MANCHESTER 1824

The University of Manchester

Description Logic Reasoning

Ian Horrocks <horrocks@cs.man.ac.uk>
University of Manchester
Manchester, UK

Talk Outline

- Introduction to Description Logics
- Ontologies
- Ontology Reasoning
  - Why do we want it?
  - How do we do it?
- Tableaux Algorithms for Description Logic Reasoning
- Research Challenges
- Summary

MANCHESTER
1824

The University
of Manchester

Introduction to Description Logics

MANCHESTER 1824

The University of Manchester

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

DL Basics

- Concept names are equivalent to unary predicates
  - In general, concepts equiv to formulae with one free variable
- Role names are equivalent to binary predicates
  - In general, roles equiv to formulae with two free variables
- Individual names are equivalent to constants
- Operators restricted so that:
  - Language is decidable and, if possible, of low complexity
  - No need for explicit use of variables
    - Restricted form of ∃ and ∀ (direct correspondence with ◊ and [])
  - Features such as counting can be succinctly expressed

DL System Architecture

Knowledge Base

Tbox (schema)

Man ≡ Human ⊓ Male

Happy-Father ≡ Man ⊓ ∃ has-child Female ⊓ …

Abox (data)

John : Happy-Father

⟨John, Mary⟩ : has-child

John: ⩽ 1 has-child

Inference System

Interface

The DL Family

- Given DL defined by set of concept and role forming operators
- Smallest propositionally closed DL is $\mathcal{ALC}$ (equiv modal $K_{(m)}$)
  - Concepts constructed using $\sqcap$, $\sqcup$, $\neg$, $\exists$ and $\forall$
- $\mathcal{S}$ often used for $\mathcal{ALC}$ with transitive roles (R_+)
- Additional letters indicate other extension, e.g.:
  - $\mathcal{H}$ for role inclusion axioms (role hierarchy)
  - $\mathcal{O}$ for nominals (singleton classes, written $\{x\}$)
  - $\mathcal{I}$ for inverse roles
  - $\mathcal{N}$ for number restrictions (of form $\leqslant nR$, $\geqslant nR$)
  - $\mathcal{Q}$ for qualified number restrictions (of form $\leqslant nR.C$, $\geqslant nR.C$)
- E.g., $\mathcal{ALC}$ + R_+ + role hierarchy + inverse roles + QNR = $\mathcal{SHIQ}$
- Have been extended in many directions
  - Concrete domains, fixpoints, epistemic, n-ary, fuzzy, …

DL Semantics

- Semantics defined by interpretations
- An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where
  - $\Delta^{\mathcal{I}}$ is the domain (a non-empty set)
  - $\cdot^{\mathcal{I}}$ is an interpretation function that maps:
    - Concept (class) name $\mathrm{A} \rightarrow$ subset $\mathrm{A}^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$
    - Role (property) name $\mathrm{R} \rightarrow$ binary relation $\mathrm{R}^{\mathcal{I}}$ over $\Delta^{\mathcal{I}}$
    - Individual name $\mathrm{i} \rightarrow \mathrm{i}^{\mathcal{I}}$ element of $\Delta^{\mathcal{I}}$

DL Semantics (cont.)

- Interpretation function $\cdot^{\mathcal{I}}$ extends to concept (and role) expressions in the obvious way, e.g.:

$$
\begin{array}{rcl}
(C \sqcap D)^{\mathcal{I}} & = & C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} & = & C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\neg C)^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
\{x\}^{\mathcal{I}} & = & \{x^{\mathcal{I}}\} \\
(\exists R.C)^{\mathcal{I}} & = & \{x \mid \exists y. \langle x, y \rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \\
(\forall R.C)^{\mathcal{I}} & = & \{x \mid \forall y. (x, y) \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\} \\
(\leqslant n R)^{\mathcal{I}} & = & \{x \mid \#\{y \mid \langle x, y \rangle \in R^{\mathcal{I}}\} \leqslant n\} \\
(\geqslant n R)^{\mathcal{I}} & = & \{x \mid \#\{y \mid \langle x, y \rangle \in R^{\mathcal{I}}\} \geqslant n\} \\
(R^{-})^{\mathcal{I}} & = & \{(x, y) \mid (y, x) \in R^{\mathcal{I}}\}
\end{array}
$$

