Introduction to OWL

Matthew Horridge

Ontologies

An ontology is a machine processable artefact
that describes the concepts in some domain of interest
and the relationships between them

Ontologies

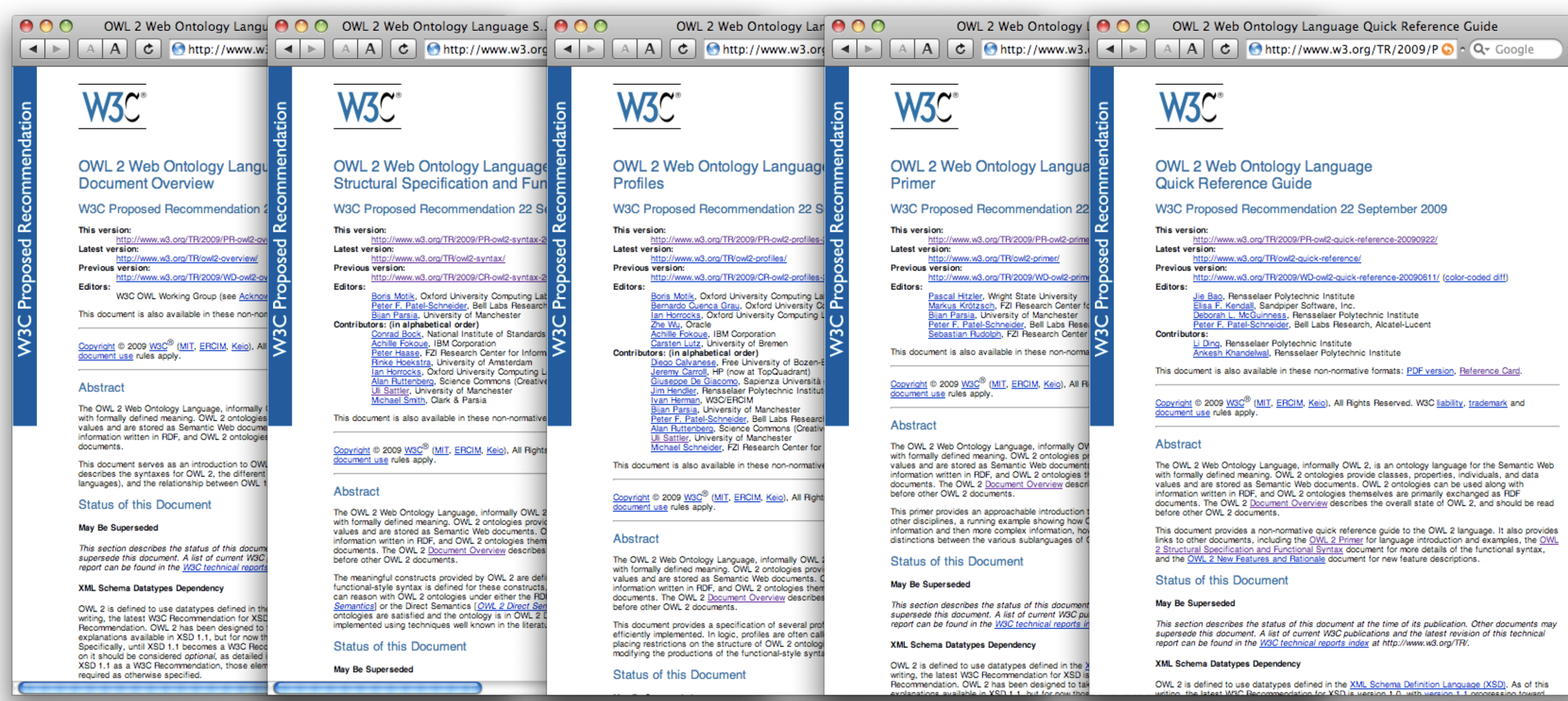
Vocabulary	Fixed set of terms. e.g. DublinCore http://dublincore.org/
Thesaurus	Vocabulary plus relationships between vocabulary terms. e.g. WordNet http://wordnet.princeton.edu/
Ontology	Vocabulary plus structured descriptions of terms, generalisation, specialisation of terms. e.g. Foundational Model of Anatomy http://sig.biostr.washington.edu/projects/fm/ e.g. Early versions of the Gene Ontology
Logic based Ontology	Ontology written in a language that is underpinned by a logic, giving it precisely specified semantics and computable relationships between terms. e.g. SNOMED CT http://www.ihtsdo.org/ e.g. Current versions of the Gene Ontology

OWL

The latest standard in ontology languages

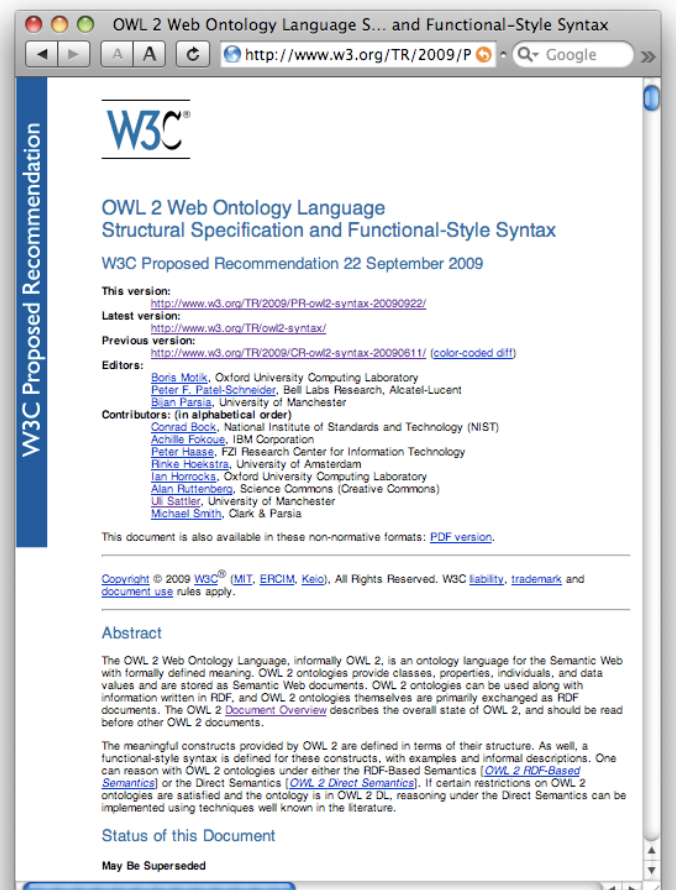
OWL 2 Web Ontology Language
Structural Specification and Functional-Style Syntax

W3C Recommendation 27 October 2009



Overview | Structural Spec. | Profiles Spec. | Primer | Quick Reference

OWL Key Features I


OWL 2 Web Ontology Language S... and Functional-Style Syntax

http://www.w3.org/TR/2009/P

W3C Proposed Recommendation

OWL 2 Web Ontology Language
Structural Specification and Functional-Style Syntax

W3C Proposed Recommendation 22 September 2009

This version:
http://www.w3.org/TR/2009/PR-owl2-syntax-20090922/
Latest version:
http://www.w3.org/TR/owl2-syntax/
Previous version:
http://www.w3.org/TR/2009/CR-owl2-syntax-20090611/ (color-coded diff)
Editors:
Boris Motik, Oxford University Computing Laboratory
Peter F. Patel-Schneider, Bell Labs Research, Alcatel-Lucent
Bijan Parsia, University of Manchester
Contributors: (in alphabetical order)
Conrad Bock, National Institute of Standards and Technology (NIST)
Achille Fokoue, IBM Corporation
Peter Haase, FZI Research Center for Information Technology
Rinke Hoekstra, University of Amsterdam
Ian Horrocks, Oxford University Computing Laboratory
Alan Ruttenberg, Science Commons (Creative Commons)
Uli Sattler, University of Manchester
Michael Smith, Clark & Parsia

This document is also available in these non-normative formats: PDF version.

Copyright © 2009 W3C® (MIT, ERCIM, Keio), All Rights Reserved. W3C liability, trademark and document use rules apply.

Abstract

The OWL 2 Web Ontology Language, informally OWL 2, is an ontology language for the Semantic Web with formally defined meaning. OWL 2 ontologies provide classes, properties, individuals, and data values and are stored as Semantic Web documents. OWL 2 ontologies can be used along with information written in RDF, and OWL 2 ontologies themselves are primarily exchanged as RDF documents. The OWL 2 Document Overview describes the overall state of OWL 2, and should be read before other OWL 2 documents.

