Aalto University
School of Science

SeCo

Web Ontology Language OWL
Background and Context

Eero Hyvönen
Aalto University, Department of Computer Science
University of Helsinki, HELDIG-centre
Semantic Computing Research Group (SeCo), http://seco.cs.aalto.fi
eero.hyvonen@aalto.fi

Learning objective

- Learn Web Ontology Language OWL

Contents

- OWL: background and context
- The OWL language

OWL: Background and Context

Component Technologies and Tools for the Semantic Web

Languages

- *Data exchange language:*	*RDF*
- *Vocabulary/schema languages:*	*SKOS, OWL*
- *Data/ontology query language:*	*SPARQL*
- *Rules for reasoning:*	*RIF, SWRL, SPIN, …*

Storages and querying

- Triplestore systems (Fuseki, Sesame, Redland, Virtuoso, …)
  - *http://en.wikipedia.org/wiki/Triplestore*

Development tools

- Ontology editors
  - *Protégé https://protege.stanford.edu/*
  - *TopBraid Composer https://www.topquadrant.com/topbraid-composer-install/*
- Software development tools
  - *Java: Apache Jena https://jena.apache.org/*
  - *Python: RDFLib https://pypi.org/project/rdflib/*

What is OWL

- Standard language (W3C) for representing vocabularies/ontologies/schemas
- Much richer than RDF Schema and SKOS (W3C recommendations for light-weight vocabularies)
- Original OWL
  - *Published as W3C recommendation on 10.2.2004*
- OWL 2 = OWL
  - *Latest W3C recommendation on 11.12.2012*
  - *Extends and replaces the old recommendation*

Requirements for Ontology Languages

- **Syntax**, convenience for expressing knowledge
- Formal **semantics**
- Sufficient **expressive** power
- Efficient **reasoning** support

OWL Syntaxes

- Based on RDF(S)
  - *Turtle and RDF/XML*
- Specific OWL/XML schema
- More user-friendly notations
  - *Functional-style syntax* *(for specifications)*
  - *Manchester syntax* *(simple notation for non-logicians)*

Formal Semantics

- Defines precise meaning for syntactic expressions
- As with RDF Schema two alternative semantics are provided
  - *Direct semantics (formal logic formulation)*
  - *RDF-based semantics (rules for inferring new triples)*
- Basis for interpreting expressions in reasoning:

```
:p rdfs:range :D .
:x :p :y .
⇒:y rdf:type :D .
```

