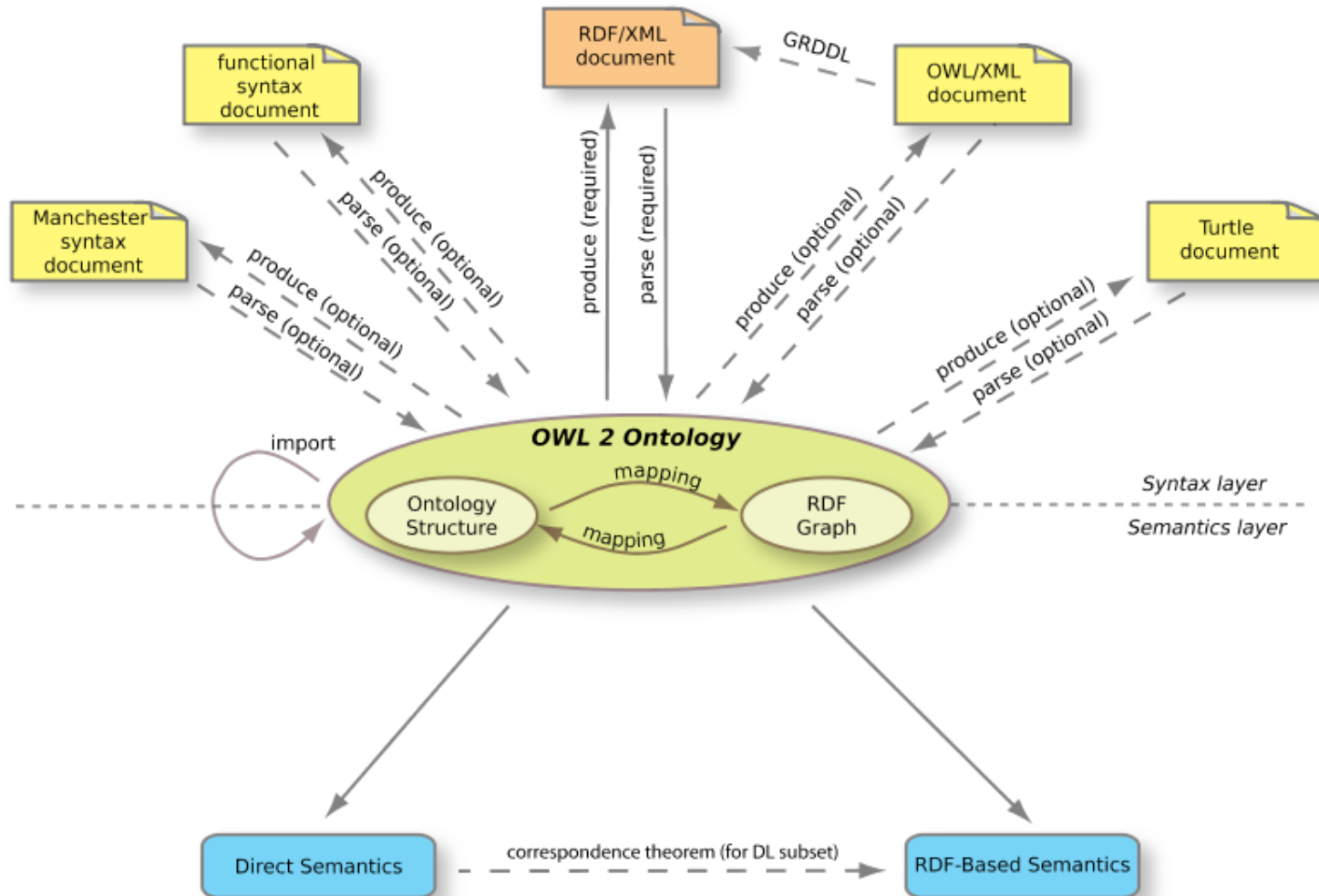


Formati OWL2

Serializzazione
e di OWL

OWL and his friends



Dall'ontologia astratta alla sua espressione

- L'ovale al centro rappresenta la nozione astratta di ontologia, che può essere pensata sia come *struttura astratta* sia come *grafo RDF*
- La sintassi di OWL nelle specifiche è descritta come sintassi **astratta**
- L'ontologia può essere **serializzata** in varie sintassi (parte in alto): per es. RDF/XML e Turtle
- La **semantica** dell'ontologia (parte in basso) può essere specificata direttamente (Direct Mapping, semantica model-teoretica) oppure a seguito della conversione in un grafo RDF (RDF-based semantics)

Mapping verso RDF

- Il mapping di OWL verso RDF consiste nel rappresentare l'ontologia come triple RDF.
- È specificata come un insieme di mapping, dove l'operatore **T** mappa un'ontologia OWL **O** in un grafo RDF **T(O)**.
- Esempi:
 - Declaration(Class(C)) \rightarrow T(C) rdf:type owl:Class
 - DifferentIndividuals(a1 ... an), $n > 2$
 \rightarrow
_:x rdf:type owl:AllDifferent .
_:x owl:members T(SEQ a1 ... an) .