DL Knowledge Base

- A DL Knowledge base $\mathcal{K}$ is a pair $\langle\mathcal{T},\mathcal{A}\rangle$ where
  - $\mathcal{T}$ is a set of "terminological" axioms (the Tbox)
  - $\mathcal{A}$ is a set of "assertional" axioms (the Abox)
- Tbox axioms are of the form:

  $\mathrm{C} \sqsubseteq \mathrm{D}$, $\mathrm{C} \equiv \mathrm{D}$, $\mathrm{R} \sqsubseteq \mathrm{S}$, $\mathrm{R} \equiv \mathrm{S}$ and $\mathrm{R}^+ \sqsubseteq \mathrm{R}$

  where C, D concepts, R, S roles, and R^+ set of transitive roles
- Abox axioms are of the form:

  $\mathrm{x{:}D}$, $\langle \mathrm{x,y}\rangle{:}\mathrm{R}$

  where x,y are individual names, D a concept and R a role

Knowledge Base Semantics

- An interpretation $\mathcal{I}$ satisfies (models) a Tbox axiom A ($\mathcal{I} \models \mathrm{A}$):
  - $\mathcal{I} \models \mathrm{C} \sqsubseteq \mathrm{D}$ iff $\mathrm{C}^\mathcal{I} \subseteq \mathrm{D}^\mathcal{I}$ — $\mathcal{I} \models \mathrm{C} \equiv \mathrm{D}$ iff $\mathrm{C}^\mathcal{I} = \mathrm{D}^\mathcal{I}$
  - $\mathcal{I} \models \mathrm{R} \sqsubseteq \mathrm{S}$ iff $\mathrm{R}^\mathcal{I} \subseteq \mathrm{S}^\mathcal{I}$ — $\mathcal{I} \models \mathrm{R} \equiv \mathrm{S}$ iff $\mathrm{R}^\mathcal{I} = \mathrm{S}^\mathcal{I}$
  - $\mathcal{I} \models \mathrm{R}^+ \sqsubseteq \mathrm{R}$ iff $(\mathrm{R}^\mathcal{I})^+ \subseteq \mathrm{R}^\mathcal{I}$
- $\mathcal{I}$ satisfies a Tbox $\mathcal{T}$ ($\mathcal{I} \models \mathcal{T}$) iff $\mathcal{I}$ satisfies every axiom A in $\mathcal{T}$
- An interpretation $\mathcal{I}$ satisfies (models) an Abox axiom A ($\mathcal{I} \models \mathrm{A}$):
  - $\mathcal{I} \models \mathrm{x{:}D}$ iff $\mathrm{x}^\mathcal{I} \in \mathrm{D}^\mathcal{I}$ — $\mathcal{I} \models \langle \mathrm{x,y} \rangle\mathrm{:R}$ iff $(\mathrm{x}^\mathcal{I},\mathrm{y}^\mathcal{I}) \in \mathrm{R}^\mathcal{I}$
- $\mathcal{I}$ satisfies an Abox $\mathcal{A}$ ($\mathcal{I} \models \mathcal{A}$) iff $\mathcal{I}$ satisfies every axiom A in $\mathcal{A}$
- $\mathcal{I}$ satisfies an KB $\mathcal{K}$ ($\mathcal{I} \models \mathcal{K}$) iff $\mathcal{I}$ satisfies both $\mathcal{T}$ and $\mathcal{A}$

Short History of Description Logics

Phase 1:

- Incomplete systems (Back, Classic, Loom, . . .)
- Based on structural algorithms

Phase 2:

- Development of tableau algorithms and complexity results
- Tableau-based systems for Pspace logics (e.g., Kris, Crack)
- Investigation of optimisation techniques

Phase 3:

- Tableau algorithms for very expressive DLs
- Highly optimised tableau systems for ExpTime logics (e.g., FaCT, DLP, Racer)
- Relationship to modal logic and decidable fragments of FOL

Recent Developments

Phase 4:

- Mainstream applications and tools
  - Databases
    - Consistency of conceptual schemata (EER, UML etc.)
    - Schema integration
    - Query subsumption (w.r.t. a conceptual schema)
  - Ontologies, e-Science and Semantic Web/Grid
    - Ontology engineering (schema design, maintenance, integration)
    - Reasoning with ontology-based annotations (data)
- Mature implementations
  - Research implementations
    - FaCT, FaCT++, Racer, Pellet, …
  - Commercial implementations
    - Cerebra system from Network Inference (and now Racer)

