# OWL

## *an Ontology Language for the Semantic Web*

Ian Horrocks

`horrocks@cs.man.ac.uk`

University of Manchester

Manchester, UK

# Talk Outline

# Talk Outline

## The Semantic Web

# Talk Outline

- The Semantic Web
- Web Ontology Languages

# Talk Outline

- The Semantic Web
- Web Ontology Languages
- OWL

# Talk Outline

- The Semantic Web
- Web Ontology Languages
- OWL
- Reasoning with OWL
    - OilEd Demo

# Talk Outline

- The Semantic Web
- Web Ontology Languages
- OWL
- Reasoning with OWL
  - OilEd Demo
- Research Challenges

# The Semantic Web

# The Semantic Web Vision

# The Semantic Web Vision

- Web made possible through established **standards**
  - **TCP/IP** for transporting bits down a wire
  - **HTTP & HTML** for transporting and rendering hyperlinked text

# The Semantic Web Vision

- Web made possible through established **standards**
  - **TCP/IP** for transporting bits down a wire
  - **HTTP & HTML** for transporting and rendering hyperlinked text
- **Applications** able to exploit this common infrastructure
  - Result is the WWW as we know it

# The Semantic Web Vision

- Web made possible through established **standards**
  - **TCP/IP** for transporting bits down a wire
  - **HTTP & HTML** for transporting and rendering hyperlinked text
- **Applications** able to exploit this common infrastructure
  - Result is the WWW as we know it
- **1st generation** web mostly handwritten HTML pages

# The Semantic Web Vision

- Web made possible through established **standards**
  - **TCP/IP** for transporting bits down a wire
  - **HTTP & HTML** for transporting and rendering hyperlinked text
- **Applications** able to exploit this common infrastructure
  - Result is the WWW as we know it
- **1st generation** web mostly handwritten HTML pages
- **2nd generation** (current) web often machine generated/active

# The Semantic Web Vision

- Web made possible through established **standards**
  - **TCP/IP** for transporting bits down a wire
  - **HTTP & HTML** for transporting and rendering hyperlinked text
- **Applications** able to exploit this common infrastructure
  - Result is the WWW as we know it
- **1st generation** web mostly handwritten HTML pages
- **2nd generation** (current) web often machine generated/active
- Both intended for direct human processing/interaction

# The Semantic Web Vision

- Web made possible through established **standards**
  - **TCP/IP** for transporting bits down a wire
  - **HTTP & HTML** for transporting and rendering hyperlinked text
- **Applications** able to exploit this common infrastructure
  - Result is the WWW as we know it
- **1st generation** web mostly handwritten HTML pages
- **2nd generation** (current) web often machine generated/active
- Both intended for direct human processing/interaction
- In **next generation** web, **resources** should be more accessible to automated processes

# The Semantic Web Vision

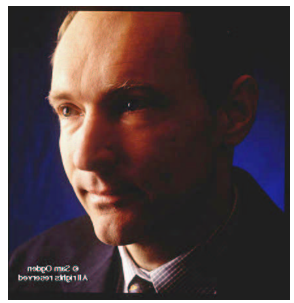
- Web made possible through established **standards**
  - **TCP/IP** for transporting bits down a wire
  - **HTTP & HTML** for transporting and rendering hyperlinked text
- **Applications** able to exploit this common infrastructure
  - Result is the WWW as we know it
- **1st generation** web mostly handwritten HTML pages
- **2nd generation** (current) web often machine generated/active
- Both intended for direct human processing/interaction
- In **next generation** web, **resources** should be more accessible to automated processes
  - To be achieved via **semantic markup**
  - **Metadata** annotations that describe content/function

# The Semantic Web Vision

- Web made possible through established **standards**
  - **TCP/IP** for transporting bits down a wire
  - **HTTP & HTML** for transporting and rendering hyperlinked text
- **Applications** able to exploit this common infrastructure
  - Result is the WWW as we know it
- **1st generation** web mostly handwritten HTML pages
- **2nd generation** (current) web often machine generated/active
- Both intended for direct human processing/interaction
- In **next generation** web, **resources** should be more accessible to automated processes
  - To be achieved via **semantic markup**
  - **Metadata** annotations that describe content/function
- Coincides with Tim Berners-Lee's vision of a **Semantic Web**

# The Semantic Web Vision

- Web made possible through established **standards**
  - **TCP/IP** for transporting bits down a wire
  - **HTTP & HTML** for transporting and rendering hyperlinked text
- **Applications** able to exploit this common infrastructure
  - Result is the WWW as we know it
- **1st generation** web mostly handwritten HTML pages
- **2nd generation** (current) web often machine generated/active
- Both intended for direct human processing/interaction
- In **next generation** web, **resources** should be more accessible to automated processes
  - To be achieved via **semantic markup**
  - **Metadata** annotations that describe content/function
- Coincides with Tim Berners-Lee's vision of a **Semantic Web**



# Ontologies

# Ontologies

☞ Semantic markup must be **meaningful** to automated processes

# Ontologies

- Semantic markup must be **meaningful** to automated processes
- Ontologies will play a key role
  - Source of **precisely defined** terms (vocabulary)
  - Can be **shared** across applications (and humans)

# Ontologies

- Semantic markup must be **meaningful** to automated processes
- Ontologies will play a key role
  - Source of **precisely defined** terms (vocabulary)
  - Can be **shared** across applications (and humans)
- Ontology typically consists of:
  - **Hierarchical** description of important **concepts** in domain
  - Descriptions of **properties** of instances of each concept

