

Sistemi di Elaborazione dell'informazione II

Corso di Laurea Specialistica in Ingegneria Telematica

II anno – 4 CFU

Università Kore – Enna – A.A. 2009-2010

Alessandro Longheu

<http://www.diit.unict.it/users/alongheu>

alessandro.longheu@diit.unict.it

Dati Semistrutturati: il linguaggio RDF Esempi ed esercitazioni

Schema Lezione

- Generalità RDF
- Serializzazione RDF
- RDF Schema
- Utilizzo di RDF
- Esempi
- Limiti di RDF

RDF

- **Resource Description Framework (RDF)** è una particolare *applicazione XML* (standardizzata dal W3C) deputata alla gestione delle **relazioni tra informazioni** ispirandosi ai principi della *logica dei predicati* (o logica predicativa del primo ordine) e ricorrendo agli strumenti tipici del Web (ad es. URI) e dell'XML (namespace).
- In estrema sintesi, secondo la logica dei predicati le informazioni sono esprimibili con **asserzioni** (statement) costituite da triple formate da **soggetto, predicato e valore** (noti anche come subject, verb e object, rispettivamente).
- RDF lavora quindi con tre tipi di elementi fondamentali:
 - **Risorse**: entità riferite attraverso URI
 - **Proprietà**: relazioni binarie tra risorse e/o valori atomici di tipo primitivo
 - **Affermazioni**: specificano il valore di una certa proprietà relativa a una risorsa

RDF

- Esempio:

affermazione

Autore(<http://www.mywebsite.com>)=JohnDoe

proprietà

risorsa soggetto

risorsa oggetto

Si può esprimere come tripla (soggetto, predicato, oggetto):
(<http://www.mywebsite.com>, Autore, JohnDoe)

- Altro esempio:

autoreDi("http://www.pinco.com","http://www.books.org/ISBN00515")
 haPrezzo("http://www.books.org/ISBN00515", "10 euro")



Serializzazione di RDF

RDF, essendo un modello di dati, può essere serializzato (rappresentato e memorizzato) in modi diversi. Ad esempio, supponiamo di avere le seguenti affermazioni sul Presidente della Repubblica:

- Il signor Napolitano vive a Roma
- Il signor Napolitano ha codice fiscale NPLGRO20T09E625V

per alcuni di questi elementi è possibile reperire arbitrariamente sul Web URI (risorse) che li identificano univocamente quali:

Il Signor Napolitano

"<http://www.quirinale.it/presidente/napolitano.htm>" (si è scelto di referenziare la relativa biografia disponibile sul sito ufficiale del Quirinale)

vive a "http://it.wiktionary.org/wiki/vivere" (si è scelta la definizione del verbo vivere su wikidizionario)

Roma "http://www.comune.roma.it/index.asp" (si è scelto di utilizzare il sito istituzionale del Comune di Roma)

ha codice fiscale "http://it.wikipedia.org/wiki/codice_fiscale" (si è scelta la definizione su wikipedia)

Serializzazione di RDF

- La serializzazione di questo permette diverse rappresentazioni.
- **La forma di base** (sequenza di triple) è la seguente:
 - (Il signor Napolitano, vive a, Roma)
 - (Il signor Napolitano, ha codice fiscale, NPLGRO20T09E625V)
- **La forma canonica di serializzazione tramite XML:**
 1. <?xml version="1.0"?>
 2. <rdf:RDF
 3. xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
 4. xmlns:wikipedia=http://it.wikipedia.org/wiki/
 5. xmlns:wikidizionario="http://it.wiktionary.org/wiki/">
 6. <rdf:Description
 7. rdf:about="http://www.quirinale.it/presidente/napolita.htm">
 8. <wikidizionario:vivere
 9. rdf:resource=http://www.comune.roma.it/index.asp
 10. />
 - 11.</rdf:Description>
 - ...

Serializzazione di RDF

```

12. <rdf:Description
13.     rdf:about="http://www.quirinale.it/presidente/napolita.htm">
14.     <wikipedia:codice_fiscale>
15.         NPLGRO20T09E625V
16.     </wikipedia:codice_fiscale>
17. </rdf:Description>
18. </rdf:RDF>

```

- Riga1 **<?xml version='1.0'?>**, *header xml* standard
- Riga2 **rdf:RDF** è il *nodo radice* di un documento *RDF* definito nel *namespace rdf* di cui alla *Riga 3*
- Riga 3 **xmlns:rdf=** riferenzia il *namespace* standard della sintassi *RDF*, identificandolo come **rdf**. Si ricorda che in XML si definisce un *namespace* per rendere più sintetica la scrittura del codice. Ogni volta che si incontra **rdf:** lo si deve (mentalmente) sostituire con quanto scritto a destra dell' = di questa espressione, come già fatto nella *Riga 2*.
- Riga4-5 **xmlns:wikipedia=** e **xmlns:wikidizionario=** riferenziano due ulteriori *namespace*, identificandoli come **wikipedia** e **wikidizionario** ⁷

Serializzazione di RDF

- Riga6 è il tag del namespace rdf di cui alla Riga 3 che consente di specificare un'asserzione (soggetto, predicato, valore)
- Riga7 è un attributo dell'elemento Description di Riga6 da utilizzare per specificare il soggetto (*Il signor Napolitano*) di un'asserzione se è un URI
- Riga8 è il tag definito nel namespace wikidizionario di cui alla Riga 4, utilizzato per definire il predicato *vive a*
- Riga9 è un identificativo del namespace rdf di cui alla Riga 3 per specificare il valore dell'asserzione (*Roma*), espresso come URI
- Riga10 definizione della chiusura di un tag XML vuoto
- Riga11 Chiude il tag Description aperto alla Riga 6
- Riga12-17 Definiscono l'asserzione Il Signor Napolitano ha codice fiscale NPLGRO20T09E625V
- Riga18 Chiude l'elemento radice RDF aperto alla Riga 2

Serializzazione di RDF

- Poiché le due frasi hanno lo stesso soggetto, allora possono essere riformulate (in termini linguistici) nella frase "Il Signor Napolitano vive a Roma e ha Codice Fiscale NPLGRO20T09E625V", alla quale corrisponde la formalizzazione RDF equivalente:

```
<?xml version='1.0'?>
  <rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:wikipedia="http://it.wikipedia.org/wiki/"
    xmlns:wikidizionario="http://it.wiktionary.org/wiki/">
  <rdf:Description
    rdf:about="http://www.quirinale.it/presidente/napolita.htm">
    <wikidizionario:vivere rdf:resource="http://www.comune.roma.it"/>
    <wikipedia:codice_fiscale>NPLGRO20T09E625V
      </wikipedia:codice_fiscale>
  </rdf:Description>
</rdf:RDF>
```

Serializzazione di RDF

■ Altro esempio:

(<http://www.mario.it>, <http://description.org/sch/creatore>, "Mario")

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
ns#">
  <rdf:Description>
    <rdf:subject resource="http://www.mario.it" />
    <rdf:predicate
resource="http://description.org/sch/creatore" />
    <rdf:object>Mario</rdf:object>
  </rdf:Description>
</rdf:RDF>
```

Serializzazione di RDF

- La serializzazione tramite XML è la più diffusa. Quando si utilizza XML, la prima riga di un file RDF è `<?xml version="1.0"?>`. Tale dichiarazione indica che il contenuto del file è scritto in un linguaggio realizzato in XML. Il resto del documento è parentesizzato con:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:...>
```

... DOCUMENTO ...

```
</rdf:RDF>
```

- l'attributo `xmlns:rdf` del tag `<rdf:RDF>` è la definizione di un namespace XML. Ogniqualvolta nel documento compare un concetto con prefisso `rdf:`, il sistema utilizzerà la definizione corrispondente a quel nome, contenuta all'URI specificato nella definizione del namespace.
- In particolare l'URI `"http://www.w3c.org/./22-rdf-syntax-ns#"` indica il sito in cui è reperibile la sintassi del linguaggio RDF. Quindi **tutte le parole che hanno prefisso `rdf:` sono parole del linguaggio.**

Serializzazione di RDF

- In generale, è possibile utilizzare diversi namespace in uno stesso documento.
- Oltre al namespace rdf: (obbligatorio per scrivere in RDF) potremo specificare le URI alle quali sono definiti i vari concetti che intendiamo utilizzare, un namespace per ogni occorrenza di xmlns:.
- Supponiamo, per esempio di voler utilizzare un secondo namespace, etichettando come "esempio:" tutti i concetti definiti all'URI "http://www.miosito.it/miei_concetti", dovremo indicare:
xmlns:esempio=http://www.miosito.it/miei_concetti
- In pratica si tratta di una notazione sintetica che evita di riscrivere ogni volta l'URI relativa al concetto.

Serializzazione di RDF

- Un altro esempio di RDF descritto con XML:

```
<rdf:Description rdf:ID="12345">
  <esempio:nome>Andrea Rossi</esempio:nome>
  <esempio:qualifica>ricercatore</esempio:qualifica>
  <esempio:luogo_lavoro rdf:resource="http://www.sito_uni.it"/>
</rdf:Description>
```

- nell'esempio è riportata una parte del documento RDF che descrive alcune proprietà di una nuova risorsa. Il fatto che si tratti di una nuova risorsa emerge dall'uso dell'attributo `rdf:ID`, l'URI della nuova risorsa è composta utilizzando l'URI che identifica il sito nel quale è contenuto il documento RDF e la stringa che compare dopo "`rdf:ID="`". Qualora si intenda invece descrivere proprietà di una risorsa già definita al posto di `rdf:ID` si usa `rdf:about="URI_risorsa"`.
- `<rdf:Description> ... </rdf:Description>` riassume i predicati che hanno la nuova risorsa come soggetto, nel caso in questione abbiamo tre proprietà.

Serializzazione di RDF

```
<rdf:Description rdf:ID="12345">  
  <esempio:nome>Andrea Rossi</esempio:nome>  
  <esempio:qualifica>ricercatore</esempio:qualifica>  
  <esempio:luogo_lavoro rdf:resource="http://www.sito_uni.it"/>  
</rdf:Description>
```

- I tag <esempio:nome> (esempio:qualifica, esempio:luogo_lavoro) sono i predicati.
 - Qualora i valori di tali proprietà siano stringhe, queste sono racchiuse fra <esempio:proprietà>...</esempio:proprietà>.
 - Se invece si tratta di risorse, si utilizza un empty tag, in cui l'URI che identifica la risorsa compare come valore dell'attributo rdf:resource.
- Nell'esempio tutte le proprietà fanno capo allo stesso namespace (esempio), in generale è però possibile utilizzare all'interno della stessa definizione proprietà contenute in namespace differenti.
- Questo meccanismo consente di distribuire nel web le descrizioni di una stessa risorsa.

