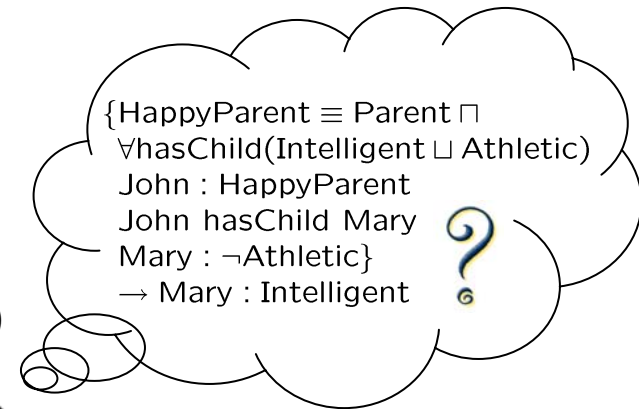
- Distinguished by:
 - **Formal semantics** (typically model theoretic)
 - Decidable fragments of FOL (often contained in C_2)
 - Closely related to Propositional Modal & Dynamic Logics, and to Guarded Fragment



[Quillian, 1967]

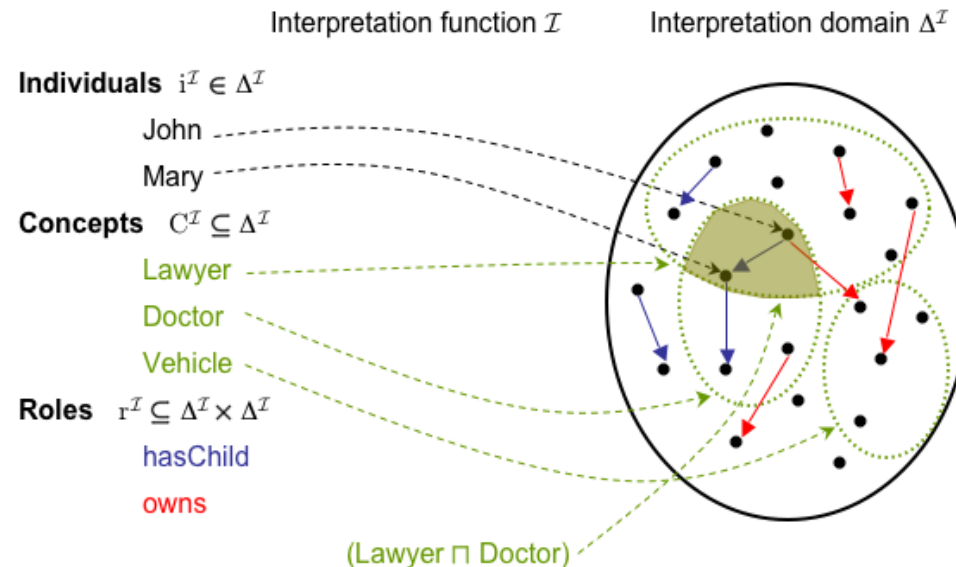
Semantics and Reasoning

- Distinguished by:
 - **Formal semantics** (typically model theoretic)
 - Decidable fragments of FOL (often contained in C_2)
 - Closely related to Propositional Modal & Dynamic Logics, and to Guarded Fragment
 - Provision of **inference services**
 - Decision procedures for key problems (satisfiability, subsumption, etc)
 - Implemented systems (highly optimised)



Why Description Logic?

- OWL exploits results of 15+ years of DL research
 - Well defined (model theoretic) **semantics**



Why Description Logic?

