DAML+OIL: an Ontology Language for the Semantic Web

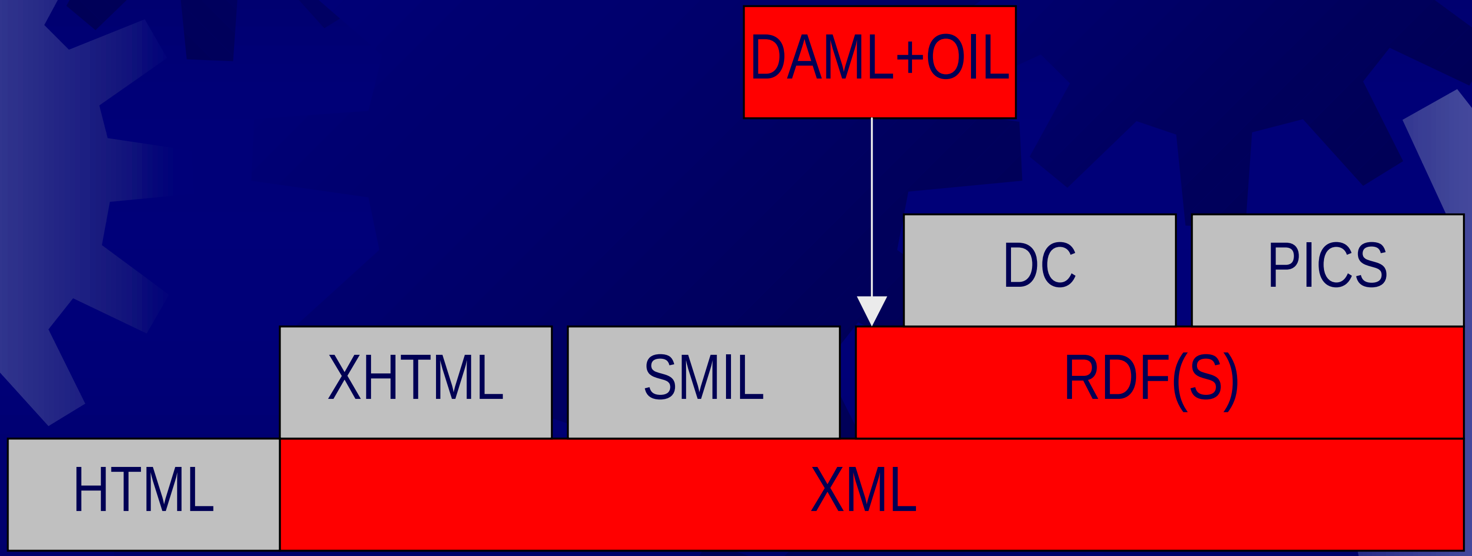
DAML+OIL Design Objectives

- Well ***designed***
  - Intuitive to (human) users
  - Adequate expressive power
  - Support machine understanding/reasoning
- Well ***defined***
  - Clearly specified syntax (obviously)
  - Formal semantics (equally important)
- ***Extend*** existing web standards
  - DAML+OIL is built on top of RDF(S)

Why Build on RDF

- Provides basic ontological primitives
  - Classes and relations (properties)
  - Class (and property) hierarchy
- Can exploit existing RDF infrastructure
- Provides mechanism for using ontologies
  - RDF triples assert facts about resources
  - Use vocabulary from DAML+OIL ontologies

The Cake!



Why RDF Is Not Enough

- Expressive inadequacy
  - Only range/domain constraints (on properties)
  - No properties of properties (unique, transitive, inverse etc.)
  - No equivalence, disjointness, coverings etc.
  - No necessary and sufficient conditions (for class membership)
- Poorly (un) defined semantics

How DAML+OIL Builds ON RDFS (1)

- Extends expressive power
  - Constraints (restrictions) on properties of classes (existential/universal/cardinality)
  - Boolean combinations of classes and restrictions
  - Equivalence, disjointness, coverings
  - Necessary and sufficient conditions
  - Constraints on properties

How DAML+OIL Builds ON RDFS (2)

- Provides well defined semantics
  - Meaning of DAML+OIL statements is formally specified
  - Both model theoretic and axiomatic specifations provided
  - Allows for machine understanding and automated reasoning

DAML+OIL ↔ RDF

- DAML+OIL ontology is a set of RDF statements
- DAML+OIL defines semantics for certain statements
- Does **NOT** restrict what can be said
  - Ontology can include arbitrary RDF
- But no semantics for non-DAML+OIL statements

Well Designed(?)