- Goal: OWL can be used without being a formal logician

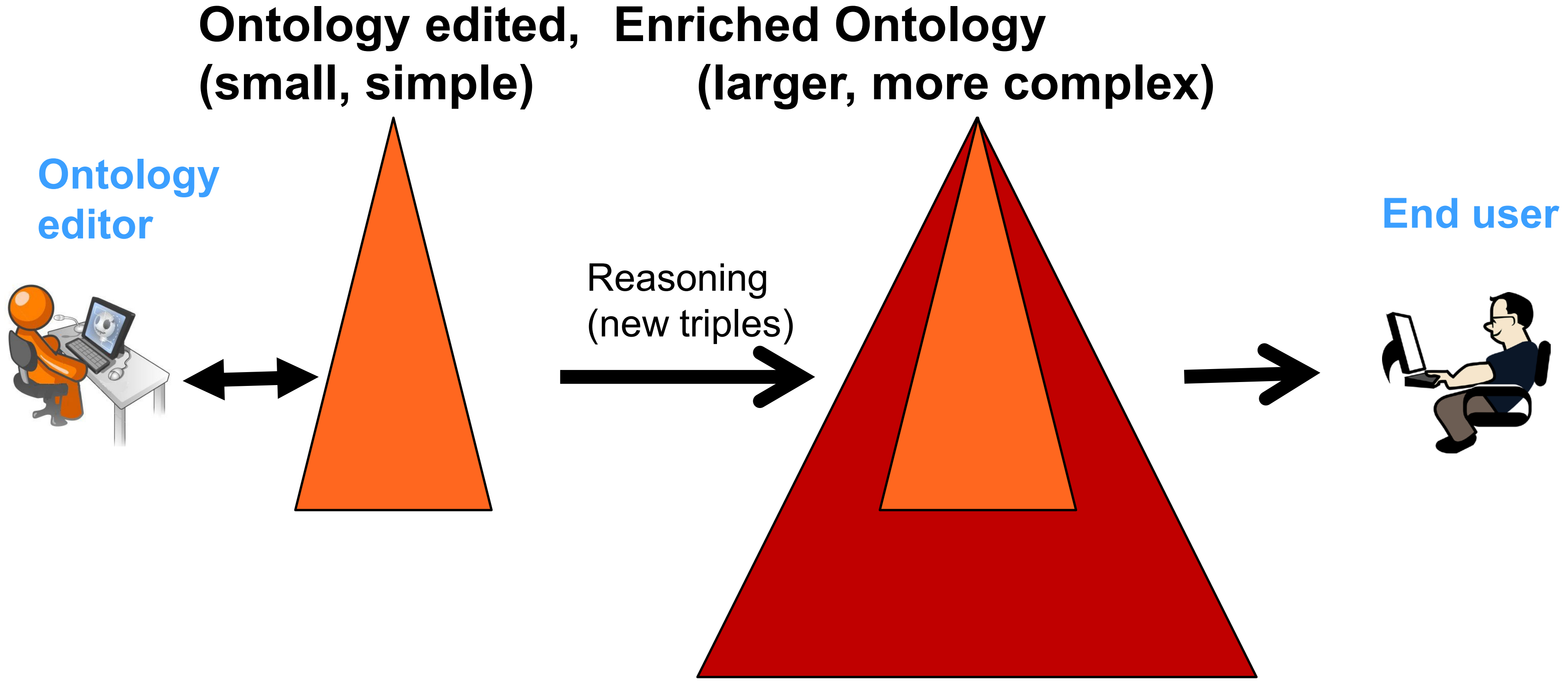
Sufficient Expressive Power

- RDF(S) semantics is limited
  - *Class membership (instance-class relations)*
  - *Class and property hierarchies*
  - *Domain and range of properties*
- OWL introduces lots of new features, e.g.
  - *Equivalence (classes) & Equality (individuals)*
  - *Disjointness (classes) & Difference (individuals)*
  - *Boolean combination of classes*
  - *Local (class-wise) scope of properties*
  - *Special relational characteristics of properties*
  - *Cardinality of properties*

Reasoning Support

- Enriching ontology and metadata with new facts
- Checking consistency
- Finding unintended relations
- Tradeoff between expressive power and efficient reasoning
    - Different versions “profiles” of OWL are available

