

Introduzione a RDF (Resource Description Framework)

Abbiamo visto che uno degli obiettivi del Web Semantico è quello di rendere le risorse comprensibili e utilizzabili da agenti software.

Un ruolo molto importante in questo panorama è ricoperto dai Metadati, ovvero le informazioni aggiunte ai dati, per renderli più facilmente utilizzabili.

Definendo una sintassi ed una struttura per rappresentare i Metadati abbiamo a disposizione un modo per descrivere le informazioni, relative ad una risorsa sul Web, che possono essere comprese da una macchina.

Mentre il linguaggio XML permette di inserire metadati all'interno dei documenti attraverso i tag il linguaggio RDF (Resource Description Framework) è lo strumento proposto dal W3C per descrivere i metadati relativi ad una risorsa, mantenedoli all'esterno e separati dalla risorsa stessa. Questo è necessario quando si hanno risorse non di testo o "opache" come immagini, filmati, file audio.

L'utilizzo della descrizione semantica delle risorse può portare numerosi benefici in molti settori del Web: ad esempio si potrebbero sviluppare motori di ricerca più efficienti in grado di basare la loro ricerca non solo sull'occorrenza di una determinata parola all'interno di un documento ma anche in base alla caratterizzazione semantica del documento stesso; oppure si potrebbe realizzare un agente software per il filtraggio dei contenuti di una risorsa in funzione di determinati criteri impostati dall'utente.

La specifica di RDF è costituita da due componenti: *RDF Model and Syntax* e *RDF Schema*.

La prima componente riguarda la definizione del data model RDF (modello dei dati), tramite il quale descrivere le risorse, e della sintassi XML utilizzata per specificare questo modello.

RDF Schema invece permette di definire il significato e le caratteristiche delle proprietà e delle relazioni che esistono tra queste e le risorse descritte nel data model RDF.

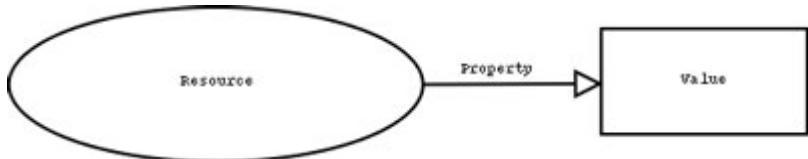
Una risorsa, identificata univocamente da un URI, viene descritta utilizzando il data model RDF.

Questo modello è basato su tre oggetti:

- Resource (risorsa): indica ciò che viene descritto mediante RDF e può essere una risorsa Web (ad esempio una pagina HTML, un documento XML o parti di esso) o anche una risorsa esterna al Web (ad esempio un libro, un quadro, etc.);
- Property (proprietà): indica una proprietà, un attributo o una relazione utilizzata per descrivere una risorsa. Il significato e le caratteristiche di questa componente vengono definite tramite RDF Schema;
- Statement (espressione): è l'elemento che descrive la risorsa ed è costituito da un soggetto (che rappresenta la Resource), un predicato (che esprime la Property) e da un oggetto (chiamato Value) che indica il valore della proprietà.

Lo statement RDF che descrive la risorsa è del tipo: <soggetto> HAS <predicato> <oggetto>.

Nella seguente figura è raffigurato un modo per esprimere lo statement RDF in forma grafica utilizzando un grafo etichettato orientato nel quale la risorsa è rappresentata da un'ellisse, la proprietà da un arco orientato, che parte dalla risorsa e punta all'oggetto, e l'oggetto da un rettangolo.



Per cercare di comprendere meglio il data model RDF consideriamo un esempio in cui andiamo a descrivere, mediante RDF, il metadato riguardante l'autore di una pagina HTML.

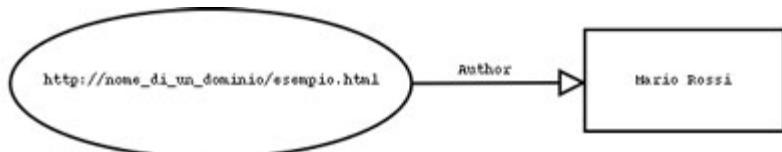
L'informazione che vogliamo descrivere è la seguente: Mario Rossi è l'autore della pagina http://nome_di_un_dominio/esempio.html

Applicando il data model RDF (basato su Resource, Property e Value) otteniamo la seguente tabella:

