Intelligent Internet Technologies

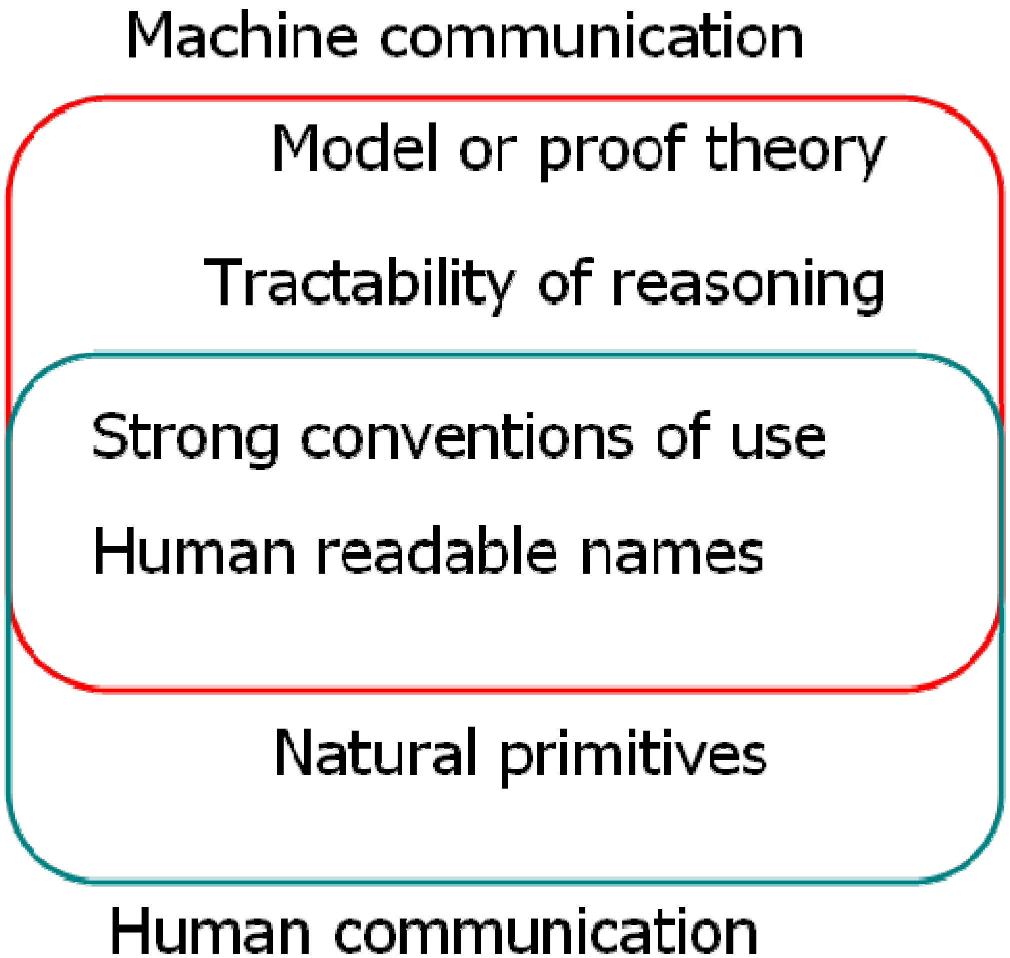
Lecture 20

Ontology Representation Languages: History

Alexandra V. Vitko

KNURE, AI Department, alexandra_vitko@yahoo.com

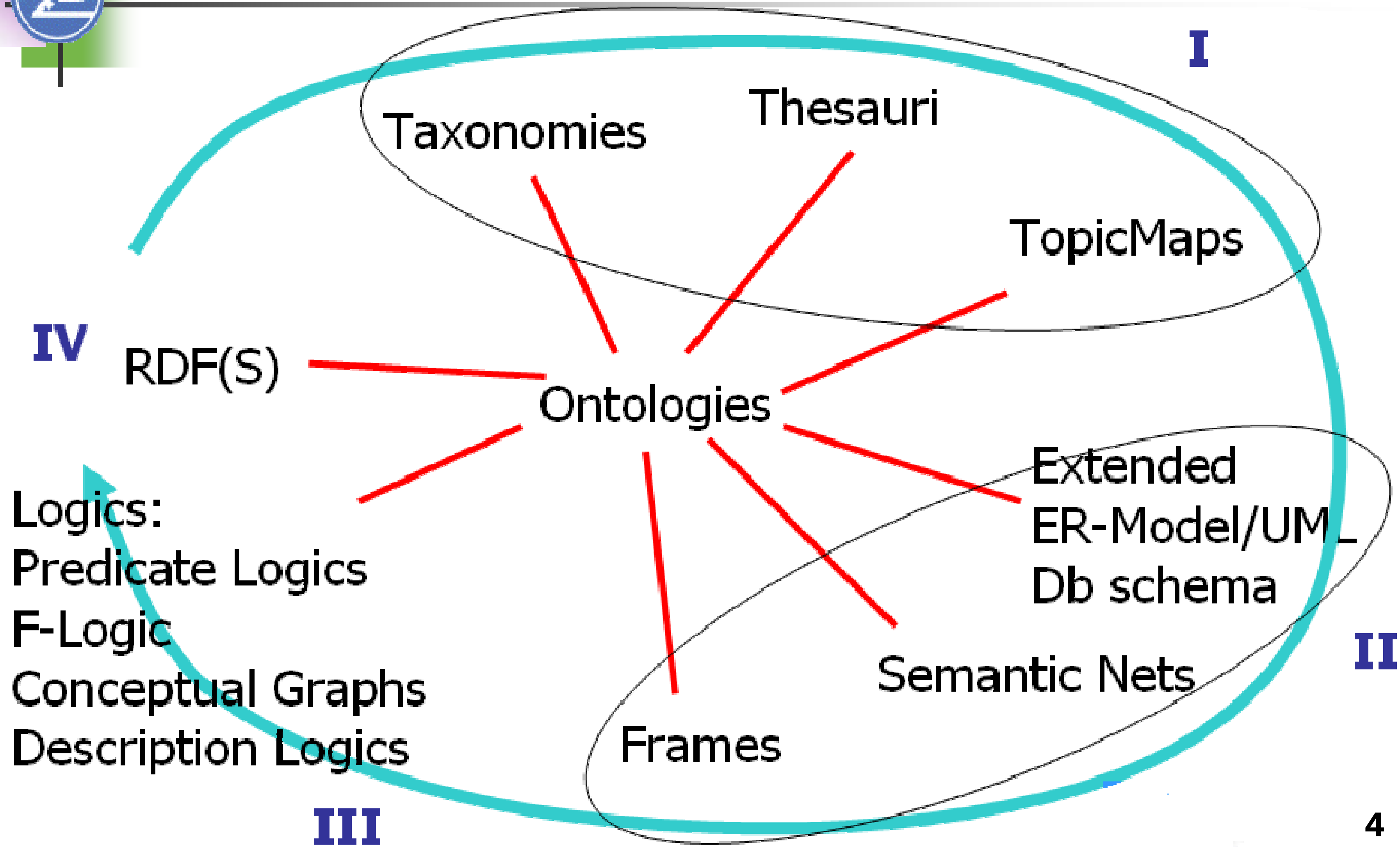
Language: desirable properties