The meaningful constructs provided by OWL 2 are defined in terms of their structure. As well, a functional-style syntax is defined for these constructs, with examples and informal descriptions. One can reason with OWL 2 ontologies under either the RDF-Based Semantics [OWL 2 RDF-Based Semantics] or the Direct Semantics [OWL 2 Direct Semantics]. If certain restrictions on OWL 2 ontologies are satisfied and the ontology is in OWL 2 DL, reasoning under the Direct Semantics can be implemented using techniques well known in the literature.

Status of this Document

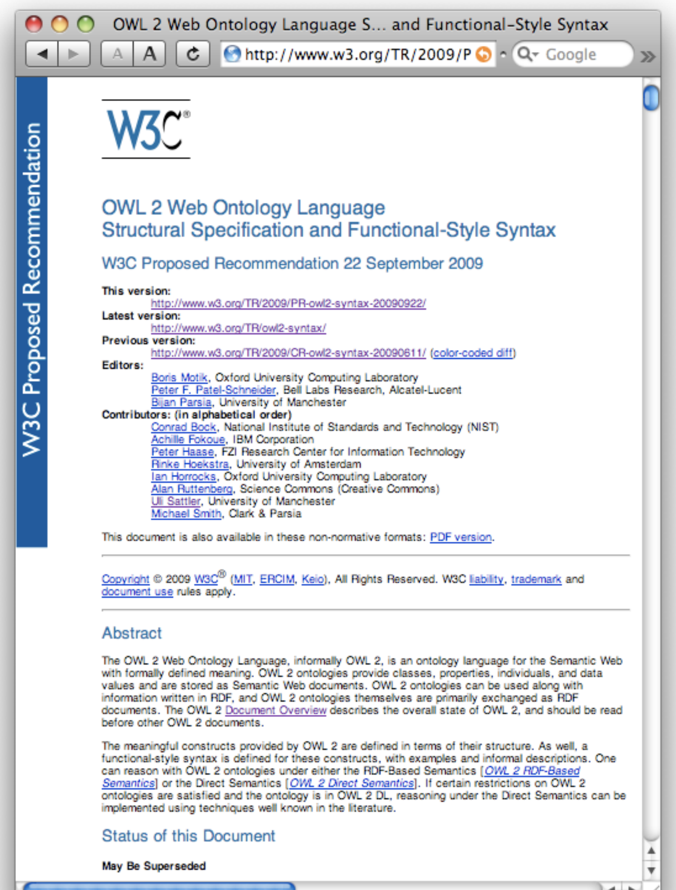
May Be Superseded

A standardised language for writing logic based ontologies

It provides an "exchange syntax" for sharing ontologies

It provides a rich set of modelling constructors for describing things

OWL Key Features II


OWL 2 Web Ontology Language S... and Functional-Style Syntax

http://www.w3.org/TR/2009/P

W3C Proposed Recommendation

W3C®

OWL 2 Web Ontology Language
Structural Specification and Functional-Style Syntax

W3C Proposed Recommendation 22 September 2009

This version:
http://www.w3.org/TR/2009/PR-owl2-syntax-20090922/
Latest version:
http://www.w3.org/TR/owl2-syntax/
Previous version:
http://www.w3.org/TR/2009/CR-owl2-syntax-20090611/ (color-coded diff)
Editors:
Boris Motik, Oxford University Computing Laboratory
Peter F. Patel-Schneider, Bell Labs Research, Alcatel-Lucent
Bijan Parsia, University of Manchester
Contributors: (in alphabetical order)
Conrad Bock, National Institute of Standards and Technology (NIST)
Achille Fokoue, IBM Corporation
Peter Haase, FZI Research Center for Information Technology
Rinke Hoekstra, University of Amsterdam
Ian Horrocks, Oxford University Computing Laboratory
Alan Ruttenberg, Science Commons (Creative Commons)
Uli Sattler, University of Manchester
Michael Smith, Clark & Parsia

This document is also available in these non-normative formats: PDF version.

Copyright © 2009 W3C® (MIT, ERCIM, Keio), All Rights Reserved. W3C liability, trademark and document use rules apply.

Abstract

The OWL 2 Web Ontology Language, informally OWL 2, is an ontology language for the Semantic Web with formally defined meaning. OWL 2 ontologies provide classes, properties, individuals, and data values and are stored as Semantic Web documents. OWL 2 ontologies can be used along with information written in RDF, and OWL 2 ontologies themselves are primarily exchanged as RDF documents. The OWL 2 Document Overview describes the overall state of OWL 2, and should be read before other OWL 2 documents.

The meaningful constructs provided by OWL 2 are defined in terms of their structure. As well, a functional-style syntax is defined for these constructs, with examples and informal descriptions. One can reason with OWL 2 ontologies under either the RDF-Based Semantics [*OWL 2 RDF-Based Semantics*] or the Direct Semantics [*OWL 2 Direct Semantics*]. If certain restrictions on OWL 2 ontologies are satisfied and the ontology is in OWL 2 DL, reasoning under the Direct Semantics can be implemented using techniques well known in the literature.