Resource (soggetto)	http://nome_di_un_dominio/esempio.html
Property (predicato)	author
Value (oggetto)	Mario Rossi

Lo statement RDF che descrive il nostro esempio è quindi:

`http://nome_di_un_dominio/esempio.html HAS author Mario Rossi` ed in forma grafica diventa:

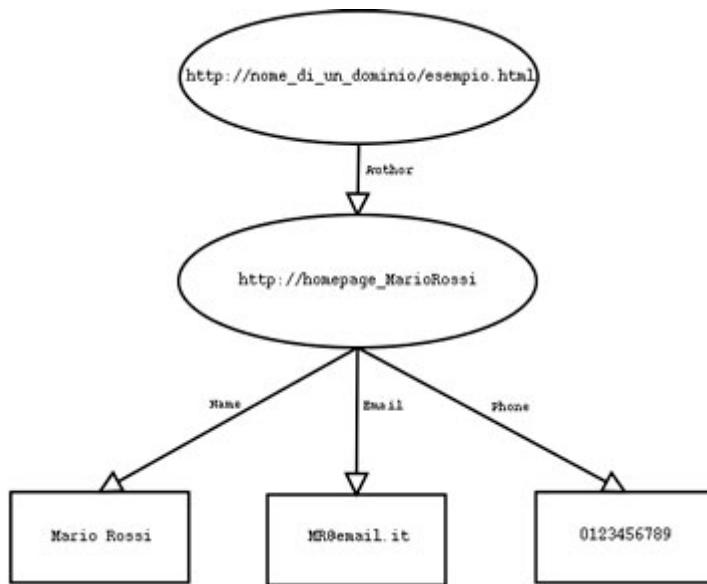


L'oggetto di uno statement RDF può essere a sua volta una risorsa, consentendo in questo modo di descrivere in maniera più approfondita il valore della proprietà.

Consideriamo l'esempio precedente e vediamo di aggiungere maggiori informazioni riguardanti l'autore della pagina HTML (ad esempio email e numero di telefono).

In questo caso dobbiamo aggiungere allo statement precedente, la descrizione della risorsa autore che può essere identificata in maniera univoca utilizzando ad esempio l'URI della sua Homepage.

Lo statement RDF espresso in forma grafica diventa quindi:



In forma testuale si ha uno statement RDF di questo tipo: `http://nome_di_un_dominio/esempio.html HAS author http://homepage_MarioRossi/ AND http://homepage_MarioRossi/ HAS name Mario Rossi AND email MR@email.it AND phone 0123456789`

Il data model RDF offre un mezzo astratto e concettuale per descrivere una risorsa e per facilitare la definizione e lo scambio dei metadati, RDF prevede l'utilizzo di una sintassi basata su XML per specificare in maniera concreta gli statement RDF.

Il primo esempio di statement RDF può essere espresso, utilizzando la sintassi XML, nel seguente modo:

```

<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:a="http://nome_di_un_dominio/schema_autore/">
  <rdf:Description about="http://nome_di_un_dominio/esempio.html">
    <a:author>
      Mario Rossi
    </a:author>
  </rdf:Description>
</rdf:RDF>

```

L'elemento `<rdf:RDF>` racchiude la definizione dello statement RDF ed al suo interno troviamo la definizione di due Namespace: il primo è relativo al Namespace RDF, mentre il secondo Namespace contiene l'URI che identifica lo schema RDF utilizzato per descrivere la semantica e le convenzioni che regolano l'utilizzo delle proprietà presenti nello statement.

La descrizione del metadato è contenuta all'interno dell'elemento `<rdf:Description>` ed il suo attributo `about` identifica la risorsa alla quale si riferisce il metadato stesso.

La proprietà dello statement è descritta utilizzando il tag `<a:author>`, secondo le regole che sono espresse nel relativo schema RDF.

Il secondo esempio di statement RDF, espresso secondo la sintassi XML, diventa:

```

<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:a="http://nome_di_un_dominio/schema_autore/">
  <rdf:Description about="http://nome_di_un_dominio/esempio.html">
    <a:author rdf:resource="http://homepage_MarioRossi/">
  </rdf:Description>

```

```

<rdf:Description about="http://homepage_MarioRossi/">
  <a:name>Mario Rossi</a:name>
  <a:email>MR@email.it</a:email>
  <a:phone>0123456789</a:phone>
</rdf:Description>
</rdf:RDF>

```

In questo caso, all'interno dello statement RDF abbiamo la definizione di due risorse (identificate dai due elementi `<rdf:Description>`) che sono messe in relazione attraverso l'uso dell'attributo `rdf:resource` presente nell'elemento `<a:author>`.

