

# Introduzione al Semantic Web

*Oreste Signore*

W3C Italia e CNR

Area della Ricerca CNR di Pisa – via G. Moruzzi, 1 – 56124 Pisa (Italy)

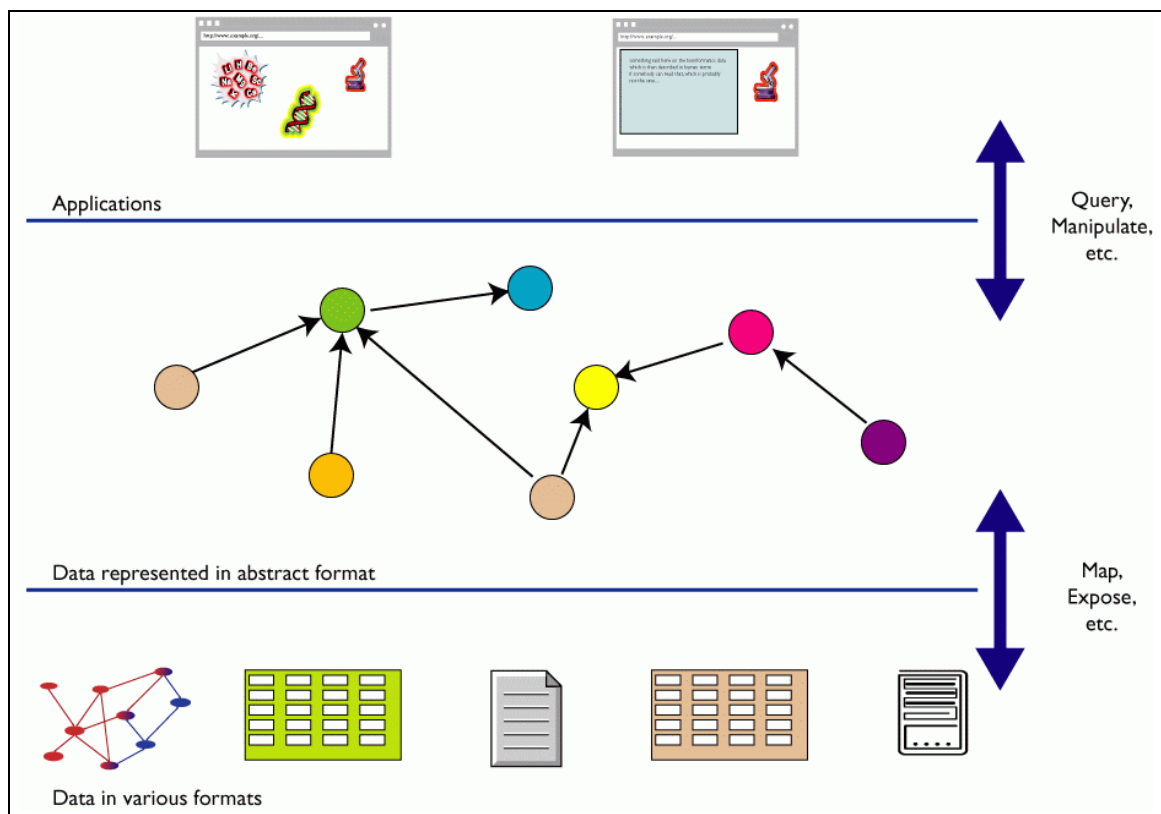
e.mail: [oreste@w3.org](mailto:oreste@w3.org)

home page: <http://www.weblab.isti.cnr.it/people/oreste/>

**Abstract.** Il Web è forse oggi il maggior contenitore di conoscenza, o, comunque, è quello più frequentemente utilizzato da una larga varietà di persone. L'informazione è spesso dispersa tra più fonti informative, e sarebbe di grande ausilio la possibilità che le macchine potessero autonomamente estrarre e dedurre conoscenza. Le tecnologie del Semantic Web mirano appunto a questo obiettivo. Questa breve presentazione illustra alcuni elementi minimi delle tecnologie di base (RDF/RDFS, OWL), per fornire un quadro delle loro potenzialità.

## Introduzione

“*Lo cerco sul Web*” è un' espressione usuale, che testimonia come il Web sia percepito come un serbatoio di conoscenza. Per gli esseri umani il processo di combinare informazioni, spesso incomplete, provenienti da fonti diverse, e memorizzate in formati diversi (pagine Web, database, fogli elettronici, etc.) per ottenere la risposta adeguata alle proprie esigenze è ragionevolmente semplice, anche se spesso noioso e ripetitivo (Figura 1). Sarebbe desiderabile che le macchine potessero automaticamente combinare la conoscenza proveniente dalle diverse fonti e, ancor meglio, derivarne di nuova



**Figura 1 - Combinazione di informazioni provenienti da fonti eterogenee**

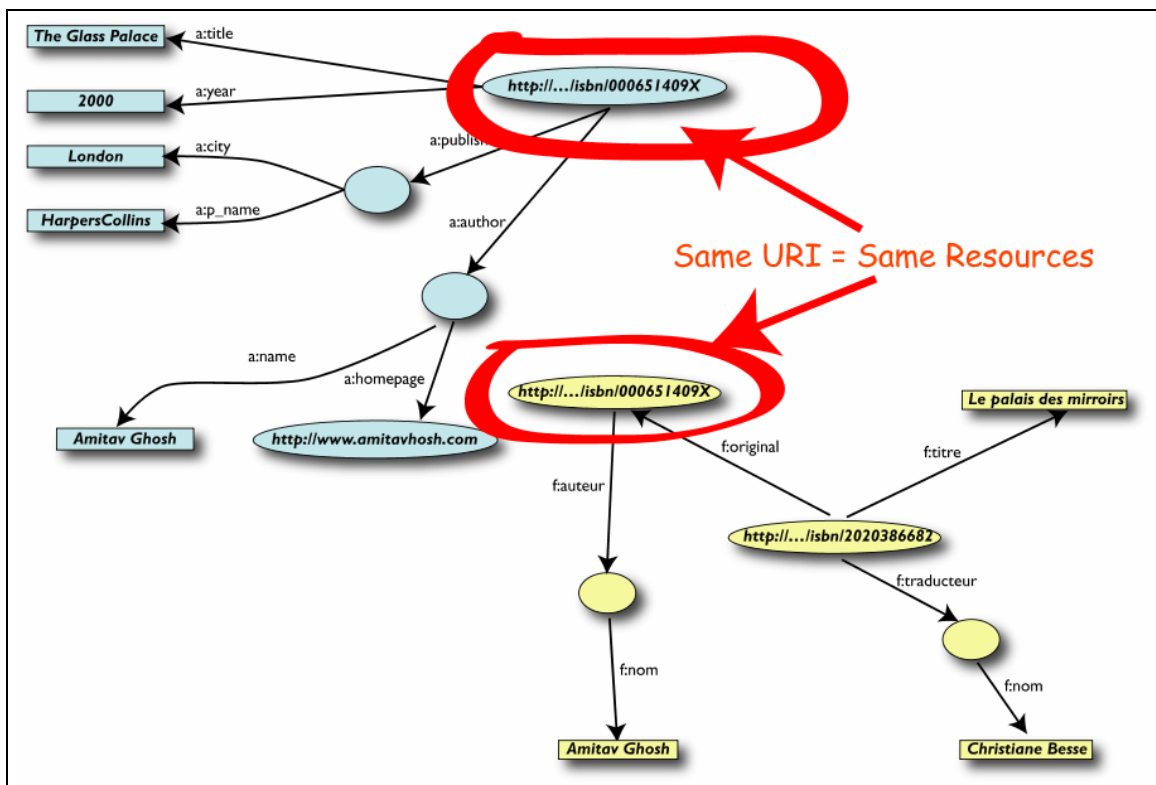
© Oreste Signore 2008

Riproduzione consentita per uso personale o didattico

Document URI: <http://www.w3c.it/papers/wsb08.pdf>

Presentation URI: <http://www.w3c.it/talks/2008/wsb08/>

Per esempio, potremmo combinare le informazioni di varie biblioteche, trovando la versione originale di un libro, e informazioni sul suo autore, o su altri libri dello stesso autore, semplicemente partendo dal suo identificatore univoco (ISBN) e percorrendo le associazioni rappresentate nelle varie fonti informative. Ciò però è impossibile, per una macchina, senza un adeguato meccanismo per la rappresentazione della conoscenza. Se rappresentiamo le informazioni sotto forma di grafo, il processo che permette di combinare le informazioni provenienti da due biblioteche (merge dei grafi) è quello rappresentato in Figura 2, dove i dati provenienti dalle due biblioteche sono contraddistinti dai prefissi “a:” e “f:” nel nome dell’ arco, e il merge dei due grafi è reso possibile dal fatto che due oggetti hanno lo stesso identificatore (URI), e quindi *sono lo stesso oggetto*.