Ontologies

Ontology: Origins and History

a philosophical discipline—a branch of philosophy that deals with the nature and the organisation of reality

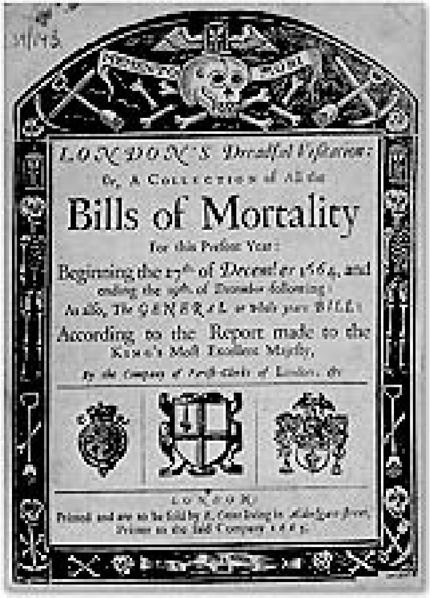
- Science of Being (Aristotle, Metaphysics, IV, 1)
- Tries to answer the questions:
  - *What characterizes being?*
  - *Eventually, what is being?*
- How should things be classified?

Classification: An Old Problem

Extract from *Bills of Mortality*, published weekly from 1664-1830s

The Diseases and Casualties this Week:

Aged	54	…	
Apoplectic	1	Suddenly	1
….		Surfeit	87
Fall down stairs	1	Teeth	113
Gangrene	1	…	
Grief	1	Ulcer	2
Griping in the Guts	74	Vomiting	7
…		Winde	8
Plague	3880	Worms	18

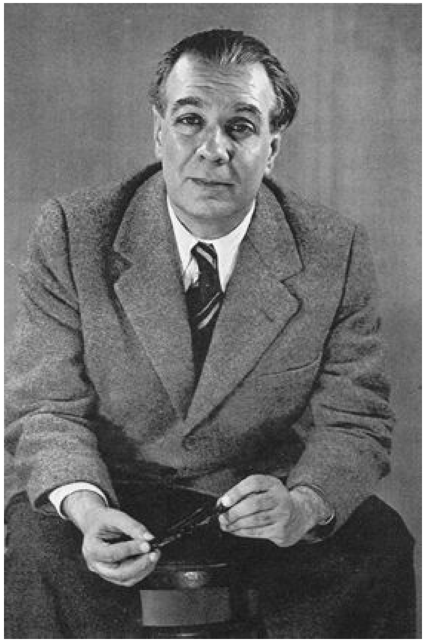


Classification: An Old Problem

Attributed to “a certain Chinese encyclopaedia entitled *Celestial Empire of benevolent Knowledge*”. Jorge Luis Borges: *The Analytical Language of John Wilkins*

On those remote pages it is written that animals are divided into:

a. those that belong to the Emperor
b. embalmed ones
c. those that are trained
d. suckling pigs
e. mermaids
f. fabulous ones
g. stray dogs
h. those that are included in this classification
i. those that tremble as if they were mad
j. innumerable ones
k. those drawn with a very fine camel's hair brush
l. others
m. those that have just broken a flower vase
n. those that from a long way off look like flies