In questo modo la descrizione della seconda risorsa (quella relativa all'autore del documento) viene assegnata come valore della proprietà `author` della prima risorsa.

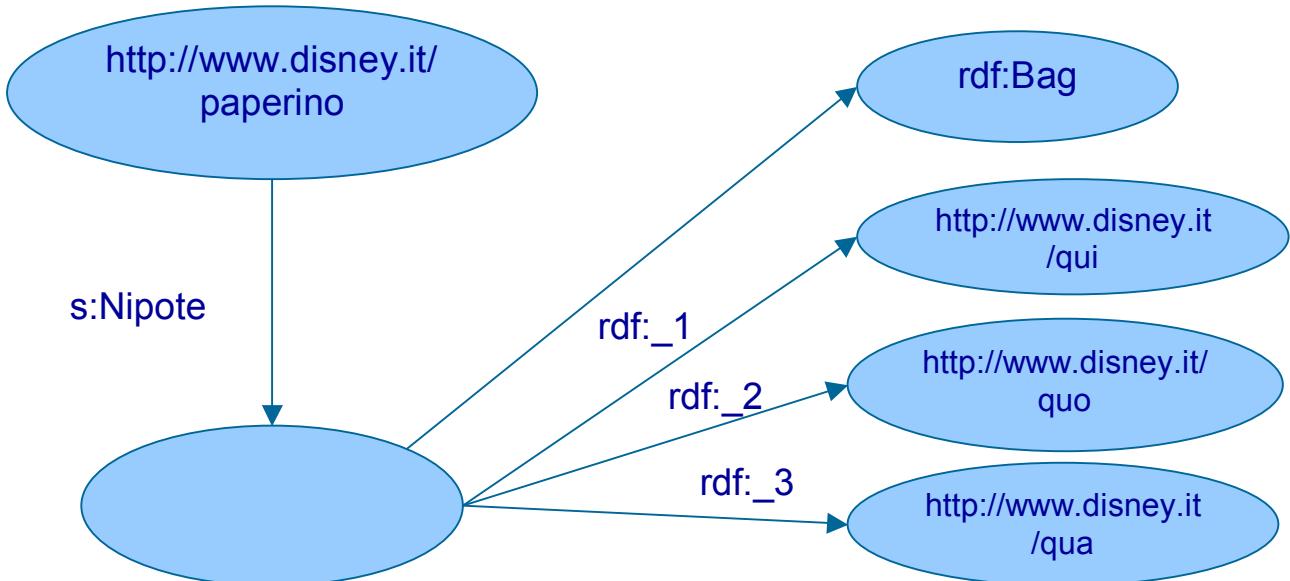
Contenitori

A volte è importante fare riferimento ad un insieme di risorse (ad esempio, se un documento è stato creato da più autori, o se lo stesso autore ha fatto più di un documento, ecc.)

In questo caso tali risorse devono essere inserite all'interno di un contenitore che sarà l'oggetto dello statement.

RDF definisce tre tipi di contenitori:

- ◆ **Bag**. E' un insieme con ripetizioni. L'ordine non è rilevante.
- ◆ **Sequence**. E' un insieme con ripetizioni ed un ordine definito tra le risorse presenti.
- ◆ **Alternative**. E' un insieme senza ripetizioni in cui può essere scelto uno solo degli elementi. L'ordine degli elementi può essere usato per esprimere preferenza.



```

<rdf:Description rdf:about="http://www.disney.it/paperino">
  <s:Nipote>
    <rdf:Description>
      <rdf:type rdf:resource=
        "http://www.w3.org/1999/02/22-rdf-syntax-ns#Bag" />

```

```

<rdf:_1 rdf:resource="http://www.disney.it/qui"/>
<rdf:_2 rdf:resource="http://www.disney.it/quo"/>
<rdf:_3 rdf:resource="http://www.disney.it/qua"/>
</rdf:Description>
</s:Nipote>
</rdf>Description>

```

Analogamente si useranno i tipi rdf:Seq per le sequenze e rdf:Alt per le alternative.