Serializzazione di RDF

- Una **terza modalità di rappresentazione** della asserzioni RDF, utilizza i **grafi** scegliendo i nodi per soggetto e valore, uniti da un arco orientato da soggetto a valore per il predicato. Alcuni applicativi adottano ellissi per i nodi che sono URI (<http://www.comune.roma.it/index.asp>), altrimenti dei rettangoli (NPLGRO20T09E625V)
- Una **quarta modalità è la rappresentazione N3** (noto anche come N-triples o Notation 3), propone una forma più facile da leggere rispetto ad RDF:
 1. `<http://www.quirinale.it/presidente/napolita.htm>`
`<http://it.wiktionary.org/wiki/vivere>`
`<http://www.comune.roma.it/index.asp>.`
 2. `<http://www.quirinale.it/presidente/napolita.htm>`
`<http://it.wikipedia.org/wiki/codice_fiscale>` "
`"NPLGRO20T09E625V"`.
- Ciascuna asserzione può essere scritta anche su un'unica linea. In N3 è significativo il punto contrassegna la fine di ciascuna asserzione.

Serializzazione di RDF

- Ultima modalità, **N3 con prefissi**, ancor più sintetico di N3 e riprende l'idea dei namespace XML per agevolare la compilazione utilizzando dei semplici editor. In questo caso l'esempio è tradotto come:
 - @prefix presidente: <http://www.quirinale.it/presidente/>
 - @prefix wpedia: <http://it.wikipedia.org/wiki/>
 - @prefix wdiz: <http://it.wiktionary.org/wiki/>
 - @prefix comune_roma: <http://www.comune.roma.it/>
 - presidente:napolita.html wdiz:vivere comune_roma:index.asp.
 - presidente:napolita.html wpedia:cod_f "NPLGRO20T09E625V".
- Con il comando @prefix si definiscono costanti di sostituzione che specifici strumenti automatici si occupano di operare per completare le asserzioni che le utilizzano.
- Così come per N3, anche per N3 con prefissi è significativo il . (punto) alla fine di ciascuna asserzione.

RDF Schema

- RDF consente di definire un semplice modello dei dati per descrivere proprietà di e relazioni fra risorse. In RDF però non esistono livelli di astrazione: risorse e relazioni sono organizzate in un grafo piatto. In altri termini non è possibile definire tipi (o classi) di risorse con loro proprietà specifiche.
- Vista l'utilità di poter definire classi di risorse, RDF è stato arricchito con un semplice sistema di tipi detto **RDF Schema**. Il sistema di tipi RDF Schema ricorda i sistemi di tipi dei linguaggi di programmazione object-oriented (tipo Java). Una risorsa può, per esempio, essere definita come istanza di una classe (o di più classi) e le classi possono essere organizzate in modo gerarchico. RDF Schema utilizza il modello RDF stesso per definire il sistema di tipi RDF, fornendo un insieme di **risorse e proprietà predefinite** che possono essere utilizzate per definire classi e proprietà a livello utente. L'insieme delle risorse e delle proprietà di base delle risorse è detto vocabolario dell'RDF Schema.

RDF Schema

- Le definizioni di classi in RDF Schema si basano
 - su due classi predefinite (**rdf:Resource** e **rdf:Class**)
 - due proprietà predefinite (**rdf:type** e **rdf:subClassOf**).
- La classe **rdf:Resource** è la classe di base dell'RDF Schema, in quanto ogni oggetto definito in RDF è una risorsa. Attraverso la proprietà **rdf:type** è possibile specificare che una risorsa è un'istanza di una classe. Si osservi che anche le classi sono risorse. Quando si definisce una nuova classe si specifica come valore di **rdf:type** o la risorsa predefinita <http://www.w3c.org/2000/01/rdf-schema#Class>.
- Una risorsa può essere istanza di più classi. **rdf:subClassOf** consente di definire relazioni di sovra/sotto-insieme fra le classi. La relazione **rdf:subClassOf** è transitiva.

RDF Schema

■ Esempio:

```
<rdf:Description rdf:ID="Mammifero">
  <rdf:type rdf:resource="http://www.w3c.org/2000/01/rdf-schema#Class"/>
  <rdf:subClassOf rdf:resource="http://www.w3c.org/2000/01/rdf-
schema#Resource"/>
</rdf:Description>
```

```
<rdf:Description rdf:ID="Orso">
  <rdf:type rdf:resource="http://www.w3c.org/2000/01/rdf-schema#Class"/>
  <rdf:subClassOf rdf:resource="#Mammifero"/>
</rdf:Description>
```

- tutte le classi sono sottoclasse di qualcosa. Nel caso in cui definiamo una classe che apparentemente non è sottoclasse di nulla, occorre comunque specificare che è sottoclasse di Resource. Nell'esempio "Mammifero" è la classe base (sottoclasse di rdf:Resource) mentre "Orso" è sottoclasse di "Mammifero". Poiché una classe può avere molte sovra-classi, <rdf:subClassOf ... /> può comparire molte volte all'interno di una stessa descrizione.

RDF Schema

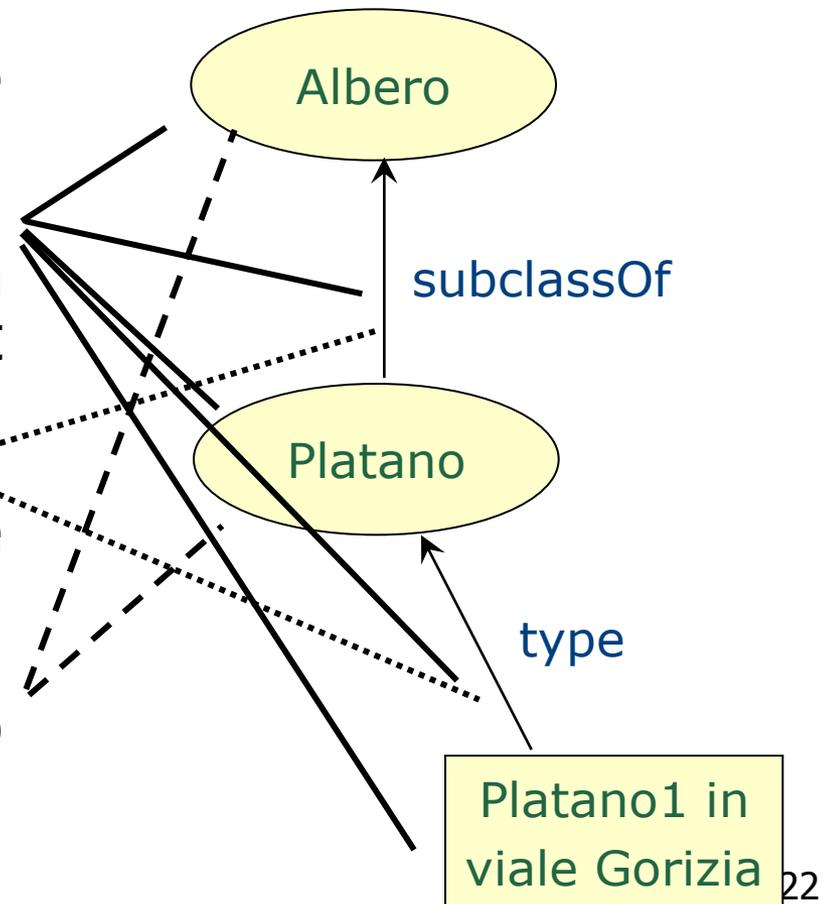
- Fin qui abbiamo visto le classi descritte in termini di altre classi. In generale le classi sono caratterizzate da proprietà. A differenza di quanto accade in molti linguaggi, in RDF Schema proprietà e classi hanno definizioni separate e, in particolare, le proprietà sono definite in termini sia delle classi che costituiscono il loro range sia delle classi a cui tali proprietà si applicano (e non le classi in termini di proprietà).
- Le proprietà possono essere definite utilizzando la risorsa predefinita **rdf:Property** con le proprietà predefinite:
 - **rdf:domain**
 - **rdf:range**
 - **rdf:subPropertyOf**

RDF Schema

- Il valore di **rdf:range** indica la classe alla quale appartengono le risorse che la proprietà definita può assumere come valori.
- **rdf:domain** specifica che la proprietà si applica a risorse di una certa classe. Una proprietà può avere diversi **rdf:domain** specificati ma se non ne ha nessuno è valida in generale e può essere usata per tutte le risorse. Quindi di default lo scope di una proprietà è l'universo; è poi possibile restringerlo tramite opportune specifiche. Sebbene questa scelta riduca il controllo sull'appropriatezza d'uso delle proprietà, consente di estendere facilmente l'uso di proprietà a situazioni non previste da chi le ha inizialmente definite. Supponiamo ad esempio che una proprietà possa essere applicata a risorse di tipo "Persona" e sia erroneamente applicata a risorse di tipo diverso. L'azione da intraprendere in questo caso è lasciata al programma applicativo in uso, il quale (a seconda del tipo di applicativo e del tipo di inferenze che è in grado di effettuare) potrebbe semplicemente segnalare un errore, suggerire alternative corrette, fare inferenze sui dati a disposizione.
- Infine tramite **rdf:subPropertyOf** è possibile specializzare proprietà indicando che si tratta di sottoproprietà di altre già definite.

RDF Schema

- Una visione diversa di quanto detto finora:
- Sono definite le classi (risorse di tipo `rdfs:Class`):
- **rdfs:Resource**: rappresenta tutto ciò che viene descritto in RDF (corrispondente di Object in Java)
- **rdfs:Property**: rappresenta il sottoinsieme di risorse che sono proprietà
- **rdfs:Class**: corrisponde al concetto di tipo (corrispondente alla nozione di classe nei linguaggi OO)



RDF Schema

- RDFS definisce le proprietà (ovvero risorse di tipo `rdfs:Property`):
 - `rdfs:type`: indica che una risorsa è istanza di una classe (1 risorsa, molti tipi)
 - `rdfs:subClassOf`: relazione “sottoinsieme - sovrainsieme” tra classi (una classe può essere sottoclasse di molte classi)
 - `rdfs:subPropertyOf`: una proprietà è una specializzazione di un’altra (1 proprietà, 0-n specializzazioni)
 - `rdfs:seeAlso`: una risorsa contiene informazioni su un’altra risorsa
 - `rdfs:isDefinedBy`: sottoproprietà di `rdfs:seeAlso`, indica quale risorsa definisce un’altra risorsa (es. quale schema?)