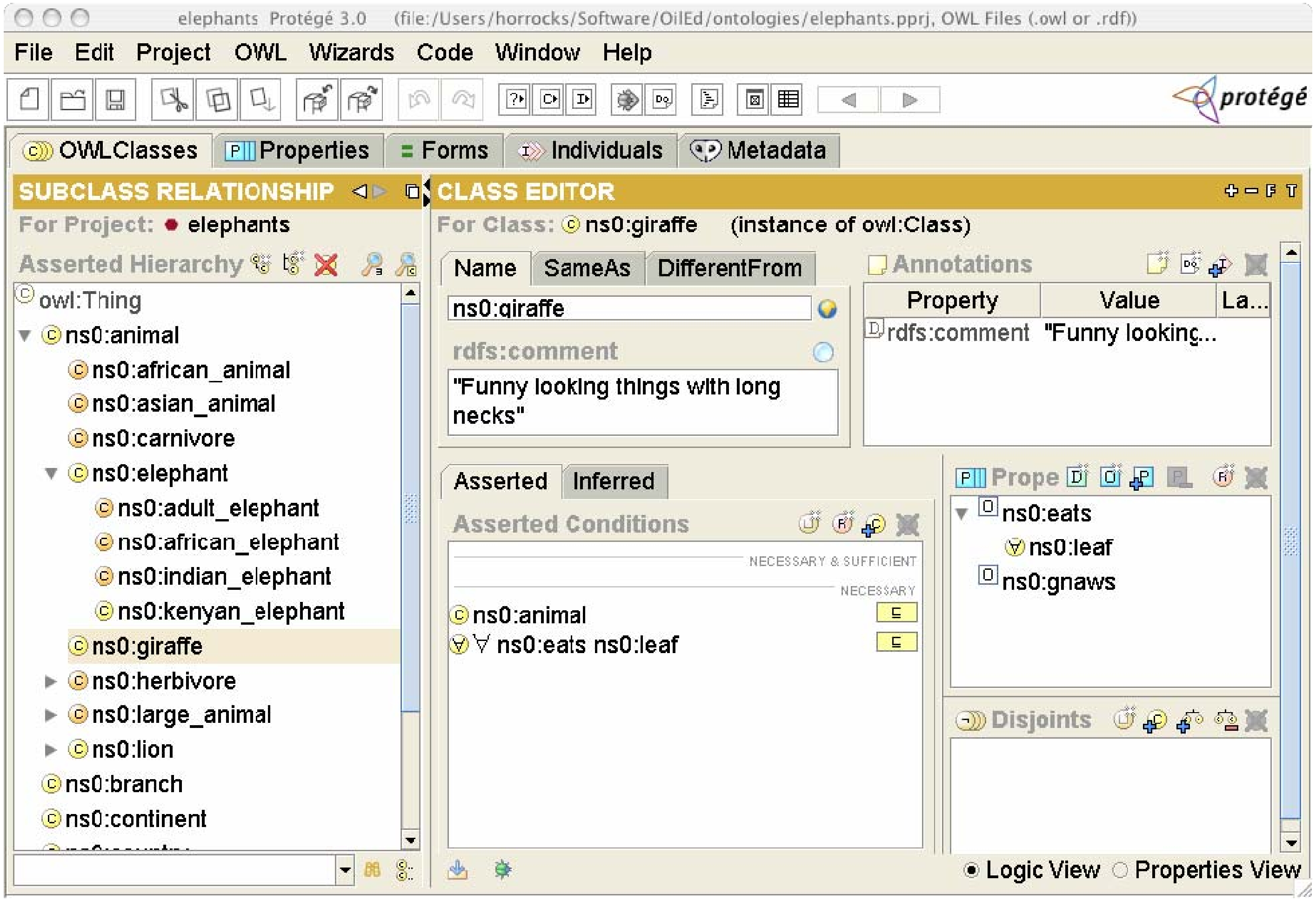


Ontology in Computer Science

- 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.
    - almost always includes how concepts should be classified
  - Constraints capturing additional knowledge about the domain

- Ideally, an ontology should:
  - Capture a shared understanding of a domain of interest
  - Provide a formal and machine manipulable model of the domain