**Figura 2 - Il meccanismo del merge dei grafi**

Proprio per superare i limiti del Web attuale, da qualche anno i ricercatori stanno lavorando intensamente per la realizzazione del Semantic Web, che può essere definito come un’ infrastruttura basata su *metadati* per svolgere ragionamenti sul Web. Nel Semantic Web la conoscenza è rappresentata in maniera elaborabile dalla macchina, e può essere utilizzata da componenti automatizzati, denominati agenti software [SemWeb].

I metadati sono informazioni, elaborabili in modo automatico, relative alle “risorse” Web, identificate univocamente da un URI (Uniform Resource Identifier). La tecnologia di riferimento per la codifica, lo scambio e il riutilizzo di metadati strutturati è *Resource Description Framework* (RDF), basata su un modello molto semplice di “statement”, rappresentabili come triple (in termini più formali, una tripla forma un “grafo orientato etichettato”).

Per esprimere le restrizioni sulle associazioni, quindi per evitare che possano essere codificati degli “statement” sintatticamente corretti, ma privi di senso, è necessario un meccanismo per rappresentare “classi di oggetti”. Da questa esigenza nasce “*RDF Vocabulary Description Language*”, che mantiene anche, per ragioni storiche, il nome di “RDF Schema” (RDFS).

Per poter effettuare dei ragionamenti, per definire le classi, e per varie altre esigenze, però, RDFS non è sufficiente, e occorre un modo per rappresentare la conoscenza e le regole che permettono di dedurre ulteriore conoscenza: l’ *ontologia*.

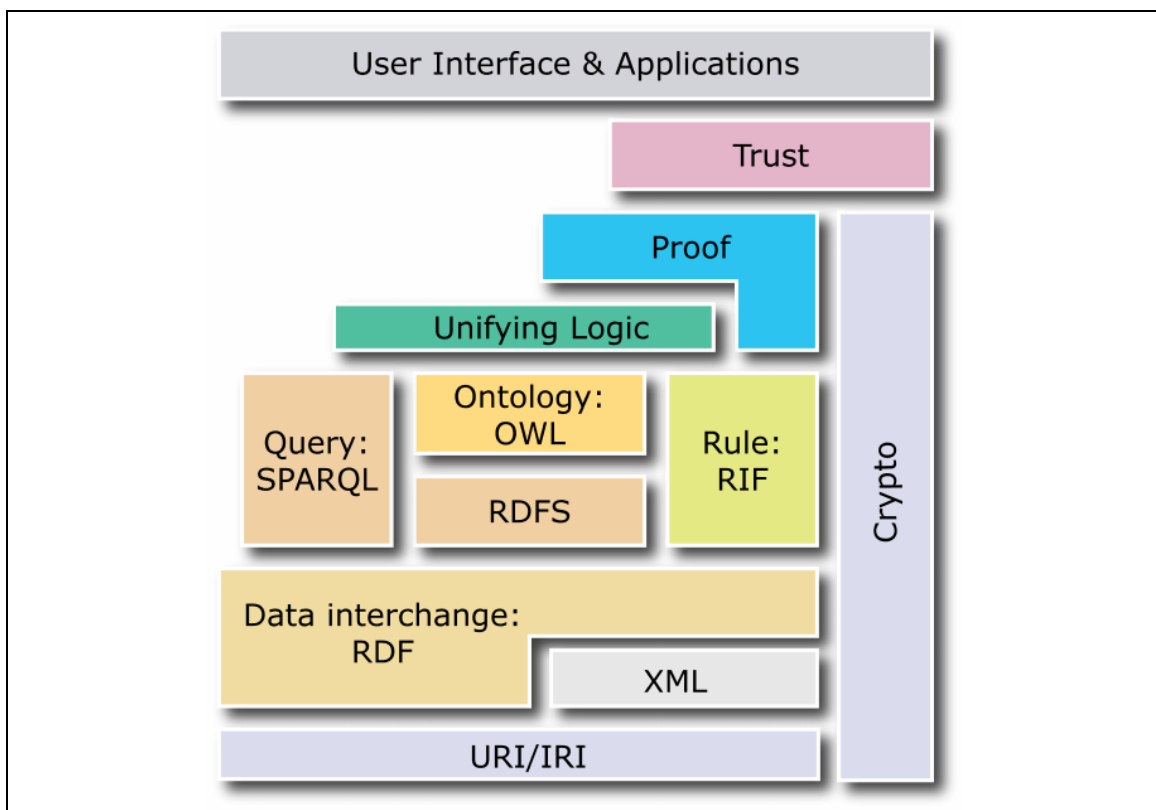
Ma il Web è intrinsecamente distribuito, e quindi occorre un linguaggio che non solo permetta di esprimere dati e regole sui dati, ma che consenta anche di esportare questa conoscenza (ontologia) per renderla disponibile a qualunque applicazione. Il W3C ha definito, per questa esigenza, il *Web Ontology Language* (OWL).

Nel resto di questo lavoro verranno illustrate brevemente l’ architettura del Semantic Web e queste tecnologie.

## Il Semantic Web

### L’ architettura

Nella visione di Berners-Lee, il Semantic Web, ha una architettura a livelli (Figura 3).



**Figura 3 - L’ architettura del Semantic Web**

Va ricordato che la filosofia di base del Web è quella di uno spazio informativo universale, navigabile, con un mapping da *Uniform Resource Identifier* (URI) alle risorse. Nel contesto del Semantic Web, il termine semantico assume la valenza di “*elaborabile dalla macchina*” e non intende fare riferimento alla semantica del

linguaggio naturale e alle tecniche di intelligenza artificiale. Il Semantic Web è, come l'XML, un ambiente dichiarativo, in cui si specifica il significato dei dati, e non il modo in cui si intende utilizzarli. La semantica dei dati consiste nelle informazioni utili perché la macchina possa utilizzarli nel modo corretto, eventualmente convertendoli.

Come chiaramente descritto in [TBL2001], il Semantic Web potrà funzionare solo se le macchine potranno accedere ad un *insieme strutturato di informazioni* e a un insieme di *regole di inferenza* da utilizzare per il ragionamento automatico. La sfida del semantic Web, quindi, è fornire un linguaggio per esprimere *dati* e *regole* per ragionare sui dati, che consenta l' *esportazione sul Web* delle regole da qualunque sistema di rappresentazione della conoscenza.

Un ruolo di base è giocato da XML (con *Name Space* e *xmldata*), che consente di dare ai documenti una *struttura* arbitraria, mentre RDF può essere usato per esprimere il *significato*, asserendo che alcuni particolari elementi hanno delle proprietà (p.es. *autore-di*).