Ontology Representation Languages

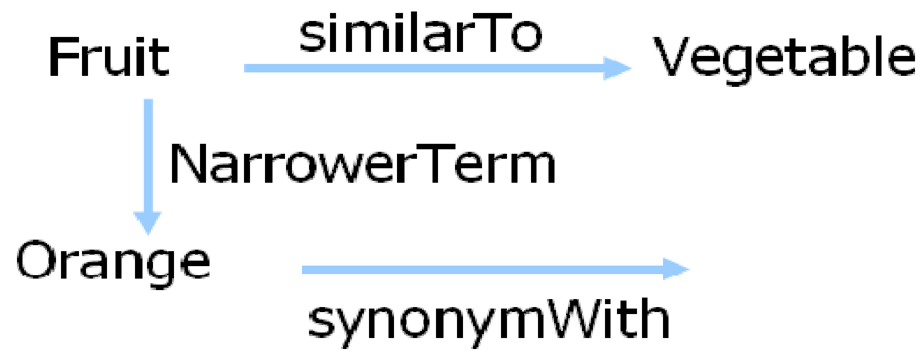
- Machines need communication with formal content to restrict meaning
- What makes a language formal?
  - model theory (1st order predicate logic)
  - proof theory (Gentzen calculus)
  - conventions (e.g. Java)