RDF Schema

- `rdfs:ConstraintProperty` è una sottoclasse di `rdf:Property`. Le sue istanze sono proprietà utilizzate all'interno di vincoli.
- RDF Schema permette di imporre:
- Vincoli di dominio (`rdfs:domain`)
 - Istanze di `rdfs:ConstraintProperty`
 - Vincolano l'applicazione di una proprietà a una o più classi
- Vincoli di intervallo (`rdfs:range`)
 - Istanze di `rdfs:ConstraintProperty`
 - Vincolano il valore di una proprietà a un determinato intervallo scelto su istanze di classi

RDF Schema

■ Esempio di vincolo:

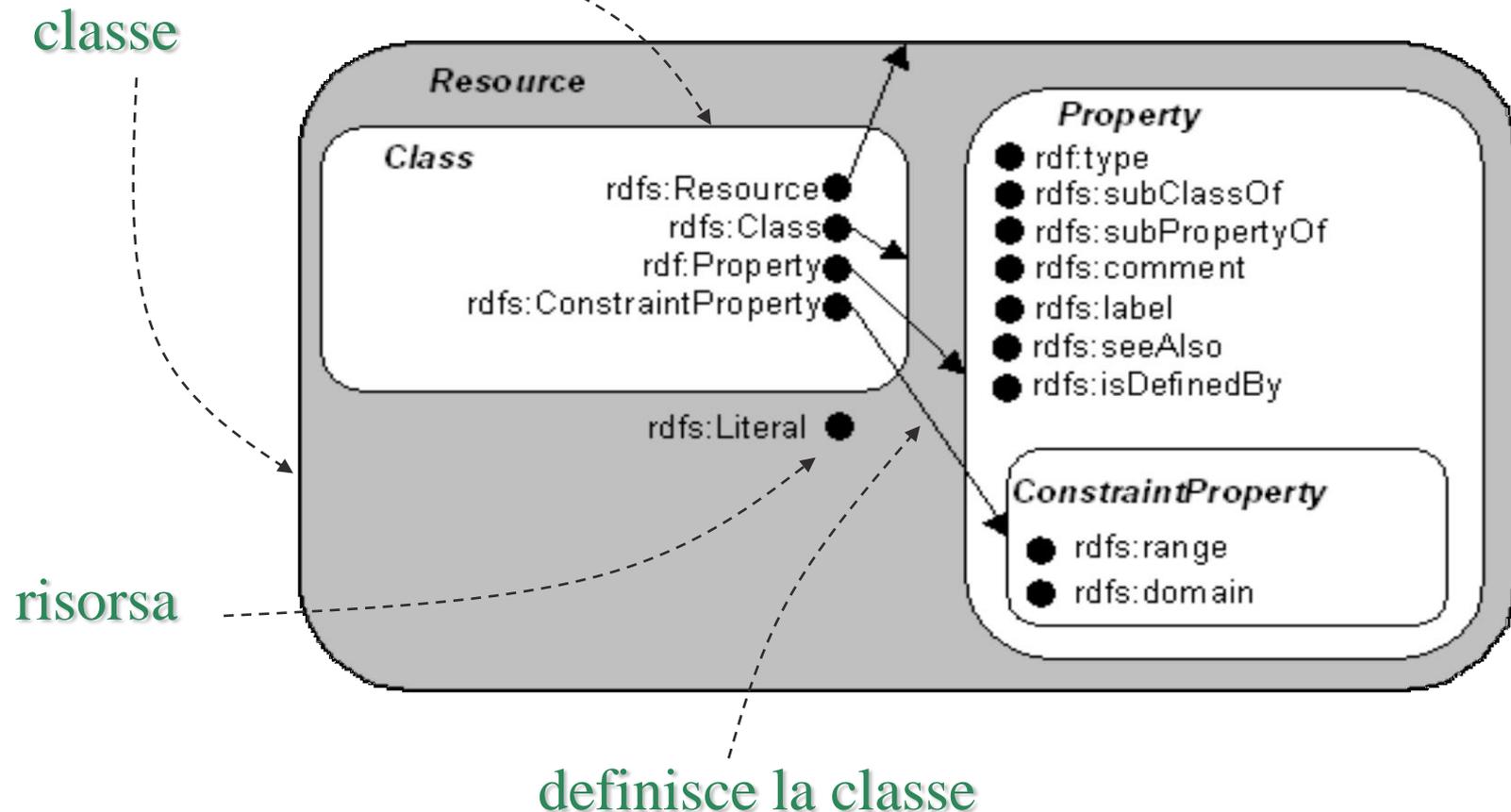
- `<rdf:Description ID="appartieneA">`
- `<rdf:type`
`resource="http://www.w3.org/1999/02/22-rdf-`
`syntax-ns#Property"/>`
- `<rdfs:domain rdf:resource="Oggetto"/>`
- `<rdfs:range rdf:resource="Persona"/>`
- `</rdf:Description>`

■ Vincola la proprietà appartieneA:

- All'applicazione su istanze della classe Oggetto
- Ad assumere valori che siano istanze della classe Persona

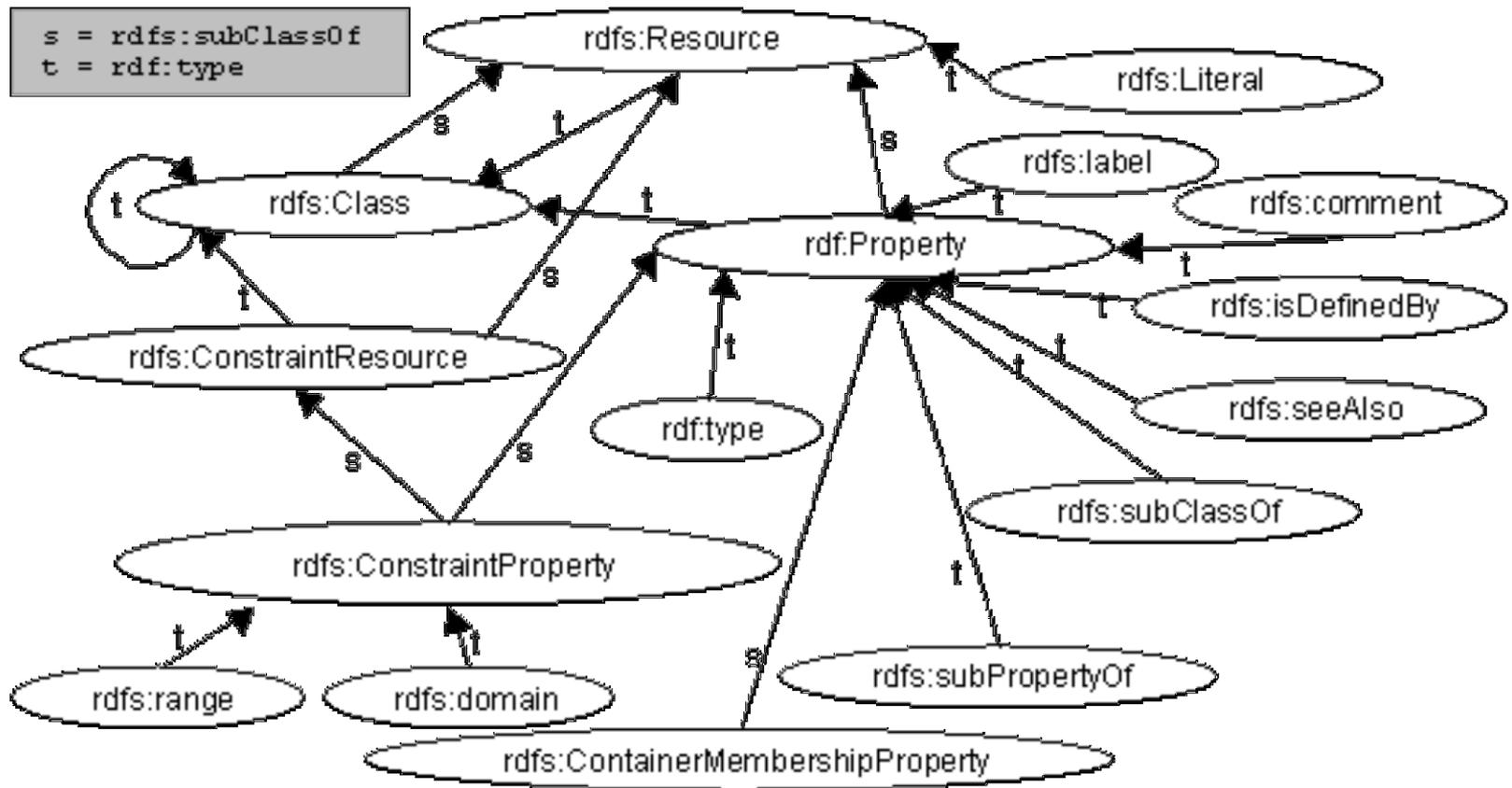
RDF Schema

- Relazione fra classi, sottoclassi e risorse
sottoclasse



RDF Schema

- Relazione fra classi, sottoclassi e tipi



Utilizzo di RDF

- Condizione necessaria per il buon utilizzo di RDF è la **disponibilità on-line di riferimenti di qualità** alle URI utilizzate/referenziate. In particolare, è importante che le risorse siano **note, condivise e stabili nel tempo**. Ad es., il riferimento utilizzato per il Presidente Napolitano non è dei migliori perché valido solo durante il mandato, dopodiché sarà spostato in http://www.quirinale.it/ex_presidenti/Napolitano/napolita.htm dove già si trovano quelle dei suoi predecessori. Dopo questa data l'asserzione [RDF] continuerà a valere, ma si perderà il contributo informativo della pagina web, utile per una completa interpretazione.
- Meglio utilizzare la biografia in wikipedia oppure il Codice Fiscale con <http://www.agenziaentrate.it/servizi/CF#NPLGRO20T09E625V> Sebbene questa URI oggi non referenzi alcunché sul Web, la si potrebbe comunque utilizzare allo scopo perché RDF non presuppone alcuna verifica sulla effettiva disponibilità della risorsa.
- In questo modo tutte le persone fisiche e giuridiche contemporanee, personaggi, aziende, enti ed istituzioni 'importanti' potrebbero essere univocamente identificati, potenziando notevolmente le possibilità di RDF (nulla vieta all'Agenzia delle Entrate di mettere a disposizione un servizio che a partire dalla URI citata restituisca automaticamente le relative informazioni anagrafiche).