# Ontologies

- ☞ Semantic markup must be **meaningful** to automated processes
- ☞ Ontologies will play a key role
  - Source of **precisely defined** terms (vocabulary)
  - Can be **shared** across applications (and humans)
- ☞ Ontology typically consists of:
  - **Hierarchical** description of important **concepts** in domain
  - Descriptions of **properties** of instances of each concept
- ☞ Degree of formality can be quite variable (NL–logic)

# Ontologies

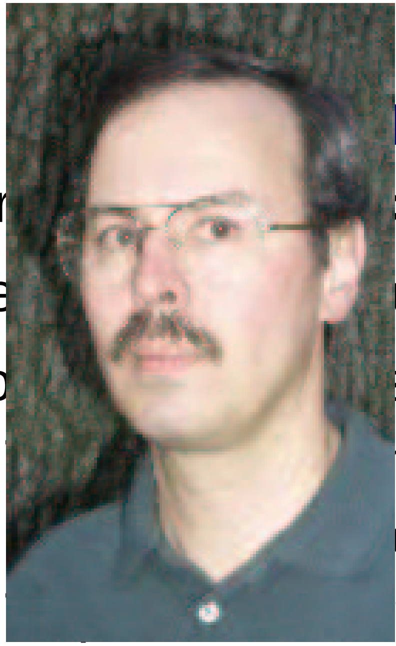
- ☞ Semantic markup must be **meaningful** to automated processes
- ☞ Ontologies will play a key role
  - Source of **precisely defined** terms (vocabulary)
  - Can be **shared** across applications (and humans)
- ☞ Ontology typically consists of:
  - **Hierarchical** description of important **concepts** in domain
  - Descriptions of **properties** of instances of each concept
- ☞ Degree of formality can be quite variable (NL–logic)
- ☞ Increased formality and regularity facilitates machine understanding

# Ontologies

- Semantic markup must be **meaningful** to automated processes
- Ontologies will play a key role
  - Source of **precisely defined** terms (vocabulary)
  - Can be **shared** across applications (and humans)
- Ontology typically consists of:
  - **Hierarchical** description of important **concepts** in domain
  - Descriptions of **properties** of instances of each concept
- Degree of formality can be quite variable (NL–logic)
- Increased formality and regularity facilitates machine understanding
- Ontologies can be used, e.g.:
  - To facilitate agent-agent communication in **e-commerce**
  - In semantic based **search**
  - To provide richer **service descriptions** that can be more flexibly interpreted by intelligent agents

# Ontologies

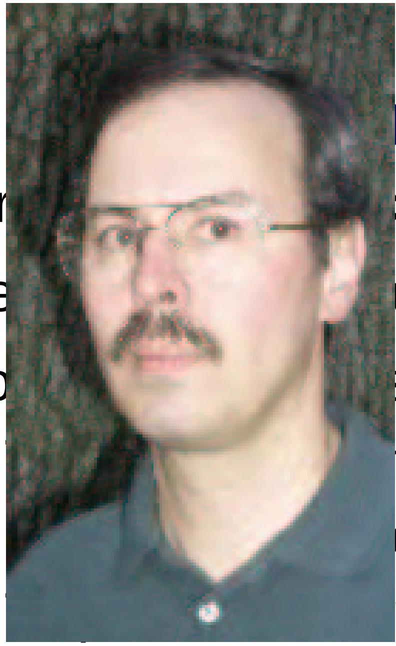
- Semantic markup must be **meaningful** to automated processes
- Ontologies will play a key role
  - Source of **precisely defined** terms (vocabulary)
  - Can be **shared** across applications (and humans)
- Ontology typically consists of:

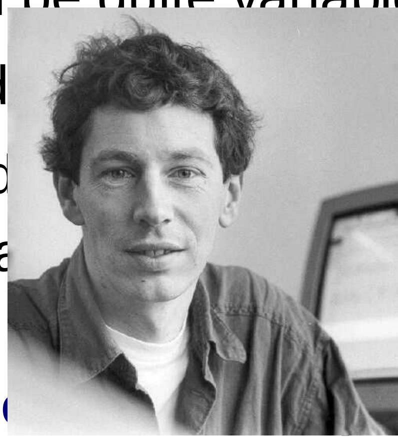


  - scription of important **concepts** in domain
  - **properties** of instances of each concept
- Degr an be quite variable (NL–logic)
- Incre nd regularity facilitates machine understanding
- Onto sed, e.g.:
  - nt-agent communication in **e-commerce**
  - ed **search**
  - r **service descriptions** that can be more flexibly interpreted by intelligent agents

# Ontologies

- Semantic markup must be **meaningful** to automated processes
- Ontologies will play a key role
  - Source of **precisely defined** terms (vocabulary)
  - Can be **shared** across applications (and humans)
- Ontology typically consists of:
  - scription of important **concepts** in domain
  - **properties** of instances of each concept

