



RDF

Resource Description Framework

Sommario

1) Cos'è l'RDF

- RDF Model and Syntax
- RDF Schema

2) Il data model RDF

- definizione di risorsa, proprietà e statement
- esempio 1
- esempio 2

3) Combinazione RDF – XML

4) I namespace

5) I contenitori

- tipi di contenitori
- sintassi per i contenitori

6) Perché RDF /XML è meglio di XML

Cos'è il Resource Description Framework

Il Resource Description Framework descrive un linguaggio utile ai metadati. I metadati hanno una funzione essenziale nell'individuazione, nel reperimento e nel trattamento delle informazioni. Un esempio classico è l'insieme dei riferimenti bibliografici di una pubblicazione (titolo, autore, editore).

L'obiettivo è la definizione di un sistema per la specificazione dei metadati associati alle risorse informative. Il risultato di tale attività ha condotto il W3C alla realizzazione di RDF, un metalinguaggio basato sulla sintassi XML. Esso permette sia di associare informazioni a dati, sia di definire lo schema di classificazione

RDF Model and Syntax

L' RDF Model and Syntax definisce il **data model RDF** (modello dei dati), che descrive le risorse, e la sintassi XML utilizzata per specificare questo modello.

RDF Schema

L' RDF Schema 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.

RDF Schema fornisce un insieme di risorse e proprietà predefinite. L'insieme delle risorse e delle proprietà di base delle risorse è detto **vocabolario** dell'RDF Schema.

Il data model RDF

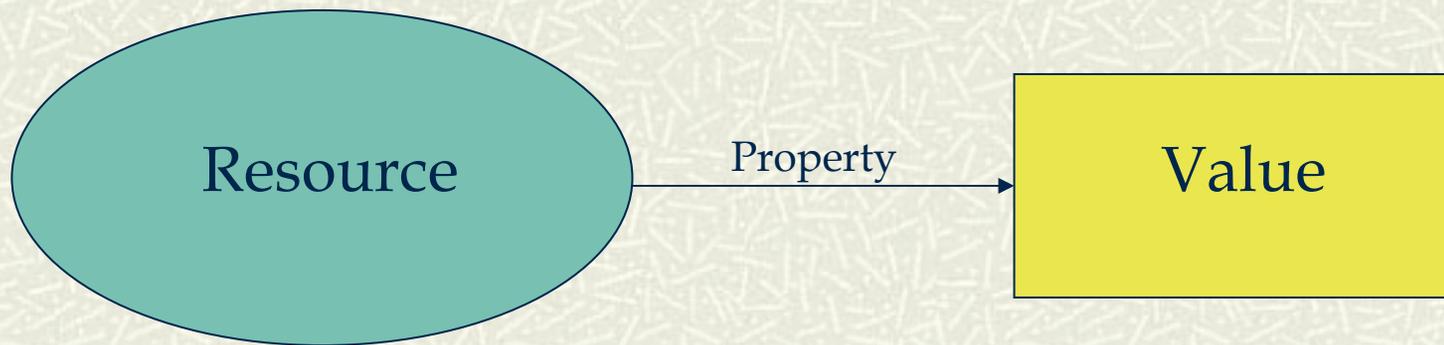
Questo modello è basato su tre oggetti:

- 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 (asserzione): 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>.



Esempio 1

L'informazione che vogliamo descrivere è la seguente: Elena Ferrari è l'autrice della pagina http://nome_di_un_dominio/esempio.html.

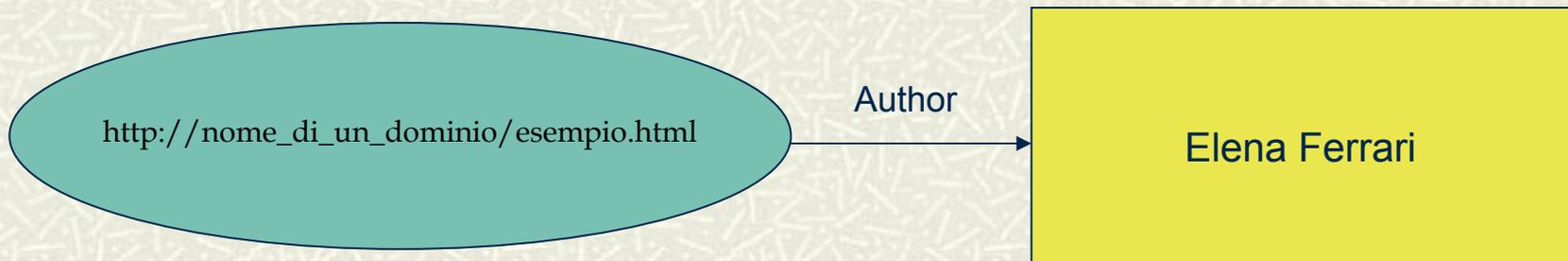
Applicando il data model RDF (basato su Resource, Property e Value) otteniamo la seguente tabella:

Resource (soggetto)	http://nome_di_un_dominio/esempio.html
Property (predicato)	Author
Value (oggetto)	Elena Ferrari

Lo statement RDF che descrive il nostro esempio è quindi:

http://nome_di_un_dominio/esempio.html

HAS author Elena Ferrari ed in forma grafica diventa:





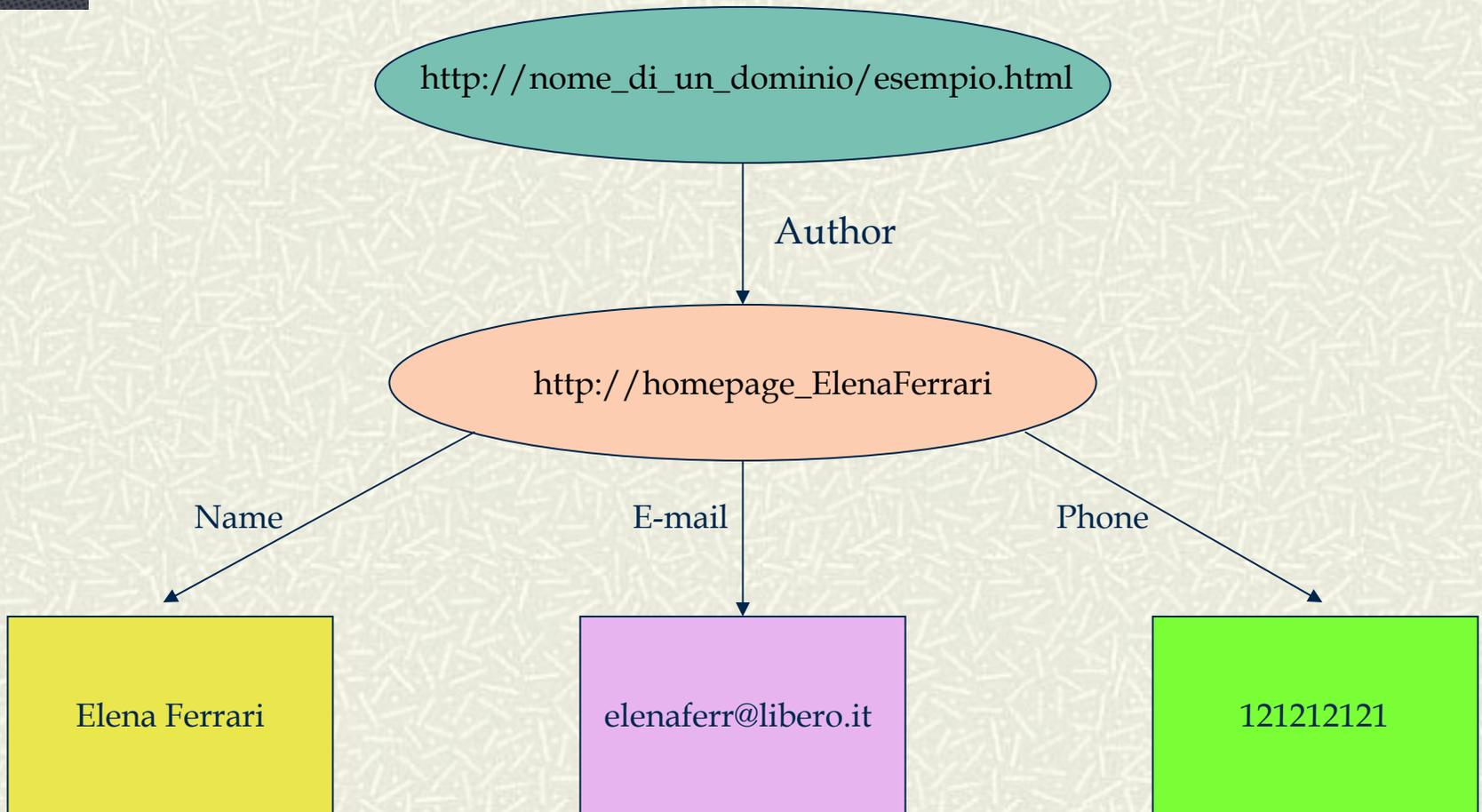
L'oggetto di uno statement RDF può essere a sua volta una risorsa, consentendo in questo modo di descrivere in maniera più approfondita il valore della proprietà.