Utilizzo di RDF

- Cio' detto, si potrebbe affermare che La risorsa individuata dal codice fiscale NPLGRO20T09E625V vive a Roma, ed ha due biografie:

- *http://it.wikipedia.org/wiki/giorgio_napolitano*
- *<http://www.quirinale.it/presidente/napolita.htm>*

ovvero:

@prefix agenzia: <http://www.agenziaentrate.it/servizi/>

@prefix comune_roma: <http://www.comune.roma.it/>

@prefix presidente: <http://www.quirinale.it/presidente/>

@prefix wikipedia: <http://it.wikipedia.org/wiki/>

@prefix wikidizionario: <http://it.wiktionary.org/wiki/>

agenzia:CF#NPLGRO20T09E625V wikidizionario:vivere

comune_roma:index.asp

agenzia:CF#NPLGRO20T09E625V wikidizionario:biografia

presidente:napolita.htm

agenzia:CF#NPLGRO20T09E625V wikidizionario:biografia

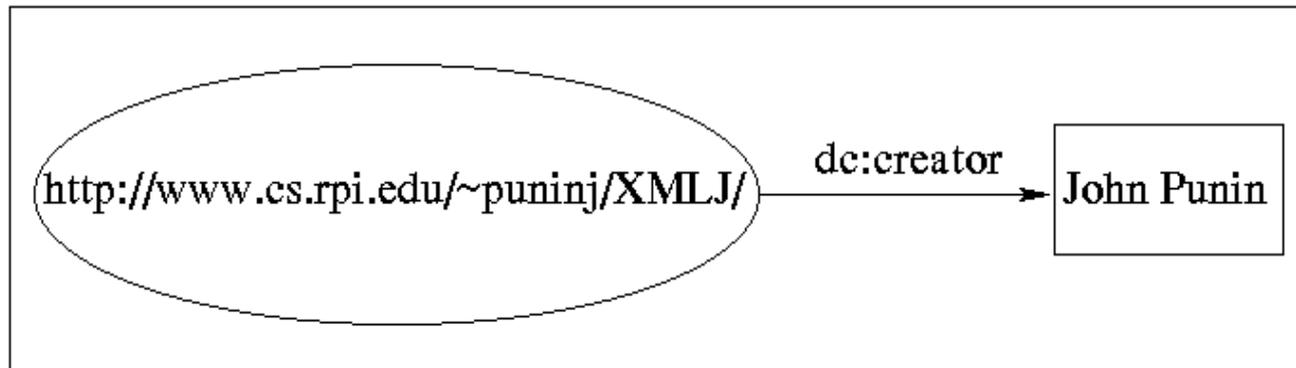
wikipedia:giorgio_napolitano

Utilizzo di RDF

- Un altro accorgimento da tenere presente nella scelta dei termini da utilizzare per la definizione delle relazioni è di **ricorrere a dizionari già noti e diffusi**, invece che inventarne ogni volta di nuovi. Ad es. per le informazioni anagrafiche personali tipiche dei biglietti da visita quali nome, cognome, indirizzo, e-mail, ruolo aziendale,... è già disponibile **vCard**. Non sono da dimenticare neppure le numerose applicazioni XML definite per abilitare l'EDI (Electronic Data Interchange, scambio dati elettronico) nell'ambito della pubblica amministrazione o di associazioni di settore.
- Da quanto esposto finora è facile intuire che, nella costruzione del web semantico, progetti come wikipedia o wikidizionario non sono solo funzionali ma, adeguatamente sfruttati, anche abilitanti perché forniscono lemmi ben documentati per individuare risorse e predicati, garantendone la stabilità nel tempo e, grazie alla possibilità di dichiarare equivalenze nella stessa lingua e tra lingue diverse, di accrescere ulteriormente l'estensione di un'indagine automatica.

Esempi di RDF

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:dc="http://purl.org/dc/elements/1.1/"
         <rdf:Description about="http://www.cs.rpi.edu/~puninj/XMLJ/">
           <dc:creator>John Punin</dc:creator>
         </rdf:Description>
</rdf:RDF>
```

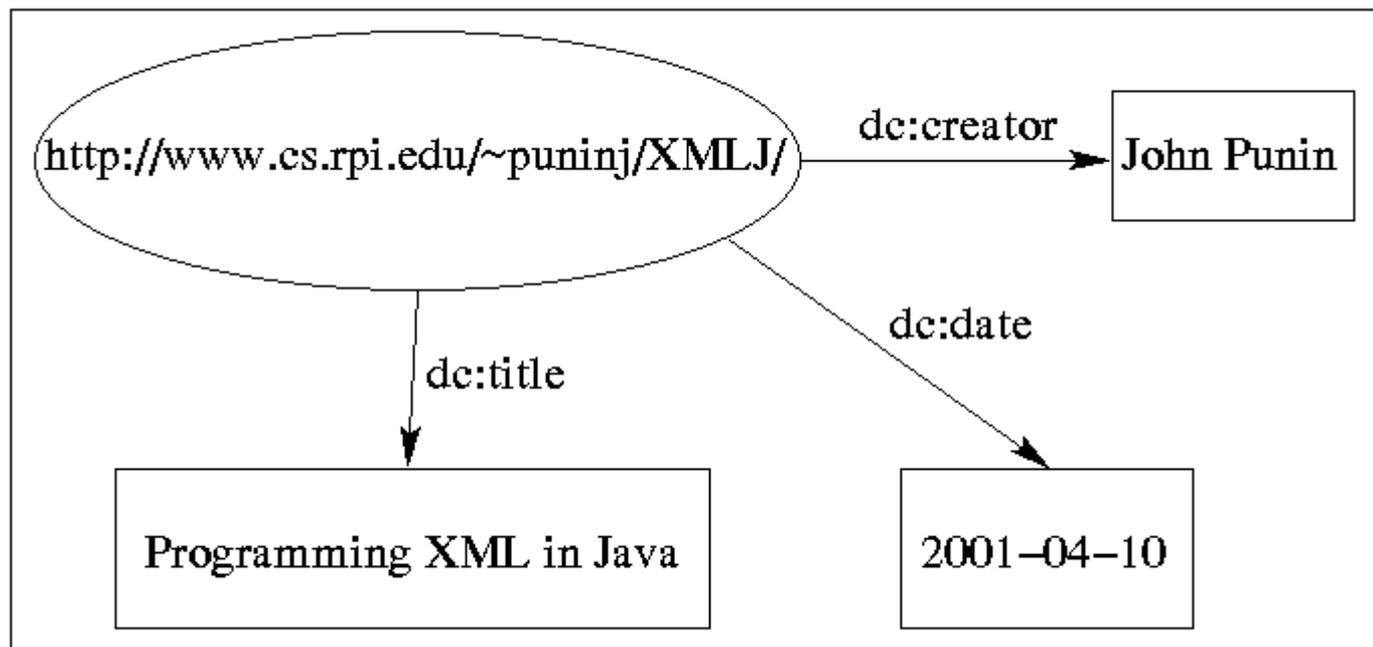


Esempi di RDF

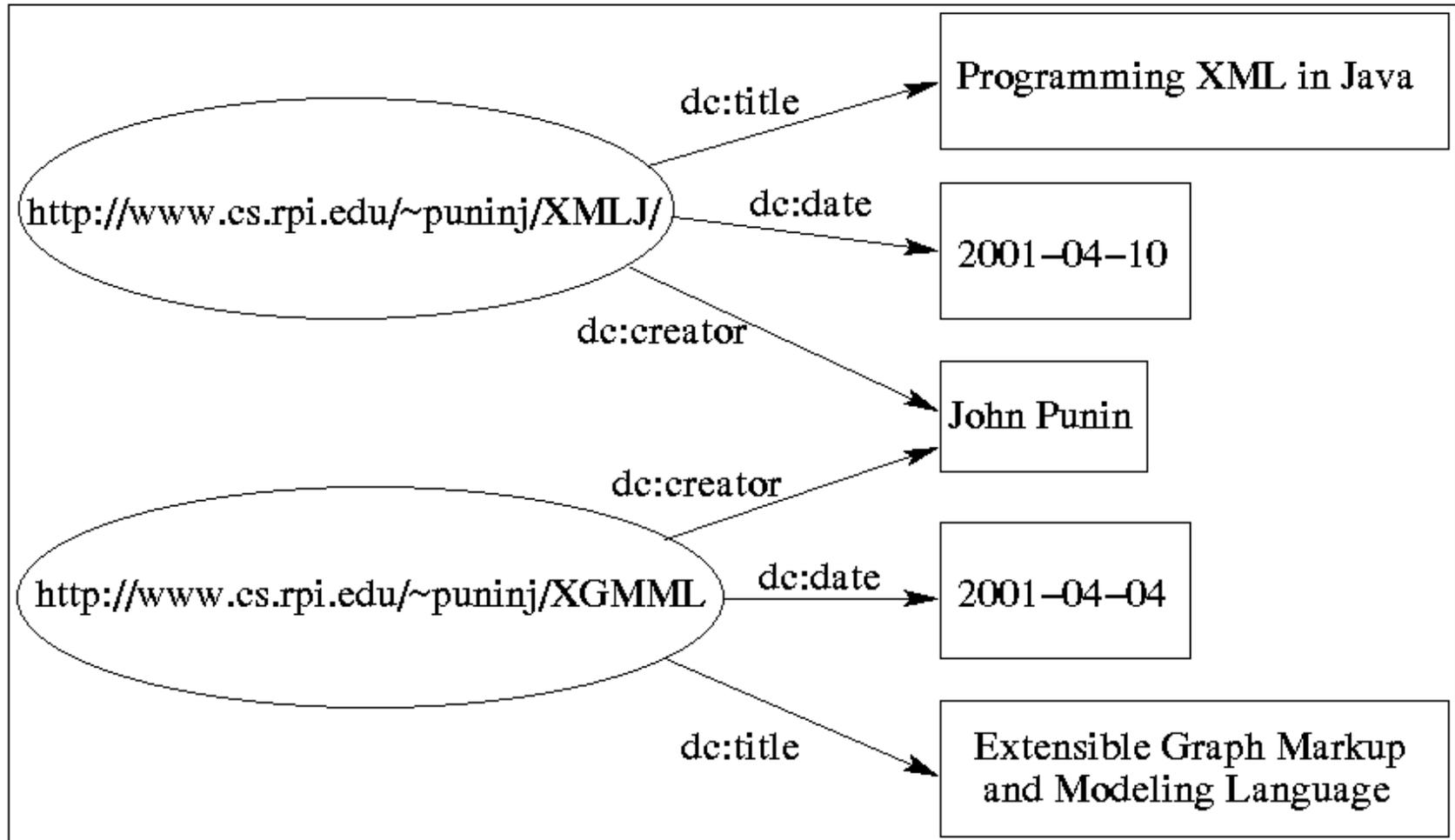
```