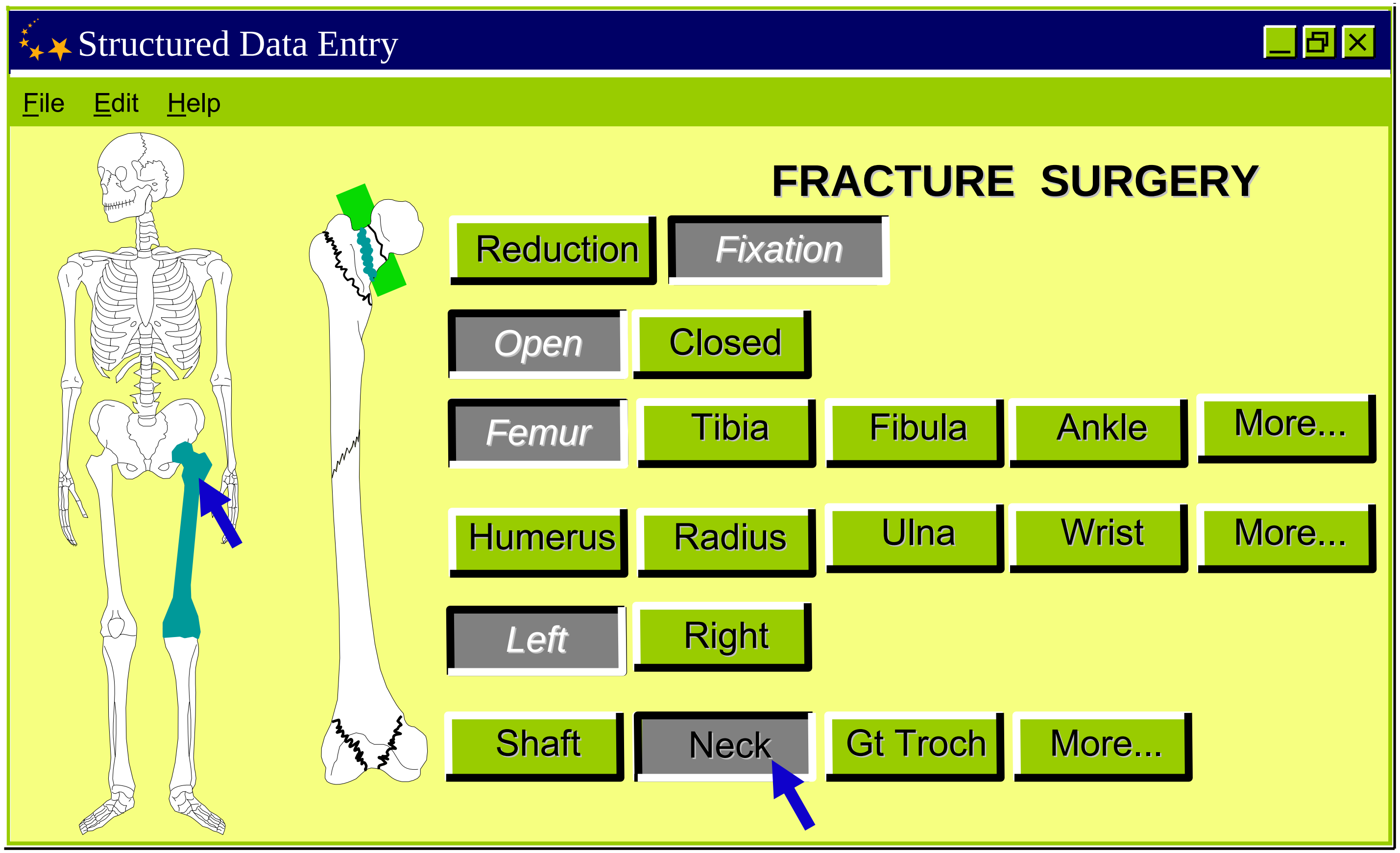
Example Ontology (Protégé)



Where are ontologies used?

- e-Science, e.g., Bioinformatics
  - The Gene Ontology
  - The Protein Ontology (MGED)
  - "in silico" investigations relating theory and data
- Medicine
  - Terminologies
- Databases
  - Integration
  - Query answering
- User interfaces
- Linguistics
- The Semantic Web

Ontology Driven User Interface



- Fixation of open fracture of neck of left femur

Scientific American, May 2001:

Beware of the Hype

MANCHESTER 1824

The University of Manchester

Ontology Reasoning:

Why do We Want It?

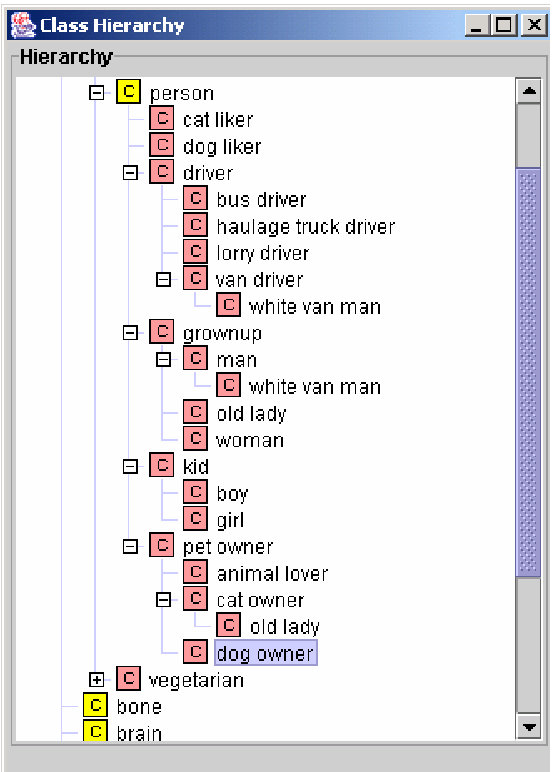
Why Ontology Reasoning?