Il *livello ontologico*, è il contenitore che definisce in modo formale le relazioni fra i termini. Un' ontologia permette di descrivere le relazioni tra i tipi di elementi (per es. "questa è una proprietà transitiva") senza però fornire informazioni su come utilizzare queste relazioni dal punto di vista computazionale. Il linguaggio definito dal W3C per definire ontologie strutturate, in architettura Web, per consentire una migliore integrazione dei dati tra applicazioni in settori diversi è OWL (Ontology Web Language).

Il *livello logico* è il livello immediatamente superiore al livello ontologico. A questo livello le asserzioni esistenti sul Web possono essere utilizzate per derivare nuova conoscenza. Tuttavia, i sistemi deduttivi non sono normalmente interoperabili, per cui, secondo Berners-Lee, invece di progettare un unico sistema onnicomprensivo per supportare il ragionamento (*reasoning system*), si potrebbe pensare di definire un linguaggio universale per rappresentare le dimostrazioni. I sistemi potrebbero quindi autenticare con la firma digitale queste dimostrazioni ed esportarle ad altri sistemi che le potrebbero incorporare nel Semantic Web.

La firma digitale (*digital signature*) è di significativa importanza in diversi strati nel modello astratto del Semantic Web. La crittografia a chiave pubblica è una tecnica nota da qualche anno, ma non si è ancora diffusa su larga scala come ci si poteva attendere. Nella visione di Berners-Lee, un elemento che potrebbe aver giocato contro la diffusione di questa tecnica è che essa è a "grana grossa", imponendo una scelta binaria tra fiducia o non fiducia (trusted/not trusted), mentre sarebbe necessaria una infrastruttura in cui le parti possano essere riconosciute e accettate come credibili in specifici domini. Con una granularità più fine come questa, la firma digitale potrebbe essere utilizzata per stabilire la provenienza delle ontologie e delle deduzioni, oltre che dei dati.

## **I metadati**

Nel navigare sul Web, si seguono dei link, che portano a quella che formalmente viene detta *risorsa* (resource) identificata univocamente da un URI. Nel linguaggio corrente una risorsa viene anche detta "documento", per mettere in evidenza il fatto che sia leggibile da un essere umano, o "oggetto", per mettere in evidenza che è leggibile da una macchina. Qualunque sia il termine utilizzato, la risorsa non è una entità a sé, ma è accompagnata da informazioni che la descrivono. Le informazioni sulla risorsa vengono generalmente dette Metadati.

Si può quindi dire che *i metadati sono informazioni, comprensibili dalla macchina, relative a una risorsa Web o a qualche altra cosa*. Il punto chiave è costituito appunto dal fatto che i metadati sono comprensibili dalla macchina (*machine understandable*). Di conseguenza, i metadati costituiscono un tipo di informazione che può essere utilizzata dai *software agent*, per fare un uso appropriato delle risorse, rendendo più semplice e veloce il funzionamento del Web, aumentando la nostra fiducia in esso. A titolo di esempio, quando si reperisce un documento (o un oggetto) sul Web, utilizzando il protocollo HTTP, è possibile che il server invii alcune informazioni sulla risorsa, quali la sua data di aggiornamento, la data massima di validità dell'informazione, il suo autore, etc. Quindi il Web, come insieme di risorse e di informazioni sulle risorse (cioè metadati) è già una realtà alla quale siamo abituati.

Va tenuto presente che *i metadati sono dati*, e questo fatto ha alcune conseguenze:

- possono essere *memorizzati come dati*, in una risorsa, che può quindi contenere informazioni relative a se stessa o ad un'altra risorsa. I metadati relativi ad un documento possono essere contenuti nel documento, oppure contenuti in un documento separato, oppure essere trasferiti a corredo del documento.
- possono essere *descritti da altri metadati*, e così via.

I limiti pratici e logici per il numero di livelli dipendono essenzialmente dalle caratteristiche e dalle tradizioni delle singole comunità che definiscono la descrizione delle risorse.

## **Il Resource Description Framework**

Automatizzare il Web restando ancorati alla sua architettura originaria, in cui tutte le informazioni erano *machine-readable*, ma non *machine-understandable*, era un obiettivo molto difficilmente raggiungibile, mentre i *metadati* sembrano offrire una soluzione al problema. L'uso efficace dei metadati, tuttavia, richiede che vengano stabilite delle convenzioni per la *semantica*, la *sintassi* e la *struttura*. Le singole comunità interessate alla descrizione delle loro risorse specifiche definiscono la semantica dei metadati pertinenti alle loro esigenze. La sintassi, cioè l'organizzazione sistematica dei data element per l'elaborazione automatica, facilita lo scambio e l'utilizzo dei metadati tra applicazioni diverse. La struttura può essere vista come un vincolo formale sulla sintassi, per una rappresentazione consistente della semantica.

*Resource Description Framework* (RDF) è lo strumento base per la codifica, lo scambio e il riutilizzo di metadati strutturati, e consente l'interoperabilità tra applicazioni che si scambiano sul Web informazioni *machine-understandable*. RDF consente l'elaborazione automatica delle risorse reperibili sul Web, e può essere utilizzato e portare vantaggi sono in molti settori.

RDF fornisce un modello per descrivere le risorse, che hanno delle proprietà (o anche attributi o caratteristiche). RDF definisce una risorsa come un qualsiasi oggetto che sia identificabile univocamente mediante un Uniform Resource Identifier (URI).

Il data model RDF, che consente di rappresentare statement RDF in modo sintatticamente neutro, è molto semplice ed è basato su tre tipi di oggetti:

*Resources* Qualunque cosa descritta da una espressione RDF viene detta risorsa (*resource*). Una risorsa può essere una pagina Web, o una sua parte, o un elemento XML all'interno del documento sorgente. Una risorsa può anche essere un'intera collezione di pagine Web, o anche un oggetto non direttamente accessibile via Web (per es. un libro, un dipinto, etc.). Le

risorse sono sempre individuate da un URI, eventualmente con un *anchor id* (riferimento interno). Qualunque cosa può essere identificata da un URI.

*Properties* Una *property* (proprietà) è un aspetto specifico, una caratteristica, un attributo, o una relazione utilizzata per descrivere una risorsa. Ogni proprietà ha un significato specifico, definisce i valori ammissibili, i tipi di risorse che può descrivere, e le sue relazioni con altre proprietà. Le proprietà associate alle risorse sono identificate da un *nome*, e assumono dei *valori*.

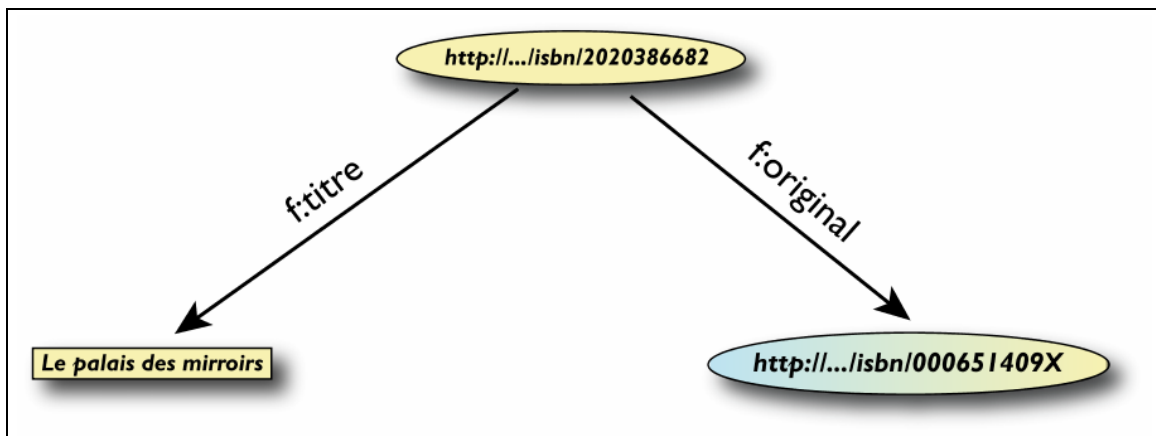
*Statements* Una risorsa, con una proprietà distinta da un nome, e un valore della proprietà per la specifica risorsa, costituisce un *RDF statement* (asserzione). Uno statement è quindi una *tripla* composta da un *soggetto* (risorsa), un *predicato* (proprietà) e un *oggetto* (valore). L'oggetto di uno statement (cioè il *property value*) può essere un'espressione (sequenza di caratteri o qualche altro tipo primitivo definito da XML) oppure un'altra risorsa.