Dublin Core

Il Dublin Core è uno schema di meta informazioni ideato per assegnare etichette ragionevoli alle risorse della rete.

Si chiama Dublin Core perché è considerato il nucleo (core) delle meta-informazioni interessanti per qualunque risorsa, e perché è nato da un'iniziativa di bibliotecari, archivisti, fornitori di contenuto e esperti di markup svoltasi nel 1995 a Dublino.

Dublin Core è indipendente da qualunque sintassi, ma ben si adatta a RDF.

Dublin Core versione 1 ha introdotto esattamente quindici categorie di meta-informazioni utili per la catalogazione di risorse di rete.

La versione 2 ha aggiunto un meccanismo di sottoclassi (detti *qualificatori*) delle categorie, ed ha introdotto un elenco iniziale di qualificatori.

Le quindici categorie descrivono meta-informazioni di tre tipi:

Contenuto	Proprietà intellettuale	Istanza
Title	Creator	Date
Subject	Publisher	Format
Description	Contributor	Identifier
Type	Rights	Language
Source		
Relation		
Coverage		

I qualificatori permettono di specificare ulteriormente informazioni di queste categorie, secondo questi criteri:

- ◆ **Raffinamento dello schema:** fornisce alcuni significati più precisi sui termini. Ad esempio, "Date" ha come qualificatori: "created", "valid", "available", "issued", "modified").
- ◆ **Supporto per codifiche specifiche:** permette di usare i valori di particolari codifiche all'interno del Dublin Core. Ad esempio, "Subject" ha come qualificatori: "LCSH" (Library of Congress Subject Headings), "MeSH" (Medical Subject Headings), "DDC" (Dewey Decimal Classification), ecc.

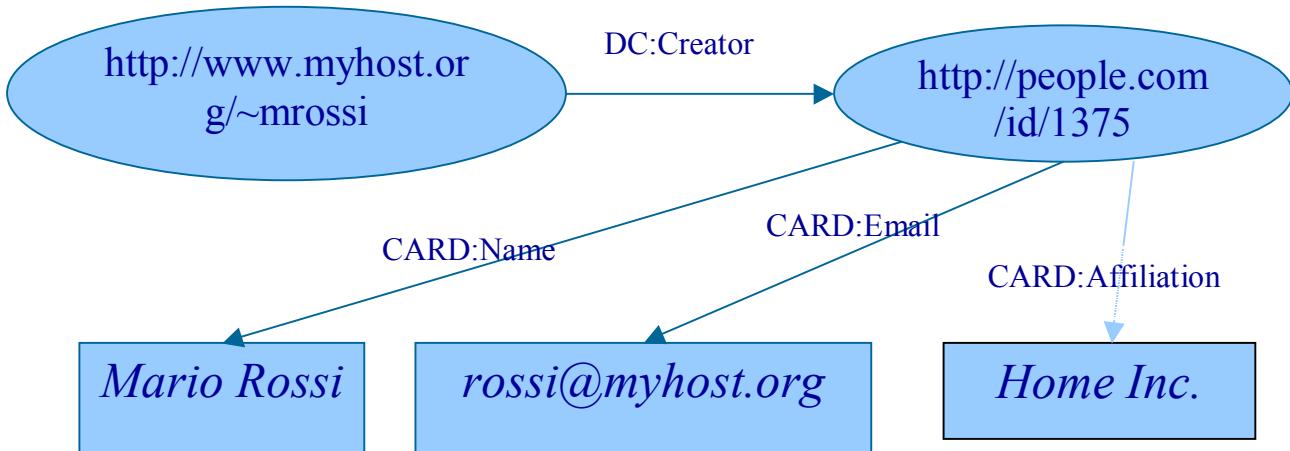
I metadata del Dublin Core sono anche parte dei metadata usati per descrivere i learning objects (IEEE Lom e Scorm).