Status of this Document

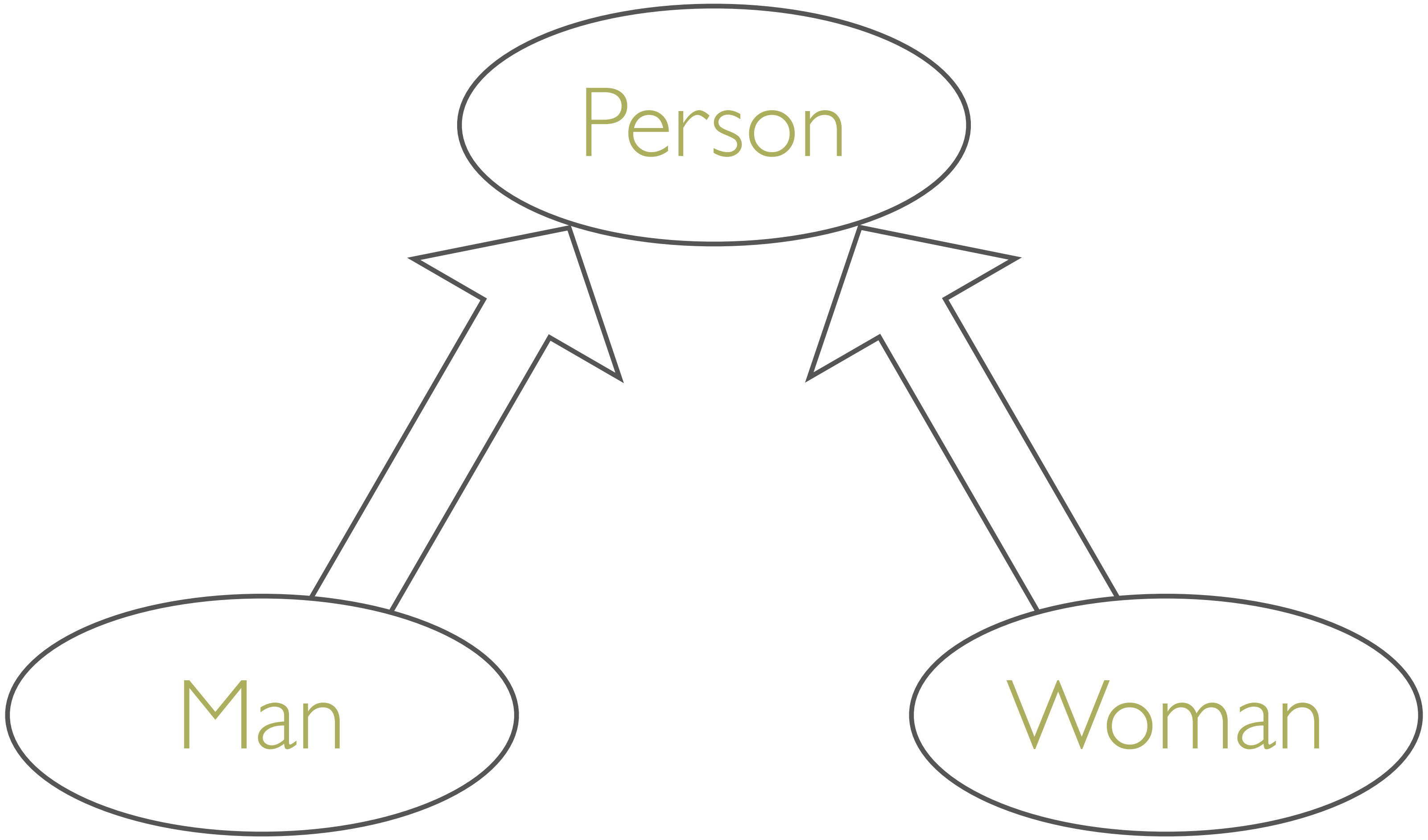
May Be Superseded

It has a precisely defined semantics

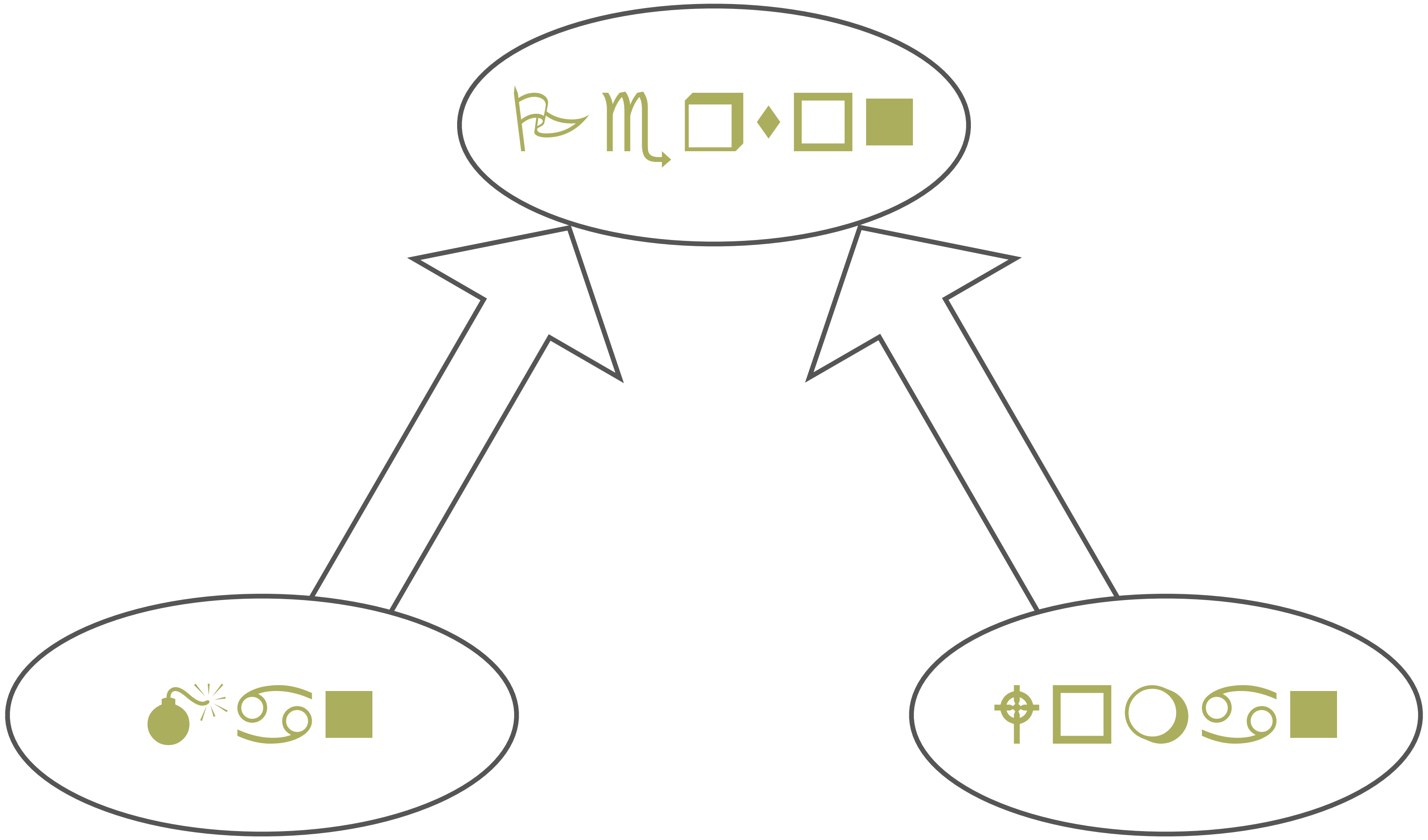
Every statement made in an OWL ontology has an unambiguous meaning