Knowledge Representation Paradigms (incomplete)



Thesauri

Example:



- Graph with labels edges (similar, nt, bt, synonym)
- Fixed set of edge labels (aka relations)
- no instances

Well known in library science

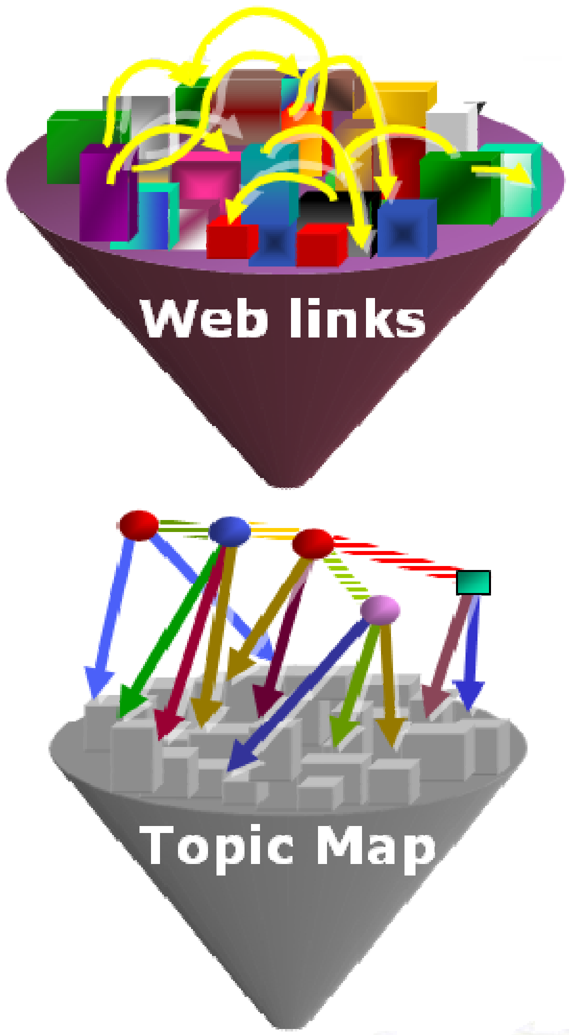
Topic Maps (I)

- Standardized: ISO/IEC 13250:2000 (Jan 2000)
  - enabling standard to describe knowledge structures, electronic indices, classification schemes, …
- To enable information resources to be classified and navigated in a consistent manner by representing knowledge structures for indexes
  - build valuable information networks above any kind of resources / data objects
  - enable the structuring of unstructured information
- To make subjects addressable.
  - The “GPS of the Web”