Esempio 2

All' esempio precedente vogliamo ora aggiungere maggiori informazioni riguardanti l'autore della pagina HTML (ad esempio e-mail e numero di telefono). Dobbiamo allora 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_ElenaFerrari/ AND
http://homepage_ElenaFerrari/ HAS name
ElenaFerrari AND email elenaferr@libero.it
AND phone 121212121`

Combinazione RDF - XML

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 che abbiamo visto 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>
      Elena Ferrari
    </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_ElenaFerrari/" />
  </rdf:Description>

  <rdf:Description about="http://homepage_ElenaFerrari/">
    <a:name>Elena Ferrari</a:name>
    <a:email>elenafer@libero.it</a:email>
    <a:phone>121212121</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 riguardante l'autore, viene assegnata come valore della proprietà author della prima risorsa.

RDF ci consente di definire la semantica. Affidare la semantica semplicemente a un nome può essere rischioso perchè questo potrebbe avere più significati diversi. RDF identifica in modo univoco le proprietà mediante il meccanismo dei namespace (XMLNs).

I namespace

I namespace XML forniscono un metodo per identificare in maniera non ambigua la semantica e le convenzioni che regolano l'utilizzo delle proprietà identificando l'autorità che gestisce il vocabolario (ad esempio la Dublin Core Initiative).

I namespace vengono implementati aggiungendo un prefisso al nome di ogni elemento e attributo. La sintassi per introdurre un prefisso namespace è la seguente:

<prefisso:nome_elemento>

I contenitori

Spesso è necessario riferirsi ad una collezione di risorse; per dire per esempio che un'opera è stata creata da più di una persona, o per elencare gli studenti di un corso. Per contenere queste liste di risorse si usano i **contenitori RDF**.

RDF definisce tre tipi di oggetti contenitore:

- **Bag:** lista non ordinata di risorse o letterali. Si usano per dichiarare che una proprietà ha valori multipli e che l'ordine con cui questi valori sono inseriti non ha alcun significato. Sono ammessi valori duplicati.

- **Sequence:** lista ordinata di risorse o letterali. Il contenitore Sequence si usa per dichiarare che una proprietà ha valori multipli e che l'ordine dei valori è significativo. Il contenitore Sequence potrebbe essere usato, per esempio, per conservare un ordine alfabetico dei valori. Sono ammessi valori duplicati.

- **Alternative:** una lista di risorse o letterali che rappresentano delle alternative al valore (singolo) di una proprietà. Il contenitore Alternative potrebbe essere impiegato, ad esempio, per indicare traduzioni in altre lingue del titolo di un'opera. Un'applicazione che usa una proprietà il cui valore è un raccolta di tipo Alternative sa di poter selezionare uno fra gli oggetti presenti nella lista.

La sintassi per i contenitori è la seguente:

```
container ::= sequence | bag | alternative
sequence ::= '<rdf:Seq' idAttr? '>' member*
           '</rdf:Seq>'
bag ::= '<rdf:Bag' idAttr? '>'
       member* '</rdf:Bag>'
alternative ::= '<rdf:Alt'
               idAttr? '>' member+ '</rdf:Alt>'
member ::= referencedItem | inlineItem
referencedItem ::= '<rdf:li' resourceAttr '/>'
inlineItem ::= '<rdf:li>'
              value '</rdf:li>'
```

Perché RDF/XML è meglio di XML

XML consente varianti diverse per esprimere gli stessi concetti. I programmi applicativi per interpretare queste variazioni hanno bisogno di una conoscenza **a priori** dello schema/DTD.

RDF impone ulteriori vincoli sintattici su come i concetti devono essere espressi.

Noi vogliamo rappresentare la relazione tra un oggetto ed un'informazione associata con esso. In XML, potremmo rappresentare questa relazione in modi diversi. Utilizzando RDF definiamo univocamente la struttura: c'è un soggetto un predicato ed un oggetto.

RDF < FLESSIBILITA' > INTEROPERABILITA'

Con RDF/XML quindi i programmi possono interpretare le descrizioni in modo consistente.