- Intuitive to (human) users
  - Supports common ontological idioms
- Adequate expressive power
  - Extends RDF in several directions
- Support for machine understanding/reasoning
  - Designed to be “implementable”
  - No features for which it is difficult or impossible to define clear semantics (e.g., defaults)
  - Decidable and (empirically) tractable reasoning

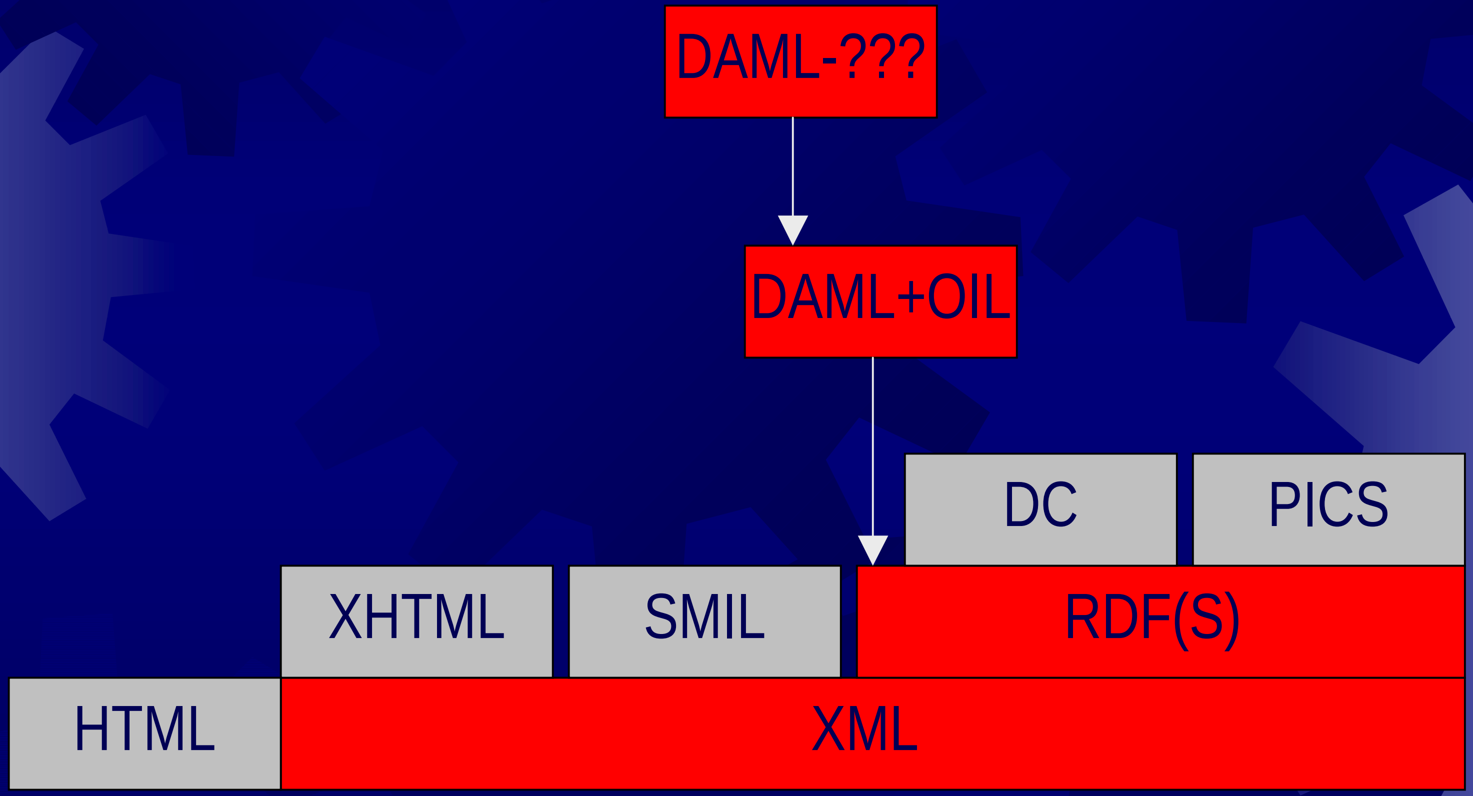
Why Automated Reasoning?

- Semantic web requires machine understanding (of resource descriptions)
  - Reasoning is integral to understanding
- Supports design and use of ontologies
  - Checking class consistency (e.g., *Skyscraper*)
  - Checking/deriving subClassOf hierarchy
  - Particularly useful when ontologies are large, multi-authored and rapidly evolving
  - Also useful when integrating/sharing ontologies
- Does not tell us how to deal with inconsistencies
  - But we should be able to determine when they exist

Extending DAML+OIL

- Work in progress on Datatypes
  - Plan to support (some of) XMLS datatypes
  - Datatypes will be disjoint from “abstract” classes and only accessible via properties
  - Maintains “implementability” of language
- Further extensions in new language layers
  - E.g., DAML-RULES
  - Layers will use DAML+OIL as it uses RDF

New Language Layers


DAML-???

DAML+OIL

DC

PICS

XHTML

SMIL

RDF(S)

HTML

XML

DAML+OIL Infrastructure

- Can exploit existing RDF tools/services
- Ontology editors being built/adapted
  - OilEd (Manchester)
  - Protégé (Stanford)
  - OntoEdit (Karlsruhe)
- Ontology integration tools being built/adapted
  - Chimera (Stanford)
- Reasoning services
  - DL derived reasoners, e.g., FaCT (used by OilEd)
  - Rule based reasoners, e.g. SiLri (Karlsruhe)
- Markup tools
- Additional tools/infrastructure urgently required

DAML+OIL Summary

- Ontology language for Semantic Web
- Extends RDF
  - More expressive power
  - Well defined semantics
- Implementable
  - Decidable and tractable reasoning
  - Cost is some restriction on expressive power
- Extensible
  - Cost may be loss of (some of) above properties