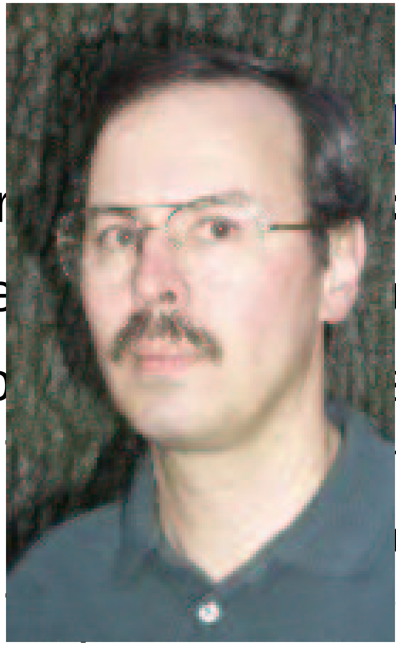


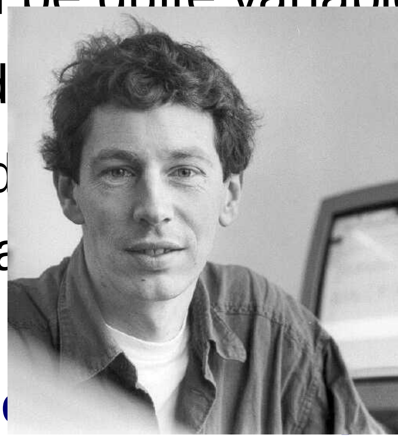


- Degr can be quite variable (NL–logic)
- Incre nd ates machine understanding
- Onto sed
  - nt-a ation in **e-commerce**
  - ed
  - r **s** **ions** that can be more flexibly interpreted by intelligent agents

# Ontologies

- Semantic markup must be **meaningful** to automated processes
- Ontologies will play a key role
  - Source of **precisely defined** terms (vocabulary)
  - Can be **shared** across applications (and humans)
- Ontology typically consists of:





  - scription of important **concepts** in domain
  - **properties** of instanc pt
- Degr can be quite variable (I
- Incre and ate rstanding
- Onto sed
  - nt-a atic **ce**
  - ed
  - r **s** **ion** ore flexibly interpreted by intelligent agents

# Web Ontology Languages

# Web Languages

# Web Languages

☞ Web languages already extended to facilitate **content description**

- XML Schema (XMLS)
- RDF and RDF Schema (RDFS)

# Web Languages

- ☞ Web languages already extended to facilitate **content description**
  - XML Schema (XMLS)
  - RDF and RDF Schema (RDFS)
- ☞ RDFS recognisable as an **ontology language**
  - Classes and properties
  - Range and domain
  - Sub/super-classes (and properties)

# Web Languages

- ☞ Web languages already extended to facilitate **content description**
  - XML Schema (XMLS)
  - RDF and RDF Schema (RDFS)
- ☞ RDFS recognisable as an **ontology language**
  - Classes and properties
  - Range and domain
  - Sub/super-classes (and properties)
- ☞ But RDFS not a suitable foundation for Semantic Web
  - **Too weak** to describe resources in sufficient detail

# Web Languages

- Web languages already extended to facilitate **content description**
  - XML Schema (XMLS)
  - RDF and RDF Schema (RDFS)
- RDFS recognisable as an **ontology language**
  - Classes and properties
  - Range and domain
  - Sub/super-classes (and properties)
- But RDFS not a suitable foundation for Semantic Web
  - **Too weak** to describe resources in sufficient detail
- Requirements for web ontology language:
  - **Compatible** with existing Web standards (XML, RDF, RDFS)
  - **Easy to understand** and use (based on familiar KR idioms)
  - **Formally specified** and of “adequate” expressive power
  - Possible to provide **automated reasoning** support

# OIL, DAML-ONT, DAML+OIL and OWL

# OIL, DAML-ONT, DAML+OIL and OWL

- Two languages developed to satisfy above requirements
  - **OIL**: developed by group of (largely) European researchers
  - **DAML-ONT**: developed in DARPA DAML programme

# OIL, DAML-ONT, DAML+OIL and OWL

- Two languages developed to satisfy above requirements
  - **OIL**: developed by group of (largely) European researchers
  - **DAML-ONT**: developed in DARPA DAML programme
- Efforts merged to produce **DAML+OIL**

# OIL, DAML-ONT, DAML+OIL and OWL

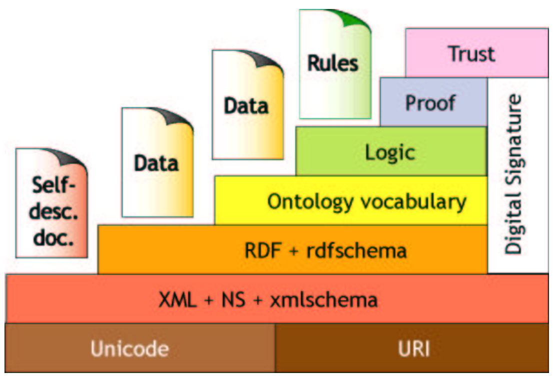
- Two languages developed to satisfy above requirements
  - **OIL**: developed by group of (largely) European researchers
  - **DAML-ONT**: developed in DARPA DAML programme
- Efforts merged to produce **DAML+OIL**
- **Submitted to W3C** as basis for standardisation
  - **WebOnt working group** developing **OWL** language standard

# OIL, DAML-ONT, DAML+OIL and OWL

- Two languages developed to satisfy above requirements
  - **OIL**: developed by group of (largely) European researchers
  - **DAML-ONT**: developed in DARPA DAML programme
- Efforts merged to produce **DAML+OIL**
- **Submitted to W3C** as basis for standardisation
  - **WebOnt working group** developing **OWL** language standard
- OWL **“layered”** on top of RDFS
  - RDFS based **syntax** and ontological primitives (subclass etc.)
  - Adds **much** richer set of primitives (transitivity, cardinality, . . . )