OWL ontologies are amenable to automated reasoning

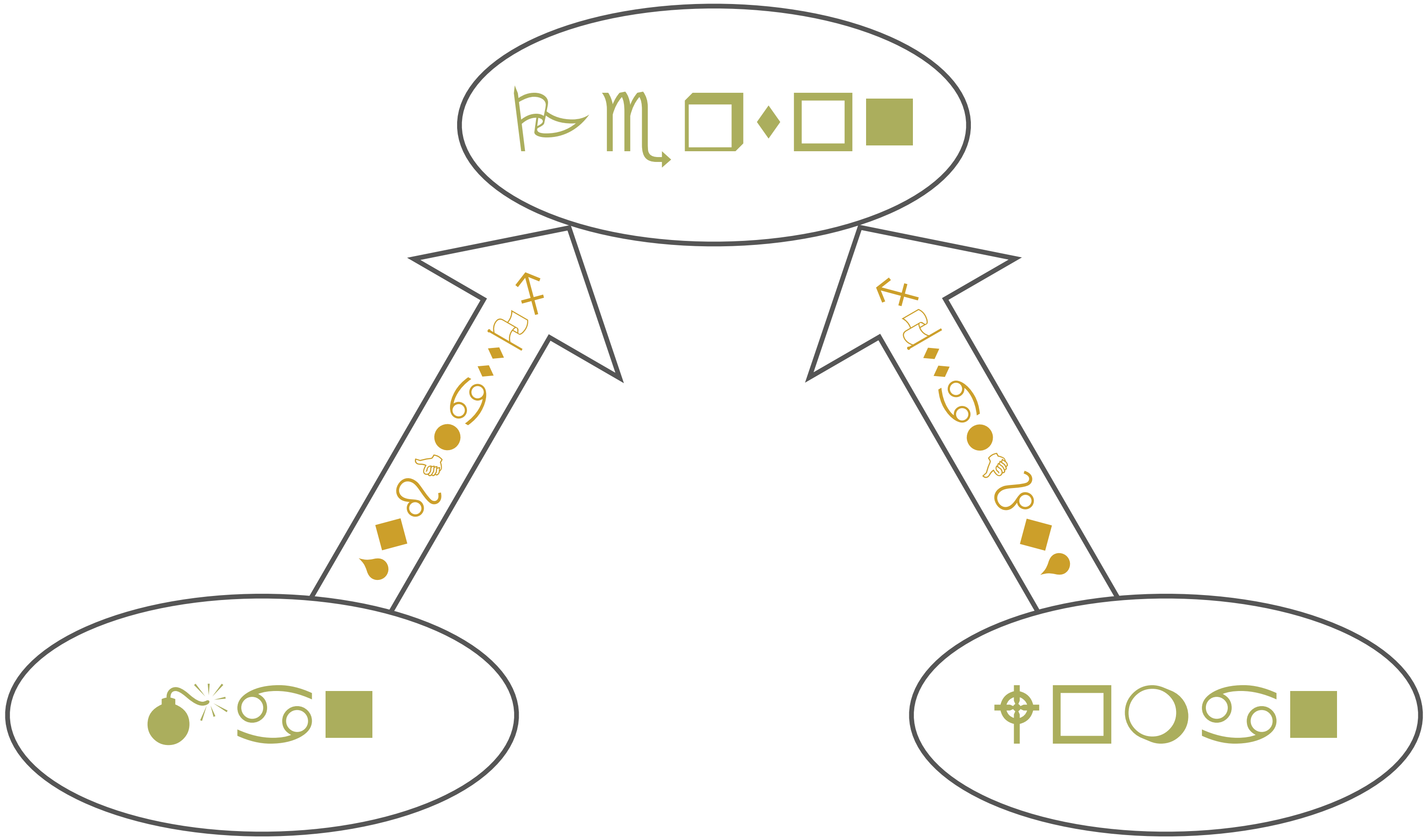
Semantics



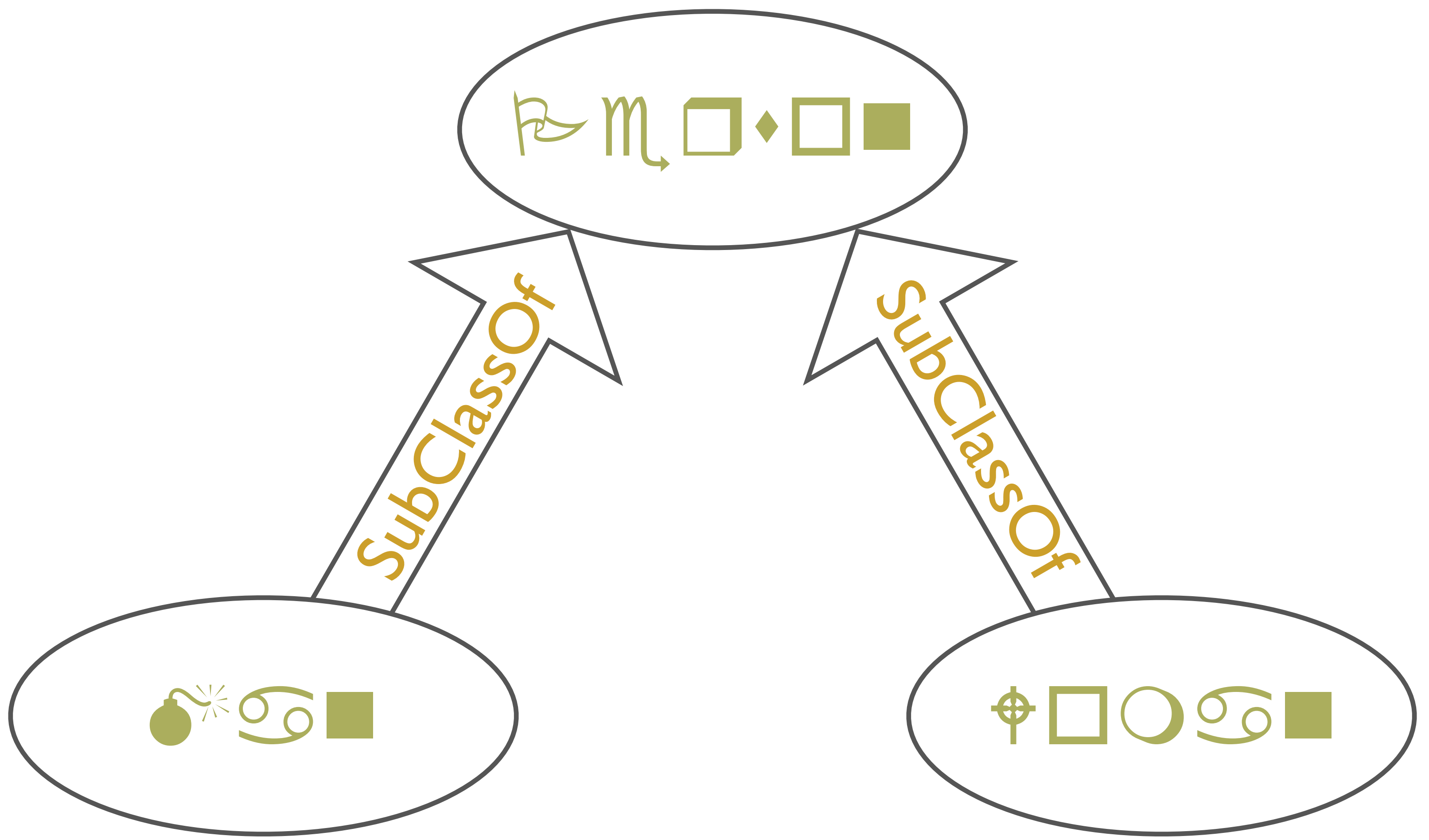
Semantics



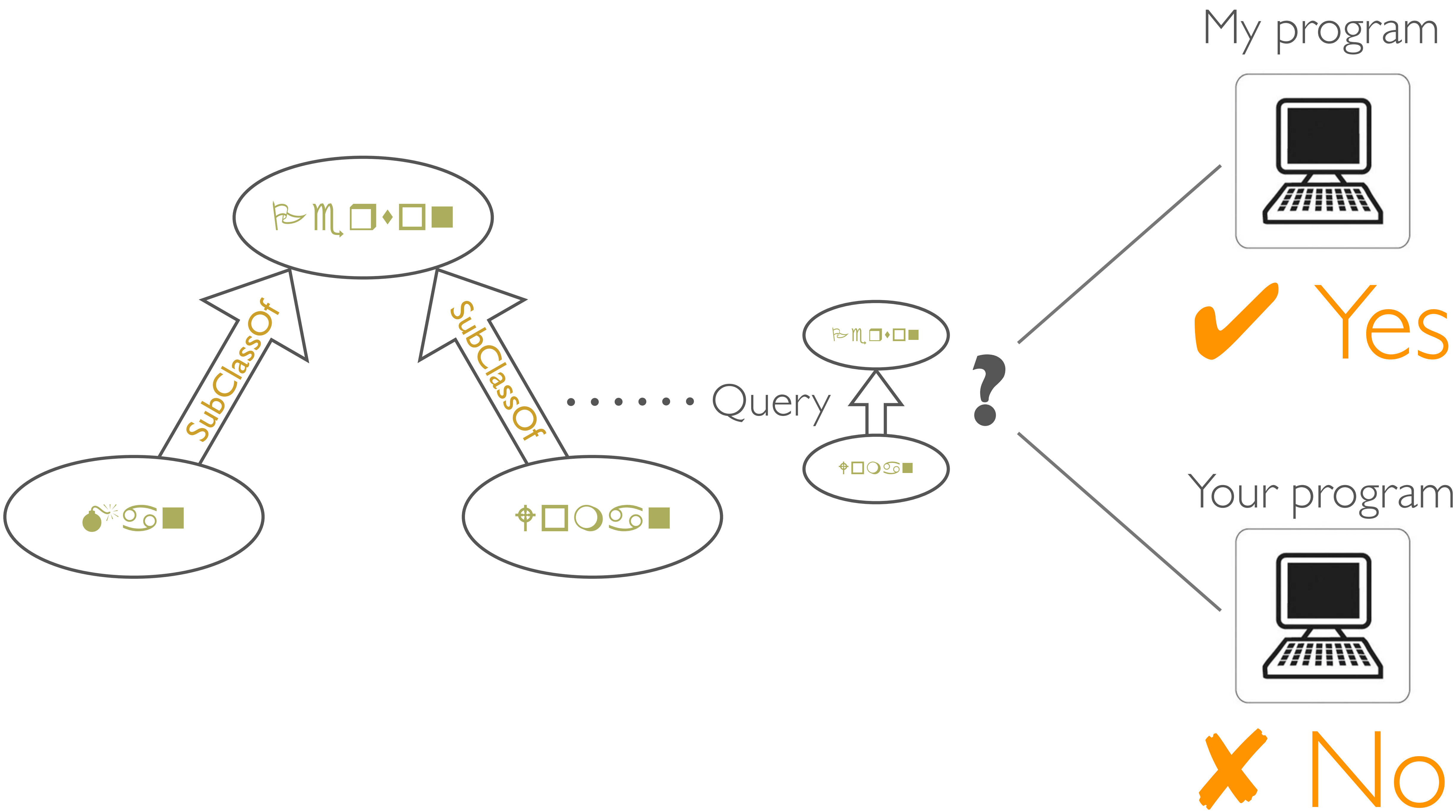
Semantics



Semantics

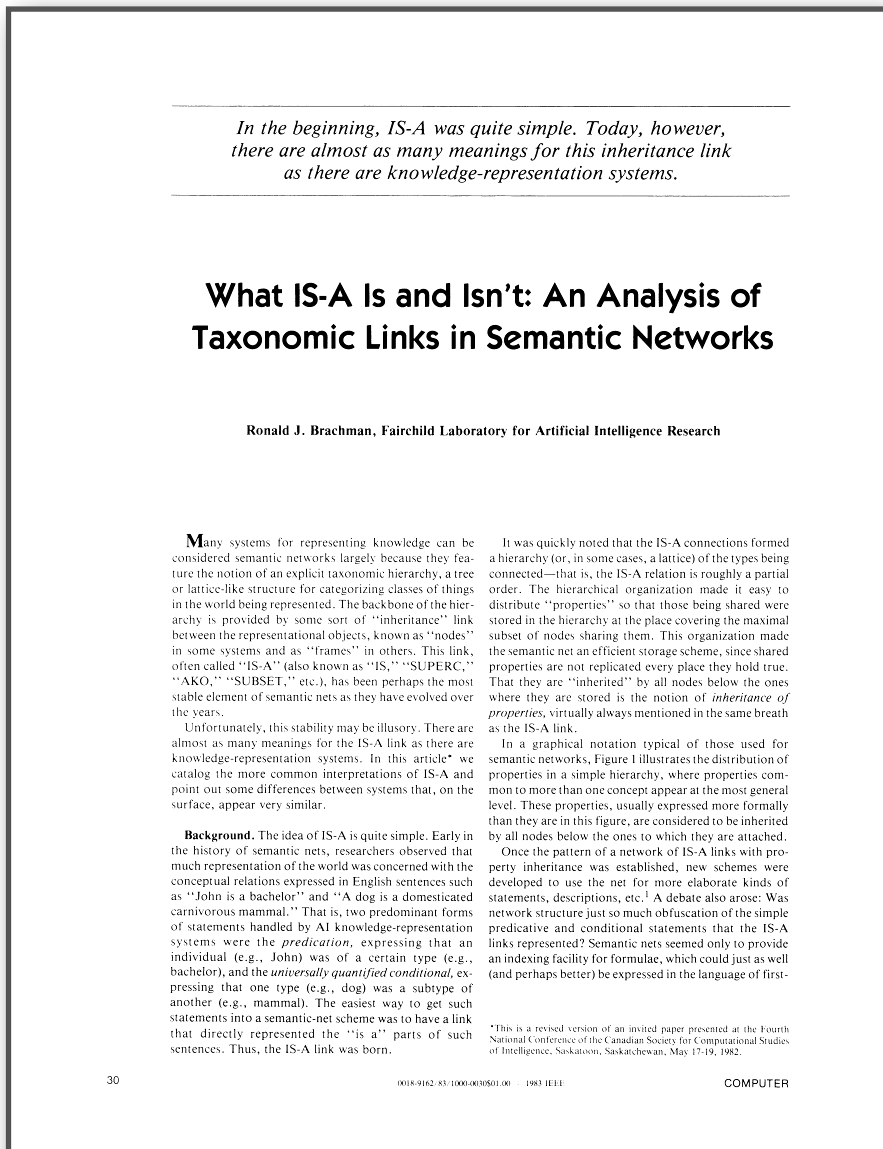


Semantics



SubClassOf
SubClassOf
Query
?
My program
✔ Yes
Your program
✘ No

What IS-A Is and Isn't



In the beginning, IS-A was quite simple. Today, however, there are almost as many meanings for this inheritance link as there are knowledge-representation systems.

What IS-A Is and Isn't: An Analysis of Taxonomic Links in Semantic Networks

Ronald J. Brachman, Fairchild Laboratory for Artificial Intelligence Research

Many systems for representing knowledge can be considered semantic networks largely because they feature the notion of an explicit taxonomic hierarchy, a tree or lattice-like structure for categorizing classes of things in the world being represented. The backbone of the hierarchy is provided by some sort of "inheritance" link between the representational objects, known as "nodes" in some systems and as "frames" in others. This link, often called "IS-A" (also known as "IS," "SUPERC," "AKO," "SUBSET," etc.), has been perhaps the most stable element of semantic nets as they have evolved over the years.

Unfortunately, this stability may be illusory. There are almost as many meanings for the IS-A link as there are knowledge-representation systems. In this article* we catalog the more common interpretations of IS-A and point out some differences between systems that, on the surface, appear very similar.

Background. The idea of IS-A is quite simple. Early in the history of semantic nets, researchers observed that much representation of the world was concerned with the conceptual relations expressed in English sentences such as "John is a bachelor" and "A dog is a domesticated carnivorous mammal." That is, two predominant forms of statements handled by AI knowledge-representation systems were the *predication,* expressing that an individual (e.g., John) was of a certain type (e.g., bachelor), and the *universally quantified conditional,* expressing that one type (e.g., dog) was a subtype of another (e.g., mammal). The easiest way to get such statements into a semantic-net scheme was to have a link that directly represented the "is a" parts of such sentences. Thus, the IS-A link was born.

