Reasoning with DAML+OIL:

What can it do for YOU?

Ian Horrocks

horrocks@cs.man.ac.uk

University of Manchester
Manchester, UK

DAML+OIL Language Overview

DAML+OIL is an **ontology** language

DAML+OIL Language Overview

DAML+OIL is an **ontology** language

- Describes **structure** of the domain (i.e., a schema)
    - RDF used to describe specific **instance** of domain (data)

DAML+OIL Language Overview

DAML+OIL is an **ontology** language

- Describes **structure** of the domain (i.e., a schema)
    - RDF used to describe specific **instance** of domain (data)
- Structure described in terms of **classes** and **properties**

DAML+OIL Language Overview

DAML+OIL is an **ontology** language

- Describes **structure** of the domain (i.e., a schema)
  - RDF used to describe specific **instance** of domain (data)
- Structure described in terms of **classes** and **properties**
- Ontology consists of set of **axioms**
  - E.g., asserting class subsumption/equivalence

DAML+OIL Language Overview

DAML+OIL is an **ontology** language

- ☞ Describes **structure** of the domain (i.e., a schema)
  - RDF used to describe specific **instance** of domain (data)
- ☞ Structure described in terms of **classes** and **properties**
- ☞ Ontology consists of set of **axioms**
  - E.g., asserting class subsumption/equivalence
- ☞ Classes can be names or **expressions**
  - Various **constructors** provided for building class expressions

DAML+OIL Language Overview

DAML+OIL is an **ontology** language

- ☞ Describes **structure** of the domain (i.e., a schema)
  - RDF used to describe specific **instance** of domain (data)
- ☞ Structure described in terms of **classes** and **properties**
- ☞ Ontology consists of set of **axioms**
  - E.g., asserting class subsumption/equivalence
- ☞ Classes can be names or **expressions**
  - Various **constructors** provided for building class expressions
- ☞ **Expressive power** determined by
  - Kinds of axiom supported
  - Kinds of class (and property) constructor supported

DAML+OIL Class Constructors

Constructor	Abbreviation	Example
intersectionOf	$C_1 \wedge \ldots \wedge C_n$	Human $\wedge$ Male
unionOf	$C_1 \vee \ldots \vee C_n$	Doctor $\vee$ Lawyer
complementOf	$\neg C$	$\neg$Male
oneOf	$\{x_1 \ldots x_n\}$	$\{$john, mary$\}$
toClass	$\forall P.C$	$\forall$hasChild.Doctor
hasClass	$\exists P.C$	$\exists$hasChild.Lawyer
hasValue	$\exists P.\{x\}$	$\exists$citizenOf.$\{$USA$\}$
minCardinalityQ	$\geqslant n\, P.C$	$\geqslant 2$ hasChild.Lawyer
maxCardinalityQ	$\leqslant n\, P.C$	$\leqslant 1$ hasChild.Male
cardinalityQ	$= n\, P.C$	$= 1$ hasParent.Female

DAML+OIL Class Constructors

Constructor	Abbreviation	Example
intersectionOf	$C_1 \wedge \ldots \wedge C_n$	Human $\wedge$ Male
unionOf	$C_1 \vee \ldots \vee C_n$	Doctor $\vee$ Lawyer
complementOf	$\neg C$	$\neg$Male
oneOf	$\{x_1 \ldots x_n\}$	$\{$john, mary$\}$
toClass	$\forall P.C$	$\forall$hasChild.Doctor
hasClass	$\exists P.C$	$\exists$hasChild.Lawyer
hasValue	$\exists P.\{x\}$	$\exists$citizenOf.$\{$USA$\}$
minCardinalityQ	$\geqslant n\, P.C$	$\geqslant 2$ hasChild.Lawyer
maxCardinalityQ	$\leqslant n\, P.C$	$\leqslant 1$ hasChild.Male
cardinalityQ	$= n\, P.C$	$= 1$ hasParent.Female

- ☞ Arbitrarily complex **nesting** of constructors
  - E.g., $\forall$hasChild.(Doctor $\vee$ $\exists$hasChild.Doctor)