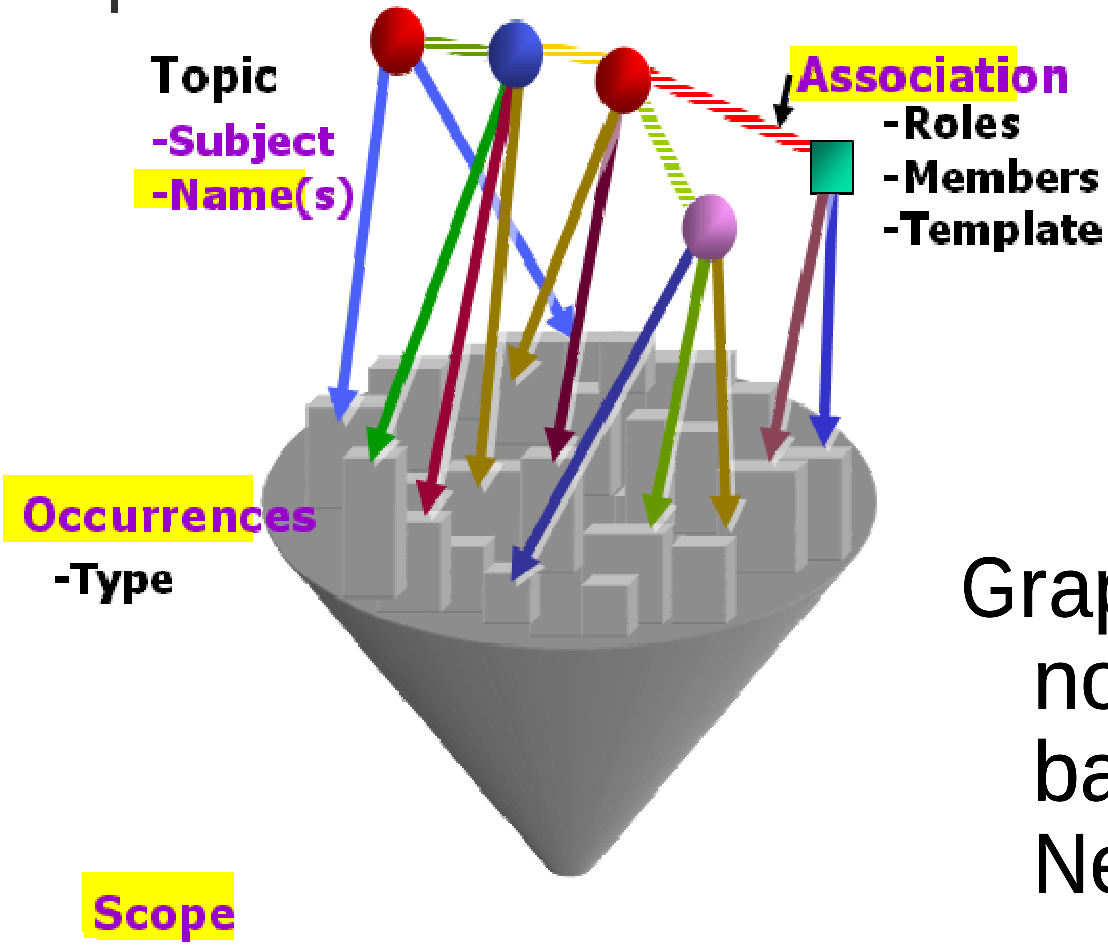
http://www.topicmaps.org

Topic Maps (II)

The electronic equivalent of table of contents, glossaries, thesauri, cross references



High-level Topic Maps concepts



Knowledge,
Occurrences,
Associations,
Names

Graph made of nodes and arcs, based on Semantic Nets

In-/Semi-formal approaches: Topic Maps, Thesauri

Advantages

- Intuitive.
- Interesting primitives that are not available in other approaches (TM).
- Topic Maps Web enabled:
  - XML Topic Maps (XTM) are ready to use.

Disadvantages

- Dependent from particular implementation.
- May be misinterpreted (TM) / few primitives (Thesauri).
- No formal interpretation.
- No formal restrictions.
- Hard to build and maintain large and coherent schemes.
- Pre-enumerate concepts.

Frame/OO model summary

Advantages

- Intuitive and popular modelling style.
- Many tools and examples.
- Some reasoning.

Disadvantages

- Extending/evolving problematic
- Hand crafting taxonomies and asserted properties.
- Static classifications.
- Pre-enumerate concepts.
- Little reasoning support
- Difficult to build large coherent and complete ontologies (e.g. multiple classifications)

First-order predicate logic languages CycL and KIF

- CycL has been developed in the Cyc project [Lenat & Guha, 1990] for the purpose of specifying the large common-sense ontology that should provide Artificial Intelligence to computers
- Cyc provides the worldwide largest formalized ontology
- CycL is a formal language whose syntax is derived from first-order predicate calculus (it extends first-order logic through the use of second order concepts)

Main concepts of CycL

- **Constants** are the vocabulary of the CycL language (the "words" that are used in writing the axioms)
- **Predicates** express relationships between terms
- **Variables** stand for terms or formulas
- **Formulas** combine terms into meaningful expressions
- **Complex formulas**

  (and...(or...(xor A (and...))...)...)
- **Quantification:** one universal quantifier *forAll*, four existential quantifiers *thereExists*, *thereExistAtLeast, thereExistAtMost, thereExistAxactly*
- **Second-order Quantification**
- **Functions**
- **Microtheories** – set of formulas in the knowledge base