# OIL, DAML-ONT, DAML+OIL and OWL

- Two languages developed to satisfy above requirements
  - **OIL**: developed by group of (largely) European researchers
  - **DAML-ONT**: developed in DARPA DAML programme
- Efforts merged to produce **DAML+OIL**
- **Submitted to W3C** as basis for standardisation
  - **WebOnt working group** developing **OWL** language standard
- OWL **“layered”** on top of RDFS
  - RDFS based **syntax** and ontological primitives (subclass etc.)
  - Adds **much** richer set of primitives (transitivity, cardinality, . . . )



# OIL, DAML-ONT, DAML+OIL and OWL

- Two languages developed to satisfy above requirements
  - **OIL**: developed by group of (largely) European researchers
  - **DAML-ONT**: developed in DARPA DAML programme
- Efforts merged to produce **DAML+OIL**
- **Submitted to W3C** as basis for standardisation
  - **WebOnt working group** developing **OWL** language standard
- OWL **“layered”** on top of RDFS
  - RDFS based **syntax** and ontological primitives (subclass etc.)
  - Adds **much** richer set of primitives (transitivity, cardinality, . . . )
- Describes **structure** of domain in terms of Classes and Properties
  - Ontology is set of **axioms** describing classes and properties
  - E.g., Person **subclass of** Animal whose parents are all Persons

# OIL, DAML-ONT, DAML+OIL and OWL

- Two languages developed to satisfy above requirements
  - **OIL**: developed by group of (largely) European researchers
  - **DAML-ONT**: developed in DARPA DAML programme
- Efforts merged to produce **DAML+OIL**
- **Submitted to W3C** as basis for standardisation
  - **WebOnt working group** developing **OWL** language standard
- OWL **"layered"** on top of RDFS
  - RDFS based **syntax** and ontological primitives (subclass etc.)
  - Adds **much** richer set of primitives (transitivity, cardinality, . . . )
- Describes **structure** of domain in terms of Classes and Properties
  - Ontology is set of **axioms** describing classes and properties
  - E.g., Person **subclass of** Animal whose parents are all Persons
- Uses RDF for class/property membership assertions (ground facts)
  - E.g., john **instance of** Person; $\langle \text{john}, \text{mary} \rangle$ instance of parent

# OWL Language

# Foundations

# Foundations

☞ Three species of OWL

- OWL full is union of OWL syntax and RDF
- OWL DL restricted to FOL fragment ($\approx$DAML+OIL)
- OWL Lite is “easier to implement” subset of OWL DL

# Foundations

- Three species of OWL
  - OWL full is union of OWL syntax and RDF
  - OWL DL restricted to FOL fragment ($\approx$DAML+OIL)
  - OWL Lite is “easier to implement” subset of OWL DL
- Semantic layering
  - OWL DL $\equiv$ OWL full **within DL fragment**
  - DL semantics officially **definitive**

# Foundations

- ☞ Three species of OWL
  - OWL full is union of OWL syntax and RDF
  - OWL DL restricted to FOL fragment ($\approx$DAML+OIL)
  - OWL Lite is “easier to implement” subset of OWL DL
- ☞ Semantic layering
  - OWL DL $\equiv$ OWL full **within DL fragment**
  - DL semantics officially **definitive**
- ☞ OWL DL based on $\mathcal{SHIQ}$ **Description Logic**

# Foundations

- Three species of OWL
  - OWL full is union of OWL syntax and RDF
  - OWL DL restricted to FOL fragment (≈DAML+OIL)
  - OWL Lite is "easier to implement" subset of OWL DL
- Semantic layering
  - OWL DL $\equiv$ OWL full **within DL fragment**
  - DL semantics officially **definitive**
- OWL DL based on $\mathcal{SHIQ}$ **Description Logic**
- Benefits from many years of DL research
  - Well defined **semantics**
  - **Formal properties** well understood (complexity, decidability)
  - Known **reasoning algorithms**
  - **Implemented systems** (highly optimised)

# OWL Class Constructors

| Constructor | DL Syntax | Example | (Modal Syntax) |
|---|---|---|---|
| intersectionOf | $C_1 \sqcap \ldots \sqcap C_n$ | Human $\sqcap$ Male | $C_1 \wedge \ldots \wedge C_n$ |
| unionOf | $C_1 \sqcup \ldots \sqcup C_n$ | Doctor $\sqcup$ Lawyer | $C_1 \vee \ldots \vee C_n$ |
| complementOf | $\neg C$ | $\neg$Male | $\neg C$ |
| oneOf | $\{x_1 \ldots x_n\}$ | {john, mary} | $x_1 \vee \ldots \vee x_n$ |
| allValuesFrom | $\forall P.C$ | $\forall$hasChild.Doctor | $[P]C$ |
| someValuesFrom | $\exists P.C$ | $\exists$hasChild.Lawyer | $\langle P \rangle C$ |
| maxCardinality | $\leqslant n P$ | $\leqslant 1$hasChild | $[P]_{n+1}$ |
| minCardinality | $\geqslant n P$ | $\geqslant 2$hasChild | $\langle P \rangle_n$ |

# OWL Class Constructors

| Constructor | DL Syntax | Example | (Modal Syntax) |
|---|---|---|---|
| intersectionOf | $C_1 \sqcap \ldots \sqcap C_n$ | Human $\sqcap$ Male | $C_1 \wedge \ldots \wedge C_n$ |
| unionOf | $C_1 \sqcup \ldots \sqcup C_n$ | Doctor $\sqcup$ Lawyer | $C_1 \vee \ldots \vee C_n$ |
| complementOf | $\neg C$ | $\neg$Male | $\neg C$ |
| oneOf | $\{x_1 \ldots x_n\}$ | {john, mary} | $x_1 \vee \ldots \vee x_n$ |
| allValuesFrom | $\forall P.C$ | $\forall$hasChild.Doctor | $[P]C$ |
| someValuesFrom | $\exists P.C$ | $\exists$hasChild.Lawyer | $\langle P \rangle C$ |
| maxCardinality | $\leqslant n P$ | $\leqslant$1hasChild | $[P]_{n+1}$ |
| minCardinality | $\geqslant n P$ | $\geqslant$2hasChild | $\langle P \rangle_n$ |