DAML+OIL Class Constructors

Constructor	Abbreviation	Example
intersectionOf	$C_1 \wedge \ldots \wedge C_n$	Human $\wedge$ Male
unionOf	$C_1 \vee \ldots \vee C_n$	Doctor $\vee$ Lawyer
complementOf	$\neg C$	$\neg$Male
oneOf	$\{x_1 \ldots x_n\}$	$\{$john, mary$\}$
toClass	$\forall P.C$	$\forall$hasChild.Doctor
hasClass	$\exists P.C$	$\exists$hasChild.Lawyer
hasValue	$\exists P.\{x\}$	$\exists$citizenOf.$\{$USA$\}$
minCardinalityQ	$\geqslant n\, P.C$	$\geqslant 2$ hasChild.Lawyer
maxCardinalityQ	$\leqslant n\, P.C$	$\leqslant 1$ hasChild.Male
cardinalityQ	$= n\, P.C$	$= 1$ hasParent.Female

- Arbitrarily complex **nesting** of constructors
  - E.g., $\forall$hasChild.(Doctor $\vee$ $\exists$hasChild.Doctor)
- XMLS **datatypes** as well as classes

DAML+OIL Axioms

Axiom	Abbreviation	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human $\sqsubseteq$ Animal $\wedge$ Biped
sameClassAs	$C_1 \doteq C_2$	Man $\doteq$ Human $\wedge$ Male
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter $\sqsubseteq$ hasChild
samePropertyAs	$P_1 \doteq P_2$	cost $\doteq$ price
sameIndividualAs	$x_1 \doteq x_2$	President_Bush $\doteq$ G_W_Bush
disjointWith	$C_1 \sqsubseteq \neg C_2$	Male $\sqsubseteq$ $\neg$Female
differentIndividualFrom	$\{x_1\} \sqsubseteq \neg\{x_2\}$	{john} $\sqsubseteq$ $\neg${peter}
inverseOf	$P_1 \doteq P_2^-$	hasChild $\doteq$ hasParent$^-$
transitiveProperty	$P^+ \sqsubseteq P$	ancestor$^+$ $\sqsubseteq$ ancestor
uniqueProperty	Thing $\sqsubseteq \leqslant 1 P$	Thing $\sqsubseteq$ $\leqslant 1$hasMother
UnambiguousProperty	Thing $\sqsubseteq \leqslant 1 P^-$	Thing $\sqsubseteq$ $\leqslant 1$isMotherOf$^-$

DAML+OIL Axioms

Axiom	Abbreviation	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human $\sqsubseteq$ Animal $\wedge$ Biped
sameClassAs	$C_1 \doteq C_2$	Man $\doteq$ Human $\wedge$ Male
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter $\sqsubseteq$ hasChild
samePropertyAs	$P_1 \doteq P_2$	cost $\doteq$ price
sameIndividualAs	$x_1 \doteq x_2$	President_Bush $\doteq$ G_W_Bush
disjointWith	$C_1 \sqsubseteq \neg C_2$	Male $\sqsubseteq \neg$Female
differentIndividualFrom	$\{x_1\} \sqsubseteq \neg\{x_2\}$	{john} $\sqsubseteq \neg${peter}
inverseOf	$P_1 \doteq P_2^-$	hasChild $\doteq$ hasParent$^-$
transitiveProperty	$P^+ \sqsubseteq P$	ancestor$^+$ $\sqsubseteq$ ancestor
uniqueProperty	Thing $\sqsubseteq \leqslant 1P$	Thing $\sqsubseteq \leqslant 1$hasMother
UnambiguousProperty	Thing $\sqsubseteq \leqslant 1P^-$	Thing $\sqsubseteq \leqslant 1$isMotherOf$^-$

☞ Axioms (mostly) **reducible to subClass/PropertyOf**

Decidable Reasoning

Set of operators/axioms restricted so that reasoning is **decidable**

Decidable Reasoning

Set of operators/axioms restricted so that reasoning is **decidable**

- ☞ Significant point on tractability -v- expressiveness scale

Decidable Reasoning

Set of operators/axioms restricted so that reasoning is **decidable**

- ☞ Significant point on tractability -v- expressiveness scale
- ☞ Consistent with Semantic Web’s **layered architecture**
  - XML provides syntax transport layer
  - RDF provides basic ontological primitives
  - DAML+OIL provides (decidable) logical layer
  - Further layers (e.g., **rules**) will extend DAML+OIL

Decidable Reasoning

Set of operators/axioms restricted so that reasoning is **decidable**

- ☞ Significant point on tractability -v- expressiveness scale
- ☞ Consistent with Semantic Web's **layered architecture**
  - XML provides syntax transport layer
  - RDF provides basic ontological primitives
  - DAML+OIL provides (decidable) logical layer
  - Further layers (e.g., **rules**) will extend DAML+OIL
- ☞ Facilitates provision of **reasoning services**
  - Known algorithms
  - Implemented systems
  - Evidence of **empirical tractability**

Why Reasoning Services?