Un esempio di metadati:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/metadata/dublin_core#">
  <rdf:Description rdf:about="http://www.dlib.org">
    <dc:Title>D-Lib Program</dc:Title>
    <dc:Description>
      The D-Lib program supports the community of people
      with research interests in digital libraries and
      electronic publishing.
    </dc:Description>
    <dc:Publisher>
      Corporation For National Research Initiatives
    </dc:Publisher>
    <dc:Date>1995-01-07</dc:Date>
    <dc:Subject><rdf:Bag>
      <rdf:_1>Research; statistical methods</rdf:_1>
      <rdf:_2>Education, research, related topics</rdf:_2>
      <rdf:_3>Library use Studies</rdf:_3>
    </rdf:Bag></dc:Subject>
    <dc>Type>World Wide Web Home Page</dc>Type>
    <dc:Format>text/html</dc:Format>
    <dc:Language>en</dc:Language>
  </rdf:Description>
</rdf:RDF>
```

Namespaces

Invece che usare delle proprietà che abbiamo definito noi come autore, nome, Email possiamo riferirci a dei namespace per dare delle definizioni univoche. Possiamo usare il Dublin Core per sostituire Autore con Creator nella definizione delle Business Card (per realizzare biglietti da visita elettronici).



```
<?xml:namespace ns = "http://www.w3.org/RDF/RDF/" prefix = "RDF" ?>
<?xml:namespace ns = "http://purl.oclc.org/DC/" prefix = "DC" ?>
<?xml:namespace ns = "http://person.org/BusinessCard/" prefix = "CARD" ?>

<RDF:RDF>
  <RDF:Description RDF:HREF = "http://www.myhost.org/~mrossi ">
    <DC:Creator RDF:HREF = " http://people.com/id/1375"/>
```

```
</RDF:Description>

<RDF:Description ID="Creator_001">
  <CARD:Name>Mario Rossi</CARD:Name>
  <CARD:Email>rossi@myhost.org</CARD:Email>
  <CARD:Affiliation>Home Inc.</CARD:Affiliation>
</RDF:Description>
</RDF:RDF>
```

Il prefisso DC viene usato come abbreviazione del Dublin Core. Il prefissi CARD viene usato come abbreviazione del Business Card. Quindi in un documento RDF si può fare riferimento a più namespace. Una definizione come quella di “Creator” del Dublin Core può essere ampliata ricorrendo ad altre definizioni come “Name”, “Email” e “Affiliation” di Business Card. Possiamo pensare a Dublin Core e Business Caed come a due autorità esterne che fissano la semantica di un vocabolario controllato.

Reificazione

Come è possibile fornire meta-informationi su una meta-informatione? Ad esempio come posso esprimere la frase «*Andrea afferma che Mario Rossi è l'autore della risorsa “http://www.myhost.org/~mrossi”*»?

Questo in breve significa attribuire la proprietà «*afferma*» allo statement «*Mario Rossi è l'autore della risorsa “http://www.myhost.org/~mrossi”*». Occorre pertanto considerare la meta-informatione come una risorsa da descrivere.

Questa procedura si chiama *reificazione* (riduzione ad oggetto) della asserzione (o statement). Dopo avere reificato l'asserzione potrà esprimere ulteriori proprietà su di essa.

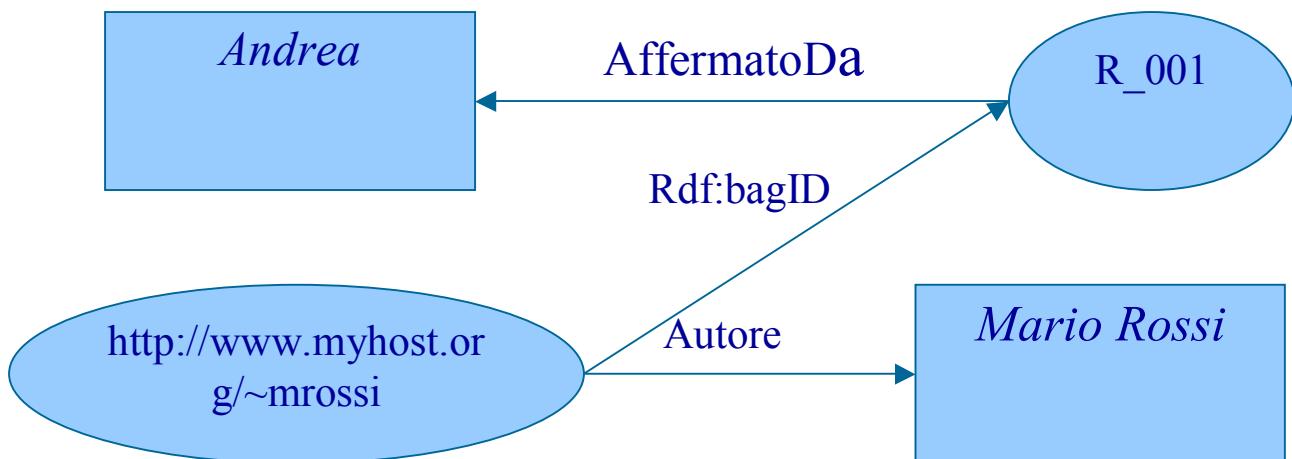
L'attributo rdf:bagID permette di considerare uno statement esplicito come se fosse reificato.

La stessa description può avere un rdf:ID, utile per indicarla come fonte di meta-informationi, ed un rdf:bagID utile per esprimere meta-informationi su di essa.

La frase precedente è equivalente a:

```
<rdf:Description rdf:about="http://www.myhost.org/~mrossi"
    rdf:bagID="#R_001">
    <s:Autore>Mario Rossi</s:Autore>
</rdf:Description>

<rdf:Description rdf:about="#R_001">
    <s:AffermatoDa>Andrea</s:AffermatoDa>
</rdf:Description>
```



Linguaggi di query per RDF

Il linguaggio RDQL permette di esprimere query sui documenti RDF usando una sintassi simile a quella di SQL.

Una query RDQL ha la forma:

```
SELECT vars
FROM documents
WHERE Expressions
AND Filters
USING Namespace declarations
```

Vediamo un esempio

```
SELECT ?x,?y
FROM <http://example.com/sample.rdf>
WHERE (?x,<dc:name> ,?y)
USING dc for <http://www.dc.com#>
```

Questa query restituisce le tuple $x? y?$ che indicano le risorse e le loro proprietà `dc:name`

Un'altro esempio: selezionare le risorse il cui DC:Creator si chiama (CARD:name) "Rossi"
La parte AND indica un filtro.

```
SELECT ?x
FROM <doc.rdf>
WHERE (?x,<DC:creator> ,?y) ,
      (?y,<CARD:name> ,?z)
AND ?z=="Rossi"
USING DC for <http:.....
USING CARD for <http:
```

RDF Schema

Il modello di RDF non permette di effettuare **validazione** di un valore o **restrizione** di un dominio di applicazione di una proprietà. Questo compito è svolto da *RDF Schema*.

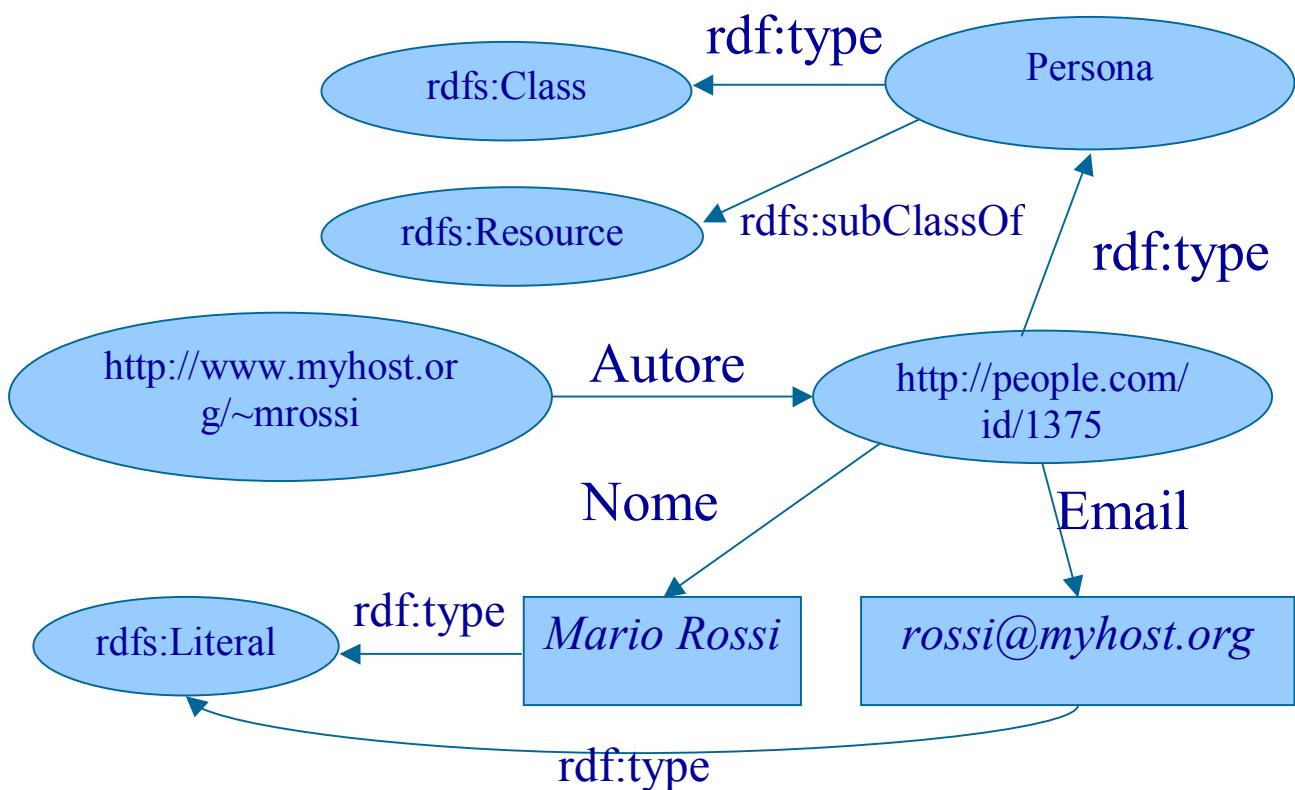
A differenza di XML Schema o di un DTD, RDF Schema non vincola la struttura del documento, ma fornisce informazioni utili all'interpretazione del documento stesso.

Fornisce un meccanismo di base per un sistema di tipizzazione da utilizzare in modelli RDF.

Lo schema è definito in termini di RDF stesso.

RDF Schema definisce un insieme di risorse RDF da usare per descrivere caratteristiche di altre risorse e proprietà RDF.

- ◆ *rdfs:Resource* Tutto ciò che viene descritto in RDF è detto *risorsa*. Ogni risorsa è istanza della classe rdfs:Resource.
- ◆ *rdfs:Literal* Sottoclasse di rdfs:Resource, rappresenta un letterale, una stringa di testo.