Sintassi di OWL (serializzazione)



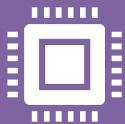
La sintassi RDF/XML di OWL è un formato XML che traduce in RDF i costrutti di OWL2

Deve essere supportata da tutti i tool per OWL 2!



La Manchester syntax è una sintassi di OWL progettata per chi non ha un background in logica

Si trova negli editor (per esempio nelle *class expression* di Protégé)



La sintassi OWL/XML è la traduzione di OWL in uno schema XML.

OWL2 vs. OWL1

Funzionalità aggiuntive di OWL

- property chains;
- datatypes più ricchi, data ranges;
- qualified cardinality restrictions;
- asymmetric, reflexive, and disjoint properties;
- enhanced annotation capabilities;
- keys

- <https://www.w3.org/TR/2012/REC-owl2-quick-reference-20121211/>

Profili di OWL2

- OWL 2 EL progettato per **grandi ontologies**, per es. SNOMED-CT, il thesaurus NCI, e Galen (famiglia di logiche EL, existential quantification)
 - Limitazioni:
 - No negazione e disgiunzione, no quantificazione universale sulle proprietà.
 - No proprietà inverse
- OWL QL (Query Language) concepito per **l'integrazione con le basi di dati** relazionali
 - Limitazioni:
 - No quantificazione esistenziale in una class expression
 - No property chains
- OWL RL (Rule Language) concepito per applicazioni che richiedono l'utilizzo del **ragionamento automatico** senza sacrificare troppa espressività
 - Specifici tipi di espressione sono vietati: per esempio, non si può dire “ogni persona ha un genitore”

Esempio (da Prov-O)

Characteristics ? [] [] [] [] []	Description: prov:hadActivity
<input type="checkbox"/> Functional	Equivalent To +
<input type="checkbox"/> Inverse functional	SubProperty Of +
<input type="checkbox"/> Transitive	Inverse Of +
<input type="checkbox"/> Symmetric	Domains (intersection) +
<input type="checkbox"/> Asymmetric	prov:Influence
<input type="checkbox"/> Reflexive	prov:Delegation or prov:Derivation or prov:End or prov:Start
<input type="checkbox"/> Irreflexive	Ranges (intersection) +
	prov:Activity

Dalle annotazioni:
This property has multiple RDFS domains to suit multiple OWL Profiles.

Da <https://www.w3.org/TR/prov-o/#owl-profile> :

"The PROV-O axioms that do not suit the OWL 2 RL profile use an anonymous class union for the domain or range of a property, while OWL 2 RL requires the classes to be explicitly named. "