☞ XMLS **datatypes** as well as classes in $\forall P.C$ and $\exists P.C$

- E.g., $\exists$hasAge.nonNegativeInteger

# OWL Class Constructors

| Constructor | DL Syntax | Example | (Modal Syntax) |
|---|---|---|---|
| intersectionOf | $C_1 \sqcap \ldots \sqcap C_n$ | Human $\sqcap$ Male | $C_1 \wedge \ldots \wedge C_n$ |
| unionOf | $C_1 \sqcup \ldots \sqcup C_n$ | Doctor $\sqcup$ Lawyer | $C_1 \vee \ldots \vee C_n$ |
| complementOf | $\neg C$ | $\neg$Male | $\neg C$ |
| oneOf | $\{x_1 \ldots x_n\}$ | {john, mary} | $x_1 \vee \ldots \vee x_n$ |
| allValuesFrom | $\forall P.C$ | $\forall$hasChild.Doctor | $[P]C$ |
| someValuesFrom | $\exists P.C$ | $\exists$hasChild.Lawyer | $\langle P \rangle C$ |
| maxCardinality | $\leqslant n P$ | $\leqslant$1hasChild | $[P]_{n+1}$ |
| minCardinality | $\geqslant n P$ | $\geqslant$2hasChild | $\langle P \rangle_n$ |

- XMLS **datatypes** as well as classes in $\forall P.C$ and $\exists P.C$
  - E.g., $\exists$hasAge.nonNegativeInteger
- Arbitrarily complex **nesting** of constructors
  - E.g., Person $\sqcap$ $\forall$hasChild.(Doctor $\sqcup$ $\exists$hasChild.Doctor)

# RDFS Syntax

```
<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>
```

# OWL DL Semantics

# OWL DL Semantics

☞ Semantics defined by **interpretations**: $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$

- concepts $\longrightarrow$ subsets of $\Delta^{\mathcal{I}}$
- roles $\longrightarrow$ binary relations over $\Delta^{\mathcal{I}}$ (subsets of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$)
- individuals $\longrightarrow$ elements of $\Delta^{\mathcal{I}}$

# OWL DL Semantics

☞ Semantics defined by **interpretations**: $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$

- concepts $\longrightarrow$ subsets of $\Delta^{\mathcal{I}}$
- roles $\longrightarrow$ binary relations over $\Delta^{\mathcal{I}}$ (subsets of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$)
- individuals $\longrightarrow$ elements of $\Delta^{\mathcal{I}}$

☞ Interpretation function $\cdot^{\mathcal{I}}$ **extended** to concept expressions

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

# OWL Axioms

| Axiom | 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 |
| disjointWith | $C_1 \sqsubseteq \neg C_2$ | Male $\sqsubseteq$ $\neg$Female |
| sameIndividualAs | $\{x_1\} \equiv \{x_2\}$ | {President_Bush} $\equiv$ {G_W_Bush} |
| differentFrom | $\{x_1\} \sqsubseteq \neg\{x_2\}$ | {john} $\sqsubseteq$ $\neg${peter} |
| subPropertyOf | $P_1 \sqsubseteq P_2$ | hasDaughter $\sqsubseteq$ hasChild |
| equivalentProperty | $P_1 \equiv P_2$ | cost $\equiv$ price |
| inverseOf | $P_1 \equiv P_2^-$ | hasChild $\equiv$ hasParent$^-$ |
| transitiveProperty | $P^+ \sqsubseteq P$ | ancestor$^+$ $\sqsubseteq$ ancestor |
| functionalProperty | $\top \sqsubseteq \leqslant 1 P$ | $\top \sqsubseteq \leqslant$1hasMother |
| inverseFunctionalProperty | $\top \sqsubseteq \leqslant 1 P^-$ | $\top \sqsubseteq \leqslant$1hasSSN$^-$ |

# OWL Axioms

| Axiom | 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 |
| disjointWith | $C_1 \sqsubseteq \neg C_2$ | Male $\sqsubseteq \neg$Female |
| sameIndividualAs | $\{x_1\} \equiv \{x_2\}$ | {President_Bush} $\equiv$ {G_W_Bush} |
| differentFrom | $\{x_1\} \sqsubseteq \neg\{x_2\}$ | {john} $\sqsubseteq \neg${peter} |
| subPropertyOf | $P_1 \sqsubseteq P_2$ | hasDaughter $\sqsubseteq$ hasChild |
| equivalentProperty | $P_1 \equiv P_2$ | cost $\equiv$ price |
| inverseOf | $P_1 \equiv P_2^-$ | hasChild $\equiv$ hasParent$^-$ |
| transitiveProperty | $P^+ \sqsubseteq P$ | ancestor$^+$ $\sqsubseteq$ ancestor |
| functionalProperty | $\top \sqsubseteq \leqslant 1 P$ | $\top \sqsubseteq \leqslant$1hasMother |
| inverseFunctionalProperty | $\top \sqsubseteq \leqslant 1 P^-$ | $\top \sqsubseteq \leqslant$1hasSSN$^-$ |