It was quickly noted that the IS-A connections formed a hierarchy (or, in some cases, a lattice) of the types being connected—that is, the IS-A relation is roughly a partial order. The hierarchical organization made it easy to distribute "properties" so that those being shared were stored in the hierarchy at the place covering the maximal subset of nodes sharing them. This organization made the semantic net an efficient storage scheme, since shared properties are not replicated every place they hold true. That they are "inherited" by all nodes below the ones where they are stored is the notion of *inheritance of properties,* virtually always mentioned in the same breath as the IS-A link.

In a graphical notation typical of those used for semantic networks, Figure 1 illustrates the distribution of properties in a simple hierarchy, where properties common to more than one concept appear at the most general level. These properties, usually expressed more formally than they are in this figure, are considered to be inherited by all nodes below the ones to which they are attached.

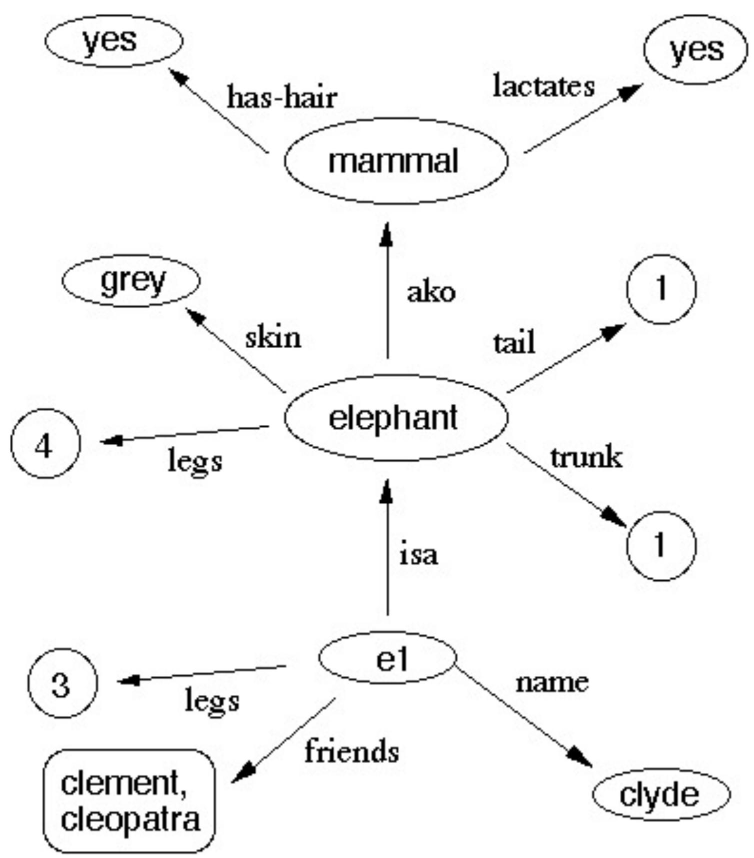
Once the pattern of a network of IS-A links with property inheritance was established, new schemes were developed to use the net for more elaborate kinds of statements, descriptions, etc.[1] A debate also arose: Was network structure just so much obfuscation of the simple predicative and conditional statements that the IS-A links represented? Semantic nets seemed only to provide an indexing facility for formulae, which could just as well (and perhaps better) be expressed in the language of first-

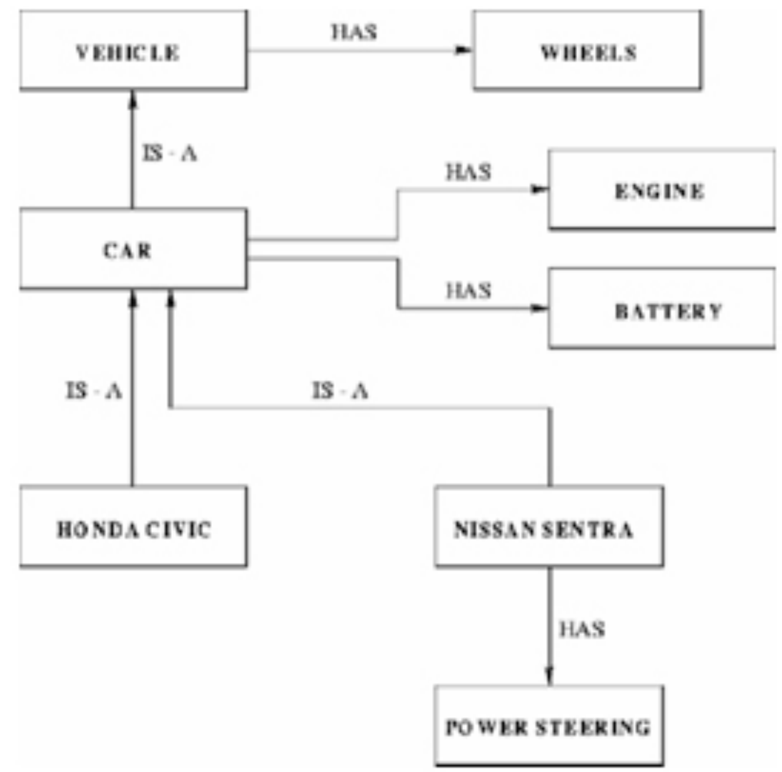
*This is a revised version of an invited paper presented at the Fourth National Conference of the Canadian Society for Computational Studies of Intelligence, Saskatoon, Saskatchewan, May 17-19, 1982.