Knowledge Interchange Format KIF

- KIF [Genesereth & Fikes, 1992] is a language designed for use in the exchange of knowledge between disparate computer systems (created by different programmers at different times, in different languages, and so forth)
- KIF can be used as a language for expressing and exchanging ontologies
- Categorical features that are essential to the design of KIF:
  - The language has *declarative semantics*
  - The language is *logically comprehensive* – at its most general, it provides for the expression of arbitrary logic sentences
  - The language provides *means for the representation of knowledge about knowledge*

Main concepts of KIF

- Four categories of **constants**: object constants, function constants, relation constants and logical constants
- Three disjoint types of **expressions**: *terms*, *sentences* and *definitions*. A knowledge base is a finite set of sentences and definitions
- Six types of **sentences**: constant, equation, inequality, relsent, logsent, quantsent
- Three types of **definitions**: unrestricted, complete and partial

Common features of CycL and KIF

- Both languages are oriented on predicate logics
- provide an important extension of first-order logic
- Allow meta-level statements
- CycL provides richer modeling primitives than KIF
- CycL is a modeling language for ontologies whereas KIF was designed as an exchange format for ontologies

Frame-based Approaches: Ontolingua and Frame Logic (1)

- The central modeling primitives of *frame-based* and *object-oriented* approaches are classes with certain properties called attributes
- Ontolingua [Gruber, 1993] was designed to support the design and specification of ontologies with a clear logical semantics based on KIF
- An Ontolingua ontology is made up of definitions of classes, relations, functions, distinguished objects and axioms that relate these terms
- The frame ontology is expressed as second-order axioms in Ontolingua. It contains a complete axiomatization of classes and instances, slots and slot constraints, class and relation specialization, relation inverses, relation composition, and class partitions

http://ontolingua.stanford.edu/

Frame-based Approaches: Ontolingua and Frame Logic (2)

- Frame logic is a language for specifying object-oriented databases, Frame systems, and logical programs
- Realised in Ontobroker system
- Main concepts of Frame-Logic approach:
  - Classes and Subclasses – C1::C2
  - Instance of – O:C
  - Attribute Declaration – C1[A=>>C2]
  - Attribute Value – O[A->>V]
  - Part-of – O1<:O2
  - Relations – p(a1,...,a2)

Frame-based Approaches: Ontolingua and Frame Logic (3)

- Ontolingua and Frame logic integrate frames (i.e., classes) into a logical framework
- The main difference between Ontolingua and Frame logic is the manner in which they realize frame-based modeling primitives in a logical language
- Ontolingua characterizes the frame-based modeling primitives via axioms in the language
- Ontolingua applies standard semantics of predicate logic
- Frame logic provides a more complex semantics compared to predicate logic. The modeling primitives are explicitly defined in the semantics of Frame logic

Description Logics

- *Description Logics* (DL) [Brachman & Schmolze, 1985], also known as *terminological logics*, form an important powerful class of logic-based knowledge representation languages
- Description Logics try to find a fragment of first-order logic with high expressive power and efficient inference procedure
- A distinguishing feature of Description logics is tha classes (usually called concepts) can be defined intensionally in terms of descriptions that specify the properties that objects must satisfy to belong to the concept
- Essential concepts (in CLASSIC):
  - **Primitive concept** – (PRIMITIVE THING car), (PRIMITIVE(AND CAR EXPENSIVE-THING) sports-car)
  - **Value restriction** – (ALL thing-driven CAR)
  - **Cardinality bounds** restrict the number of fillers for roles – (AT-MOST 4 thing-driven)
  - **Co-reference constraints** specify simple equalities between single-valued roles – (SAME-AS (driver) (insurance payer))

http://dl.kr.org/

Description Logics

- Many years of DL research
  - Well defined semantics
  - Formal properties well understood (complexity, decidability)
  - Known reasoning algorithms
  - Implemented systems (highly optimised)