<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description about="http://www.cs.rpi.edu/~puninj/XMLJ/">
    <dc:creator>John Punin</dc:creator>
    <dc:title>Programming XML in Java</dc:title>
    <dc:date>2001-04-10</dc:date>
  </rdf:Description>
</rdf:RDF>

```



Esempi di RDF

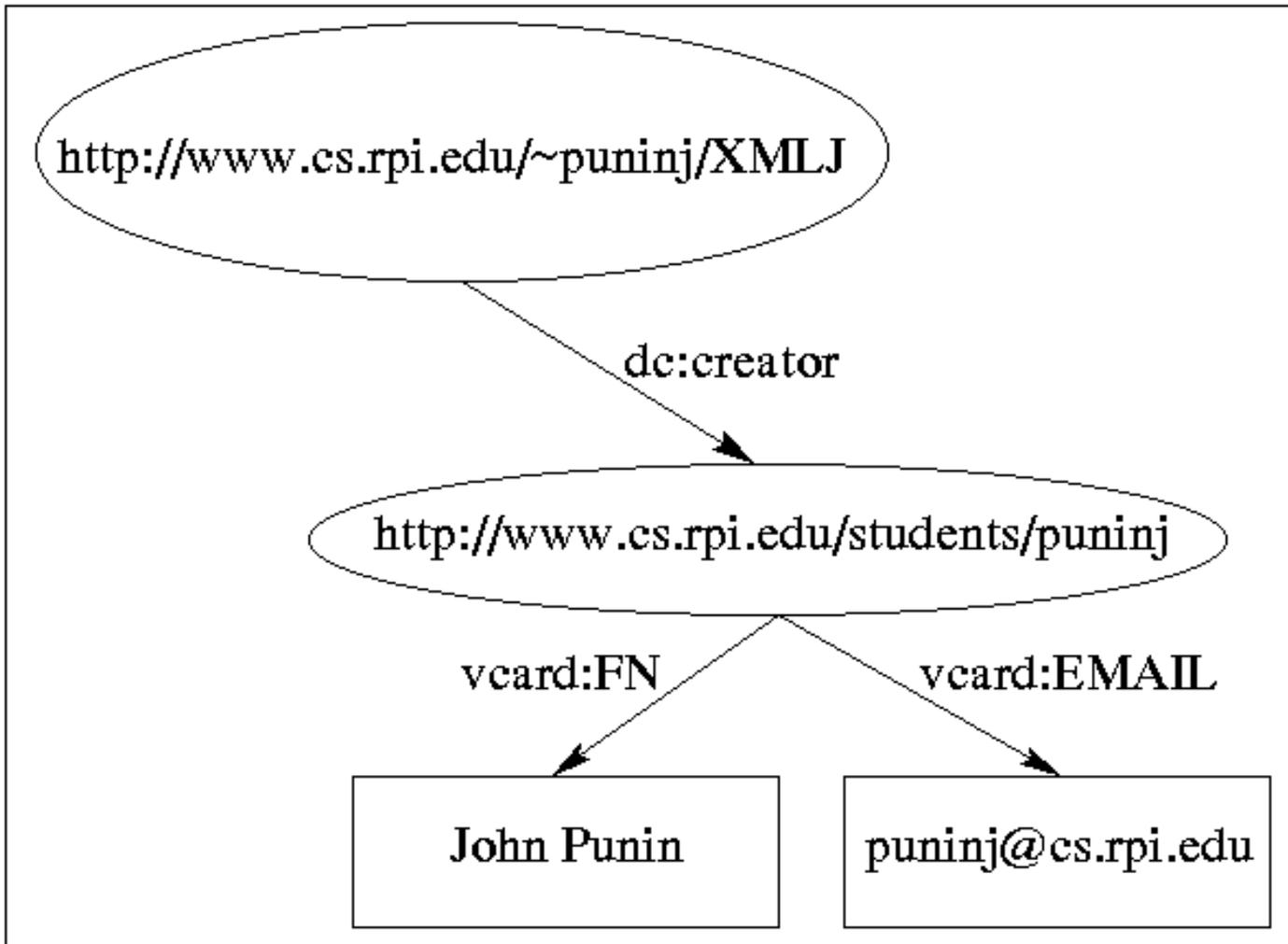


Esempi di RDF

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description about="http://www.cs.rpi.edu/~puninj/XMLJ/">
    <dc:creator>John Punin</dc:creator>
    <dc:title>Programming XML in Java</dc:title>
    <dc:date>2001-04-10</dc:date>
  </rdf:Description>

  <rdf:Description about="http://www.cs.rpi.edu/~puninj/XGML/">
    <dc:creator>John Punin</dc:creator>
    <dc:title>Extensible Graph Markup and Modeling Language</dc:title>
    <dc:date>2001-04-04</dc:date>
  </rdf:Description>
</rdf:RDF>
```

Esempi di RDF



Esempi di RDF

- Production propertyElt :

```
<PropertyName> Value </PropertyName> or
```

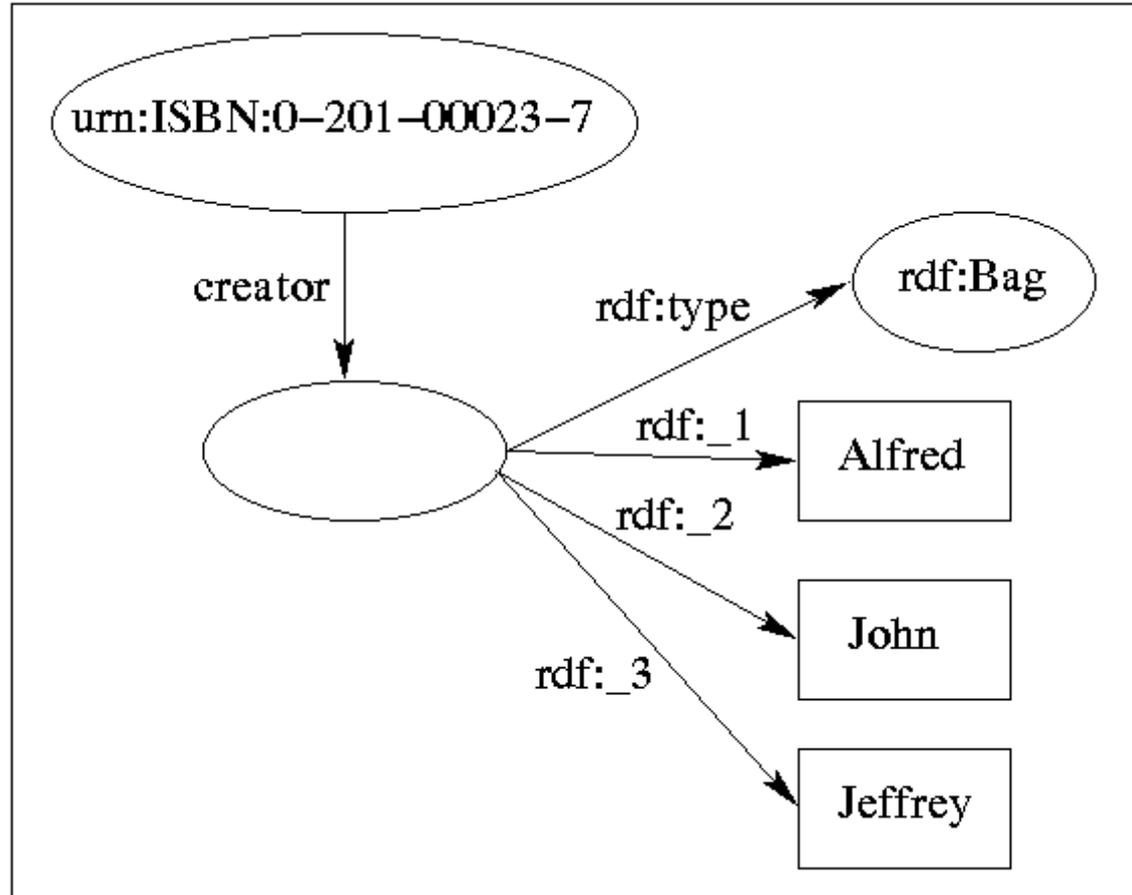
```
<PropertyName rdf:resource="URI" />
```

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:vcard="http://imc.org/vCard/3.0#">
  <rdf:Description about="http://www.cs.rpi.edu/~puninj/XMLJ/">
    <dc:creator rdf:resource="http://www.cs.rpi.edu/students/puninj"/>
  </rdf:Description>

  <rdf:Description about="http://www.cs.rpi.edu/students/puninj">
    <vcard:FN>John Punin</vcard:FN>
    <vcard:EMAIL>puninj@cs.rpi.edu</vcard:EMAIL>
  </rdf:Description>
</rdf:RDF>
```

Esempi di RDF

- in RDF è possibile specificare **collezioni** di risorse (un libro ha molti autori, molti studenti fanno parte di un corso, ecc.). In alcuni casi più elementi debbono essere posti in sequenza o in alternativa. A questi scopi in rdf è possibile utilizzare
 - rdf:Bag per le collezioni non ordinate di oggetti,
 - rdf:Seq per le sequenze ordinate,
 - rdf:Alt per le alternative.

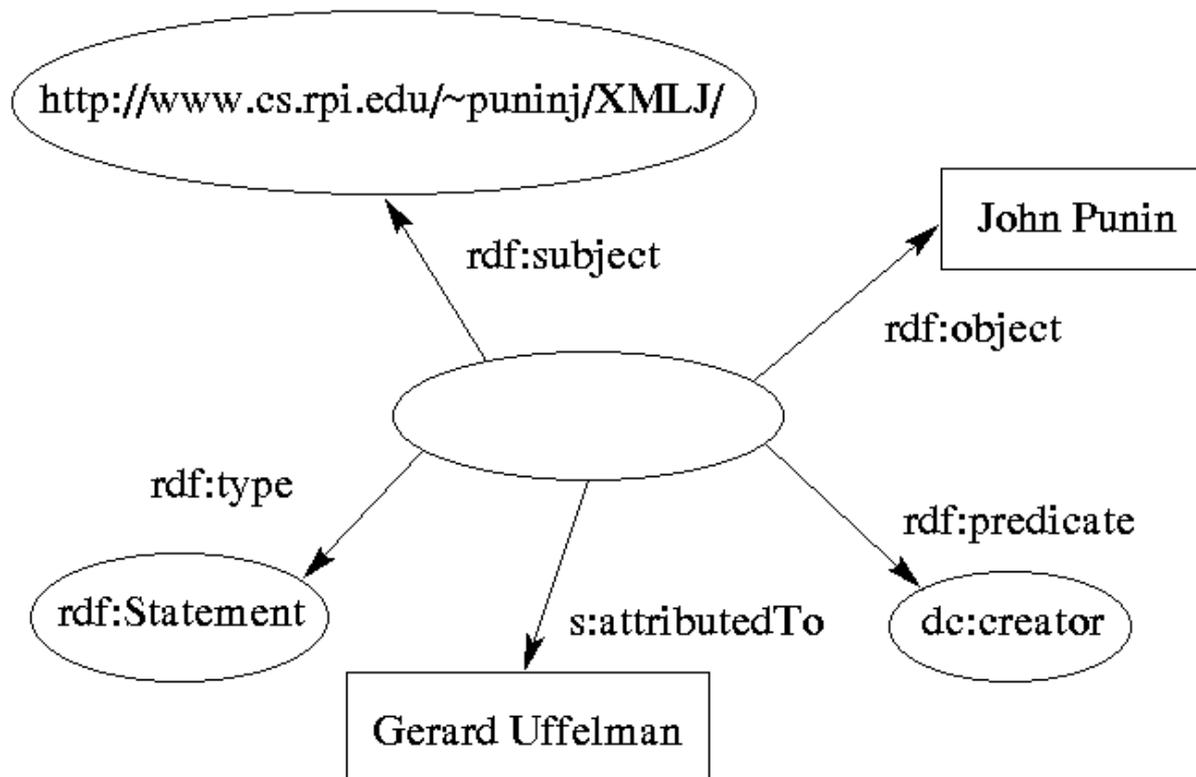


Esempi di RDF

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description about="urn:ISBN:0-201-00023-7">
    <dc:creator>
      <rdf:Bag>
        <rdf:li>Alfred</rdf:li>
        <rdf:li>John</rdf:li>
        <rdf:li>Jeffrey</rdf:li>
      </rdf:Bag>
    </dc:creator>
  </rdf:Description>
</rdf:RDF>
```