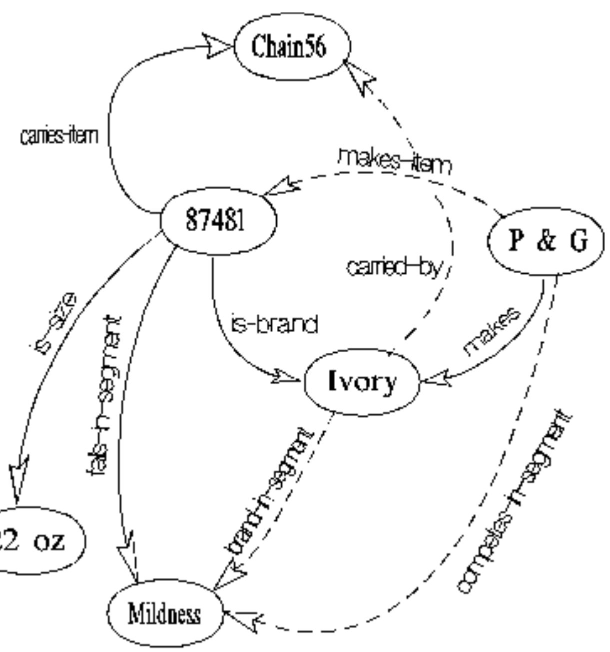
0018-9162/83/1000-0030$01.00 © 1983 IEEE

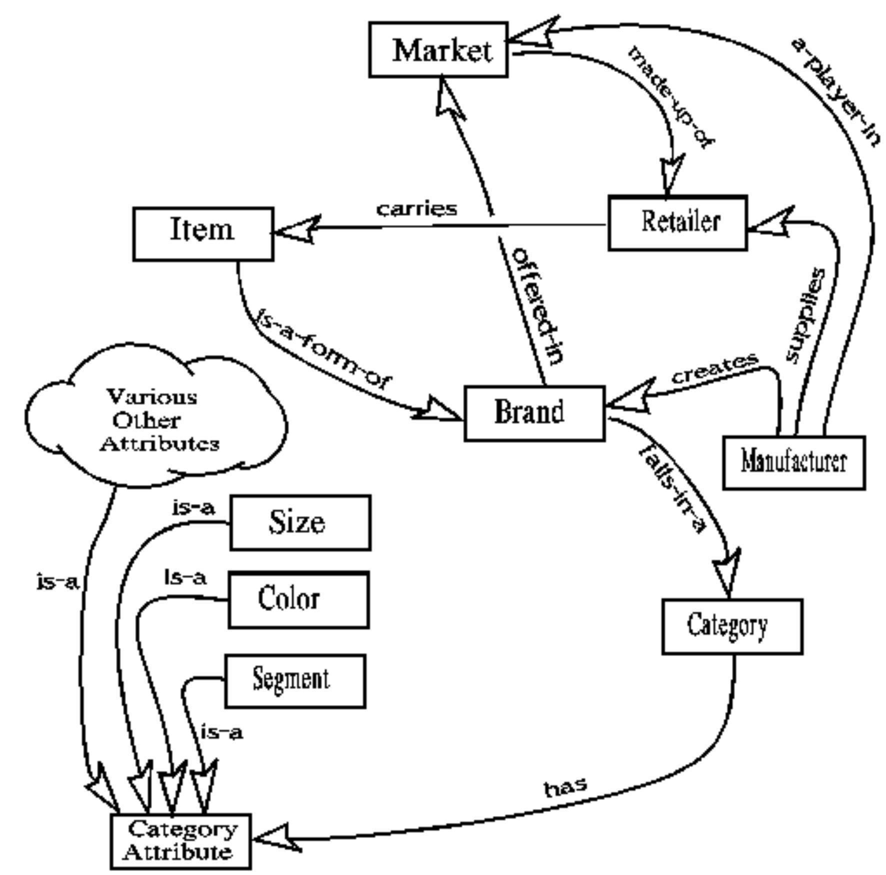
In the beginning, IS-A was quite simple. Today, however, there are almost as many meanings for this inheritance link as there are knowledge-representation systems.

Ronald J. Brachman - In Computer, October 1983


yes
has-hair
mammal
lactates
yes
grey
skin
ako
tail
1
4
legs
elephant
trunk
1
isa
3
legs
e1
name
friends
clement, cleopatra
clyde


VEHICLE
HAS
WHEELS
IS - A
HAS
ENGINE
CAR
HAS
BATTERY
IS - A
IS - A
HONDA CIVIC
NISSAN SENTRA
HAS
POWER STEERING


Chain56
carries-item
makes-item
87481
carried-by
P & G
is-size
falls-in-segment
is-brand
makes
Ivory
brand-in-segment
competes-in-segment
22 oz
Mildness


Market
made-up-of
a-player-in
Item
carries
Retailer
offered-in
supplies
is-a-form-of
creates
Various Other Attributes
Brand
Manufacturer
falls-in-a
is-a
Size
is-a
Is-a
Color
Segment
is-a
Category
has
Category Attribute

OWL to the rescue...

Basic Terminology

Axioms	Basic statements in an ontology. **An ontology is a set of axioms**
Entities	Used to refer to basic things in the domain of interest
Class Expressions	Combinations of entities that form more complex descriptions out of simpler ones

Axioms specify the **relationships** between entities and class expressions

Axioms

(statements)

Some examples...

Cat SubClassOf Animal	SubClassOf Cats are Animals
Cat DisjointWith Dog	DisjointClasses Cats are not Dogs
Tibbs Type Cat	ClassAssertion Tibbs is a Cat
Betty hasPet Tibbs	PropertyAssertion Betty has Tibbs as a pet
hasPet Domain Person	Domain Anything that has a pet is Person

Entities

(basic things in the domain)

Cat SubClassOf Animal

Cat DisjointWith Dog

Tibbs Type Cat

Betty hasPet Tibbs

hasPet Domain Person

Entities

(basic things in the domain)

Cat
Animal
Dog
Person

Classes
(Concepts)

Properties
(Roles, Slots)

hasPet

Individuals
(Instances)

Tibbs
Betty

Datatypes
(Concrete Domains)

Class Expressions

(complex "descriptions" built up from simpler ones)

Some examples...

Cat or Dog

The class of individuals that instances of Cat or Dog (or both!)

Person and PetOwner

The class of individuals that are both instances of Person and PetOwner

hasPet some Cat

The class of individuals that have at least one hasPet relationship to an individual that is an instance of Cat

Person and hasPet some Cat

The class of individuals that are both instances of Person and hasPet some Cat

Person and not (hasPet some (Cat or Dog))

The class of individuals that are instances of Person but not instances of the class of individuals that have at least one hasPet relationship to and individual that is an instance of the class Cat or Dog

Class Expressions

(complex "descriptions" built up from simpler ones)

In any position where we can use a class name we can use a class expression

Person and PetOwner

Person and (hasPet some Cat)

Person and (hasPet some (Cat or Dog))

We can nest class expressions to arbitrary depths

Class Expressions

Like entities, class expressions then get used to build axioms
which describe our domain classes in more detail

```
PetOwer SubClassOf Person and (hasPet some Animal)
```

Entailment

(consequences)

OWL has a precisely defined notion of **entailment**

We can unambiguously determine whether or not
an axiom (or set of axioms) follows as as consequence of what we've stated in our ontology

Entailment

(consequences)

Ontology		Example entailments
Dog SubClassOf Animal	⇒	Dalmatian SubClassOf Animal
Dalmatian SubClassOf Dog		Patch Type Dog
Patch Type Dalmatian		hasPet some Dog SubClassOf Person
Pete hasPet Patch		Pete Type Person
hasPet Domain Person		Pete Type hasPet some Dog
		Dog SubClassOf Animal
		Dalmatian SubClassOf Dog
		⋮

Some Terminology

Asserted Axioms

Axioms that have been explicitly stated (or "written down") in an ontology

Entailed Axioms
(or simply **entailments**)

Axioms that follow as a consequence of what has been stated.
Note: Asserted axioms are (trivially) entailed axioms!

Inferred Axioms

A synonym of entailed axioms.
(Can mean axioms which have been computed to be entailed via reasoning)

Reasoning

The process used to compute whether or not an axiom is entailed by an ontology

Automated Reasoning Reasoning performed by a computer

Reasoner A program that performs automated reasoning

"Off the shelf" OWL reasoners can be plugged into Protege

Automated Reasoning

Design-Time reasoning

For ontology construction

We can check our ontology for logical bugs and to ensure it means what we think it means

Run-Time reasoning

For application querying

We can make sure that we get the correct answers, and the same answers as everyone else

OWL 2 Profiles

Designed to supporting highly scalable reasoning

OWL 2 EL

Efficient polynomial time reasoning
Designed for very large (biomedical) ontologies.
e.g. SNOMED. Highly scalable.