Quindi è usata come alternativa la classe (prov:Influence) che li sussume

Reasoning plugins per Protégé

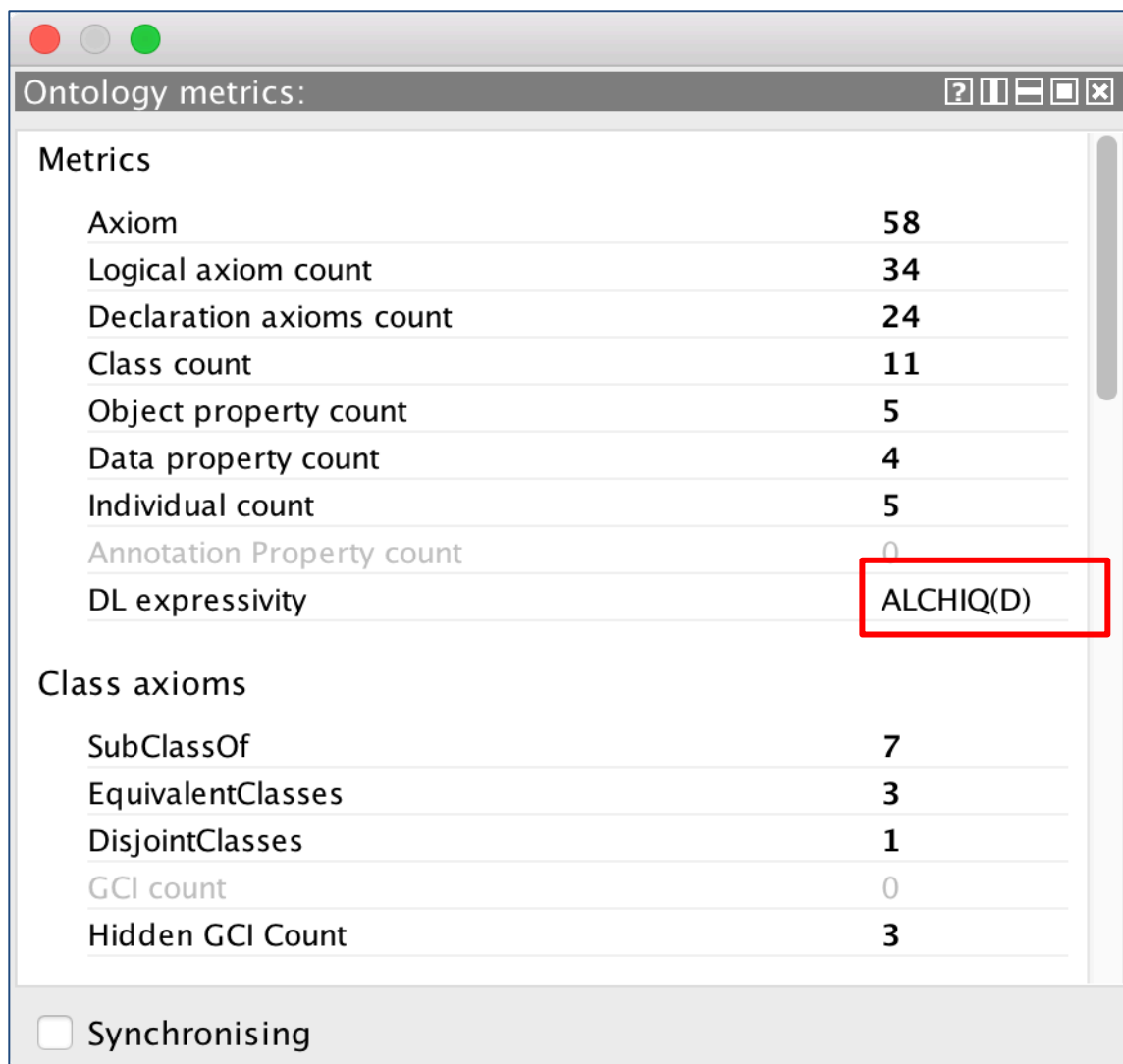
(basati su OWL API)

- ELK è un reasoner specifico per ontologie light weight (profilo EL)
 - Non supporta, ad esempio, le enumerazioni, le restrizioni sulla cardinalità minima, le inverse object properties, le data property assertions.
 - <https://www.cs.ox.ac.uk/isg/tools/ELK/>

Reasoners per OWL e OWL2 basati su logiche descrittive (DL) – tutti i profili di OWL

- Hermit
 - Tecnologia hyper-tableaux, più veloce
 - Ottimo anche per sviluppo di ontologie
 - <http://www.hermit-reasoner.com>
- Fact++
 - Manca del supporto per alcuni datatypes (supporta solo: datatypes are Literal, string, anyURI, boolean, float, double, integer, int, nonNegativeInteger, dateTime)
 - <https://code.google.com/archive/p/factplusplus/>
- Pellet
 - Non supporta il ragionamento su data property ranges
 - Supporta le regole SWRL in Protege, è incrementale
 - Disponibile per Jena
 - Diffuso anche per ragioni storiche
 - <https://www.w3.org/2001/sw/wiki/Pellet>

Espressività di OWL



Ontology metrics:	
Metrics	
Axiom	58
Logical axiom count	34
Declaration axioms count	24
Class count	11
Object property count	5
Data property count	4
Individual count	5
Annotation Property count	0
DL expressivity	ALCHIQ(D)
Class axioms	
SubClassOf	7
EquivalentClasses	3
DisjointClasses	1
GCI count	0
Hidden GCI Count	3
<input type="checkbox"/> Synchronising	

L'espressività dell'ontologia in termini logici viene calcolata da Protégé sulla base delle caratteristiche dell'ontologia stessa

In questo caso è uguale a AL (default) + C (Disgiunzione), H (Gerarchia delle proprietà), I (Proprietà inverse), Q (Restrizioni numeriche), D (Datatype).