☞ $\mathcal{I}$ **satisfies** $C_1 \sqsubseteq C_2$ iff $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$; satisfies $P_1 \sqsubseteq P_2$ iff $P_1^{\mathcal{I}} \subseteq P_2^{\mathcal{I}}$

# OWL Axioms

| Axiom | 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 |
| disjointWith | $C_1 \sqsubseteq \neg C_2$ | Male $\sqsubseteq \neg$Female |
| sameIndividualAs | $\{x_1\} \equiv \{x_2\}$ | {President_Bush} $\equiv$ {G_W_Bush} |
| differentFrom | $\{x_1\} \sqsubseteq \neg\{x_2\}$ | {john} $\sqsubseteq \neg${peter} |
| subPropertyOf | $P_1 \sqsubseteq P_2$ | hasDaughter $\sqsubseteq$ hasChild |
| equivalentProperty | $P_1 \equiv P_2$ | cost $\equiv$ price |
| inverseOf | $P_1 \equiv P_2^-$ | hasChild $\equiv$ hasParent$^-$ |
| transitiveProperty | $P^+ \sqsubseteq P$ | ancestor$^+$ $\sqsubseteq$ ancestor |
| functionalProperty | $\top \sqsubseteq \leqslant 1 P$ | $\top \sqsubseteq \leqslant$1hasMother |
| inverseFunctionalProperty | $\top \sqsubseteq \leqslant 1 P^-$ | $\top \sqsubseteq \leqslant$1hasSSN$^-$ |

- $\mathcal{I}$ **satisfies** $C_1 \sqsubseteq C_2$ iff $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$; satisfies $P_1 \sqsubseteq P_2$ iff $P_1^{\mathcal{I}} \subseteq P_2^{\mathcal{I}}$
- $\mathcal{I}$ satisfies ontology $\mathcal{O}$ (is a **model** of $\mathcal{O}$) iff satisfies every axiom in $\mathcal{O}$

# Reasoning with OWL DL

# Reasoning

# Reasoning

- ☞ Why do we want it?

# Reasoning

- ☞ Why do we want it?
  - Semantic Web aims at “machine understanding”
  - **Understanding** closely related to **reasoning**

# Reasoning

- Why do we want it?
  - Semantic Web aims at “machine understanding”
  - **Understanding** closely related to **reasoning**
- What can we do with it?

# Reasoning

- ☞ Why do we want it?
  - Semantic Web aims at “machine understanding”
  - **Understanding** closely related to **reasoning**
- ☞ What can we do with it?
  - **Design and maintenance** of ontologies
    - Check class consistency and compute class hierarchy
    - Particularly important with large ontologies/multiple authors

# Reasoning

- Why do we want it?
  - Semantic Web aims at “machine understanding”
  - **Understanding** closely related to **reasoning**
- What can we do with it?
  - **Design and maintenance** of ontologies
    - Check class consistency and compute class hierarchy
    - Particularly important with large ontologies/multiple authors
  - **Integration** of ontologies
    - Assert inter-ontology relationships
    - Reasoner computes integrated class hierarchy/consistency

# Reasoning

- Why do we want it?
  - Semantic Web aims at "machine understanding"
  - **Understanding** closely related to **reasoning**
- What can we do with it?
  - **Design and maintenance** of ontologies
    - Check class consistency and compute class hierarchy
    - Particularly important with large ontologies/multiple authors
  - **Integration** of ontologies
    - Assert inter-ontology relationships
    - Reasoner computes integrated class hierarchy/consistency
  - **Querying** class and instance data w.r.t. ontologies
    - Determine if set of facts are consistent w.r.t. ontologies
    - Determine if individuals are instances of ontology classes
    - Retrieve individuals/tuples satisfying a query expression
    - Check if one class subsumes (is more general than) another w.r.t. ontology
    - ...

# Why Decidable Reasoning?

# Why Decidable Reasoning?

- ☞ OWL DL constructors/axioms restricted so reasoning is **decidable**

# Why Decidable Reasoning?

- OWL DL 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 DL provides powerful but still decidable **ontology language**
  - Further layers may (will) extend OWL
    - Will almost certainly be undecidable