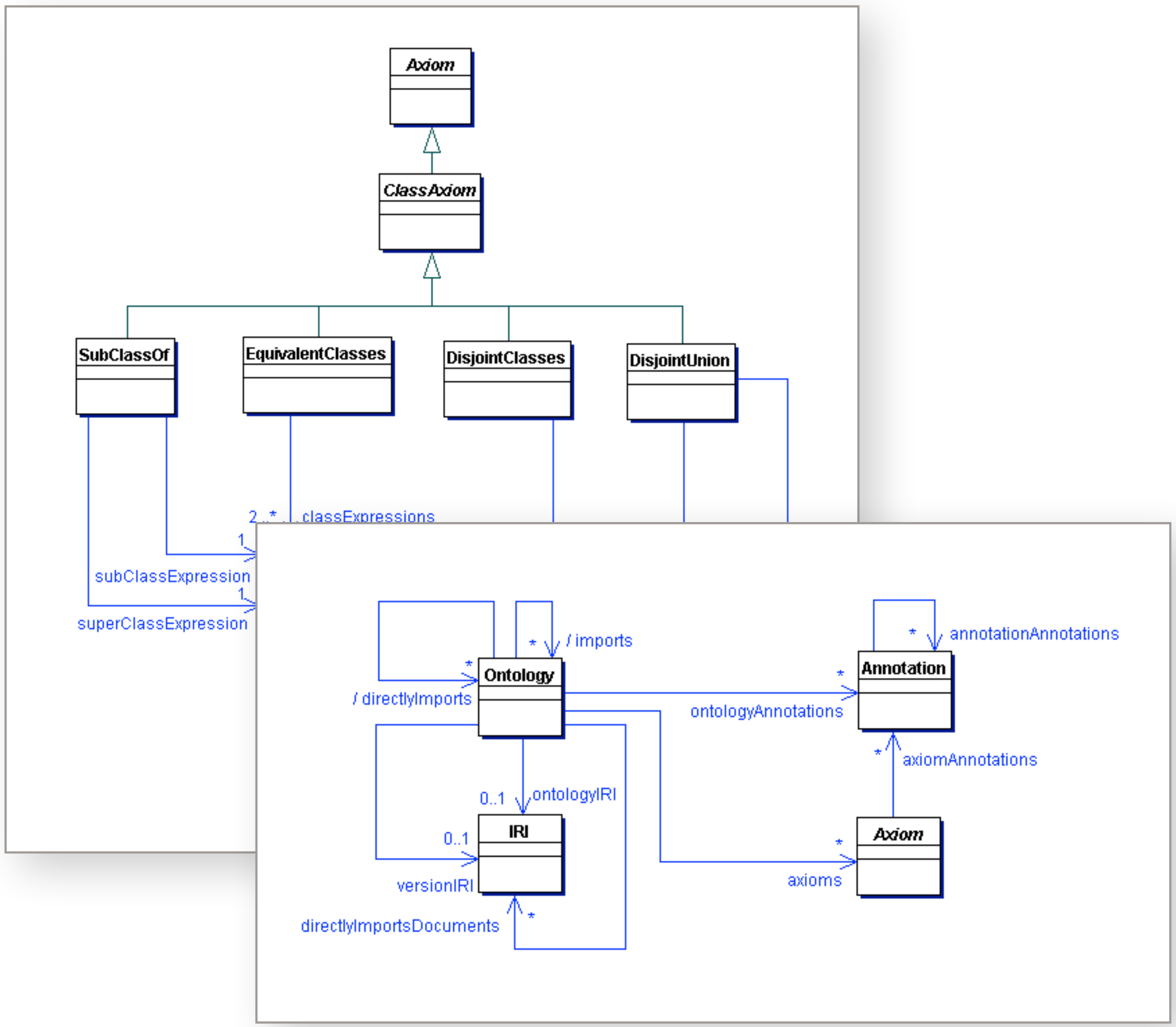
OWL 2 QL

Query data (held in a database) through an ontology. Sound and complete query answering is very low complexity w.r.t. size of data. (Rewrite query into multiple SQL queries).

OWL 2 RL

Sound and complete reasoning using rule based technologies (for certain kinds of entailments).

Syntaxes



The structure of an OWL ontology is specified at an abstract level, using UML style diagrams, that is independent of any particular concrete syntax

An OWL ontology can be serialised in a variety of alternative exchange syntaxes

Syntaxes

RDF/XML

The official (normative) exchange syntax

All OWL 2 tools must support this syntax

Manchester Syntax

Designed to be easy to read and write by hand

This is the syntax used in Protege

Turtle

An RDF based syntax

OBO

A readable flat text file format
(for biomedical ontologies)

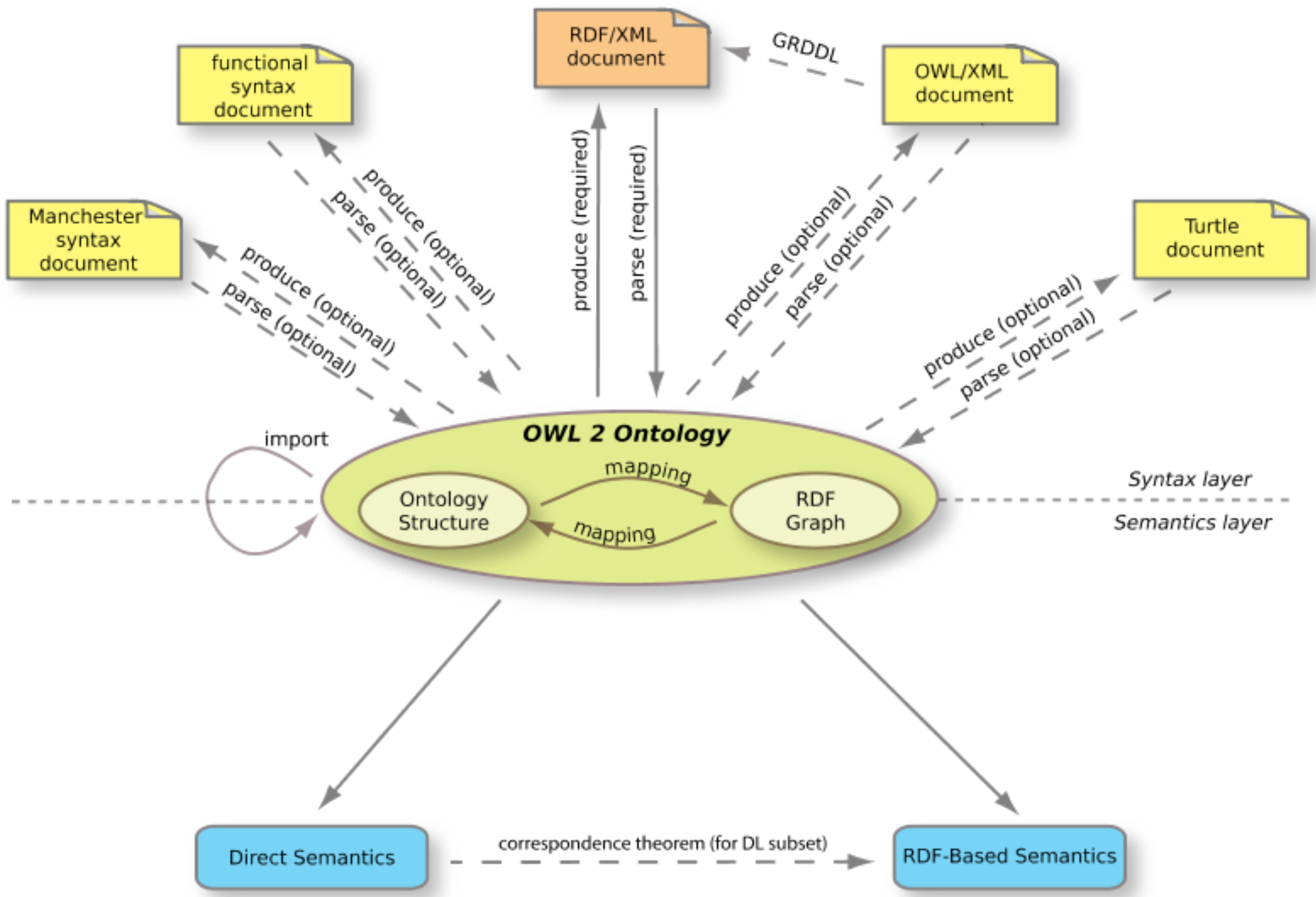
OWL/XML

An XML format defined by
an XML Schema.
(Works well with XML tool chains)

