

OWL網路本體語言基本概念

周國華

屏東商業技術學院

初版：2010/3/18

本次修訂：2011/3/16

Ontology

- **Ontology**(本體論 或簡稱本體)是源自哲學領域的名詞，它是一門探討如何描述存在於世界上的各種事物及其彼此間關係的科學。
- **OWL (Web Ontology Language)**：網路本體語言。
- 一份OWL本體可包含對類別(class)、屬性(property)及案例(instance或individual)的描述，然後，根據**OWL formal semantics (OWL正規語意)**的規範，推論出未明示於本體內但可推論得到的邏輯結果。

Why OWL?

- 已經有了XML及XML schema，為何還要OWL？
 - OWL是用來讓軟體處理資訊的內涵，而非只是把資訊呈現出來。
 - OWL本體是一個知識表達方式，XML schema則是一種訊息格式。雖然已有許多特定領域專屬XML標準(例如XBRL)可以在一定程度上表達操作性語意，但在專屬情境之外並不支援一般性的推論功能，所以無法使用一般性的推論工具。
 - OWL本體可使用一般性的推論工具，所以任何領域的使用者都可以專注於OWL本體的建立，然後使用一般性的工具對本體進行推論。

OWL的三種子語言

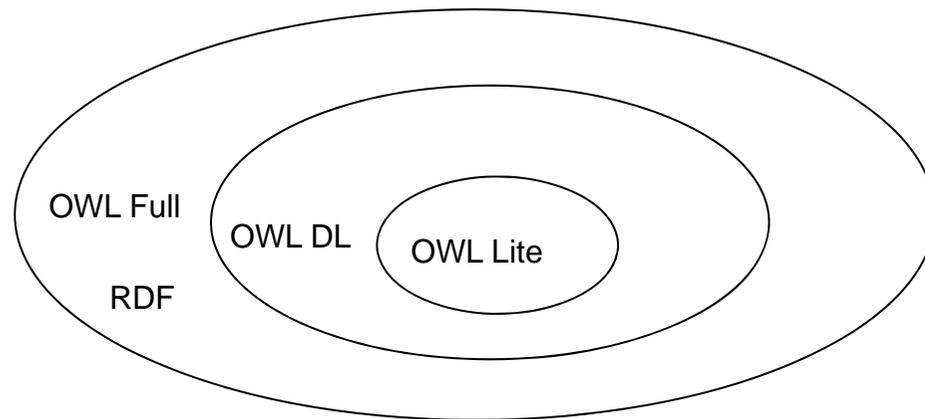
- 根據表達上的約束性，OWL分成三種子語言：
 - OWL Lite：約束性最大，表達性最受限。但軟體開發較容易。
 - OWL DL：可在確保計算完整性及可決定性的前提下，提供最大程度的表達性。
 - OWL Full：約束性最小，可提供最大程度的表達性及RDF語法自由性，但無法確保計算完整性及可決定性。推論軟體不太可能支援OWL Full的每一項特質。
 - OWL Full是OWL DL的延伸，OWL DL是OWL Lite的延伸。每一個合法(legal)或合格(valid)的OWL Lite本體，也是一個合法或合格的OWL DL本體；每一個合法或合格的OWL DL本體，也是一個合法或合格的OWL Full本體。

Semantic Web

- OWL是W3C的語意網(semantic web)技術架構的一部份：
 - XML：提供結構化文件的基礎語法。
 - XML Schema：為XML文件增添結構性限制，並大量增加資料型態。
 - RDF：為物件(即資源)及其間的關係提供資料模型及該模型的簡單語意，並以XML語法表達此模型。
 - RDF Schema：提供一個可描述RDF資源類別(class)及屬性(property)的語彙，內含一般化及階層化的語意概念。
 - OWL：對類別及屬性的描述提供更豐富的語彙，以讓軟體進行推論工作。

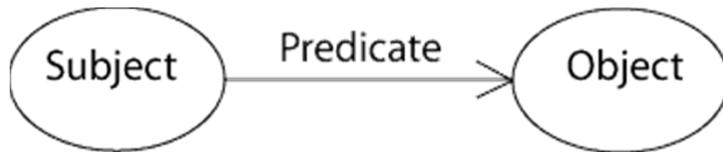
相容性

- 每一份OWL文件都是一份RDF文件。
- 每一份RDF文件都是一份OWL Full文件，但只有部分RDF文件是一份OWL DL或OWL Lite文件。



OWL 語法：RDF Graph

- 一份OWL本體是由OWL抽象語法(abstract syntax)建構而成，然後對映成一份RDF Graph，後者則是由許多RDF triple組成。RDF Graph也稱為OWL本體的交換語法(exchange syntax)。
- RDF triple：每個triple包含主詞(subject)、述語(predicate)及受詞(object)。述語又稱屬性(property)，負責連結主詞及受詞。