Reasoning is important for:

Why Reasoning Services?

Reasoning is important for:

- Ontology **design**
  - Check class consistency and (unexpected) implied relationships
  - Particularly important with large ontologies/multiple authors

Why Reasoning Services?

Reasoning is important for:

- Ontology **design**
  - Check class consistency and (unexpected) implied relationships
  - Particularly important with large ontologies/multiple authors
- Ontology **integration**
  - Assert inter-ontology relationships
  - Reasoner computes integrated class hierarchy/consistency

Why Reasoning Services?

Reasoning is important for:

- ☞ Ontology **design**
  - Check class consistency and (unexpected) implied relationships
  - Particularly important with large ontologies/multiple authors
- ☞ Ontology **integration**
  - Assert inter-ontology relationships
  - Reasoner computes integrated class hierarchy/consistency
- ☞ Ontology **deployment**
  - Determine if set of facts are consistent w.r.t. ontology
  - Determine if individuals are instances of ontology classes
  - No point in having semantics unless exploited by “agents”

Why Reasoning Services?

Reasoning is important for:

- ☞ Ontology **design**
  - Check class consistency and (unexpected) implied relationships
  - Particularly important with large ontologies/multiple authors
- ☞ Ontology **integration**
  - Assert inter-ontology relationships
  - Reasoner computes integrated class hierarchy/consistency
- ☞ Ontology **deployment**
  - Determine if set of facts are consistent w.r.t. ontology
  - Determine if individuals are instances of ontology classes
  - No point in having semantics unless exploited by “agents”

“**The Semantic Web needs a logic on top**” (Henry Thompson)

OilEd

OilEd is a DAML+OIL **ontology editor** with reasoning support

OilEd

OilEd is a DAML+OIL **ontology editor** with reasoning support

- ☞ Frame based interface (inspired by Protegé)

OilEd

OilEd is a DAML+OIL **ontology editor** with reasoning support

- ☞ Frame based interface (inspired by Protegé)
- ☞ Extended to clarify semantics and capture whole language
  - Explicit $\exists$ (hasClass) or $\forall$ (toClass) restrictions
  - Boolean connectives ($\wedge$, $\vee$, $\neg$) and nesting
  - Transitive and unique (functional) properties

OilEd

OilEd is a DAML+OIL **ontology editor** with reasoning support

- ☞ Frame based interface (inspired by Protegé)
- ☞ Extended to clarify semantics and capture whole language
  - Explicit $\exists$ (hasClass) or $\forall$ (toClass) restrictions
  - Boolean connectives ($\wedge$, $\vee$, $\neg$) and nesting
  - Transitive and unique (functional) properties
- ☞ Reasoning support provided by FaCT system
  - Ontology translated into $\mathcal{SHIQ}$ DL
  - Communicates with FaCT via CORBA interface
  - Indicates inconsistencies and implicit subsumptions
  - Can add axioms to make implicit subsumptions explicit

Reasoning Examples — what you CAN do

E.g., DAML+OIL medical terminology ontology

Reasoning Examples — what you CAN do

E.g., DAML+OIL medical terminology ontology

- ☞ Transitive roles capture partonomy, causality, etc.

Reasoning Examples — what you CAN do

E.g., DAML+OIL medical terminology ontology

- ☞ Transitive roles capture partonomy, causality, etc.

  $\text{Smoking} \sqsubseteq \exists\text{causes.Cancer}$ plus $\text{Cancer} \sqsubseteq \exists\text{causes.Death}$

  $\Rightarrow \text{Smoking} \sqsubseteq \exists\text{causes.Death}$

Reasoning Examples — what you CAN do

E.g., DAML+OIL medical terminology ontology

☞ Transitive roles capture partonomy, causality, etc.

$\text{Smoking} \sqsubseteq \exists \text{causes.Cancer}$ plus $\text{Cancer} \sqsubseteq \exists \text{causes.Death}$

$\Rightarrow \text{Smoking} \sqsubseteq \exists \text{causes.Death}$

☞ Multiple equality/inclusion axioms

Reasoning Examples — what you CAN do

E.g., DAML+OIL medical terminology ontology

- ☞ Transitive roles capture partonomy, causality, etc.