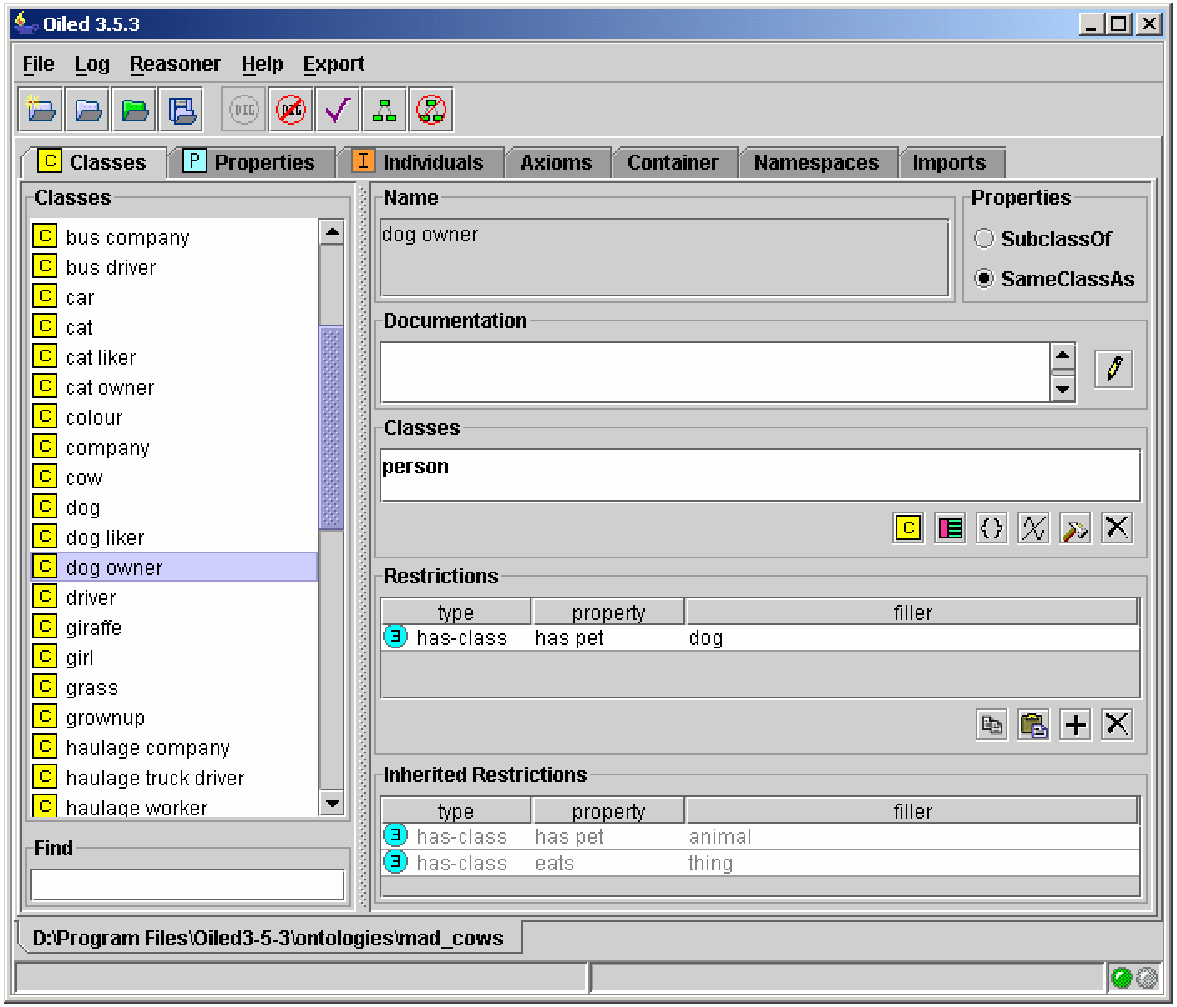
- Given key role of ontologies in many applications, it is essential to provide tools and services to help users:
  - Design and maintain high quality ontologies, e.g.:
    - Meaningful — all named classes can have instances
    - Correct — captured intuitions of domain experts
    - Minimally redundant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

Why Decidable Reasoning?