- OWL exploits results of 15+ 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 $\mathcal{L}(v) \rightarrow \mathcal{L}(v) \cup \{C_1, C_2\}$. |
| \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 $\mathcal{L}(v) \rightarrow \mathcal{L}(v) \cup \{E\}$ 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 $C \in \mathcal{L}(v_2)$, 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 $\mathcal{L}(v_2) \rightarrow \mathcal{L}(v_2) \cup \{C\}$. |
| \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 $\text{Trans}(r')$ and $r' \sqsubseteq r$ 3. there is an r' -neighbour v_2 of v_1 with $\forall r'.C \notin \mathcal{L}(v_2)$ then $\mathcal{L}(v_2) \rightarrow \mathcal{L}(v_2) \cup \{\forall r'.C\}$. |
| choose-rule | if 1. $\leq n r.C \in \mathcal{L}(v_1)$, v_1 is not indirectly blocked, and 2. there is an r -neighbour v_2 of v_1 with $\{C, \dot{C}\} \cap \mathcal{L}(v_2) = \emptyset$ then $\mathcal{L}(v_2) \rightarrow \mathcal{L}(v_2) \cup \{E\}$ for some $E \in \{C, \dot{C}\}$. |
| \geq -rule | if 1. $\geq n r.C \in \mathcal{L}(v)$, v is not blocked, and 2. there are not n safe r -neighbours v_1, \dots, v_n of v with $C \in \mathcal{L}(v_i)$ and $v_i \neq v_j$ for $1 \leq i < j \leq n$ |

Why Description Logic?

- OWL exploits results of 15+ years of DL research
 - Well defined (model theoretic) **semantics**
 - **Formal properties** well understood (complexity, decidability)
 - Known **reasoning algorithms**
 - **Implemented systems** (highly optimised)



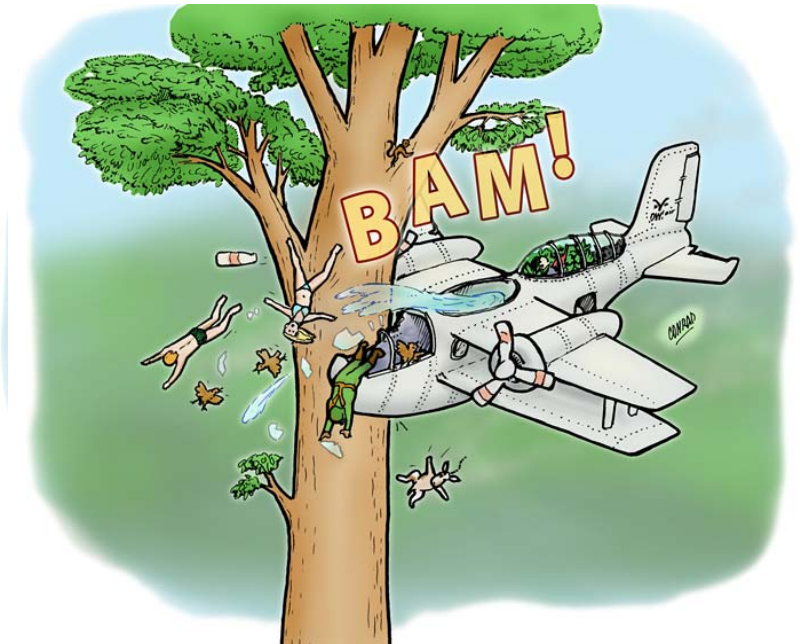
FaCT++



Pellet

Why Description Logic?

- Foundational research was **crucial** to design of OWL
 - Informed Working Group decisions at every stage, e.g.:
 - “Why not extend the language with feature \mathbf{x} , which is clearly harmless?”



- “Adding \mathbf{x} would lead to undecidability - see proof in [...]”

Why the Strange Names?

- Description Logics are a **family** of KR formalisms
 - Mainly distinguished by available operators
- **Available operators** indicated by letters in name, e.g.,
 - S** : basic DL (ALC) plus transitive roles (e.g., ancestor $\in R_+$)
 - H** : role hierarchy (e.g., hasDaughter \vee hasChild)
 - O** : nominals/singleton classes (e.g., {Italy})
 - I** : inverse roles (e.g., isChildOf $\bar{\text{ }}$ hasChild $\bar{\text{ }}$)
 - N** : number restrictions (e.g., >2 hasChild, ≤ 3 hasChild)
- Basic DL + role hierarchy + nominals + inverse + NR = **SHOIN**
 - SHOIN is the basis for W3C's **OWL** Web Ontology Language
- SHOIN is very expressive, but still decidable (just)