Graficamente, le relazioni tra Resource, Property e Value vengono rappresentate mediante *grafi orientati etichettati*, in cui le risorse vengono identificate come nodi (graficamente delle ellissi), le proprietà come archi orientati etichettati, e i valori corrispondenti a sequenze di caratteri come rettangoli. Un insieme di proprietà che fanno riferimento alla stessa risorsa viene detto descrizione (*description*).

RDF permette di descrivere anche fatti complessi. Per esempio, il fatto espresso dalla concatenazione delle due frasi:

*Il libro identificato dall' ISBN <http://.../isbn/2020386682>  
ha titolo *Le palais des miroirs*  
e ha come versione originale il libro con ISBN <http://.../isbn/000651409X>*

verrebbe rappresentato dal diagramma di Figura 4



**Figura 4 - Un esempio semplice di RDF**

E questo statement potrebbe essere codificato in vari modi, a seconda della sintassi adottata (vedi Figura 5 e Figura 6)

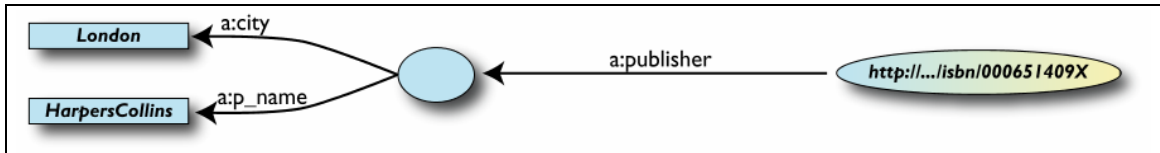
```
<rdf:Description rdf:about="http://.../isbn/2020386682">
  <f:titre xml:lang="fr">Le palais des miroirs</f:titre>
  <f:original rdf:resource="http://.../isbn/000651409X"/>
</rdf:Description>
```

**Figura 5 - Esempio di codifica in RDF/XML**

```
<http://.../isbn/2020386682>
  f:titre "Le palais des miroirs"@fr;
  f:original <http://.../isbn/000651409X>.
```

**Figura 6 - esempio di codifica in Turtle**

Non sempre i nodi hanno un nome e un identificatore, talvolta è utile, o necessario, utilizzare dei nodi “anonimi” (*blank nodes*) come illustrato in Figura 7, in cui si modella il fatto che un libro identificato da un certo ISBN ha come editore “qualcuno” che si chiama “HarpersCollins” e risiede nella città “London”.



**Figura 7 - Un esempio di "blank node"**

È importante sottolineare che sia le risorse che le proprietà sono identificate univocamente da URI. La potenza del meccanismo di RDF risiede appunto nel fatto che un documento RDF può far riferimento a proprietà e risorse che sono definite in un qualunque punto del Web, senza necessità di centralizzare le informazioni.

## RDF Schema

RDF è un linguaggio universale che consente agli utenti di usare il loro vocabolario per descrivere le risorse. Per questo motivo RDF non formula nessuna assunzione su qualunque dominio applicativo specifico, né ne definisce la semantica.

Per esprimere le restrizioni sulle associazioni, in altri termini per evitare che possano essere codificati degli “statement” sintatticamente corretti, ma privi di senso, è necessario un meccanismo per rappresentare “classi di oggetti”. Da questa esigenza nasce “*RDF Vocabulary Description Language*”, che mantiene anche, per ragioni storiche, il nome di “RDF Schema” (RDFS).

L’aspetto fondamentale è costituito dalla necessità di esprimere fatti e condizioni non solo sui singoli oggetti, ma anche sulle classi che definiscono i tipi di oggetto. Usualmente, si dice che una classe può essere vista come un insieme di elementi, che vengono indicati come istanze della classe. In RDF possiamo usare la proprietà `rdf:type` per specificare la relazione tra istanze e classi. Il meccanismo delle classi può essere utilizzato per imporre restrizioni sulle proposizioni che possono essere enunciate in un documento RDF che usi quello schema. Per esempio, potremmo dire che hanno senso proposizioni (*triple s-p-o*) come:

Leonardo	autoreDi	Gioconda.
Cimabue	maestroDi	Giotto.

altre proposizioni quali:

Michelangelo	autoreDi	Leonardo.
ritrattoDiGiulioII	autoreDi	Gioconda.

sono entrambe prive di senso, perché un artista non può essere autoreDi un altro artista, e un’ opera non può essere autoreDi un’ altra opera. Invece, un artista può

essere maestroDi un altro artista, e un' opera può essere in relazione con un' altra opera perché ne è una versione diversa o una copia.. In maniera più formale, possiamo dire che è necessario poter restringere *dominio* e *codominio* (*domain* e *range*) delle proprietà.

Una volta definite le classi, può risultare utile definire delle relazioni tra di esse. Per esempio, potremmo aver definito le classi:

- cane
- mammifero

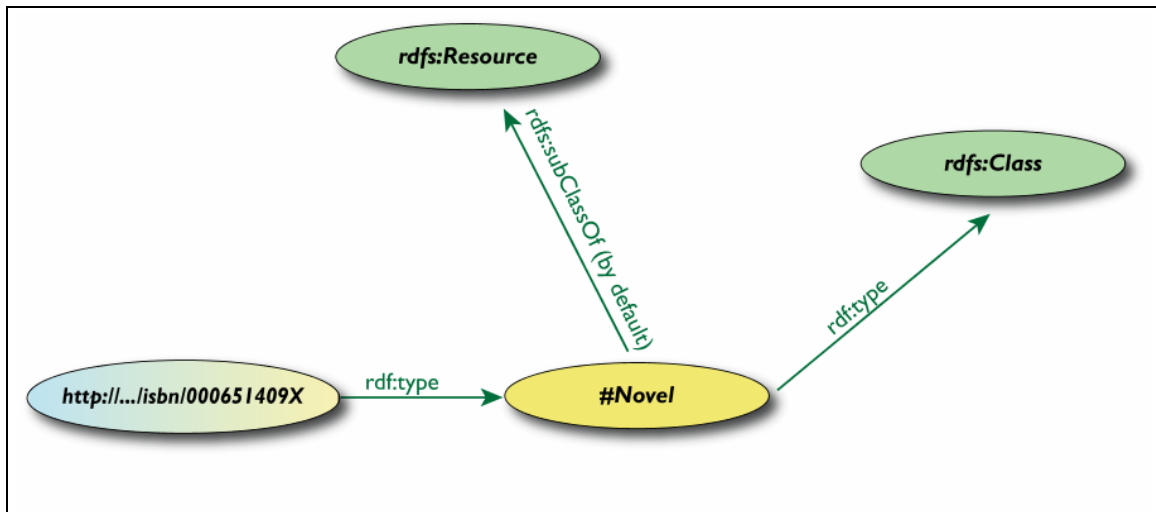
e aver asserito che: “Attila” è un cane (cioè è un' istanza di cane).

Per un essere umano è evidente a tutti che “Attila” è un mammifero, ma, se vogliamo che le macchine siano in grado di comprendere questi fatti, e operare dei ragionamenti, dobbiamo essere in grado di asserire in modo formale la conoscenza che “ogni cane è un mammifero”.