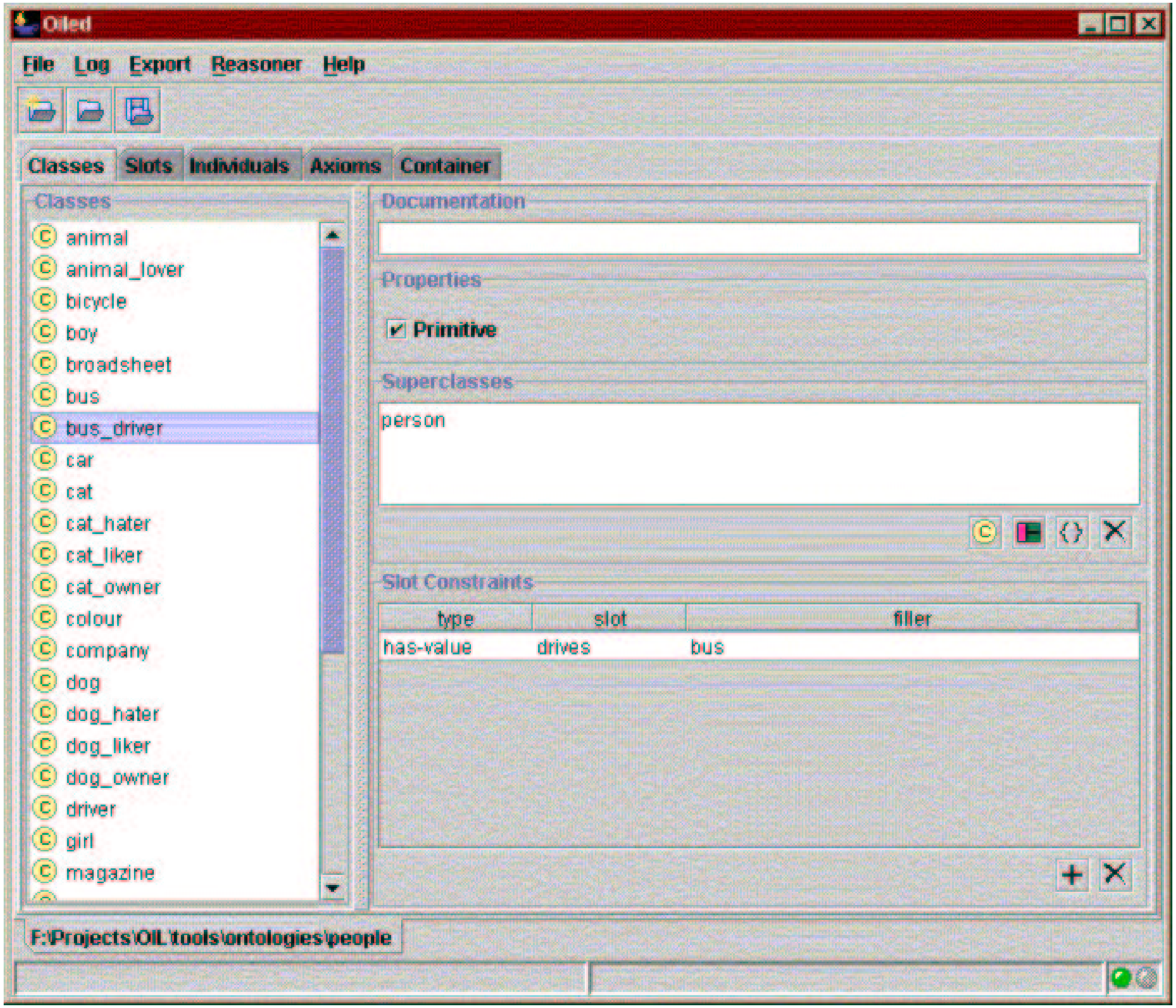
# Why Decidable Reasoning?

- OWL DL 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 DL provides powerful but still decidable **ontology language**
  - Further layers may (will) extend OWL
    - Will almost certainly be undecidable
- Facilitates provision of **reasoning services**
  - Known "practical" **algorithms**
  - Several implemented **systems**
  - Evidence of **empirical tractability**

# Why Decidable Reasoning?

- OWL DL 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 DL provides powerful but still decidable **ontology language**
  - Further layers may (will) extend OWL
    - Will almost certainly be undecidable
- Facilitates provision of **reasoning services**
  - Known "practical" **algorithms**
  - Several implemented **systems**
  - Evidence of **empirical tractability**
- Understanding dependent on **reliable & consistent** reasoning

# Reasoning Support for Ontology Design: OilEd



# Description Logic Reasoning

# Highly Optimised Implementation

# Highly Optimised Implementation

- DL reasoning based on tableaux algorithms

# Highly Optimised Implementation

- ☞ DL reasoning based on tableaux algorithms
- ☞ Naive implementation $\longrightarrow$ effective non-termination

# Highly Optimised Implementation

- ☞ DL reasoning based on tableaux algorithms
- ☞ Naive implementation $\longrightarrow$ effective non-termination
- ☞ Modern systems include **MANY** optimisations

# Highly Optimised Implementation

- ☞ DL reasoning based on tableaux algorithms
- ☞ Naive implementation $\longrightarrow$ effective non-termination
- ☞ Modern systems include **MANY** optimisations
- ☞ Optimised **classification** (compute partial ordering)
  - Use enhanced traversal (exploit information from previous tests)
  - Use structural information to select classification order

# Highly Optimised Implementation

- ☞ DL reasoning based on tableaux algorithms
- ☞ Naive implementation $\longrightarrow$ effective non-termination
- ☞ Modern systems include **MANY** optimisations
- ☞ Optimised **classification** (compute partial ordering)
  - Use enhanced traversal (exploit information from previous tests)
  - Use structural information to select classification order
- ☞ Optimised **subsumption** testing (search for models)
  - Normalisation and simplification of concepts
  - Absorption (simplification) of general axioms
  - Davis-Putnam style semantic branching search
  - Dependency directed backtracking
  - Caching of satisfiability results and (partial) models
  - Heuristic ordering of propositional and modal expansion
  - ...

# Research and Implementation Challenges

# Challenges

# Challenges

☞ **Increased expressive power**

- Existing DL systems implement (at most) $\mathcal{SHIQ}$
- OWL extends $\mathcal{SHIQ}$ with datatypes and nominals

# Challenges

- **Increased expressive power**
  - Existing DL systems implement (at most) $\mathcal{SHIQ}$
  - OWL extends $\mathcal{SHIQ}$ with datatypes and nominals
- **Scalability**
  - Very large KBs
  - Reasoning with (very large numbers of) individuals

# Challenges

- **Increased expressive power**
  - Existing DL systems implement (at most) $\mathcal{SHIQ}$
  - OWL extends $\mathcal{SHIQ}$ with datatypes and nominals
- **Scalability**
  - Very large KBs
  - Reasoning with (very large numbers of) individuals
- **Other reasoning tasks**
  - Querying
  - Matching
  - Least common subsumer
  - ...