  Smoking $\sqsubseteq$ $\exists$causes.Cancer plus Cancer $\sqsubseteq$ $\exists$causes.Death
  $\Rightarrow$ Smoking $\sqsubseteq$ $\exists$causes.Death

- ☞ Multiple equality/inclusion axioms

  Stomach-Ulcer $\doteq$ Ulcer $\wedge$ $\exists$hasLocation.Stomach plus
  Stomach-Ulcer $\sqsubseteq$ $\exists$hasLocation.Lining-Of-Stomach
  $\Rightarrow$ Ulcer $\wedge$ $\exists$hasLocation.Stomach $\sqsubseteq$ OrganLiningLesion

Reasoning Examples — what you CAN do

E.g., DAML+OIL medical terminology ontology

- Transitive roles capture partonomy, causality, etc.

  $\text{Smoking} \sqsubseteq \exists\text{causes.Cancer}$ plus $\text{Cancer} \sqsubseteq \exists\text{causes.Death}$
  $\Rightarrow \text{Smoking} \sqsubseteq \exists\text{causes.Death}$

- Multiple equality/inclusion axioms

  $\text{Stomach-Ulcer} \doteq \text{Ulcer} \wedge \exists\text{hasLocation.Stomach}$ plus
  $\text{Stomach-Ulcer} \sqsubseteq \exists\text{hasLocation.Lining-Of-Stomach}$
  $\Rightarrow \text{Ulcer} \wedge \exists\text{hasLocation.Stomach} \sqsubseteq \text{OrganLiningLesion}$

- Inverse roles capture e.g. causes/causedBy relationship

Reasoning Examples — what you CAN do

E.g., DAML+OIL medical terminology ontology

- ☞ Transitive roles capture partonomy, causality, etc.

  Smoking $\sqsubseteq$ $\exists$causes.Cancer plus Cancer $\sqsubseteq$ $\exists$causes.Death

  $\Rightarrow$ Smoking $\sqsubseteq$ $\exists$causes.Death

- ☞ Multiple equality/inclusion axioms

  Stomach-Ulcer $\doteq$ Ulcer $\wedge$ $\exists$hasLocation.Stomach plus

  Stomach-Ulcer $\sqsubseteq$ $\exists$hasLocation.Lining-Of-Stomach

  $\Rightarrow$ Ulcer $\wedge$ $\exists$hasLocation.Stomach $\sqsubseteq$ OrganLiningLesion

- ☞ Inverse roles capture e.g. causes/causedBy relationship

  Death $\wedge$ $\exists$causedBy.Smoking $\sqsubseteq$ PrematureDeath

  $\Rightarrow$ Smoking $\sqsubseteq$ $\exists$causes.PrematureDeath

Reasoning Examples — what you CAN do

E.g., DAML+OIL medical terminology ontology

- ☞ Transitive roles capture partonomy, causality, etc.

  $\text{Smoking} \sqsubseteq \exists\text{causes.Cancer}$ plus $\text{Cancer} \sqsubseteq \exists\text{causes.Death}$

  $\Rightarrow \text{Smoking} \sqsubseteq \exists\text{causes.Death}$

- ☞ Multiple equality/inclusion axioms

  $\text{Stomach-Ulcer} \doteq \text{Ulcer} \wedge \exists\text{hasLocation.Stomach}$ plus

  $\text{Stomach-Ulcer} \sqsubseteq \exists\text{hasLocation.Lining-Of-Stomach}$

  $\Rightarrow \text{Ulcer} \wedge \exists\text{hasLocation.Stomach} \sqsubseteq \text{OrganLiningLesion}$

- ☞ Inverse roles capture e.g. causes/causedBy relationship

  $\text{Death} \wedge \exists\text{causedBy.Smoking} \sqsubseteq \text{PrematureDeath}$

  $\Rightarrow \text{Smoking} \sqsubseteq \exists\text{causes.PrematureDeath}$

- ☞ Cardinality restrictions add consistency constraints

Reasoning Examples — what you CAN do

E.g., DAML+OIL medical terminology ontology

- ☞ Transitive roles capture partonomy, causality, etc.