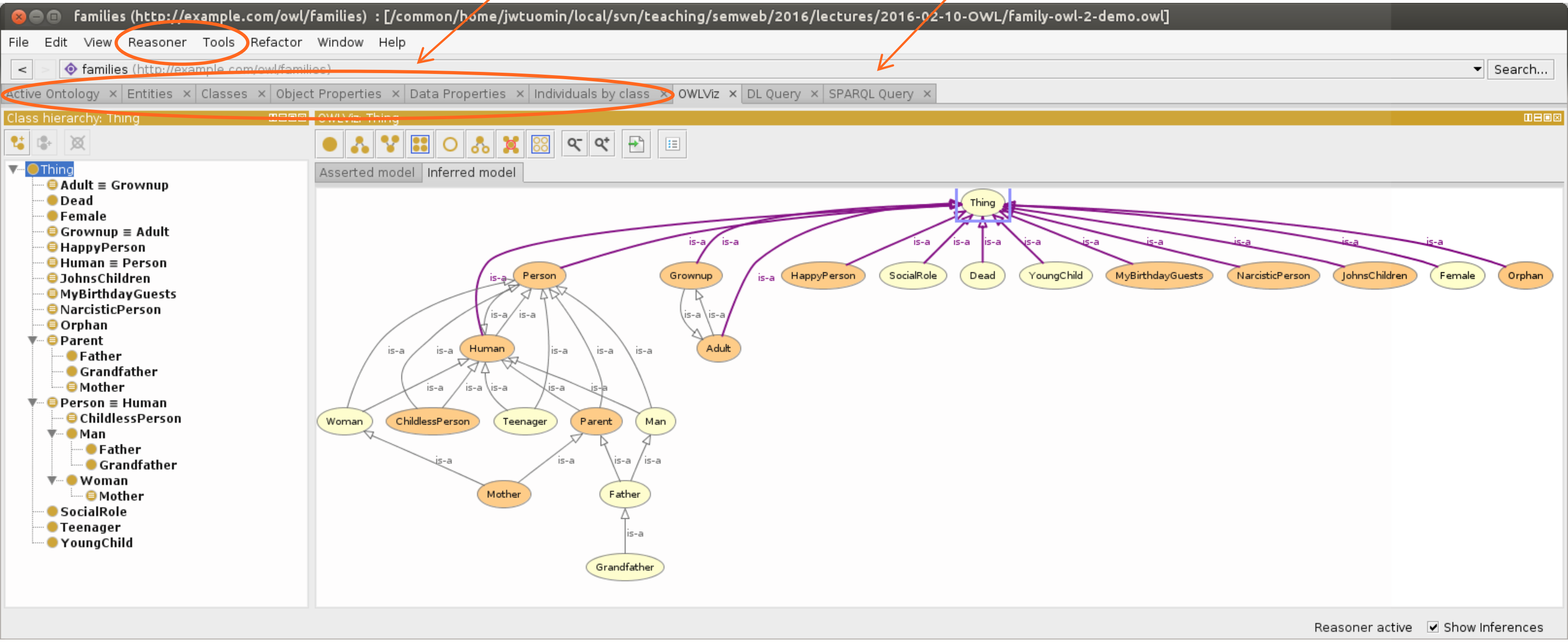
Enriching Ontology by Reasoning



Protégé Editor for Editing Ontologies

OWL constructs

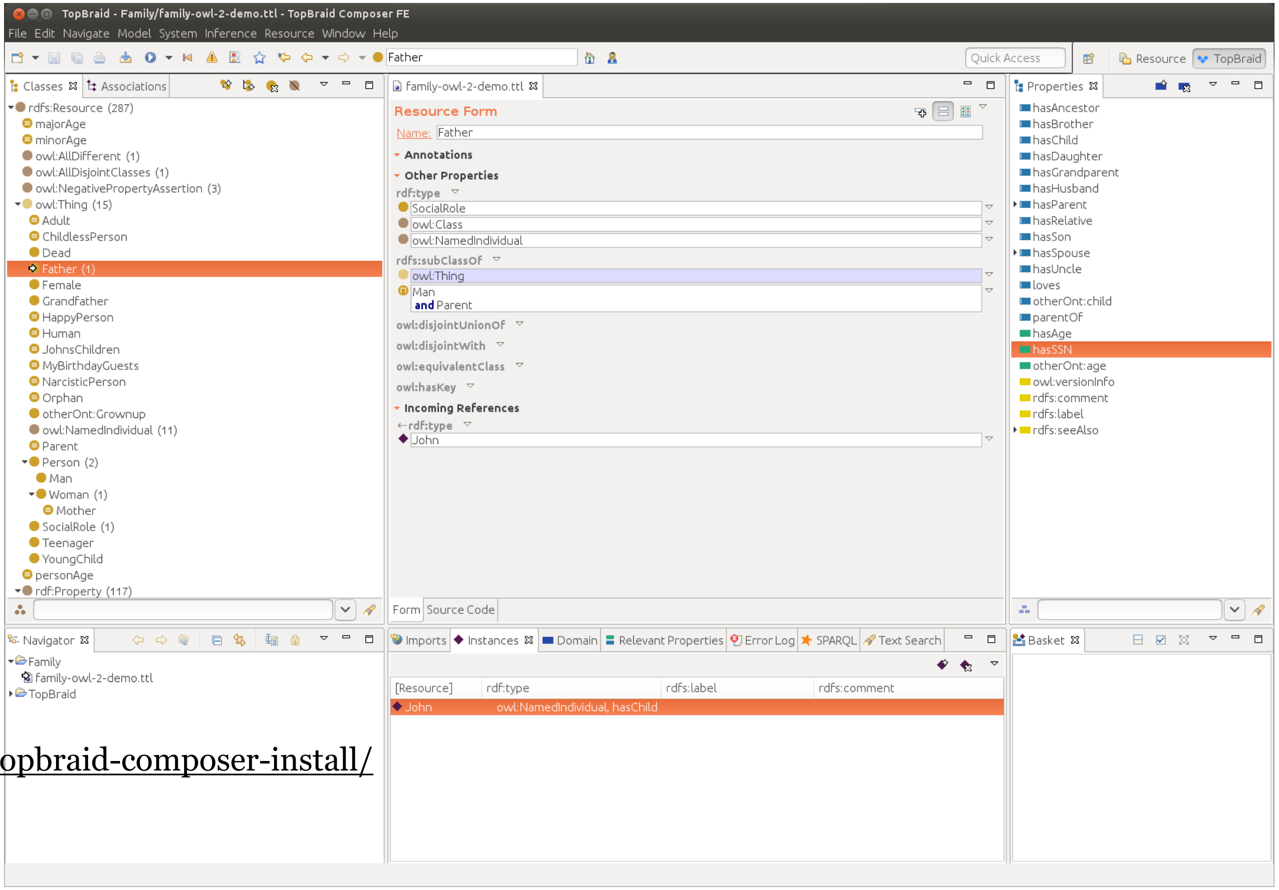
Plugins



https://protege.stanford.edu/

TopBraid Composer

- Commercial product with a free edition option
- SPIN rules for reasoning
- SPARQL querying



rdfs:Resource (287)
- majorAge
- minorAge
- owl:AllDifferent (1)
- owl:AllDisjointClasses (1)
- owl:NegativePropertyAssertion (3)
- owl:Thing (15)
  - Adult
  - ChildlessPerson
  - Dead
  - Father (1)
  - Female
  - Grandfather
  - HappyPerson
  - Human
  - JohnsChildren
  - MyBirthdayGuests
  - NarcisticPerson
  - Orphan
  - otherOnt:Grownup
  - owl:NamedIndividual (11)
  - Parent
  - Person (2)
    - Man
    - Woman (1)
      - Mother
  - SocialRole (1)
  - Teenager
  - YoungChild
- personAge
- rdf:Property (117)