Class/Concept Constructors

| Constructor | DL Syntax | Example | FOL Syntax |
|----------------|---------------------------------------|---------------------------|--------------------------------------|
| intersectionOf | $C_1 \sqcap \dots \sqcap C_n$ | Human \sqcap Male | $C_1(x) \wedge \dots \wedge C_n(x)$ |
| unionOf | $C_1 \sqcup \dots \sqcup C_n$ | Doctor \sqcup Lawyer | $C_1(x) \vee \dots \vee C_n(x)$ |
| complementOf | $\neg C$ | \neg Male | $\neg C(x)$ |
| oneOf | $\{x_1\} \sqcup \dots \sqcup \{x_n\}$ | {john} \sqcup {mary} | $x = x_1 \vee \dots \vee x = x_n$ |
| 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 | $\leq_n P$ | ≤ 1 hasChild | $\exists \leq_n y.P(x, y)$ |
| minCardinality | $\geq_n P$ | ≥ 2 hasChild | $\exists \geq_n y.P(x, y)$ |

C is a concept (class); P is a role (property); x is an individual name

Knowledge Base / Ontology

- A **TBox** is a set of “schema” axioms (sentences), e.g.:

```
{Parent  $\forall$  Person  $u$   $>1$ hasChild,  
HappyParent  $\sqsubset$  Parent  $u$   $\exists$ hasChild.(Intelligent  $\sqcup$  Athletic)}
```

- An **ABox** is a set of “data” axioms (ground facts), e.g.:

```
{John:HappyParent,  
John hasChild Mary}
```

- An OWL ontology is just a **SHOIN** KB

OWL RDF/XML Exchange Syntax

E.g., $\text{Parent} \sqcap \text{hasChild}(\text{Intelligent} \sqcup \text{Athletic})$:

```
<owl:Class>
  <owl:intersectionOf rdf:parseType=" collection">
    <owl:Class rdf:about="#Parent"/>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasChild"/>
      <owl:allValuesFrom>
        <owl:unionOf rdf:parseType=" collection">
          <owl:Class rdf:about="#Intelligent"/>
          <owl:Class rdf:about="#Athletic"/>
        </owl:unionOf>
      </owl:allValuesFrom>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
```

Why Ontology Reasoning?

- Given key role of ontologies in many applications, it is essential to provide **tools** and **services** to help users:
 - Design and maintain high quality ontologies, e.g.:
 - Meaningful** — all named classes can have instances

The screenshot shows a software interface for ontology reasoning. At the top, there are two buttons: "Concise Format" and "Abstract Syntax", followed by a play button. Below these is a label "OWL-Class: [mad+cow](#)".

The main window is titled "Explanation" and contains the following text:

Axioms causing the inference
mad+cow = owl:Nothing:

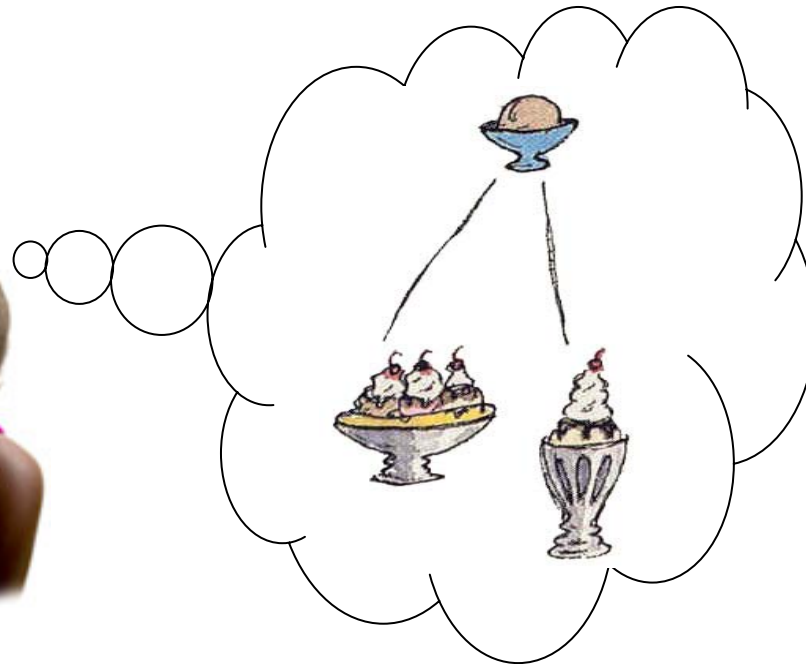
- 1) ([mad+cow](#) = (([∃eats . \(\(\[∃part+of . sheep\\) ∩ \\[brain\\]\\(#\\)\\)\]\(#\)\) ∩ \[cow\]\(#\)\)\)](#)
- 2) |_([sheep](#) ⊆ [animal](#))
- 3) |_([cow](#) ⊆ [vegetarian](#))
- 4) |_([vegetarian](#) = ([animal](#) ∩ ([∀eats . \(¬ \[animal\]\(#\)\)](#)) ∩ ([∀eats . \(¬ \(\[∃part+of . animal\]\(#\)\)\)\)\)\)\)](#)