  Smoking $\sqsubseteq \exists$causes.Cancer plus Cancer $\sqsubseteq \exists$causes.Death

  $\Rightarrow$ Smoking $\sqsubseteq \exists$causes.Death

- ☞ Multiple equality/inclusion axioms

  Stomach-Ulcer $\doteq$ Ulcer $\wedge \exists$hasLocation.Stomach plus

  Stomach-Ulcer $\sqsubseteq \exists$hasLocation.Lining-Of-Stomach

  $\Rightarrow$ Ulcer $\wedge \exists$hasLocation.Stomach $\sqsubseteq$ OrganLiningLesion

- ☞ Inverse roles capture e.g. causes/causedBy relationship

  Death $\wedge \exists$causedBy.Smoking $\sqsubseteq$ PrematureDeath

  $\Rightarrow$ Smoking $\sqsubseteq \exists$causes.PrematureDeath

- ☞ Cardinality restrictions add consistency constraints

  BloodPressure $\sqsubseteq \exists$hasValue.$($High $\vee$ Low$) \wedge \leqslant 1$hasValue plus

  High $\sqsubseteq \neg$Low $\Rightarrow$ HighLowBloodPressure $\sqsubseteq \bot$

Reasoning Examples — what you CAN’T do

Where to begin!

Reasoning Examples — what you CAN'T do

Where to begin!

☞ Robust decidability largely due to **tree model property**

- For any consistent class there exists a tree (like) model

Reasoning Examples — what you CAN'T do

Where to begin!

- ☞ Robust decidability largely due to **tree model property**
  - For any consistent class there exists a tree (like) model
- ☞ No **property constructors**, e.g.:
  - parent $\circ$ brother $\sqsubseteq$ uncle
  - ancestor $\doteq$ parent^{+}

Reasoning Examples — what you CAN'T do

Where to begin!

☞ Robust decidability largely due to **tree model property**

- For any consistent class there exists a tree (like) model

☞ No **property constructors**, e.g.:

- parent $\circ$ brother $\sqsubseteq$ uncle
- ancestor $\doteq$ parent^{+}

☞ No **variables**, e.g.:

- Ulcer $\wedge$ $\exists$hasLocation.$?x$ $\sqsubseteq$ $\exists$hasLocation.$($$\exists$LiningOf.$?x$$)$

Reasoning Examples — what you CAN'T do

Where to begin!

- ☞ Robust decidability largely due to **tree model property**
  - For any consistent class there exists a tree (like) model
- ☞ No **property constructors**, e.g.:
  - parent $\circ$ brother $\sqsubseteq$ uncle
  - ancestor $\doteq$ parent$^+$
- ☞ No **variables**, e.g.:
  - Ulcer $\wedge$ $\exists$hasLocation.$?x$ $\sqsubseteq$ $\exists$hasLocation.$($$\exists$LiningOf.$?x$$)$
- ☞ Only have **unary and binary predicates**
  - Can't express (directly) $P(x, y, z)$

Reasoning Examples — what you CAN'T do

Where to begin!

- Robust decidability largely due to **tree model property**
  - For any consistent class there exists a tree (like) model
- No **property constructors**, e.g.:
  - parent $\circ$ brother $\sqsubseteq$ uncle
  - ancestor $\doteq$ parent$^+$
- No **variables**, e.g.:
  - Ulcer $\wedge$ $\exists$hasLocation.$?x$ $\sqsubseteq$ $\exists$hasLocation.$($$\exists$LiningOf.$?x)$
- Only have **unary and binary predicates**
  - Can't express (directly) $P(x, y, z)$

Language extensions may remove some of above limitations

Reasoning Examples — what you CAN'T do

Where to begin!

- ☞ Robust decidability largely due to **tree model property**
  - For any consistent class there exists a tree (like) model
- ☞ No **property constructors**, e.g.:
  - parent $\circ$ brother $\sqsubseteq$ uncle
  - ancestor $\doteq$ parent$^+$
- ☞ No **variables**, e.g.:
  - Ulcer $\wedge$ $\exists$hasLocation.$?x$ $\sqsubseteq$ $\exists$hasLocation.$($$\exists$LiningOf.$?x$$)$
- ☞ Only have **unary and binary predicates**
  - Can't express (directly) $P(x, y, z)$

Language extensions may remove some of above limitations

- ☞ But there is **no such thing as a free lunch**