Name: Father

Annotations

Other Properties

rdf:type
- SocialRole
- owl:Class
- owl:NamedIndividual

rdfs:subClassOf
- owl:Thing
- Man **and** Parent

owl:disjointUnionOf

owl:disjointWith

owl:equivalentClass

owl:hasKey

Incoming References

←rdf:type
- John

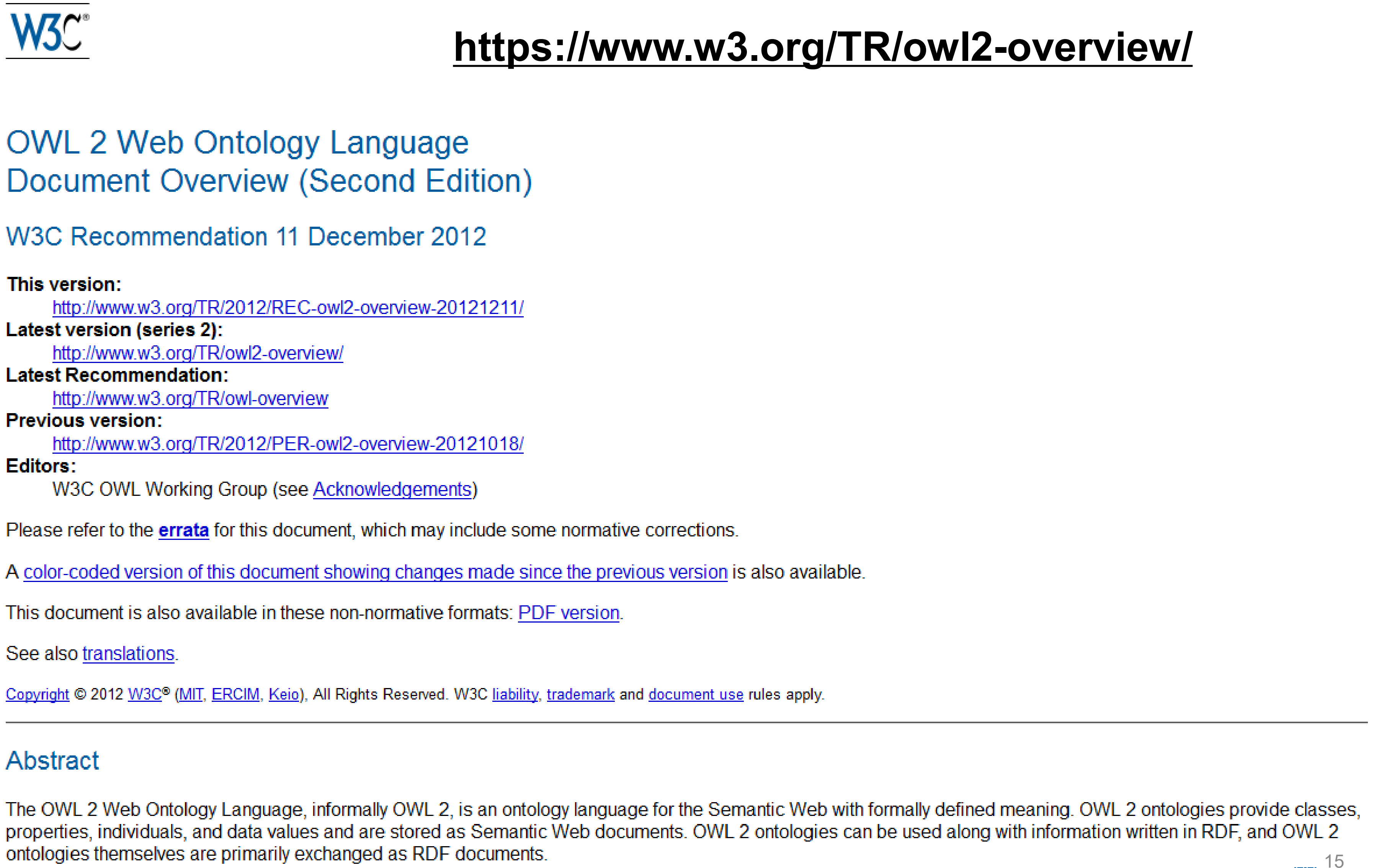
Properties:
- hasAncestor
- hasBrother
- hasChild
- hasDaughter
- hasGrandparent
- hasHusband
- hasParent
- hasRelative
- hasSon
- hasSpouse
- hasUncle
- loves
- otherOnt:child
- parentOf
- hasAge
- hasSSN
- otherOnt:age
- owl:versionInfo
- rdfs:comment
- rdfs:label
- rdfs:seeAlso

Navigator:
- Family
  - family-owl-2-demo.ttl
- TopBraid

[Resource]	rdf:type	rdfs:label	rdfs:comment
John	owl:NamedIndividual, hasChild		

https://www.topquadrant.com/topbraid-composer-install/

https://www.w3.org/TR/owl2-overview/



W3C®

OWL 2 Web Ontology Language Document Overview (Second Edition)

W3C Recommendation 11 December 2012

This version:
  http://www.w3.org/TR/2012/REC-owl2-overview-20121211/
Latest version (series 2):
  http://www.w3.org/TR/owl2-overview/
Latest Recommendation:
  http://www.w3.org/TR/owl-overview
Previous version:
  http://www.w3.org/TR/2012/PER-owl2-overview-20121018/
Editors:
  W3C OWL Working Group (see Acknowledgements)

Please refer to the **errata** for this document, which may include some normative corrections.

A color-coded version of this document showing changes made since the previous version is also available.

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

See also translations.

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.

