OIL: *the* Ontology Language for the Semantic Web

Ian Horrocks[1], Stefan Decker[2], Dieter Fensel[3], Frank van Harmelen[3,4],

Sergey Melnik[2], Michel Klein[3], Jeen Broekstra[4]

and others too numerous to mention

[1] **University of Manchester, UK**

[2] **AIFB, University of Karlsruhe, Germany**

[3] **Vrije Universiteit Amsterdam, Holland**

[4] **AIdministrator Nederland B.V.**

Requirements for an Ontology Language

- Well designed
  - Intuitive to human users
  - Adequate expressive power
- Well defined
  - Clearly specified syntax (obviously)
  - Formal semantics (equally important)
  - Adequate expressive power
- Compatible with existing (web) standards

Standards for Ontology Languages

- Proposals already exist for W3C standard schema languages
  - XMLS (XML Schema)
  - RDFS (RDF Schema)
- Both have been touted as (standard) web ontology languages
- However, both suffer from
  - Expressive inadequacy — lack of basic modelling primitives
  - Poorly (un) defined semantics

Proposed Common Core: OIL

- Simple and intuitive Frame Language syntax
  - Many users are frightened by logic-based syntax (I know I am!)
  - Rich range of modelling primitives
  - Can still function as a basic frame language
  - Facilitates construction/adaption of tools
- Semantics defined by mapping to expressive Description Logic
  - Well defined formal properties (decidability, complexity)
  - Enriched expressive power (boolean connectives, etc.)
  - Can provide reasoning services to support ontology design
- Compatibility provided by layering on top of RDFS
  - Class hierarchy etc. accessible to any RDFS-aware agent

Frames + DL + WWW $\Rightarrow$ OIL

Why Reasoning Support?

- Reasoning support is key feature of OIL
- Reasoning is important
  - as design support tool
  - for large ontologies
  - with multiple authors
  - for integrating and sharing ontologies
- Because it allows to
  - Establishing inter-ontology relationships
  - Checking for consistency
  - Checking for (unexpected) implied relationships

"The Semantic Web needs a logic on top" (Henry Thompson)

OIL Language Overview

OIL restricts frame languages:

- No defaults
- Limited axioms/rules
- Ontology only (limited form of individuals)

Main reasons for this:

- Reasoning support
- Semantics

OIL extends frame languages:

- Defined classes (necessary and sufficient conditions)
- Enhanced slot constraints
  - Restriction to class as well as value
  - Existential and universal restrictions
  - Cardinality constraints with optional class qualifier
  - Boolean expressions as well as class names
  - Sub-slots as well as sub-classes
  - Properties on slots (transitive, symmetrical)
  - Inverse slots
  - . . .
- Concrete data types
  - Integers and strings, with min, max, ranges etc.
- Additional kinds of axiom
  - Disjointness, disjoint-coverings, equivalence etc.

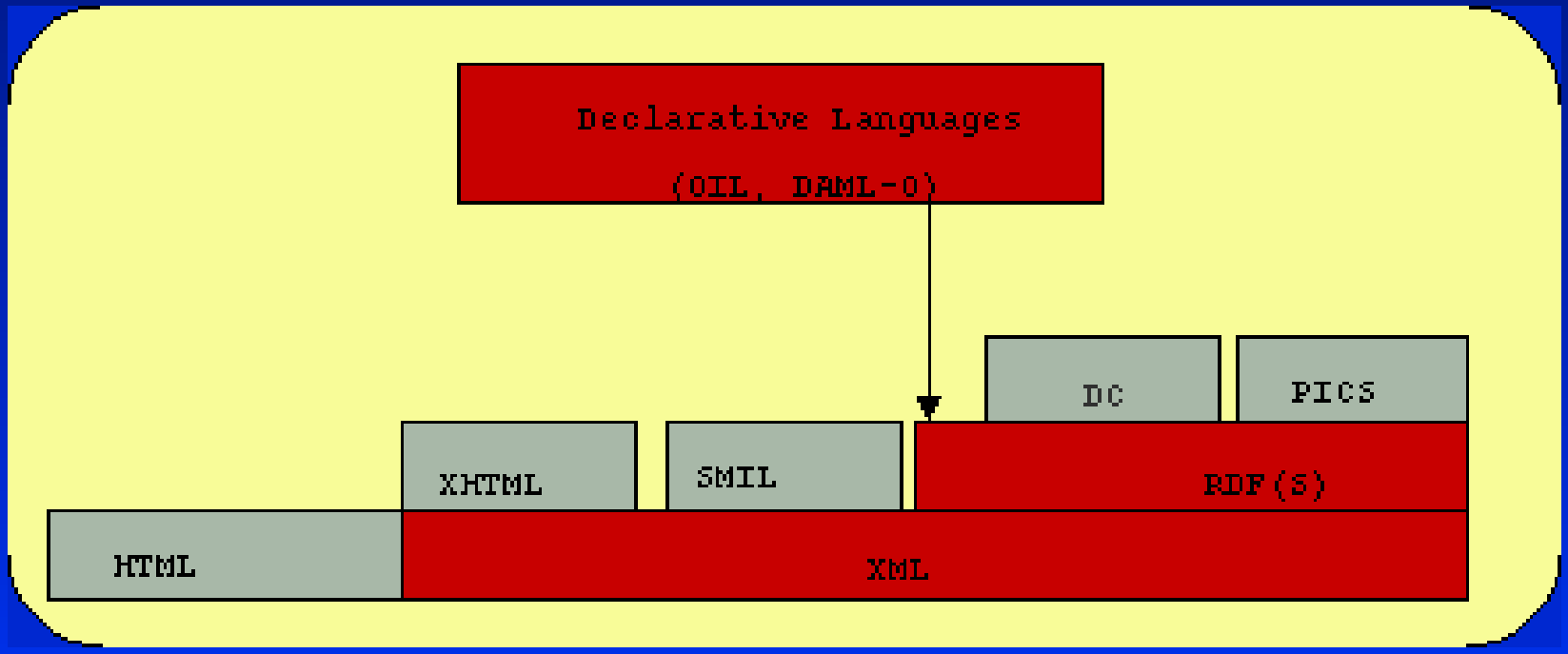
OIL by Example

```
slot-def part-of                          % part-of is a slot
   subslot-of structural-relation         % sub-slot of structural-relation
   inverse has-part                       % inverse is has-part
   properties transitive                  % it is transitive

class-def defined herbivore               % herbivore exactly defined as:
   subclass-of animal                     % sub-class of animal
   slot-constraint eats                   % that eats
      value-type plant OR                 % only plants
         slot-constraint part-of          % or parts of
            has-value plant               % plants
   min-cardinality 2 vegetable            % and ⩾2 types of vegetable

disjoint herbivore carnivore              % herbivore and carnivore disjoint
```