Esempi di RDF

- Statements about Statements (**reification**), for example: "Gerard Uffelmann says that John Punin is the creator of <http://www.cs.rpi.edu/~puninj/XMLJ/>"



Esempi di RDF

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:s="http://www.schemas.org/schema/">
  <rdf:Description>
    <rdf:subject rdf:resource="http://www.cs.rpi.edu/~puninj/XMLJ/">
    <rdf:predicate rdf:resource="http://purl.org/dc/elements/1.1/creator"/>
    <rdf:object>John Punin</rdf:object>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"/>
    <s:attributedTo>Gerard Uffelman</s:attributedTo>
  </rdf:Description>
</rdf:RDF>
```

Esempi di RDF: RSS

- Una delle applicazioni RDF più diffusa è l’RSS (RDF Site Summary), una modalità di rappresentazione di informazioni contenute nei siti web. L’esigenza nasce dal volere
 - accedere a informazioni **disseminate in piu’ siti**, quindi da integrare
 - accedere ad informazioni **variabili nel tempo**, quindi si deve mantenere aggiornato quanto desiderato;
- classico esempio, le “news” di siti informativi, variabili e distribuite
- RSS 1.0, basato su RDF (versioni precedenti di RSS non utilizzavano RDF), è un vocabolario RDF che permette la descrizione di informazioni memorizzate su siti web
- RSS è solo uno strumento descrittivo; come sempre, occorre che una qualche applicazione ne faccia uso per ottenere i risultati voluti (integrazione di informazioni variabili nel tempo).

Esempi di RDF: RSS

- Esistono tre categorie di software che gestiscono l’RSS 1.0:
 - **On-line aggregators**, come Meerkat e NewsIsFree, che effettuano l’integrazione di diversi item disseminati su vari siti, proponendola all’utente e permettendo di effettuare ricerche su tale aggregazione; in questo modo si possono avere le ultime informazioni su, ad esempio, la politica in medio oriente, senza dovere cercare in tanti siti (a patto ovviamente che i siti supportino l’RSS)
 - **Desktop Readers**, che permettono agli utenti di sottoscrivere a diversi feed RSS, tenendoli periodicamente aggiornati sul desktop del client
 - **Scripts**, software che permettono di includere feed RSS di altri siti nel proprio (scopo originale di RSS)

Limiti di RDF

■ RDF Schema è uno strumento che consente (in modo molto basilare) di definire vocabolari RDF, tuttavia sarebbe auspicabile arricchire questo linguaggio o, in alternativa, definire nuovi **linguaggi che forniscono qualche capacità in più**. Alcune di queste capacità sono state individuate ed esistono già linguaggi (es. DAML+OIL) che le forniscono; esse sono:

1. vincoli di cardinalità (es. una persona ha una sola madre e un solo padre)
2. possibilità di indicare che due classi definite in schemi differenti rappresentano lo stesso concetto
3. possibilità di indicare che due istanze, definite separatamente, rappresentano lo stesso individuo
4. possibilità di indicare che una proprietà è transitiva
5. la possibilità di definire una nuova classe come combinazione di più classi esistenti.

■ La definizione e la realizzazione di queste (e altre) capacità sono lo scopo dei gruppi di lavoro sui linguaggi per la definizione di ontologie.

Esempi RDF

Seguono adesso diversi esempi di RDF:

Riferimenti:

<http://www.w3.org/RDF/Validator/>

- permette la validazione di file RDF conformi allo schema, ed una rappresentazione tramite triple e/o grafo

<http://www.w3.org/TR/rdf-schema/>

- lo schema completo di RDF, utile per descrivere le risorse, proprietà e relazioni

Esempi RDF

- Consideriamo la frase:
 - il ICT Technology Park si trova presso la via Techniker 21a, 6020, Innsbruck, Austria
- Può essere scritta come tripla RDF senza blank node:


```

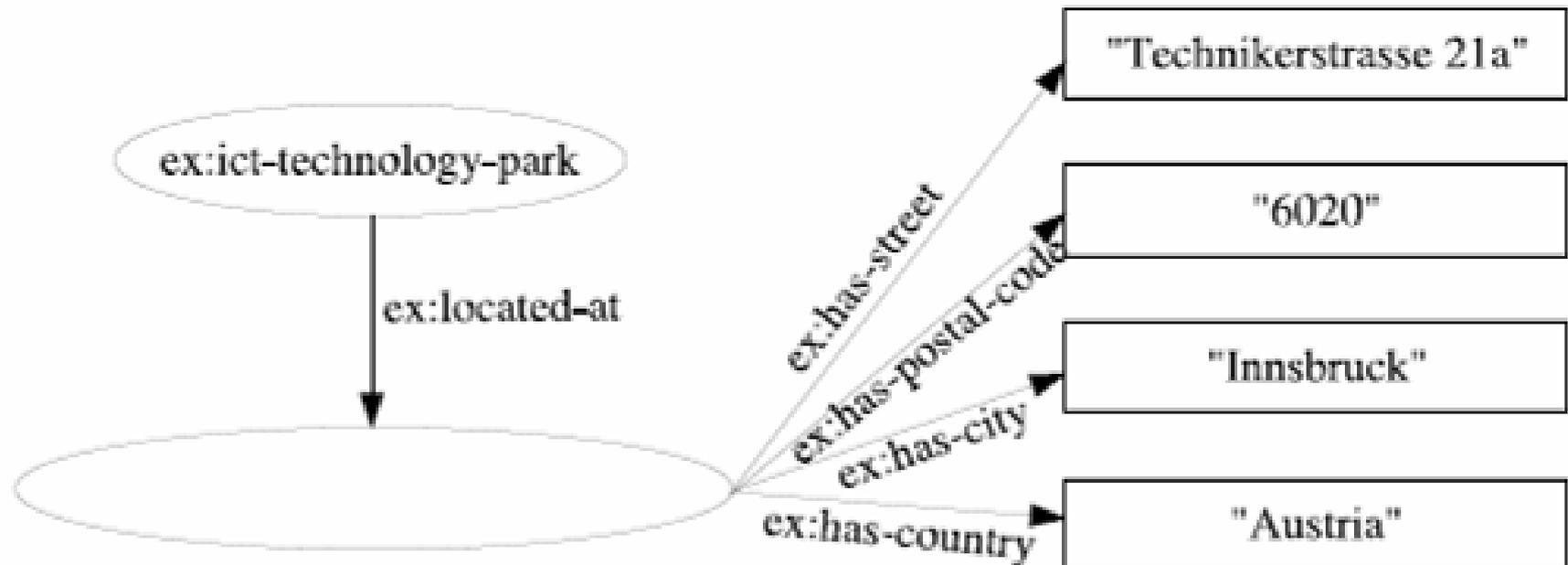
ex:ict-technology-park ex:located-at ex:ict-tech-park-location
ex:ict-tech-park-location ex:has-street "Techniker 21a"
ex:ict-tech-park-location ex:has-postal-code "6020"
ex:ict-tech-park-location ex:has-city "Innsbruck"
ex:ict-tech-park-location ex:has-country "Austria"
      
```
- o con un **blank node** (che consente di creare un aggregato stile struct del linguaggio C):


```

ex:ict-technology-park ex:located-at _:X
_:X ex:has-street "Technikerstrasse 21a"
_:X ex:has-postal-code "6020"
_:X ex:has-city "Innsbruck"
_:X ex:has-country "Austria"
      
```

Esempi RDF

- Rappresentazione grafica nella versione con blank node:



Esempi RDF

- Rappresentazione della frase “John claims that Mary is married to Bill”. Un caso simile, in cui si vuole descrivere un’affermazione relativa ad un’altra affermazione, e’ denominato **reification**
- Una possibile soluzione:
 - _:X rdf:type rdf:Statement*
 - _:X rdf:subject ex:mary*
 - _:X rdf:predicate ex:married-to*
 - _:X rdf:object ex:bill*
 - ex:john ex:claims _:X*

Esempi RDF

- Utilizzando la reification, modellare le seguenti frasi tramite triple RDF:

Il creatore della risorsa Microsoft.com e' Bill gates

Giorgio crede che il creatore della risorsa Microsoft.com sia Bill Gates

Esempi RDF

- Una possibile soluzione:

```

<?xml version=1.0?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  <rdf:Description>
    <rdf:subject rdf:resource="http://www.microsoft.com"/>
    <rdf:predicate "http://purl.org/dc/elements/1.1/creator"/>
    <rdf:object>Bill Gates</rdf:object>
    <!--
ALTERNATIVAMENTE, SE BILL E' UNA RISORSA...
    <rdf:object rdf:resource="http://www.wikipedia.org/..."/>-->
    <rdf:type rdf:resource=http://www.w3.org/1999/02/22-rdf-
syntax-ns#Statement"
    <s:attributedTo>Giorgio</s:attributedTo>
  </rdf:Description>
</rdf:RDF>