Questo genere di relazioni tra classi è conosciuto come “*gerarchia di classi*”, che non è necessariamente una gerarchia ad albero semplice, nel senso che una classe può essere una sottoclasse di più classi.

È compito dell' utente definire i vincoli sulle classi e sulle proprietà, e le eventuali gerarchie di sottoclasse. Per esempio, la Figura 8 rappresenta graficamente due tipi di fatti:

- Novel è una sottoclasse di `rdfs:Resource` (perché qualunque classe è una risorsa) ed è un' istanza di `rdfs:Class` (che è la “classe di tutte le classi”);
- il libro identificato da `http://.../isbn/000651409X` è un' istanza della classe Novel.



**Figura 8 - Un frammento di RDF Schema**

Secondo la terminologia tradizionale dei sistemi di Knowledge Representation questi due componenti (codificati formalmente in RDF/XML in Figura 9 e Figura 10) sono denominati rispettivamente “*Terminological axioms*” e “*Assertions*” (o T-box e A-box)

```
<rdf:Description rdf:ID="Novel">
  <rdf:type
    rdf:resource= "http://www.w3.org/2000/01/rdf-schema#Class"/>
</rdf:Description>
```

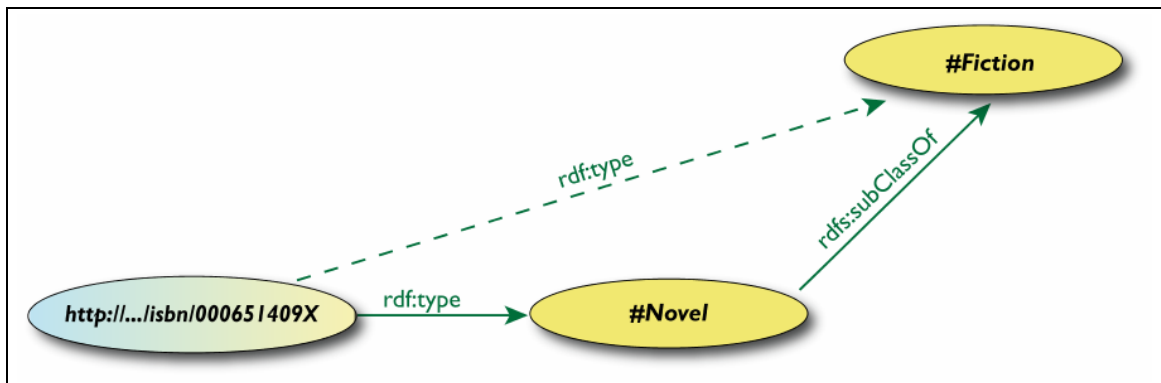
**Figura 9 - Un frammento di RDFS in RDF/XML (lo schema)**



```
<rdf:Description rdf:about="http://.../isbn/000651409X">
  <rdf:type rdf:resource="http://.../bookSchema.rdf#Novel" />
</rdf:Description>
```

**Figura 10 - I dati RDF per una specifica istanza**

Uno degli effetti più importanti di questa modellazione in termini di classi e sottoclassi è l' *ereditarietà* (inheritance), per cui gli elementi di una sottoclasse ereditano le proprietà della superclasse.



**Figura 11 - Un esempio di inferenza delle proprietà in RDF Schema**

Per esempio (vedi Figura 11), una volta definito che i romanzi (**Novel**) sono un tipo di **Fiction**, e che l' oggetto identificato dall' URI: `http://.../isbn/000651409X` è un romanzo (cioè un' istanza della classe **Novel**), ne consegue che è anche un' istanza della classe **Fiction**.

Una volta definita la semantica di una proprietà tra classi, quindi i vincoli per il dominio e il condominio, le loro sottoclassi ereditano tutte le proprietà definite.

Per esempio, supponiamo di aver definito le classi **Persona**, **Artista**, **OggettoArtistico**, **Dipinto**, **Statua**, con le ovvie gerarchie di classe (**Artista** è sottoclasse di **Persona**, e **OggettoArtistico** è superclasse sia di **Dipinto** che di **Statua**). Se definiamo la proprietà **autoreDi** con dominio **Persona** e codominio **OggettoArtistico**, ne consegue che un **Artista** può essere **autoreDi** un **Dipinto** o una **Statua**, perché le sottoclassi ereditano le proprietà delle rispettive superclassi.

In RDFS le proprietà sono definite in modo globale, e non sono ancorate alla definizione della classe, per cui è possibile definire nuove proprietà da applicare a classi già esistenti e definite da altri.

Si noti che è anche possibile definire relazioni gerarchiche tra proprietà: per esempio, **scultoreDi** è una *sottoproprietà* (`rdfs:subPropertyOf`) di **autoreDi**.

## Le ontologie

Il termine ontologia deriva dalla filosofia, dove viene inteso come una spiegazione sistematica dell' essere. Negli anni recenti il termine si è ampiamente diffuso nella comunità del Knowledge Engineering. Esistono diverse definizioni di ontologia, ognuna delle quali mette in evidenza qualche aspetto. Secondo la definizione di [Neches1991]:

*An ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary.*

Da notare che un' ontologia include non solo i termini che sono esplicitamente definiti in essa, ma anche la conoscenza che ne può essere derivata mediante un processo di inferenza.

Secondo la definizione data in Studer et al., che riprende e chiarisce quelle date precedentemente da [Gruber1993] e [Borst1997]:

*An ontology is a formal, explicit specification of a shared conceptualisation. A 'conceptualisation' refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. 'Explicit' means that the type of concepts used, and the constraints on their use are explicitly defined. For example, in medical domains, the concepts are diseases and symptoms, the relations between them are causal and a constraint is that a disease cannot cause itself. 'Formal' refers to the fact that the ontology should be machine readable, which excludes natural language. 'Shared' reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual, but accepted by a group.*

Talvolta il concetto di ontologia viene inteso in modo meno rigido, per cui anche le tassonomie vengono considerate ontologie, perché offrono una concettualizzazione di un dominio su cui c'è il consenso di una vasta comunità. Si distingue spesso tra ontologie *lightweight* (che sono sostanzialmente delle tassonomie) e le ontologie *heavyweight* (che forniscono un modello del dominio più accurato e includono delle restrizioni sulla semantica del dominio). Le prime includono i concetti e le tassonomie di concetti, le relazioni tra concetti, e le proprietà che descrivono i concetti. Le ontologie *heavyweight* aggiungono a questo assiomi e vincoli, che chiariscono il senso inteso (*intended meaning*) dei termini presenti nell' ontologia.

Abbiamo riportato solo alcune delle tante definizioni di ontologia, ma l' esistenza di un gran numero di definizioni non deve far ritenere che possa sorgere confusione sul significato che la comunità scientifica che opera nel settore attribuisce a questo termine. Le varie definizioni enfatizzano di volta in volta alcuni aspetti, ma in realtà forniscono una serie di punti di vista complementari. In alcuni casi la definizione è indipendente dal processo seguito per la costruzione di un' ontologia e il suo utilizzo nelle applicazioni, in altri casi viene invece data rilevanza al processo seguito per svilupparla. Va piuttosto posto l'accento su come le ontologie mirino a catturare la conoscenza *consensuale*, e possano essere condivise e riutilizzate tra applicazioni e gruppi di persone diversi.

Le ontologie vengono costruite in genere mediante un processo cooperativo e distribuito, e utilizzano varie tecniche di modellazione della conoscenza e diversi tipi di linguaggi. Possono perciò essere *molto informali*, *semi-informali*, *semi-formali* o *rigorosamente formali* a seconda che siano espresse in linguaggio naturale, in linguaggio naturale ristretto, in un linguaggio artificiale e definito formalmente, o fornendo una descrizione meticolosa dei termini, utilizzando una semantica formale, teoremi e dimostrazioni di proprietà. Nella classificazione delle ontologie basata sulla ricchezza della loro struttura interna, i vocabolari controllati e i thesauri si collocano nella parte bassa delle ontologie informali, mentre le ontologie in cui vengono espressi dei vincoli sui possibili valori si collocano nella parte alta delle ontologie formali.