At the bottom of the window, there is a checkbox labeled "Strike out irrelevant parts of axioms" which is currently unchecked.

Below the explanation window, there is a yellow highlighted area containing the text "[owl:Nothing](#) ([Why?](#))".

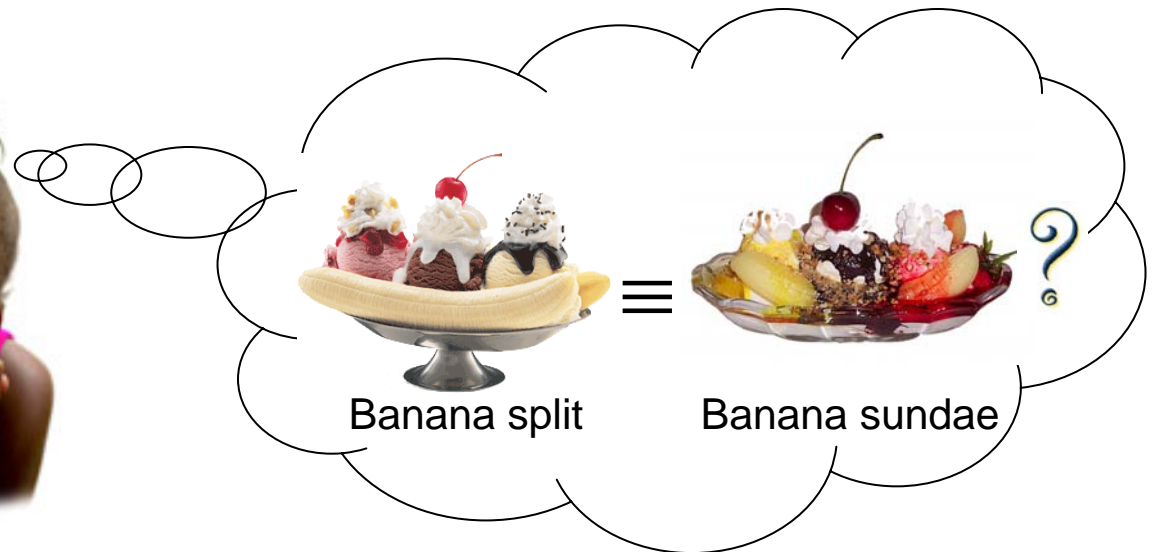
Why Ontology Reasoning?

- Given key role of ontologies in many applications, it is essential to provide **tools** and **services** to help users:
 - Design and maintain high quality ontologies, e.g.:
 - **Meaningful** — all named classes can have instances
 - **Correct** — captures intuitions of domain experts



Why Ontology Reasoning?

- Given key role of ontologies in many applications, it is essential to provide **tools** and **services** to help users:
 - Design and maintain high quality ontologies, e.g.:
 - **Meaningful** — all named classes can have instances
 - **Correct** — captures intuitions of domain experts
 - **Minimally redundant** — no unintended synonyms



Why Ontology Reasoning?