OWL 交換語法：多種選擇

- OWL本體以抽象語法建構而成，至於交換語法則有多種選擇：
 - RDF/XML：把OWL本體按照RDF Graph表達，任何OWL工具軟體都必須支援此語法。
 - OWL/XML：把OWL本體按照XML Schema的規格表達。
 - Functional Syntax：比較容易呈現OWL本體的正規結構。
 - Manchester Syntax：比較容易讀寫OWL DL本體。
 - Turtle：比較容易讀寫RDF triple，功能與RDF/XML類似。
- 上述五種交換語法中，只有RDF/XML是強制形式，其他四種均為選用形式。但每種語法形式表達的實質結果必須相同。

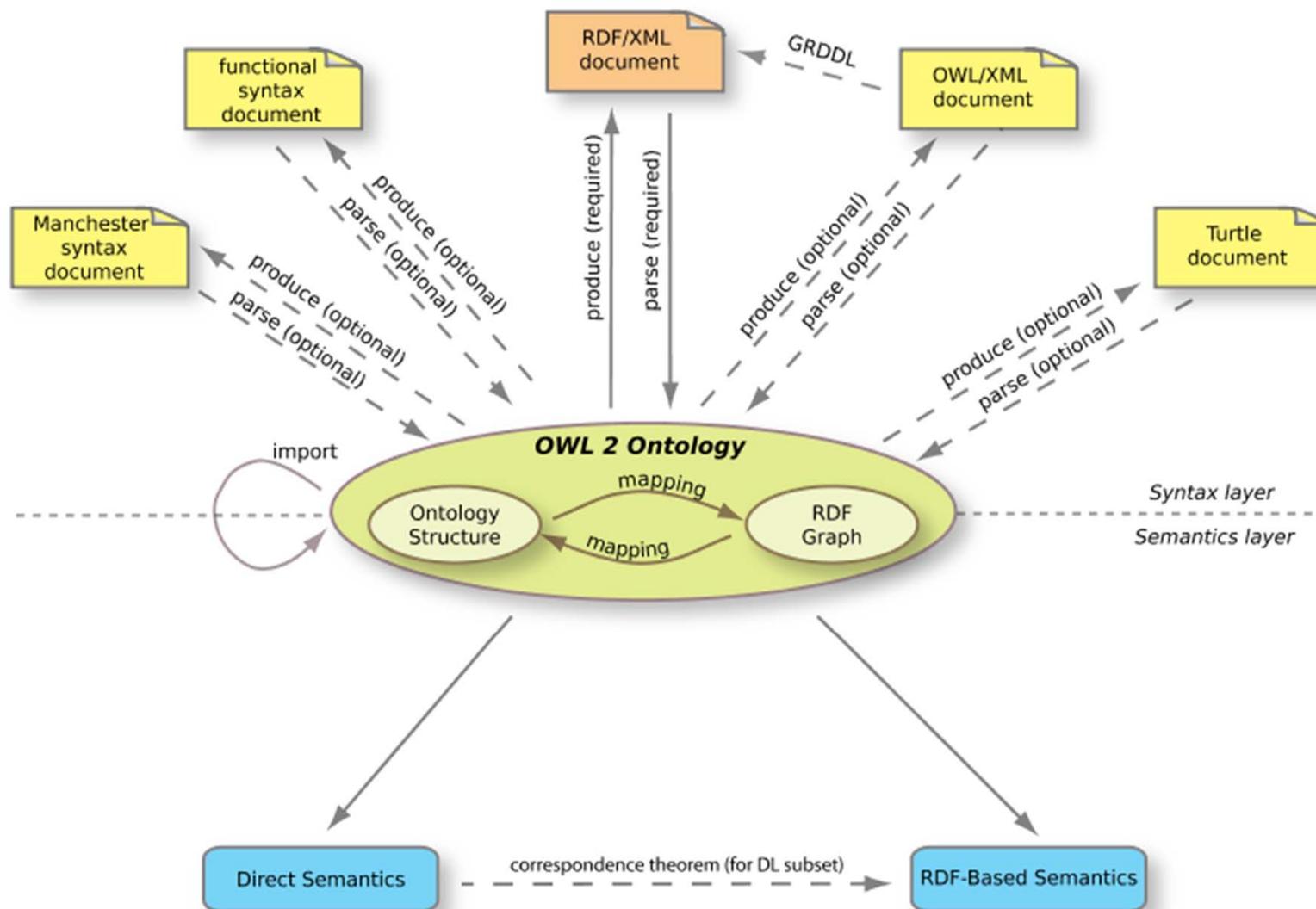
OWL 語意

- OWL本體可用兩種方式給定語意內涵：
 - Direct model-theoretic semantics：直接模式理論語意法，根據OWL本體的抽象語法給予語意內涵。
 - RDF-compatible model-theoretic semantics：RDF相容模式理論語意法，根據OWL本體所對映的RDF Graph交換語法給予語意內涵。本法是RDF語意法的延伸。比前一種語意法要複雜一些。
- 上述兩種語意法應該產生類似的語意內涵，如果某一OWL本體在兩種語意法下得到不同內涵，應該以直接模式法為準。

OWL 2 的分類

- 在W3C最新版本的OWL 2中，OWL 1的三種子語言只剩下OWL 2 DL及OWL 2 Full，而且是按照兩種語意模式來區分，而非按照表達上的約束性：
 - OWL 2 DL：按直接語意模式(direct semantics) 賦予OWL本體意義。
 - OWL 2 Full：按RDF語意模式(RDF-based semantics)賦予OWL本體意義。

OWL 2 語法及語意架構圖



OWL Lite 語法構念

RDF Schema Features:

- [Class \(Thing, Nothing\)](#)
- [rdfs:subClassOf](#)
- [rdf:Property](#)
- [rdfs:subPropertyOf](#)
- [rdfs:domain](#)
- [rdfs:range](#)
- [Individual](#)

Property Restrictions:

- [Restriction](#)
- [onProperty](#)
- [allValuesFrom](#)
- [someValuesFrom](#)

Class Intersection:

- [intersectionOf](#)

Datatypes

- [xsd datatypes](#)

(In)Equality:

- [equivalentClass](#)
- [equivalentProperty](#)
- [sameAs](#)
- [differentFrom](#)
- [AllDifferent](#)
- [distinctMembers](#)