È stato osservato come le ontologie semi-formali si siano dimostrate in pratica molto utili per raggiungere diversi obiettivi importanti, in particolare l' *information integration*. Rispetto alle ontologie rigorosamente formali, le ontologie semi-formali sono più diffuse e spesso più utili, perché possono essere realizzate a una scala adeguata

per le applicazioni reali e richiedono uno sforzo di sviluppo certamente minore. La loro diffusione è legata alla necessità di dover rappresentare informazione parziale (quindi incompleta) e talvolta non completamente coerente, in particolare per quanto concerne le asserzioni. Per esempio, la Gene Ontology (<http://www.geneontology.org/>), che è più una nomenclatura e una tassonomia che una vera e propria ontologia, e presenta alcune incoerenze, è largamente usata con molto successo per annotare grosse moli di documenti, supportando così l' interoperabilità e l' integrazione di fonti informative diverse, dimostrando che possono essere sviluppate applicazioni reali aggiungendo un po' di semantica, o venendo a un compromesso con la completezza e il rigore richiesti da una rappresentazione più formale e dalle tecniche di inferenza ("*little semantics goes a long way*" – Jim Hendler).

## OWL

RDF e RDF Schema presentano dei limiti di espressività, in quanto RDF consente unicamente di specificare predicati binari, e RDF Schema consente unicamente di definire gerarchie di classi e proprietà, e di imporre vincoli per dominio e condominio. Applicazioni sofisticate richiedono di poter "ragionare" sui dati. Il Semantic Web deve quindi essere supportato da ontologie, e disporre di un linguaggio che consenta di definire la terminologia usata, le caratteristiche logiche e i vincoli delle proprietà, l' equivalenza dei termini, le cardinalità delle associazioni, etc. [Antoniou2004] Un' ulteriore complessità deriva dal fatto che il Web è intrinsecamente distribuito, e di conseguenza applicazioni diverse possono usare ontologie diverse, o le stesse ontologie, ma espresse in lingue diverse. Il W3C, sfruttando anche i risultati di altri progetti, quali DAML e OIL, ha definito un linguaggio, denominato OWL, che permette di esportare le ontologie in modo interoperabile.

Un buon ontology language dovrebbe avere un certo numero di caratteristiche sofisticate, ma quanto più è espressivo il linguaggio, tanto meno efficiente è il ragionamento, e, in particolar modo nel definire un linguaggio da utilizzare sul Web, occorre trovare un compromesso tra espressività e computabilità. In particolare, OWL avrebbe potuto essere un' estensione di RDFS, utilizzando il significato delle classi e proprietà definite in RDF (`rdfs:Class`, `rdfs:subClassOf`, etc.), aggiungendo le primitive necessarie per supportare la maggior ricchezza espressiva richiesta. In tal modo sarebbe stata mantenuta anche la coerenza con l' architettura a strati del Semantic Web (Figura 3). Purtroppo, questo avrebbe cozzato contro l' esigenza di contenere la complessità computazionale, e assicurare una adeguata efficienza nel ragionamento. In particolare, costrutti come `rdfs:Class` e `rdf:Property` (la classe di tutte le classi e la classe di tutte le proprietà), pur essendo molto espressivi, avrebbero portato a una complessità computazionale fuori controllo.

Per questo motivo OWL offre tre sottolinguaggi, di crescente potere espressivo:

**OWL Lite** indicato principalmente per utenti che hanno bisogno di rappresentare classificazioni gerarchiche e vincoli semplici. Consente una migrazione agevole per thesauri e altre tassonomie. Ha una complessità formale inferiore a OWL DL, e non consente alcuni costrutti, come cardinalità arbitrarie o statement di disgiunzione.

**OWL DL** (OWL Description Logic) indicato per gli utenti che desiderano la massima potenza espressiva garantendo comunque che tutte le conclusioni siano computabili (*computational completeness*) e concluse in un tempo finito (*decidability*). OWL DL offre un ragionevole supporto per il ragionamento

(*reasoning*), ma non è perfettamente compatibile con RDF. Per ottenere un documento OWL DL corretto da un documento RDF, è necessario ricorrere ad alcune restrizioni e ad alcune estensioni. Tuttavia, qualunque documento OWL DL corretto è un documento RDF corretto.

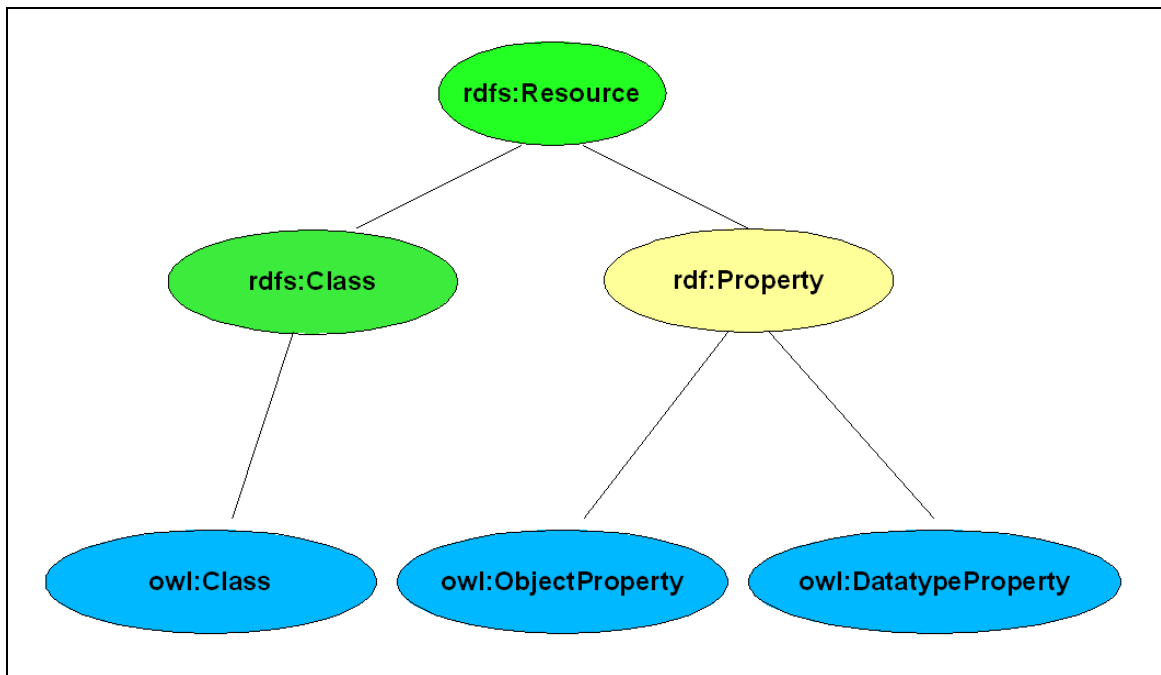
*OWL Full* indicato per gli utenti che desiderano la massima potenza espressiva e la libertà sintattica di RDF, senza garanzie sui tempi di computazione. Qualunque documento RDF corretto è un documento OWL Full corretto, perché OWL Full è perfettamente compatibile con RDF, sia sotto l'aspetto sintattico che sotto quello semantico. Tuttavia, il linguaggio è indecidibile, non esistono strumenti che supportino il ragionamento in maniera efficiente o completa, e difficilmente sarà supportato nella sua interezza da software che implementano il ragionamento.

Ognuno di questi linguaggi è un' estensione del precedente, sia in termini di ciò che può essere espresso che in termini della validità delle conclusioni.