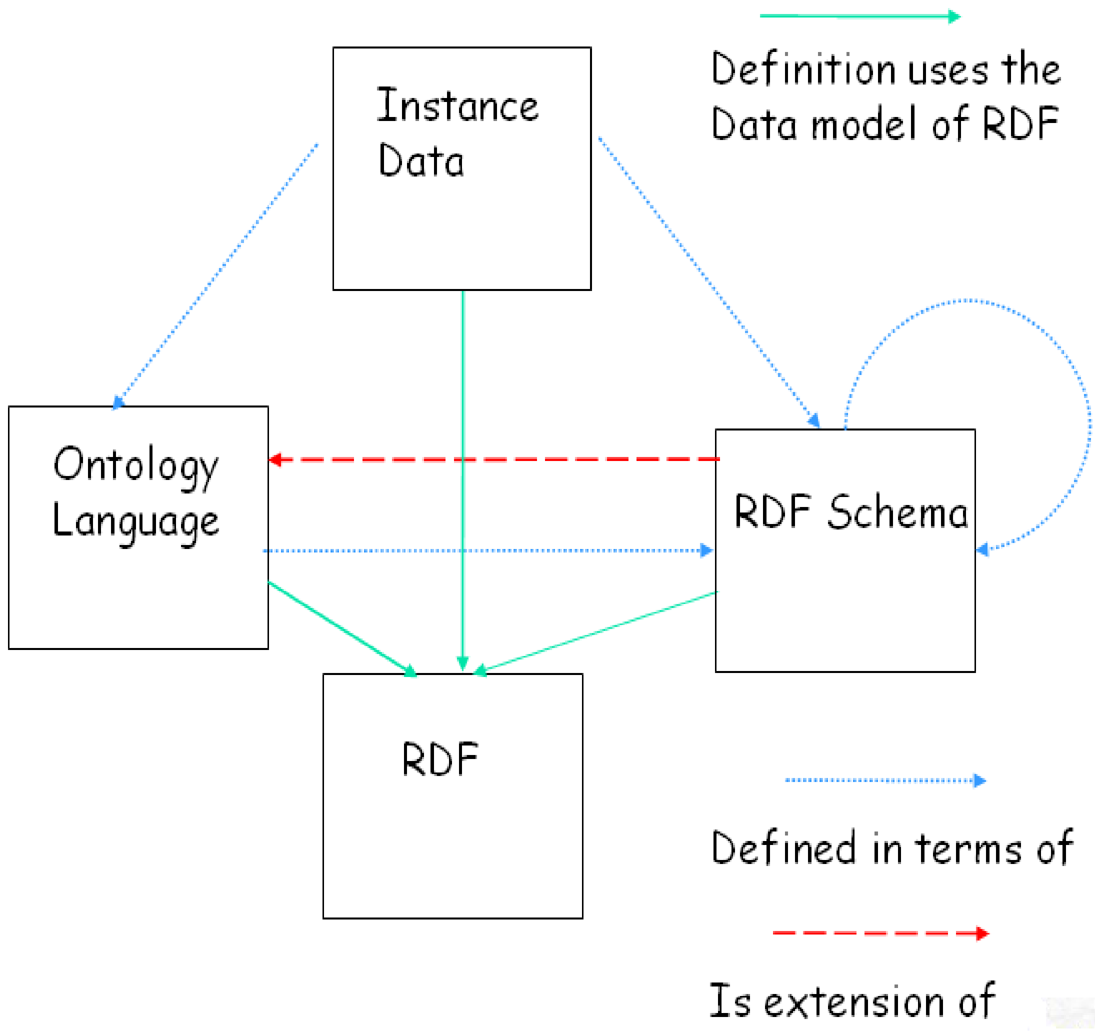
What's in a "Logic based ontology"?

- Primitive concepts - in a hierarchy
  - Described but not defined
- Properties - relations between concepts, also in a hierarchy
- Constructors – on concepts and properties
  - "some", "only", "at least", "at most", and, or, not.
- Defined concepts
  - Made from primitive concepts, constructors and descriptors
  - Enzyme ≡ protein and catalyses reaction.
  - Reason that enzyme is a kind of protein.
- "is-kind-of" = "implies"
  - "Dog is a kind of wolf" mean "All dogs are wolves"
- Axioms
  - disjointness, further description of defined concepts
- A Reasoner
  - Consistency & Taxonomy for defined concepts established though logical reasoning.

Model built up incrementally and descriptively based on concept's properties.