Functional Syntax

An abstract syntax for structurally
specifying OWL 2

Syntax



Curtesy OWL 2 Web Ontology Language Document Overview

RDF

The Resource Description Framework

For describing properties of resources on the web

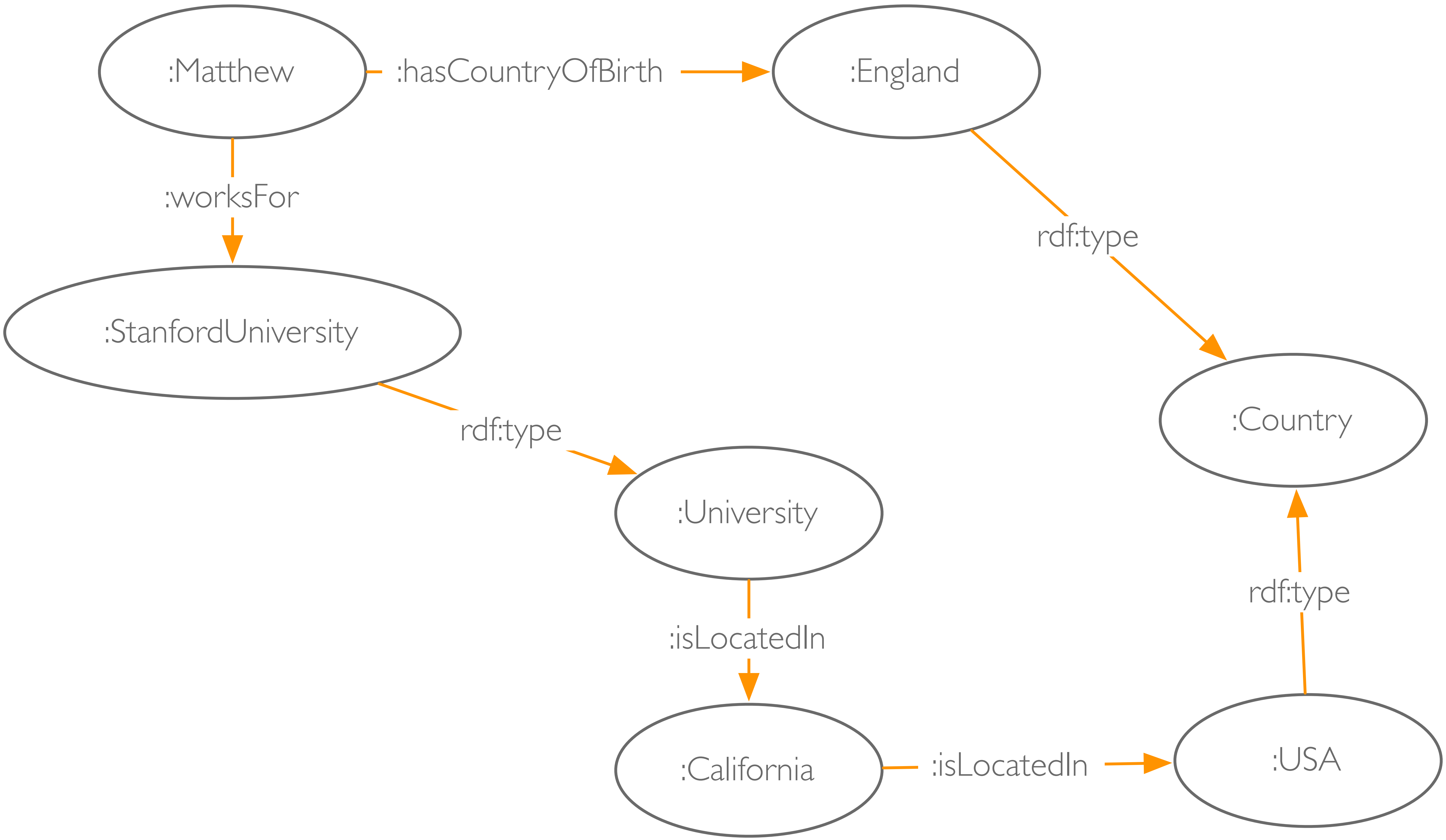
An RDF Graph is a set of statements - Triples

Subject - Predicate - Object

Graph Example

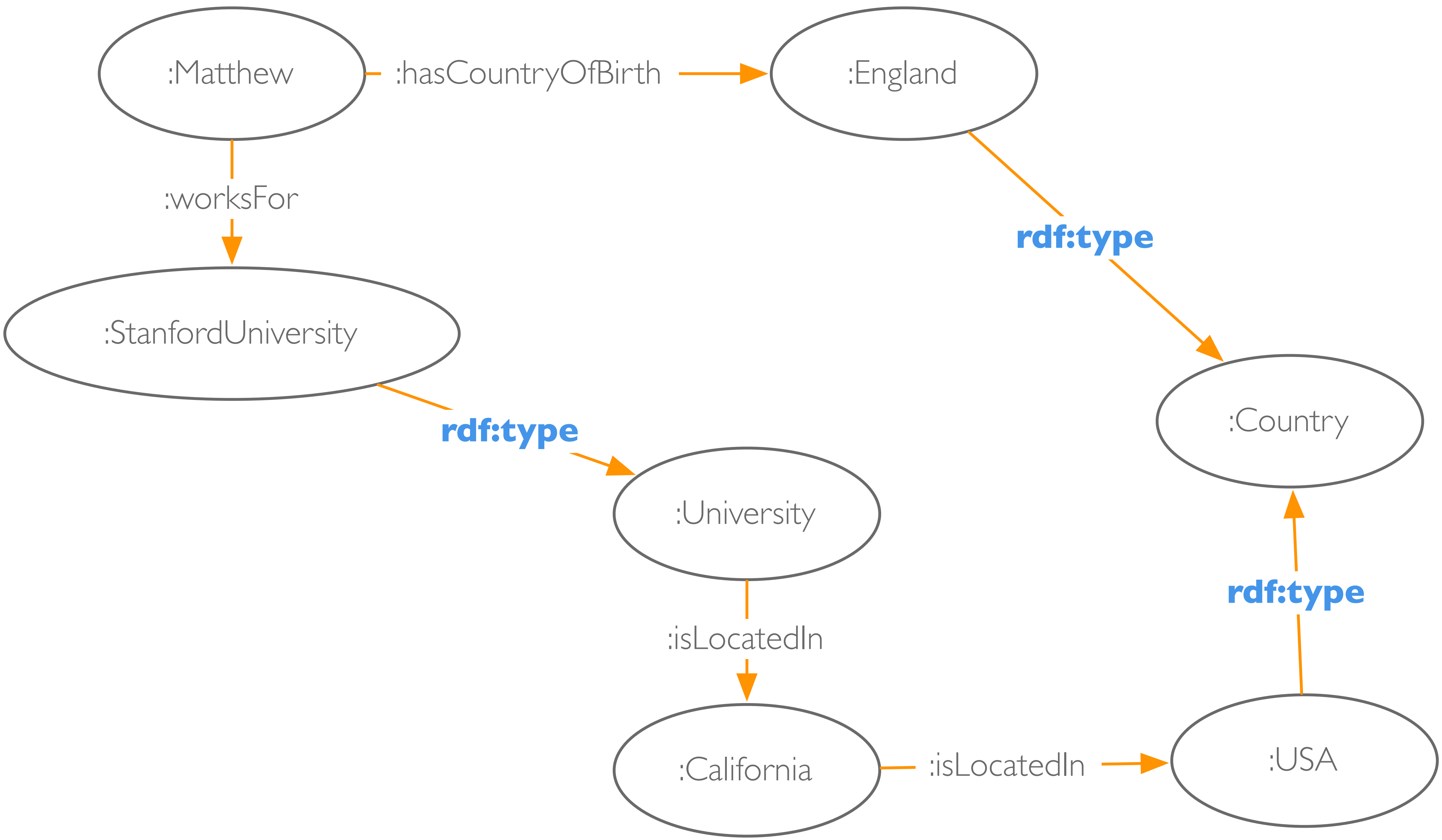
:Matthew	:hasCountryOfBirth	:England
:England	rdf:type	:Country
:Matthew	:worksFor	:StanfordUniversity
:StanfordUniversity	rdf:type	:University
:StanfordUniversity	:locatedIn	:California
:California	:locatedIn	:USA

Graph Example



rdf:type

(built in vocabulary)



rdf:type

(built in vocabulary)

:Matthew	:hasCountryOfBirth	:England
:England	**rdf:type**	:Country
:Matthew	:worksFor	:StanfordUniversity
:StanfordUniversity	**rdf:type**	:University
:StanfordUniversity	:locatedIn	:California
:California	:locatedIn	:USA

Summary

OWL is a logic based ontology language

It has a precisely defined semantics

The notion of entailment is well defined

Automated reasoning can be used to compute entailments