# Challenges

- **Increased expressive power**
  - Existing DL systems implement (at most) $\mathcal{SHIQ}$
  - OWL extends $\mathcal{SHIQ}$ with datatypes and nominals
- **Scalability**
  - Very large KBs
  - Reasoning with (very large numbers of) individuals
- **Other reasoning tasks**
  - Querying
  - Matching
  - Least common subsumer
  - ...
- **Tools and Infrastructure**
  - Support for large scale ontological engineering and deployment

# Summary

# Summary

- ☞ **Semantic Web** aims to make web resources accessible to automated processes

# Summary

- ☞ **Semantic Web** aims to make web resources accessible to automated processes
- ☞ **Ontologies** will play key role by providing vocabulary for semantic markup

# Summary

- **Semantic Web** aims to make web resources accessible to automated processes
- **Ontologies** will play key role by providing vocabulary for semantic markup
- **OWL** is an ontology language designed for the web
  - Exploits existing standards: XML, RDF(S)
  - Adds KR idioms from object oriented and frame systems
  - Formal rigor of a **logic**
  - Facilitates provision of **reasoning support**

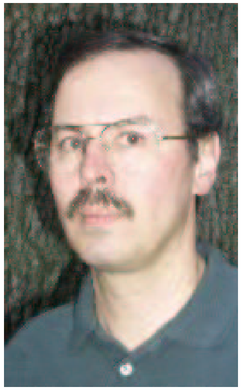
# Summary

- ☞ **Semantic Web** aims to make web resources accessible to automated processes
- ☞ **Ontologies** will play key role by providing vocabulary for semantic markup
- ☞ **OWL** is an ontology language designed for the web
  - Exploits existing standards: XML, RDF(S)
  - Adds KR idioms from object oriented and frame systems
  - Formal rigor of a **logic**
  - Facilitates provision of **reasoning support**
- ☞ **Challenges** remain
  - Reasoning with nominals
  - (Convincing) demonstration(s) of scalability
  - New reasoning tasks

# Acknowledgements

# Acknowledgements

- Members of the OIL, DAML+OIL and OWL development teams, in particular Dieter Fensel and Frank van Harmelen (Amsterdam) and Peter Patel-Schneider (Bell Labs)

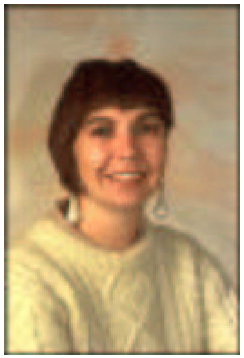
  

# Acknowledgements


- Members of the OIL, DAML+OIL and OWL development teams, in particular Dieter Fensel and Frank van Harmelen (Amsterdam) and Peter Patel-Schneider (Bell Labs)
- Franz Baader, Uli Sattler and Stefan Tobies (Dresden)


# Acknowledgements

- Members of the OIL, DAML+OIL and OWL development teams, in particular Dieter Fensel and Frank van Harmelen (Amsterdam) and Peter Patel-Schneider (Bell Labs)
- Franz Baader, Uli Sattler and Stefan Tobies (Dresden)
- Members of the Information Management, Medical Informatics and Formal Methods Groups at the University of Manchester



# Resources

Slides from this talk

`http://www.cs.man.ac.uk/~horrocks/Slides/ilash.pdf`

FaCT system (open source)

`http://www.cs.man.ac.uk/FaCT/`

OilEd (open source)

`http://oiled.man.ac.uk/`

W3C Web-Ontology (WebOnt) working group (OWL)

`http://www.w3.org/2001/sw/WebOnt/`

**Description Logic Handbook**

Baader et al., Cambridge University Press

# Select Bibliography


I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. Reviewing the design of DAML+OIL: An ontology language for the Semantic Web. In *Proc. of AAAI 2002*, 2002.

I. Horrocks and P. F. Patel-Schneider. Three theses of representation in the Semantic Web. In *Proc. of WWW 2003*, 2003. To appear

I. Horrocks and S. Tessaris. Querying the semantic web: a formal approach. In I. Horrocks and J. Hendler, editors, *Proc. of the 2002 International Semantic Web Conference (ISWC 2002)*, number 2342 in Lecture Notes in Computer Science. Springer-Verlag, 2002.

C. Lutz. *The Complexity of Reasoning with Concrete Domains*. PhD thesis, Teaching and Research Area for Theoretical Computer Science, RWTH Aachen, 2001.

# Select Bibliography


I. Horrocks and U. Sattler. Ontology reasoning in the $\mathcal{SHOQ}(\mathbf{D})$ description logic. In B. Nebel, editor, *Proc. of IJCAI-01*, pages 199–204. Morgan Kaufmann, 2001.

F. Baader, S. Brandt, and R. Küsters. Matching under side conditions in description logics. In B. Nebel, editor, *Proc. of IJCAI-01*, pages 213–218, Seattle, Washington, 2001. Morgan Kaufmann.

A. Borgida, E. Franconi, and I. Horrocks. Explaining $\mathcal{ALC}$ subsumption. In *Proc. of ECAI 2000*, pages 209–213. IOS Press, 2000.

D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, and R. Rosati. A principled approach to data integration and reconciliation in data warehousing. In *Proceedings of the International Workshop on Design and Management of Data Warehouses (DWDM'99)*, 1999.