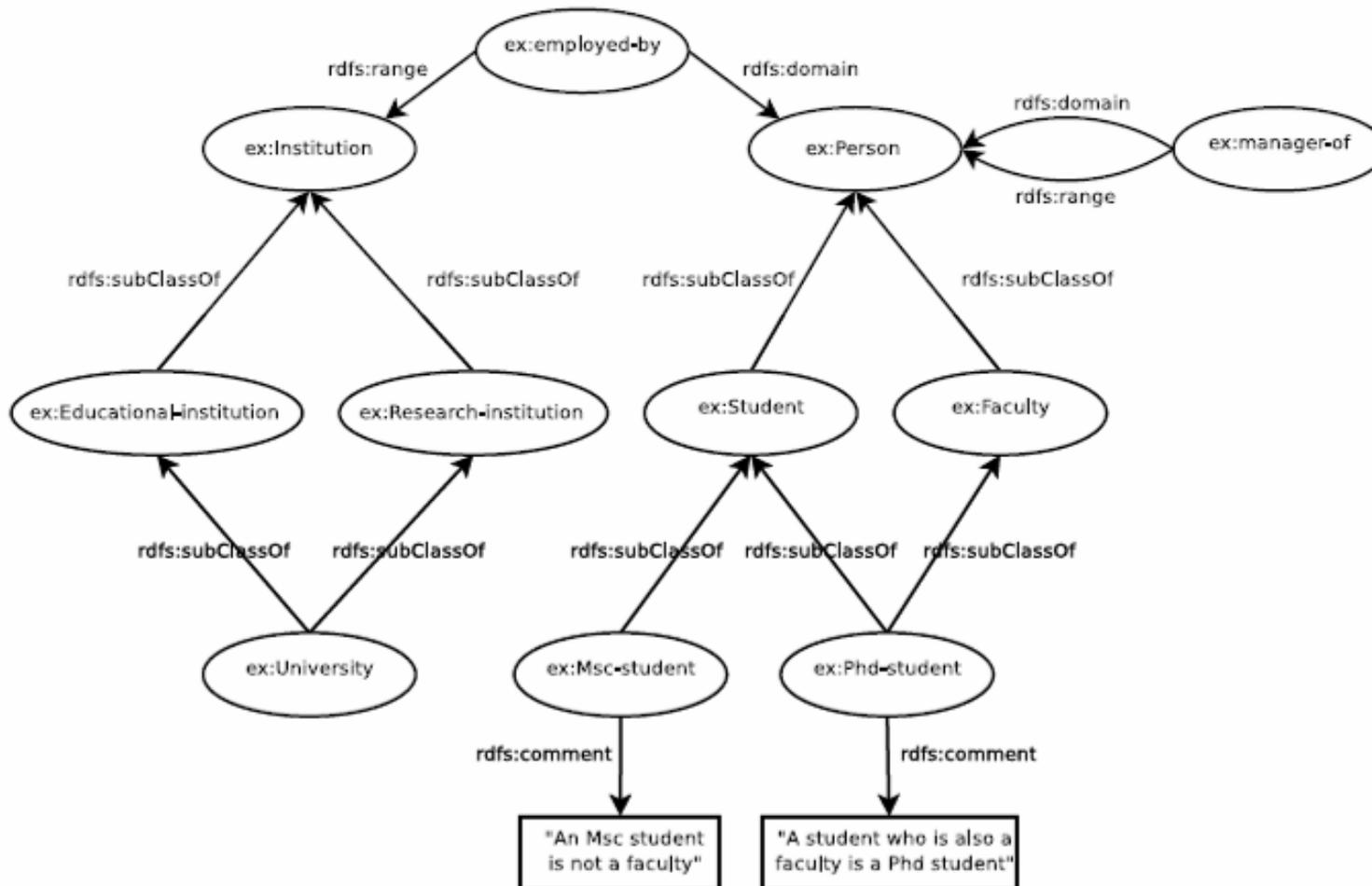
```

Esempi RDF

- Creare un'ontologia in notazione RDF per rappresentare Persone all'interno di una realtà universitaria. Una possibile soluzione in triple:
ex:Research-institution rdfs:subClassOf ex:Institution
ex:Educational-institution rdfs:subClassOf ex:Institution
ex:University rdfs:subClassOf ex:Research-institution
ex:University rdfs:subClassOf ex:Educational-institution
ex:Faculty rdfs:subClassOf ex:Person
ex:Student rdfs:subClassOf ex:Person
ex:Phd-Student rdfs:subClassOf ex:Faculty
ex:Phd-Student rdfs:subClassOf ex:Student
ex:Msc-Student rdfs:subClassOf ex:Student
ex:Msc-Student rdfs:comment "An Msc student is not a faculty"
ex:Phd-Student rdfs:comment "A student who is also a faculty is a PhD Student"
ex:employed-by rdfs:domain ex:Person
ex:employed-by rdfs:range ex:Institution
ex:manager-of rdfs:domain ex:Person
ex:manager-of rdfs:range ex:Person

Esempi RDF

- Il corrispondente grafo



Esempi RDF

- Una rappresentazione in RDF/XML:

```

<!ENTITY ex "http://example.org/#">
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:ex="http://example.org/#">

  <rdfs:Class rdf:about="&ex;Research-institution">
    <rdfs:subClassOf rdf:resource="&ex;Institution"/>
  </rdfs:Class>

  <rdfs:Class rdf:about="&ex;Educational-institution">
    <rdfs:subClassOf rdf:resource="&ex;Institution"/>
  </rdfs:Class>

```

...

Esempi RDF

- ...segue...

```
<rdfs:Class rdf:about="&ex;University">  
  <rdfs:subClassOf rdf:resource="&ex;Research-institution"/>  
  <rdfs:subClassOf                rdf:resource="&ex;Educational-  
  institution"/>  
</rdfs:Class>
```

```
<rdfs:Class rdf:about="&ex;Faculty">  
  <rdfs:subClassOf rdf:resource="&ex;Person"/>  
</rdfs:Class>
```

```
<rdfs:Class rdf:about="&ex;Student">  
  <rdfs:subClassOf rdf:resource="&ex;Person"/>  
</rdfs:Class>
```

Esempi RDF

- ...segue...

```
<rdfs:Class rdf:about="&ex;Phd-Student">  
  <rdfs:subClassOf rdf:resource="&ex;Student"/>  
  <rdfs:subClassOf rdf:resource="&ex;Faculty"/>  
  <rdfs:comment>A student who is also a faculty is a PhD  
  Student</rdfs:comment>  
</rdfs:Class>
```

```
<rdfs:Class rdf:about="&ex;Msc-Student">  
  <rdfs:subClassOf rdf:resource="&ex;Student"/>  
  <rdfs:comment>An Msc student is not a  
  faculty</rdfs:comment>  
</rdfs:Class>
```

Esempi RDF

- ...segue...

```
<rdf:Property rdf:about="&ex;employed-by">  
  <rdfs:domain rdf:resource="&ex;Person"/>  
  <rdfs:range rdf:resource="&ex;Institution"/>  
</rdf:Property>
```

```
<rdf:Property rdf:about="&ex;manager-of">  
  <rdfs:domain rdf:resource="&ex;Person"/>  
  <rdfs:range rdf:resource="&ex;Person"/>  
</rdf:Property>
```

```
</rdf:RDF>
```

Esempi RDF

- Un esempio di istanza creata a partire dall'ontologia creata in precedenza:

```
ex:UIBK rdf:type ex:University  
ex:josdebruijn rdf:type ex:Phd-Student  
ex:josdebruijn ex:employed-by ex:UIBK  
ex:josdebruijn rdfs:label "Jos de Bruijn"  
ex:dieter ex:manager-of _:x  
_:x rdf:type rdf:Bag  
_:x rdf:_1 ex:josdebruijn  
_:x rdf:_2 ex:holgerlausen
```

Esempi RDF

- Creare un'ontologia in notazione RDF per rappresentare impiegati all'interno di una realtà lavorativa. Ogni impiegato può essere responsabile di progetti presso un dato reparto. I reparti sono caratterizzati da un direttore (anch'esso impiegato) e da una o più sedi
- Rappresentare l'ontologia di cui sopra in XML
- Creare qualche istanza a partire dall'ontologia

Esempi RDF

- Una possibile soluzione

```
<?xml version=1.0?>
```

```
<!ENTITY ex "http://MY_NS#">
```

```
<rdf:RDF
```

```
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:MY_NS="http://...#"
```

```
  <rdfs:Class rdf:about="&MY_NS;RESPONSABILE">
```

```
    <rdfs:subClassOf
```

```
      rdf:resource="http://wikipedia.org/IMPIEGATO"/>
```

```
  </rdfs:Class>
```

```
  <rdfs:Class rdf:about="&MY_NS;DIRETTORE">
```

```
    <rdfs:subClassOf
```

```
      rdf:resource="http://wikipedia.org/IMPIEGATO"/>
```

```
  </rdfs:Class>
```

```
  <rdfs:Class rdf:about="&MY_NS;DIRETTORE">
```

```
    <rdfs:subClassOf
```

```
      rdf:resource="http://wikipedia.org/IMPIEGATO"/>
```

```
  </rdfs:Class>
```

```
...
```

Esempi RDF

```

...
<rdfs:Class rdf:about="&MY_NS;PROGETTO"/>
<rdfs:Class rdf:about="&MY_NS;SEDE"/>
<rdfs:Class rdf:about="&MY_NS;REPARTO"/>

<rdf:Property rdf:about="&MY_NS;gestito_da">
<rdfs:domain rdf:resource="&MY_NS;PROGETTO"/>
<rdfs:range rdf:resource="&ex;RESPONSABILE"/>
</rdf:Property>

<rdf:Property rdf:about="&MY_NS;diretto_da">
<rdfs:domain rdf:resource="&MY_NS;REPARTO"/>
<rdfs:range rdf:resource="&ex;DIRETTORE"/>
</rdf:Property>

<rdf:Property rdf:about="&MY_NS;diviso_in">
<rdfs:domain rdf:resource="&MY_NS;REPARTO"/>
<rdfs:range rdf:resource="&ex;SEDE"/>
</rdf:Property>