RDF(S) Extensibility

- Define an Ontology of your Language with RDF Schema (like RDF-Schema itself)
- Describe Instance Data using your new Vocabulary
- Advantage: all Languages use the same Data Model (simplifies Interoperability)



RDF Schema (RDFS)

- RDF just defines the data model.
- Need for definition of vocabularies for the data model - an Ontology Language!
- RDF-Schemas describe rules for using RDF properties
  - Define a domain vocabulary for RDF
  - Organise this vocabulary in a typed hierarchy
- RDF Schemas are Web resources (and have URIs) and can be described using RDF.
- Are not to be confused with XML Schemas.
- RDFS is the framework for a vocabulary.

RDF Schema Model

- Property-centric: Each property specifies what classes of subjects and objects it relates. New properties can be added to a class without modifying the class
  - resource, class, subClassOf, type
  - property, subPropertyOf
  - domain, range, constraintResource, constraintProperty
- Definitions can include constraints which express validation conditions
  - domain constraints link properties with classes
  - range constraints limit property values
- BUT poorly defined semantics

The DAML Program

- DAML: DARPA Agent Markup Language
- Defense Advanced Research Agency (DARPA) program
  - Program Managers: James Hendler, Murray Burke
  - Begin in August 2000
- Goal: achieve semantic interoperability between Web pages, databases, programs, and sensors
- Integration contractor and 16 technology development teams
  - MIT (Tim Berners-Lee, Ben Grosof)
  - Stanford (Gio Weiderhold, Richard Fikes, Deborah McGuinness)
  - UMBC (Tim Finin)
  - U West Florida (Pay Hayes)
  - Yale (Drew McDermott)
  - Cycorp (Doug Lenat)
  - Nokia (Ora Lassila)
  - Teknowledge (Bob Balzer)

  ...
- Advisors: Ramanthan Guha, Peter Patel-Schneider, …

Roots of OIL

Description Logics:
Formal Semantics &
Reasoning Support

Frame-based systems:
Epistemological Modeling
Primitives

OIL

Web languages:
XML- and RDF-based syntax

History: OIL&DAML

- OIL developed by group of (largely) European researchers: extends RDF Schema to a fully-fledged knowledge representation language
  - Logical expressions
  - Data typing
  - cardinality
  - Quantifiers
  - http://www.ontoknowledge.org
- DAML=US sister of OIL (http://www.daml.org)
- Merged as DAML+ OIL in 2001
- Becomes OWL W3C standard 10 Feb 04

DAML+OIL

- A representation language for user-defined ontologies
  - An ontology added to RDF and RDF-Schema
  - Specification document:

    http://www.daml.org/2000/12/daml+oil-index.html
- Expressive power analogous to:
  - Description logics (e.g., CLASSIC)
  - Monotonic frame languages (e.g., OKBC knowledge model)
- Designed in collaboration with the European Community

  Designers of the Ontology Inference Layer (OIL)
- Basis for OWL, the candidate W3C standard