Semantics via translation to $\mathcal{SHIQ}$ DL:

OIL	Equivalent $\mathcal{SHIQ}$
slot-def part-of	
subslot-of structural-relation	% part-of $\sqsubseteq$ structural-relation
inverse has-part	% has-part $\doteq$ part-of$^-$
properties transitive	% part-of $\in \mathbf{R}_+$
class-def defined herbivore	% herbivore $\doteq$
subclass-of animal	% animal $\sqcap$
slot-constraint eats	
value-type plant **OR**	% $\forall$eats.(plant $\sqcup$
slot-constraint part-of	
has-value plant	% $\exists$part-of.plant) $\sqcap$
min-cardinality 2 vegetable	% $\geqslant 2.eats$vegetable
disjoint herbivore carnivore	% herbivore $\sqsubseteq \neg$carnivore

How to Put Ontologies on the Web (W3C vision)



Compatibility via RDFS delivery syntax:

```
<rdf:Property rdf:ID="has-part">
  <rdfs:subPropertyOf rdf:resource="#structural-relation"/>
  <oil:inverseRelationOf rdf:resource="#is-part-of"/>
</rdf:Property>

<rdfs:Class rdf:ID="herbivore">
  <rdf:type rdf:resource="http://www.ontoknowledge.org/oil/rdfs-schema/#DefinedC
  <rdfs:subClassOf rdf:resource="#animal"/>
  <oil:hasSlotConstraint>
    <oil:valueType>
      <oil:hasProperty rdf:resource="#eats"/>
      <oil:hasClass>
        <oil:OR>
          <oil:hasOperand rdf:resource="#plant"/>
          <oil:hasOperand>
            <oil:has-value>
              <oil:hasProperty rdf:resource="#is-part-of"/>
              <oil:hasClass rdf:resource="#plant"/>
            </oil:has-value>
          </oil:hasOperand>
                  :
                  :
```

Extensible OIL

One of the key ideas behind OIL:

Don’t make the core language too large

- Core language should contain only “consensus” primitives
- Additional expressive power provided by language extensions
- These could include:
  - Rules
  - Additional algebraic properties on slots
  - Limited second order features
  - Modules, import, etc.
  - . . .

DAML and OIL

- US DAML initiative also developing RDFS based ontology language
- Largely based on OIL
- Joint US/EU Committee on Agent Markup Languages now established
- Ultimate aim is OIL/DAML based W3C standardisation proposal

OIL Infrastructure

- Reasoning services provided by CORBA FaCT system
  - Currently via OIL ↔ FaCT translators (XSL)
  - CORBA OIL coming soon
- Frame ontology editors being built/adapted to OIL
  - Protege editor (Stanford)
  - OntoEdit (Karlsruhe)
  - OilEd (Manchester)
- Additional infrastructure urgently required

WonderWeb Project

EU Project Proposal to:

"Provide ontology infrastructure for the Semantic Web"

- key objectives
  - Ontology languages and standardisation
  - Integration/reconciliation techniques for migration and sharing
  - Foundational ontologies for range of application domains
  - Technical infrastructure and tools for development and deployment
- Participants
  - University of Manchester, UK (coordinator)
  - Vrije Universiteit Amsterdam, Netherlands
  - LADSEB-CNR, Italy
  - University of Karlsruhe, Germany
  - InfoLab, Stanford University, USA
  - Interprice Technologies GmbH, Germany

More information at: www.cs.man.ac.uk/~horrocks/WonderWeb