### Alcune caratteristiche di OWL

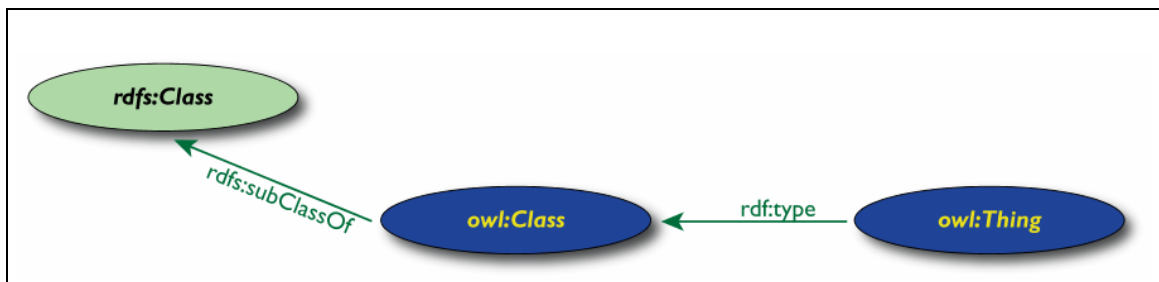
Mentre in RDFS è possibile unicamente specializzare le classi esistenti in sottoclassi, in OWL è possibile *costruire* classi a partire da quelle esistenti, mediante enumerazione, intersezione, complemento, restrizioni delle proprietà.

Alcune primitive di modellizzazione di OWL sono delle specializzazioni delle corrispondenti primitive in RDF/RDFS (vedi Figura 12).



**Figura 12 - Le relazioni di sottoclasse tra alcune primitive RDF/RDFS e OWL**

Più in dettaglio, le classi vengono definite utilizzando l' elemento `owl:Class`. Vi sono poi due classi predefinite: `owl:Thing` e `owl:Nothing`, corrispondenti rispettivamente alla classe più generale, di cui tutte le classi sono sottoclasse, e all' insieme vuoto, di cui tutte le classi sono superclasse. per distinguere le classi dalle istanze (in Figura 13 la rappresentazione grafica con evidenziazione delle proprietà RDF/RDFS utilizzate).



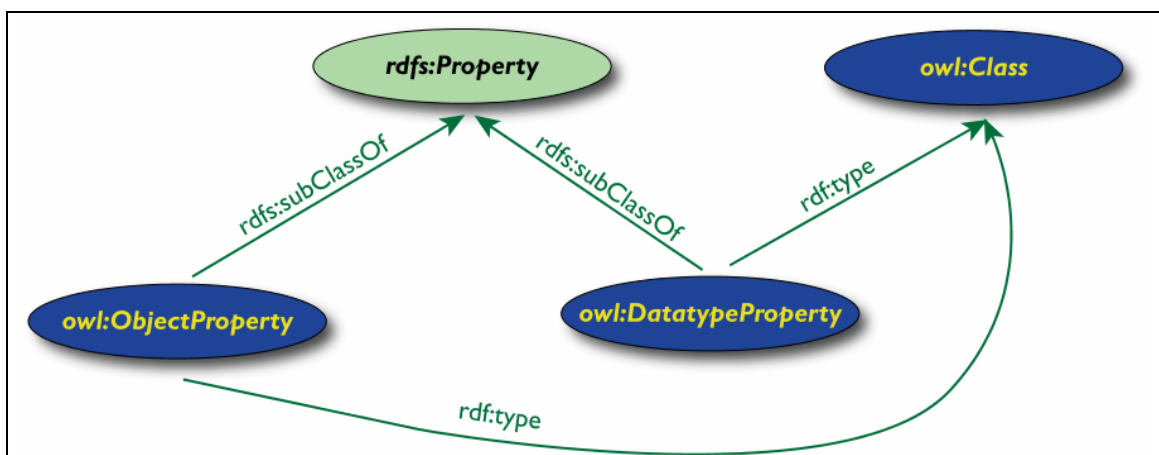
**Figura 13 - Le classi in OWL**

Le proprietà sono distinte in (Figura 14):

- `owl:ObjectProperty`, che consentono di mettere in relazione tra di loro gli oggetti (es. possiede, insegna, etc.);
- `owl:DatatypeProperty`, che consentono di mettere in relazione gli oggetti con i valori (es. numeroDiTelefono, nome, dataDiNascita, etc.). Non esistono tipi predefiniti, ma è possibile utilizzare i data type di XML Schema, restando così aderenti all'architettura a strati del Semantic Web (Figura 3).

È possibile caratterizzare il comportamento delle proprietà, specificando, per esempio:

- la cardinalità (`owl:minCardinality`, `owl:maxCardinality`);
- se è simmetrica (`owl:SymmetricProperty`);
- se è transitiva (`owl:TransitiveProperty`);
- se è funzionale, cioè ha al massimo un valore per ogni oggetto, come l'età, l'altezza, il peso, etc. (`owl:FunctionalProperty`);
- se è "funzionale inversa", cioè è una proprietà per la quale due oggetti diversi non possono avere lo stesso valore (`owl:InverseFunctionalProperty`). Esempi potrebbero essere: "èCodiceFiscaleDi", "èNumeroDiMatricolaDi", etc.



**Figura 14 - Le proprietà in OWL**

È anche possibile, e può avvenire facilmente con ontologie distribuite sul Web, che una classe o una proprietà siano definite con nomi diversi, e che sia opportuno specificare che di fatto sono identiche. OWL fornisce due elementi per definire l'equivalenza tra classi o proprietà: `owl:equivalentClass` e `owl:equivalentProperty` (Figura 15).

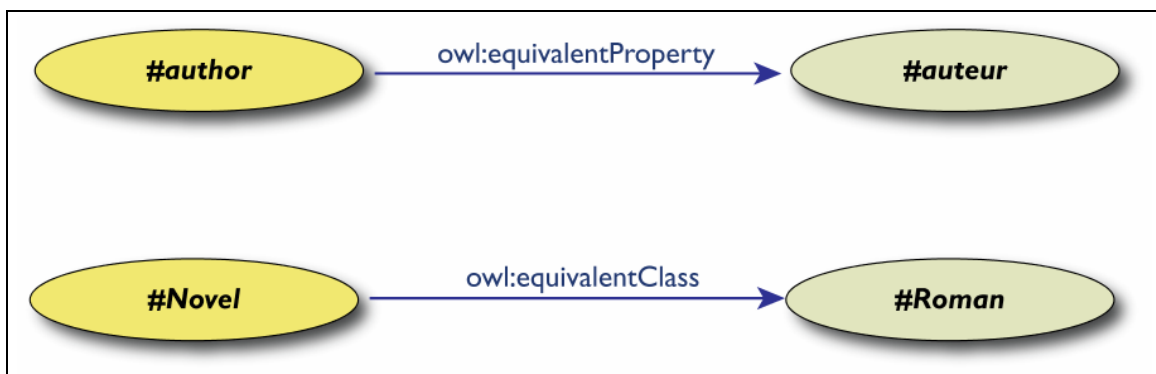


Figura 15 - Un esempio di equivalenza

## Conclusioni

L'obiettivo principale del Semantic Web è consentire alle macchine di estrarre la conoscenza disponibile sul Web, spesso disponibile in formati eterogenei, e combinarla per poter estrarre nuova conoscenza. Questo processo è possibile solo se si riesce a rappresentare, esportare e condividere la conoscenza, mediante una rappresentazione indipendente dallo specifico ambiente operativo. Le tecnologie del W3C (RDF, RDFS, OWL, SPARQL, etc.) consentono appunto di raggiungere questo obiettivo (Figura 16).