Evaluation of DAML+OIL/OWL

Advantages

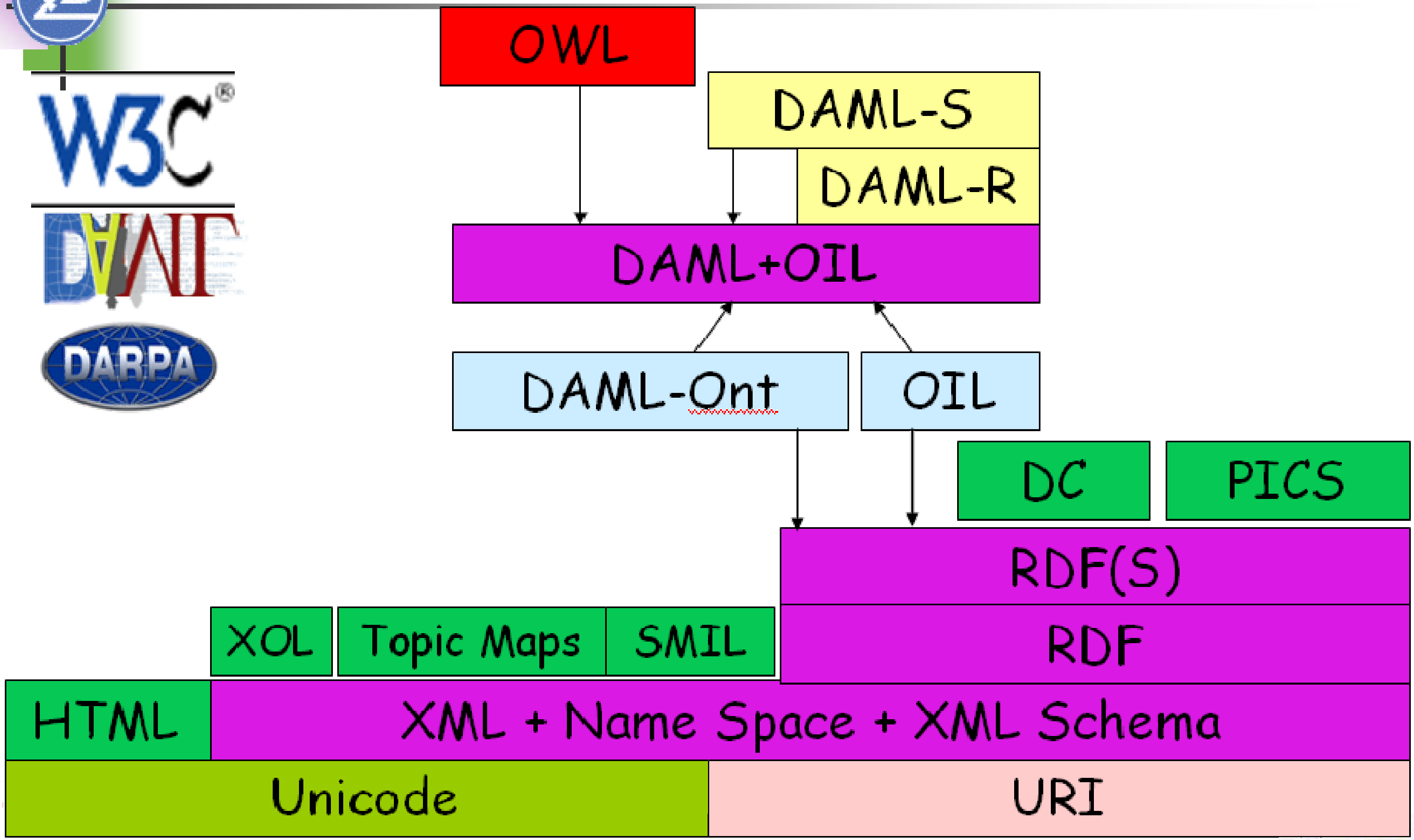
- Decidable (if choosen carefully like DAML+OIL)
- Subsumption reasoning
- Consistency checking
- Dynamically post-coordinate rather than have to pre-enumerate.
- Support for evolution, merging, large scale building
- Can publish the ontologies as static lattices.
- W3C standard (so tools etc)

Disadvantages

- Different modeling style if want to take advantage of reasoning.
- Limited support for A-Box reasoning (on instances) in tractable DL versions

"As simple as required but as complex as necessary"

The Ontology Language Stack



Web Languages Stack summary

- XML
  - Interchange syntax, no semantics
- RDF
  - Data model, some semantics & inference (recent)
- RDF Schema
  - Concept modelling, more semantics & inference
- DAML+ OIL/ OWL
  - More expressive ontology language;
  - Quite expressive; expensive inference

  Requirements for a Web Ontology Language W3C
  http://www.w3.org/TR/webont-req/

Why Reasoning?

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
- Query inclusion
- Service description matchmaking
- Classification-based querying

Languages: Summary

- Thesauri & Topic Maps
  - Hand crafted, flexible but difficult to evolve, maintain and keep consistent, with poor semantics.
- Object-based KR: e.g. frames
  - Extensively used, good structuring, intuitive.
  - Semantics defined by OKBC standard.
- Logic-based: Description Logics
  - Very expressive, model is a set of theories, well defined semantics, reasoning.
  - Automatic derived classification taxonomies.
  - Concepts are defined and primitive.
  - Expressivity vs. computational complexity balance.