Restricted Cardinality:

- [minCardinality](#) (only 0 or 1)
- [maxCardinality](#) (only 0 or 1)
- [cardinality](#) (only 0 or 1)

Versioning:

- [versionInfo](#)
- [priorVersion](#)
- [backwardCompatibleWith](#)
- [incompatibleWith](#)
- [DeprecatedClass](#)
- [DeprecatedProperty](#)

Property Characteristics:

- [ObjectProperty](#)
- [DatatypeProperty](#)
- [inverseOf](#)
- [TransitiveProperty](#)
- [SymmetricProperty](#)
- [FunctionalProperty](#)
- [InverseFunctionalProperty](#)

Header Information:

- [Ontology](#)
- [imports](#)

Annotation Properties:

- [rdfs:label](#)
- [rdfs:comment](#)
- [rdfs:seeAlso](#)
- [rdfs:isDefinedBy](#)
- [AnnotationProperty](#)
- [OntologyProperty](#)

OWL DL 及 Full 的額外構念

Class Axioms:

- [oneOf, dataRange](#)
- [disjointWith](#)
- [equivalentClass](#)
(applied to class expressions)
- [rdfs:subClassOf](#)
(applied to class expressions)

Arbitrary Cardinality:

- [minCardinality](#)
- [maxCardinality](#)
- [cardinality](#)

Boolean Combinations of Class Expressions:

- [unionOf](#)
- [complementOf](#)
- [intersectionOf](#)

Filler Information:

- [hasValue](#)

本體之間的關連性

- 基於網路資源的分散性本質，**OWL**允許從不同的來源取得資訊，並可讓**OWL**本體載入另一**OWL**本體的資訊。
- **OWL**遵守開放世界假設，關於任何資源的描述並不限於特定檔案或領域。在本體**A**中對**class C**所做的定義，可以在其他本體中做延伸。但延伸必須遵守單調性(monotonic)，亦即「可增不可減」，不能在延伸中收回之前的規定。因此，彼此關連的**OWL**本體有可能會發生抵觸現象。