- ◆ *rdf:Property* Rappresenta le proprietà. È sottoclasse di rdfs:Resource.
- ◆ *rdfs:Class* Corrisponde al concetto di *tipo* e di *classe* della programmazione object-oriented. Quando viene definita una nuova classe, la risorsa che la rappresenta deve avere la proprietà rdf:type impostata a rdfs:Class.
- ◆ *rdfs:subClassOf* Specifica la relazione di ereditarietà fra classi. Questa proprietà può essere assegnata solo a istanze di rdfs:Class. Una classe può essere sottoclasse di una o più classi (*ereditarietà multipla*).
- ◆ *rdfs:subPropertyOf* Istanza di rdf:Property, è usata per specificare che una proprietà è una specializzazione di un'altra. Ogni proprietà può essere la specializzazione di zero o più proprietà.
- ◆ *rdfs:seeAlso* Specifica una risorsa che fornisce ulteriori informazioni sul soggetto dell'asserzione.
- ◆ *rdfs:isDefinedBy* E' sottoproprietà di rdfs:seeAlso e indica una risorsa che definisce il soggetto di un'asserzione
- ◆ I predicati più utilizzati per esprimere vincoli su altre proprietà sono i seguenti:
 - ◆ *rdfs:range* (codominio) Usato come predicato di una risorsa r, indica le classi che saranno oggetto di un'asserzione che ha r come predicato.
 - ◆ *rdfs:domain* (dominio) Usato come predicato di una risorsa r, indica le classi (soggetto) a cui può essere applicata r.



Nell'esempio si dice che `Persona` è di tipo classe, ed è sottoclasse di `rdfs:resource`. (Tutto è sottoclassificazione di risorsa)

<http://people.com/id/1375> è di tipo `Persona`.

Il suo nome ed il suo Email sono di tipo Literal (stringhe).

Gli schemi RDF vengono usati da linguaggi per la descrizione di ontologie come vedremo più avanti.

Per approfondire ulteriormente l'analisi su RDF si rimanda alla lettura delle specifiche delle due componenti di RDF: RDF Model and Syntax (<http://www.w3.org/TR/REC-rdf-syntax/>) e RDF Schema (<http://www.w3.org/TR/rdf-schema/>).