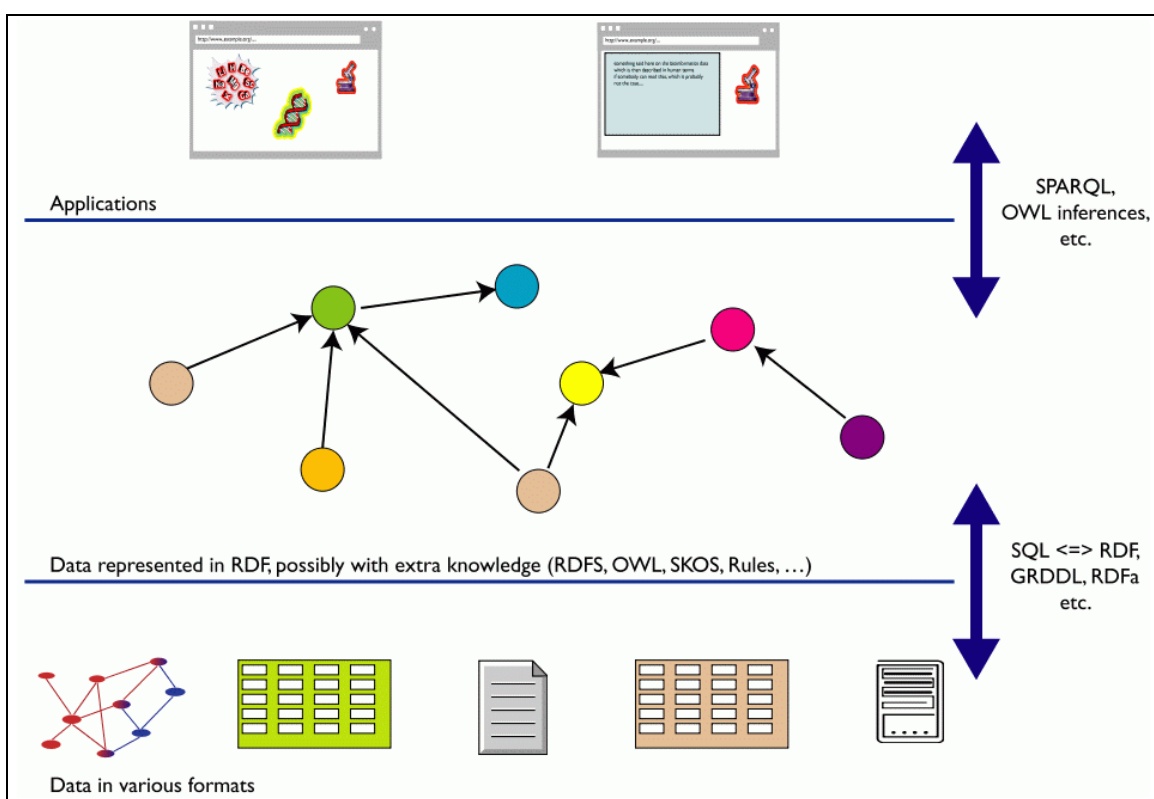


Figura 16 - Il ruolo delle tecnologie W3C per combinare informazioni provenienti da fonti eterogenee

## Ringraziamenti

Ringrazio il W3C Team e Ivan Herman, W3C Semantic Web Activity Lead, dalle cui presentazioni ho preso molto materiale (in particolare molte delle figure). Un ringraziamento anche al collega Massimo Martinelli, che mi ha aiutato nella preparazione di questo documento con discussioni chiarificatrici.

## Bibliografia

- [Antoniou2004] Grigoris Antoniou, Frank van Harmelen, *A Semantic Web Primer*, The MIT Press, Cambridge, Massachusetts, April 2004, ISBN 0-262-01210-3.
- [Borst1997] Borst WN. *Construction of Engineering Ontologies*. Centre for Telematica and Information Technology, University of Twente. Enschede, The Netherlands, 1997.
- [DC] The Dublin Core Home Page, URL: <http://dublincore.org/>
- [Gómez-Pérez2004] Asunción Gómez-Pérez, Mariano Fernández-López, Oscar Corcho, *Ontological Engineering*, Springer-Verlag (2004), ISBN 1-85233-551-3
- [Gruber1993] Gruber TR: A translation approach to portable ontology specification. *Knowledge Acquisition* 5(2):199-220
- [IRDF] Introduction to RDF Metadata, W3C NOTE 1997-11-13, Ora Lassila, URL:<http://www.w3.org/TR/NOTE-rdf-simple-intro>
- [Miller1998] Miller E.: An Introduction to the Resource Description Framework, D-Lib Magazine, May 1998, <http://www.dlib.org/dlib/may98/miller/05miller.html>
- [Naming] Naming and Addressing: URIs, URLs, ..., <http://www.w3.org/Addressing/>
- [Neches1991] Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T., and Swartout, W. R. (1991) in *Enabling Technology for Knowledge Sharing*. *AI Magazine*, 12(3) pp. 16-36
- [RDFMSS] O.Lassila, R.Swick: *Resource Description Framework (RDF) Model and Syntax Specification*, W3C Recommendation 22 February 1999, <http://www.w3.org/TR/REC-rdf-syntax>
- [RDFSS] *Resource Description Framework (RDF) Schema Specification*, W3C Recommendation 03 March 1999, <http://www.w3.org/TR/1999/PR-rdf-schema-19990303>
- [SemWeb] <http://www.semanticweb.org/>
- [Signore2001] Signore O.: *Il ruolo centrale di XML nell'evoluzione del Web*, XML Day Milan, Conference proceedings, Milan, September 21 (find this and other similar papers at: <http://www.w3c.it/papers/>)
- [Signore2003] Signore, Oreste: *Strutturare la conoscenza: XML, RDF, Semantic Web - Clinical Knowledge 2003* (1st edition) - Udine, 20-21 September 2003 <http://www.w3c.it/papers/ck2003.pdf>, <http://www.w3c.it/talks/ck2003/>
- [Signore2005] Signore, Oreste, *Semantic Web: il futuro è già qui?* - 10th Knowledge Management Forum - Siena, 24-25 Novembre 2005, <http://www.w3c.it/papers/km10.pdf>
- [Signore2007] Signore O. - *Knowledge creation: integrazione di HTML e Semantic Web*. In: The 12th Knowledge Management Forum (<http://www.jekpot.com/pagine/km12.htm>) (Milano, 26-28 novembre 2007). Proceedings, pp. 1 - 8. Jekpot Srl (ed.). Jekpot Srl, 2007., <http://www.w3c.it/papers/km12.pdf>
- [TBL1997] Tim Berners-Lee: *Metadata architecture*, (1997), <http://www.w3.org/DesignIssues/Metadata.html>
- [TBL1998] Tim Berners-Lee: *Semantic Web Road Map*, (1998), <http://www.w3.org/DesignIssues/Semantic.html>

- [TBL1999] Tim Berners-Lee: *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*, HarperSanFrancisco (1999), ISBN 0-06-251587-X
- [TBL2001] Berners-Lee T., Hendler J., Lassila O.: *The Semantic Web*, Scientific American, May 2001, <http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html>
- [W3C] World Wide Web Consortium Home Page, <http://www.w3.org>
- [XML] Extensible Markup Language (XML) , <http://www.w3.org/XML/>
- [XML1.0] Extensible Markup Language (XML) 1.0 (Second Edition) W3C Recommendation 6 October 2000, <http://www.w3.org/TR/2000/REC-xml-20001006>
- [XMLns] Namespaces in XML - World Wide Web Consortium 14-January-1999 <http://www.w3.org/TR/REC-xml-names/>
- [XMLschema0] XML Schema Part 0: Primer - W3C Recommendation - 2 May 2001 <http://www.w3.org/TR/xmlschema-0/>
- [XMLschema1] XML Schema Part 1: Structures - W3C Recommendation - 2 May 2001 <http://www.w3.org/TR/xmlschema-1/>
- [XMLschema2] XML Schema Part 2: Datatypes - W3C Recommendation - 2 May 2001 <http://www.w3.org/TR/xmlschema-2/>