- Given key role of ontologies in many applications, it is essential to provide **tools** and **services** to help users:
 - Design and maintain high quality ontologies, e.g.:
 - **Meaningful** — all named classes can have instances
 - **Correct** — captures intuitions of domain experts
 - **Minimally redundant** — no unintended synonyms
 - Answer **queries** over ontology classes and instances, e.g.:
 - Find more general/specific classes
 - Retrieve individuals/tuples matching a given query

Research Challenges

Increasing Expressive Power

- **Complex role inclusion axioms** [Horrocks&Sattler, IJCAI-03]
 - E.g., $\text{hasLocation} \pm \text{partOf} \triangleright \text{hasLocation}$
- **Concrete domains/datatypes**, e.g., [Lutz, IJCAI-99; Pan et al, ISWC-03]
 - E.g., value comparison (income > expenditure)
- **Database style keys** [Lutz et al, JAIR 2004]
 - E.g., make + model + chassis-number is a key for Vehicles
- **Rule language extensions**
 - First order extensions (e.g., SWRL) [Horrocks et al, JWS, 2005]
 - Hybrid language extensions, e.g., [Eiter et al, KR-04; Motik et al, ISWC-04]
 - LP/F-Logic/Common Logic [Chen et al, JLP, 1993; de Bruijn et al, WWW-05]

Improving Scalability

- **Optimisation techniques**
 - Improve performance of DL reasoners, e.g., [Sirin et al, KR-06]
- **Reduction to disjunctive Datalog** [Motik et al, KR-04]
 - Transform DL ontology to Datalog^c rules
 - Use LP techniques to deal with large numbers of ground facts
- **Hybrid DL-DB systems** [Horrocks et al, CADE-05]
 - Use DB to store “Abox” (individual) axioms
 - Cache inferences and use DB queries to answer/scope logical queries
- **Polynomial time algorithms** for sub- ALC logics [Baader et al, IJCAI-05]
 - Graph based techniques for subsumption computation

Tools and Infrastructure

- Editors/environments
 - Oiled, Protégé, Swoop, Construct, Ontotrack, ...

The image displays three screenshots of ontology editing environments:

- OntoTrack (left):** Shows a hierarchical class structure. The root class is **Individual**, which branches into **TemporalThing**, **SpatialThing**, **SolidTangible**, and **PartiallyIntangible**. **TemporalThing** has a restriction: "= 1 startingPoint". **SpatialThing** further branches into **SomethingExisting**, **SpatialThing-Localized**, **Agent-Generic**, **Place**, and **PartiallyTangible**. **PartiallyIntangible** branches into **Relation**, **TruthValue**, **Mass**, and **Temperature**. **Intangible** branches into **MyClass**. A **Classes** list on the left includes **Thing**, **Relation**, **TruthValue**, **Mass**, **Temperature**, **man**, **publication**, **quality broadsheet**, **red top**, **sheep**, **tabloid**, **tree**, **truck**, **van**, **van driver**, **vegetarian**, **vehicle**, **white van man**, and **woman**. A **Restrictions** table shows:

| type | property | restriction |
|-----------|----------|-------------|
| has-class | drives | ((has color |
- Swoop v2.2b (top right):** Shows a class hierarchy for **space:DistanceCategory**. The **OWL class** is **space:DistanceCategory**. It lists **Intersection of** (**space:SpatialExtentCategory** (bDelete), **space:hasDirection** (bDelete)) and **Disjoint with** (**space:SizeCategory** (Undo), **space:SpatialExtentCategory** (Undo)). It also lists **Subclass of** (**space:SpatialExtentCategory** (Undo), **Yspace:hasAssociatedQuantity** (bDelete)) and **Superclass of** (**space:DistanceCategory** (Undo)).
- Protégé (bottom right):** Shows a class hierarchy for **Symptom**. The **General** tab shows:

| Property | Value |
|------------------|----------------------------------------------------|
| Name | Symptom |
| Namespace | |
| Disjoints | |
| Equivalents | Hallucination, Anxiety, SensoryDisrupt, WeightLoss |
| Subclass | |
| Datatype Prop... | |
| Associations | Superclass P... |

Tools and Infrastructure

- **Editors/environments**
 - Oiled, Protégé, Swoop, Construct, Ontotrack, ...
- **Reasoning systems**
 - Cerebra, FaCT++, Kaon2, Pellet, Racer, ...