</rdf:RDF>

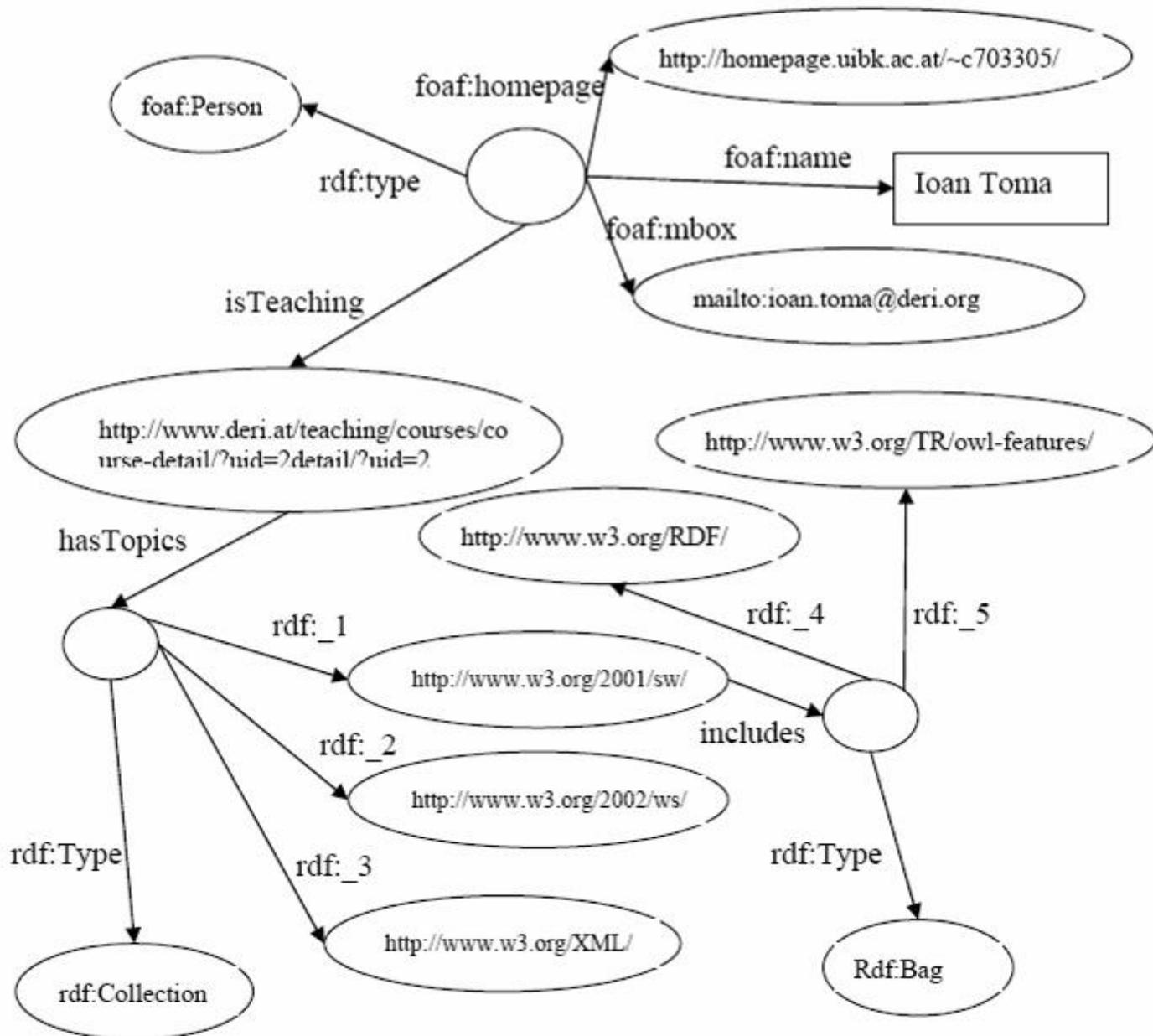
```

Esercitazione RDF

- Sia data una classe che definisce un elettrodomestico generico. Le proprietà che lo caratterizzano sono
 - le dimensioni (altezza, larghezza, profondità)
 - la tensione nominale ammessa (110/220V)
 - la potenza massima dissipata in W.
- Sia data poi una classe frigorifero, che prevede:
 - due termostati per le temperature delle sezioni frigo e congelatore, tenendo conto che la sezione frigo permette da 0 a 4 °C mentre il congelatore va da -4 a -18 °C
- L'elettrodomestico "forno a microonde" invece deve avere
 - un livello di potenza di lavoro, assegnabile come 10%, 20%, 50%, 70% o 90% della potenza massima ammessa per l'elettrodomestico.
- Scrivere una rappresentazione RDF in XML delle classi usando RDF-Schema per definire un generico elettrodomestico, i frigoriferi, ed i forni a microonde.

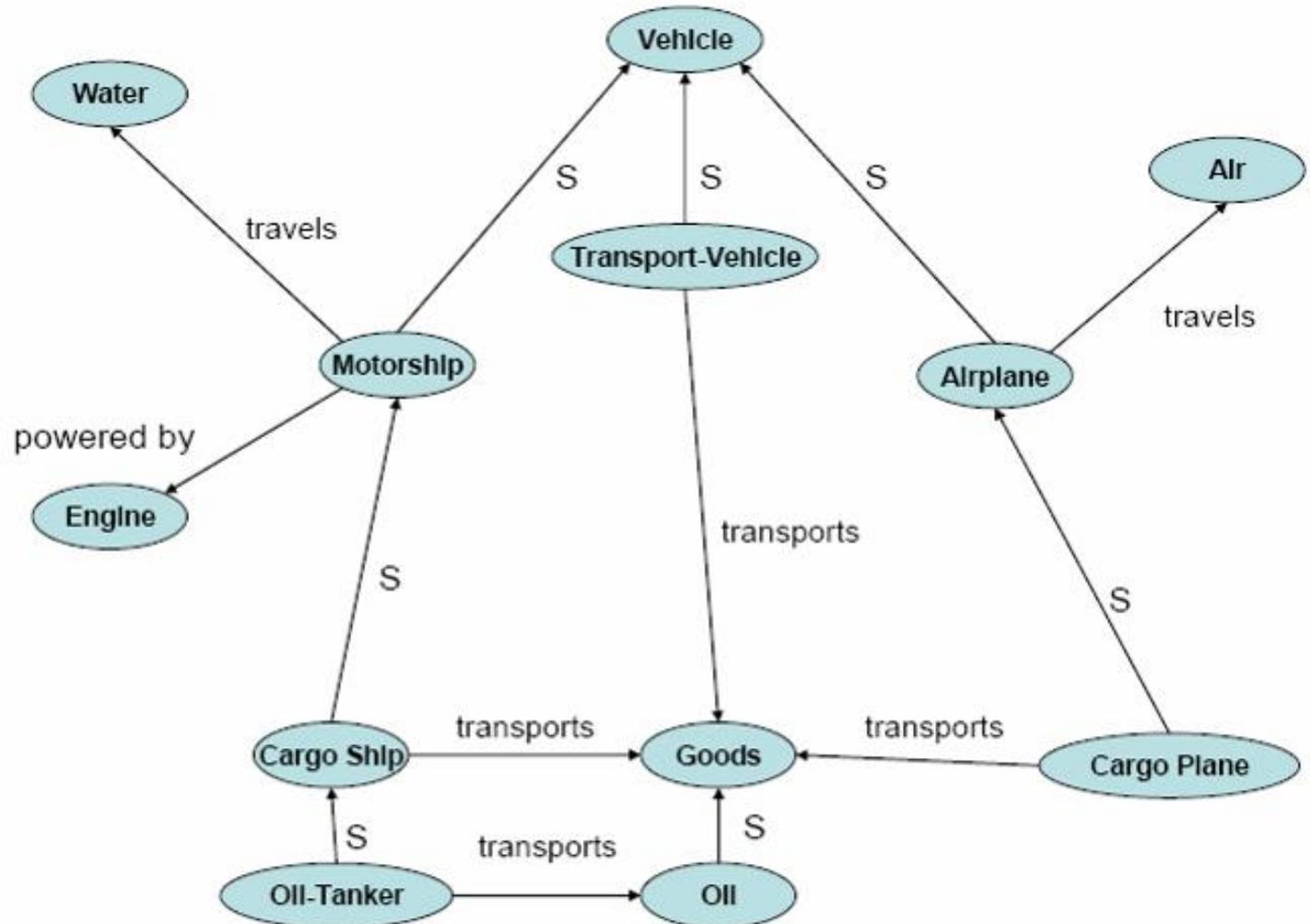
Esempi RDF

- Descrivere in XML il seguente grafo RDF:



Esempi RDF

Descrivere
in XML il
seguinte
grafo RDF:



Esempi RDF

- Sito del W3C

The screenshot shows the W3C website homepage. At the top is the W3C logo and the tagline "Leading the Web to its Full Potential...". Below this is a navigation bar with links for "Activities", "Technical Reports", "Site Index", "About W3C", and "Contact". The main content area features a paragraph about the W3C's mission and a list of news items. Two sidebars are visible: "W3C A to Z" with a list of links, and "Mission" with a list of links. The news items include "Extensible Stylesheet Language (XSL) Becomes a W3C Recommendation" and "XML Encryption Last Call Working Drafts Published".

W3C A to Z

- [Accessibility](#)
- [Amsys](#)
- [Annexes](#)
- [CG/P](#)
- [CSS](#)
- [CSS Validator](#)
- [Device Independence](#)
- [DOM](#)
- [HTML](#)
- [HTML Tidy](#)
- [HTML Validator](#)
- [HTTP](#)
- [Internationalization](#)
- [Jigsaw](#)
- [Libwww](#)
- [MathML](#)

News

- ▶ **Extensible Stylesheet Language (XSL) Becomes a W3C Recommendation**
16 October 2001: The World Wide Web Consortium today released the [Extensible Stylesheet Language \(XSL\) 1.0](#) as a W3C Recommendation. The specification has been reviewed by the W3C Membership, who favor its adoption by industry. Designers use an XSL stylesheet to express how source content should be styled, laid out, and paginated onto a presentation medium such as a browser window, a pamphlet or a book. Read the [press release](#) and [testimonials](#). ([News archive](#))
- ▶ **XML Encryption Last Call Working Drafts Published**
18 October 2001: The [XML Encryption Working Group](#) has released three Last Call Working Drafts. [XML Encryption Requirements](#) provides XML syntax and processing requirements for encrypting digital content. [XML Encryption Syntax and Processing](#) specifies a process for encrypting data and representing the result in an EncryptedData element for cipher data. [Encryption Transform for XML Signature](#) enables the repeated encryption and signing of parts of XML documents. Comments are welcome through 9 November. Read about the W3C [XML Encryption Activity](#). ([News archive](#))
- ▶ **XML Events Working Draft Published**

Mission

- [About W3C](#)
- [W3C in Seven Points](#)
- [Frequently Asked Questions](#)
- [Process Document](#)
- [Contact W3C](#)

Get Involved

- [Participate](#)
- [Mailing Lists](#)
- [Translations](#)
- [Open Source Software](#)
- [World Offices](#)
- [Employment](#)

- Descritto in RSS su

- <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>

Esercitazione RDF

- Creare un feed RSS per un sito a scelta