- OWL is an W3C standard DL based ontology language
  - OWL constructors/axioms restricted so reasoning is decidable
- Consistent with Semantic Web's layered architecture
  - XML provides syntax transport layer
  - RDF(S) provides basic relational language and simple ontological primitives
  - OWL provides powerful but still decidable ontology language
  - Further layers (e.g. SWRL) will extend OWL
    - Will almost certainly be undecidable
- W3C requirement for “implementation experience”
  - “Practical” decision procedures
  - Several implemented systems
  - Evidence of empirical tractability

System Demonstration (OilEd)





Ontology Reasoning:
How do we do it?

Use a (Description) Logic

- OWL DL based on $\mathcal{SHIQ}$ Description Logic
  - In fact it is equivalent to $\mathcal{SHOIN}(\mathbf{D}_n)$ DL
- OWL DL Benefits from many years of DL research
  - Well defined semantics
  - Formal properties well understood (complexity, decidability)
  - Known reasoning algorithms
  - Implemented systems (highly optimised)
- In fact there are three "species" of OWL (!)
  - OWL full is union of OWL syntax and RDF
  - OWL DL restricted to First Order fragment ($\approx$ DAML+OIL)
  - OWL Lite is "simpler" subset of OWL DL (equiv to $\mathcal{SHIF}(\mathbf{D}_n)$)

Class/Concept Constructors

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	$\neg C$	¬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	$\forall P.C$	∀hasChild.Doctor	$\forall y.P(x,y) \rightarrow C(y)$
someValuesFrom	$\exists P.C$	∃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	$\exists^{\geqslant n} y.P(x,y)$

- C is a concept (class); P is a role (property); x is an individual name
- XMLS datatypes as well as classes in $\forall \mathrm{P.C}$ and $\exists \mathrm{P.C}$
  - Restricted form of DL concrete domains

RDFS Syntax

E.g., Person ⊓ ∀hasChild.(Doctor ⊔ ∃hasChild.Doctor):

```
<owl:Class>
  <owl:intersectionOf rdf:parseType=" collection">
    <owl:Class rdf:about="#Person"/>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasChild"/>
      <owl:toClass>
        <owl:unionOf rdf:parseType=" collection">
          <owl:Class rdf:about="#Doctor"/>
          <owl:Restriction>
            <owl:onProperty rdf:resource="#hasChild"/>
            <owl:hasClass rdf:resource="#Doctor"/>
          </owl:Restriction>
        </owl:unionOf>
      </owl:toClass>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
```

Ontology / Tbox & Abox Axioms

OWL Syntax	DL Syntax	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human $\sqsubseteq$ Animal $\sqcap$ Biped
equivalentClass	$C_1 \equiv C_2$	Man $\equiv$ Human $\sqcap$ Male
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter $\sqsubseteq$ hasChild
equivalentProperty	$P_1 \equiv P_2$	cost $\equiv$ price
transitiveProperty	$P^+ \sqsubseteq P$	ancestor$^+ \sqsubseteq$ ancestor

OWL Syntax	DL Syntax	Example
type	$a : C$	John : Happy-Father
property	$\langle a, b \rangle : R$	$\langle$John, Mary$\rangle$: has-child

- Obvious FOL equivalences
  - E.g., DL: $C \sqsubseteq D$ FOL: $\forall x.C(x) \rightarrow D(x)$

MANCHESTER
1824

The University
of Manchester

Description Logic Reasoning

DL Reasoning: Basics (I)

- Key reasoning tasks reducible to (un)satisfiability
  - E.g., $C \sqsubseteq D$ iff $C \sqcap \neg D$ is *not* satisfiable
- Tableau algorithms used to test satisfiability (consistency)
- Try to build a tree-like model of the input concept C
- Decompose C syntactically
  - Apply tableau expansion rules
  - Infer constraints on elements of model
- Tableau rules correspond to constructors in logic ($\sqcap$, $\sqcup$ etc)
  - Some rules are nondeterministic (e.g., $\sqcup$, $\leqslant$)
  - In practice, this means search
- Stop when no more rules applicable or clash occurs
  - Clash is an obvious contradiction, e.g., $A(x)$, $\neg A(x)$

DL Reasoning: Basics (II)

- Cycle check (blocking) may be needed for termination
- C satisfiable iff rules can be applied such that a fully expanded clash free tree is constructed:

  Terminating
  - Bounds on out-degree (rule applications per node) and depth (blocking) of tree

  Sound
  - Can construct a tableau, and hence a model, from a fully expanded and clash-free tree

  Complete
  - Can use a model to guide application of non-deterministic rules and so construct a clash-free tree

DL Reasoning: Advanced Techniques

- Satisfiability w.r.t. an Ontology $\mathcal{O}$
  - For each axiom C $\sqsubseteq$ D $\in \mathcal{O}$, add $\neg$C $\sqcup$ D to every node label
- More expressive DLs
  - Basic technique can be extended to deal with
    - Role inclusion axioms (role hierarchy)
    - Number restrictions
    - Inverse roles
    - Concrete domains/datatypes
    - Aboxes
    - etc.
  - Extend expansion rules and use more sophisticated blocking strategy
  - Forest instead of Tree (for Aboxes)
    - Root nodes correspond to individuals in Abox

MANCHESTER
1824

The University of Manchester

DL Reasoning: Optimised Implementations

- Naive implementation can lead to effective non-termination
  - 10 GCIs × 10 nodes → 2^{100} different possible expansions
- Modern systems include MANY optimisations
- Optimised classification (compute partial ordering)
  - Enhanced traversal (exploits information from previous tests)
  - Use structural information to select classification order
- Optimised satisfiability/subsumption testing
  - Normalisation and simplification of concepts
  - Absorption (simplification) of axioms
  - Dependency directed backtracking
  - Caching of satisfiability results and (partial) models
  - Heuristic ordering of propositional and modal expansion
  - …

The University of Manchester

MANCHESTER 1824

Research Challenges:
What next?

MANCHESTER
1824

The University of Manchester

Increased Expressive Power

- OWL not expressive enough for some applications
  - Constructors mainly for classes (unary predicates)
  - No complex datatypes or built in predicates (e.g., arithmetic)
  - No variables
  - No higher arity predicates
- Rules language extension (SWRL) already developed
  - Horn clauses where predicates are OWL classes and properties
  - Resulting language is undecidable
- OWL-FOL also proposed
  - Extends SWRL with explicit quantification

Improved Scalability

- Reasoning is hard (NExpTime-complete for OWL-DL)
- Web ontologies may grow very large
- Good empirical evidence of scalability/tractability for DL systems
  - E.g., 5,000 (complex) classes; 100,000+ (simple) classes
  - But evidence mostly w.r.t. SHF (no inverse)
- Reasoning with individuals
  - Deployment of web ontologies will mean reasoning with (possibly very large numbers of) individuals/tuples
  - Unlikely that standard Abox techniques will be able to cope

Other Reasoning Tasks

- Querying
  - Retrieval and instantiation wont be sufficient
  - Minimum requirement will be DB style query language
  - May also need “what can I say about x?” style of query
- Explanation
  - To support ontology design
  - Justifications and proofs (e.g., of query results)
- “Non-Standard Inferences”, e.g., LCS, matching
  - To support ontology integration
  - To support “bottom up” design of ontologies

Tools and Infrastructure

- Adoption of OWL and realisation of Semantic Web will require development of wide range of tools and infrastructure
  - Not just editors, but complete ontology development environments
  - Annotation tools, including (semi-)automated annotation of existing content
  - Reasoning systems/query engines
  - …

Summary

- DLs are a family of logic based Knowledge Representation formalisms
  - Describe domain in terms of concepts, roles and individuals
- An Ontology is an engineering artefact consisting of:
  - A vocabulary of terms
  - An explicit specification their intended meaning
- Ontologies play a key role in many applications
  - e-Science, Medicine, Databases, Semantic Web, etc.

Summary

- Reasoning is important
  - Essential for design, maintenance and deployment of ontologies
- Reasoning support currently based on DL systems
  - Tableaux decision procedures
  - Highly optimised implementations
- Many challenges remain
  - Including extensions up to an including FOL

Enough work to keep logic based KR community busy for many years to come ☺

Acknowledgements

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

- Alan Rector
- Franz Baader
- Uli Sattler

Resources

- Slides from this talk
  - http://www.cs.man.ac.uk/~horrocks/Slides/iccs05.ppt
- FaCT system (open source)
  - http://www.cs.man.ac.uk/FaCT/
- OilEd (open source)
  - http://oiled.man.ac.uk/
- 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