FaCT++



Pellet

Tools and Infrastructure

- **Editors/environments**
 - Oiled, Protégé, Swoop, Construct, Ontotrack, ...
- **Reasoning systems**
 - Cerebra, FaCT++, Kaon2, Pellet, Racer, ...
- **Non-standard inferences**
 - Explanation, matching, least common subsumer, ...

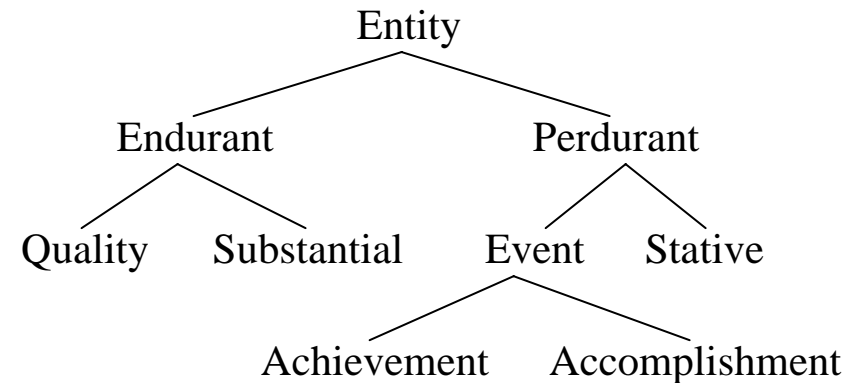


A screenshot of a window titled "Explanation" showing a list of logical axioms. The window has a standard Mac OS-style title bar with three buttons on the left and a scroll bar on the right. The text inside the window is as follows:

```
Axioms causing the inference:  
1) (mad+cow = (( $\exists$ eats x (( $\exists$ part+of y sheep)  $\cap$  brain))  $\cap$  cow))  
2)  $\perp$  (sheep  $\subseteq$  animal)  
3)  $\perp$  (cow  $\subseteq$  vegetarian)  
4)  $\perp$  (vegetarian = (animal  $\cap$  ( $\forall$ eats x ( $\neg$  ( $\exists$ part+of y animal))))  $\cap$  ( $\forall$ eats x ( $\neg$  animal))))
```

Tools and Infrastructure

- **Editors/environments**
 - Oiled, Protégé, Swoop, Construct, Ontotrack, ...
- **Reasoning systems**
 - Cerebra, FaCT++, Kaon2, Pellet, Racer, ...
- **Non-standard inferences**
 - Explanation, matching, least common subsumer, ...
- **Design methodologies**
 - Foundational ontologies, modularisation, etc.



Summary

- **Semantic Web** aims to make web content more accessible to automated processes
 - Adds semantic annotations to web resources
- **Ontologies** provide vocabulary for annotations
 - Terms have well defined meaning
- **OWL** ontology language based on (description) logic
 - Exploits results of basic research on complexity, reasoning, etc.
- Many **research challenges** remain
 - Including expressive power, scalability and tools

Acknowledgements

Thanks to my many friends in the DL and Semantic Web communities, in particular:

- Alan Rector
- Franz Baader
- Uli Sattler



Resources

- FaCT++ system (open source)
 - <http://owl.man.ac.uk/factplusplus/>
- Protégé
 - <http://protege.stanford.edu/plugins/owl/>
- W3C Web-Ontology (WebOnt) working group (OWL)
 - <http://www.w3.org/2001/sw/WebOnt/>
- DL Handbook, Cambridge University Press
 - <http://books.cambridge.org/0521781760.htm>



Thank you for listening

Any questions?

DL & KR, Windermere, 30th May — 5th June