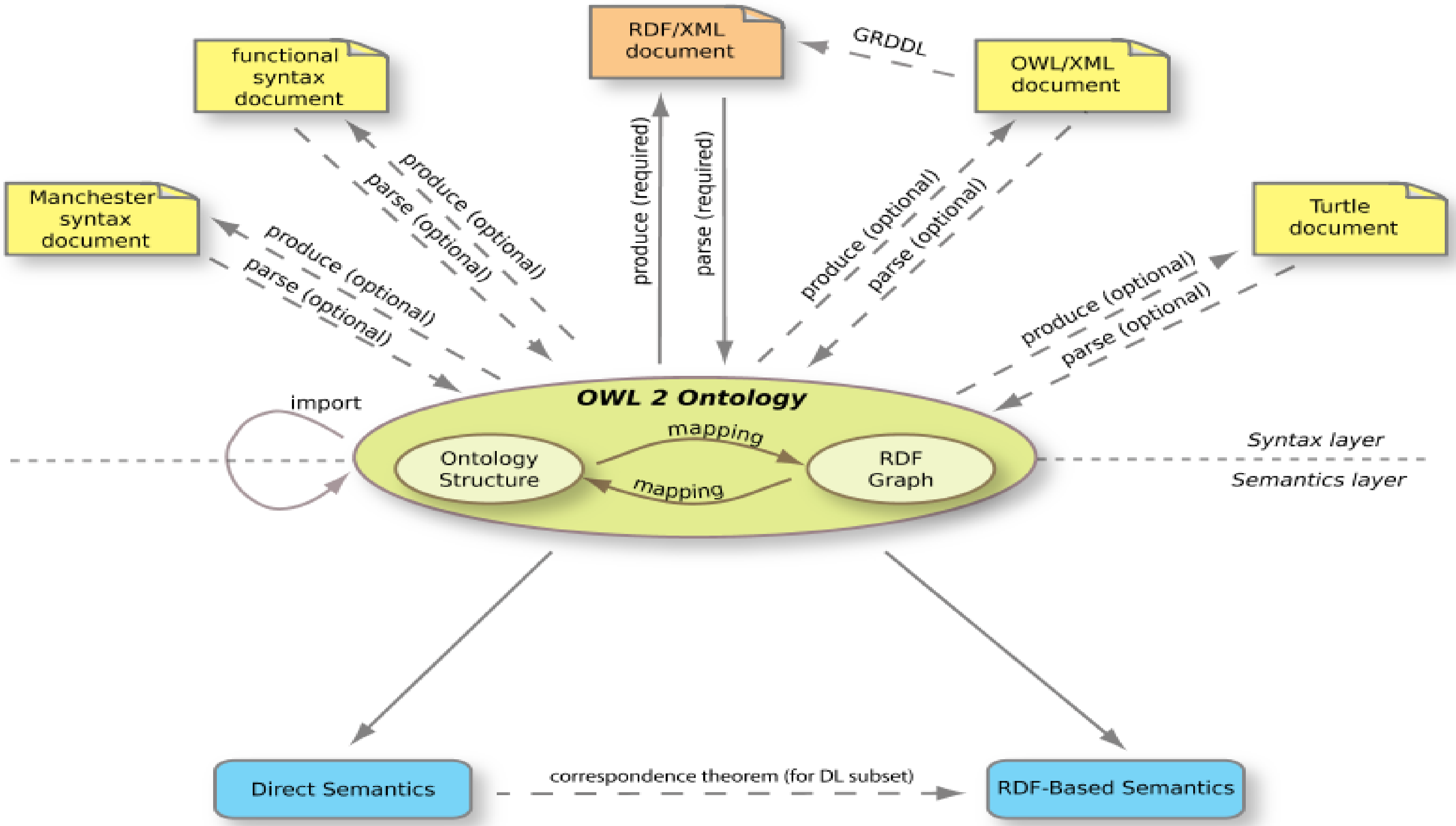


Figure 1. The Structure of OWL 2

Summary

- Background and context of OWL was specified
- See next lecture for learning the OWL language