Facoltà di Ingegneria dell'informazione

Ingegneria della conoscenza 2010–11

Appello del 25 luglio 2011

Parte II			
Cognome		Laureando	si ☐ no ☐
Nome		Matricola	

4 7 punti	Si descrivano vantaggi e svantaggi dei tre approcci all'interfaccia verso documenti XML. Si progetti il DTD di un linguaggio XML per la comunicazione di quotazioni di borsa. Ogni file indica la data di riferimento e contiene un insieme di quotazioni e portafogli. Per ogni azione si indichi il nome, il prezzo (con la valuta che può essere EUR o USD) ed una descrizione in linguaggio naturale che può contenere una parte in grassetto. Ogni azione potrà inoltre essere collegata con una o più azioni ed appartenere ad uno o più portafogli. Ogni portafoglio dovrà figurare una sola volta nel documento e dovrà riportare il titolo. Si indichino eventuali approssimazioni necessarie e dire per ciascuna se e come è eliminabile con l'uso di XSD.

I tre approcci sono: DOM, SAX e JAXB
DOM è basato sul modello, costruisce (unmarshalling) e serializza (marshalling) modelli complessi in memoria. Occupa memoria, ma permette di leggere e scrivere i file, inoltre gestisce la validazione prima dell'elaborazione
SAX è snello ed occupa poca memoria perchè è basato su eventi, però non permette di modificare il file e richiede la ricostruzione dello stato del parsing a livello applicativo
JABX è simile a DOM, ma comporta la compilazione di classi Java che riproducono a livello di codice la struttura del documento. E' più intuitivo e semplice da usare a livello di codice, ma genera classi che dipendono dal particolare linguaggio ed introduce un overhead rispetto a DOM (sul quale si basa)

```
<!ELEMENT   file          (quotazione, portafoglio)*>
<!ELEMENT   quotazione    (prezzo, descrizione)>
<!ELEMENT   prezzo        #PCDATA>
<!ELEMENT   portafoglio   EMPTY>
<!ATTLIST   file          data     CDATA          #REQUIRED>
<!ATTLIST   quotazione
        id          ID          #REQUIRED
        nome        CDATA       #REQUIRED
        collegate   IDREFS      #REQUIRED
        portafogli  IDREFS      #REQUIRED>
<!ATTLIST   prezzo        valuta          (EUR|USD)     #REQUIRED>
<!ATTLIST   portafoglio
        id          ID          #REQUIRED
        nome        CDATA       #REQUIRED>
```

- non è possibile vincolare la forma delle date e del prezzo (approssimazione eliminabile in XSD con dei SimpleType)
- non è possibile tipizzare le relazioni collegate e portafogli (approssimazione NON eliminabile in XSD)

continuare sulla pagina 4

5

7 punti

Si definisca graficamente (tramite RDF-Graph) una semplice ontologia RDF-S per la descrizione di documenti. Ogni documento ha un titolo ed un autore ed è relativo ad uno o più contenuti. Gli autori sono persone o enti. Tra i contenuti possiamo avere richieste, autorizzazioni ed informazioni.

I documenti possono essere correlati tra di loro. I documenti autorizzativi sono documenti che contengono autorizzazioni e devono essere correlati ad almeno un documento che contiene una richiesta.

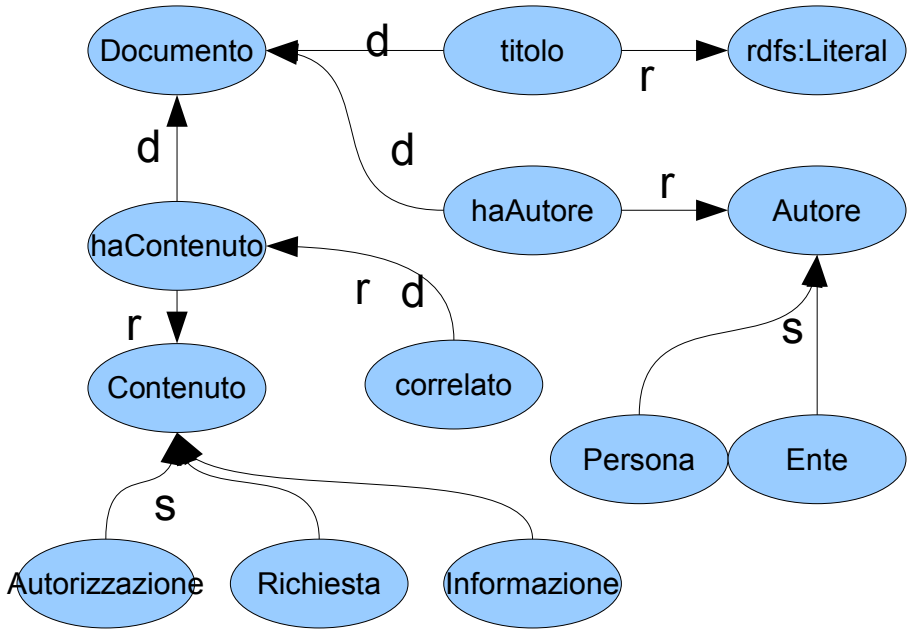
Indicare le approssimazioni e mostrare se e come possono essere superate in OWL indicando i soli frammenti da aggiungere con sintassi GrOWL.

Con riferimento all'ontologia proposta ed includendo eventuali aggiunte OWL siano date le seguenti query SPAR-QL:

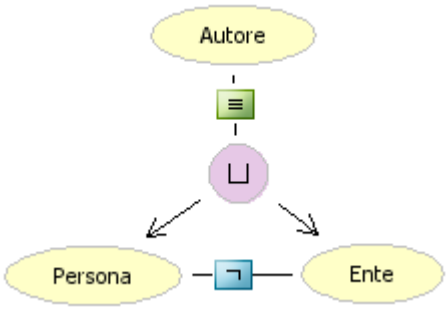
```
SELECT ?doc WHERE  {
    ?doc   contiene        ?a.
    ?a     rdf:type        Autorizzazione.
}
```

```
SELECT ?doc WHERE  {
    ?doc   rdf:type        DocumentoAutorizzativo.
}
```

Dire se sono equivalenti, se sono l'una inclusa nell'altra o nessuna delle due, giustificando la risposta

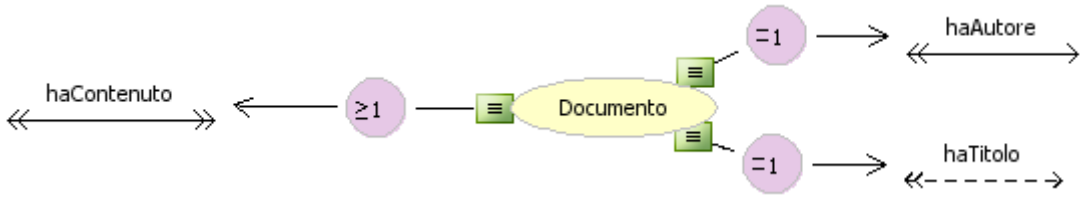


d = rdfs:domain r = rdfs:range s = rdfs:subClassOf

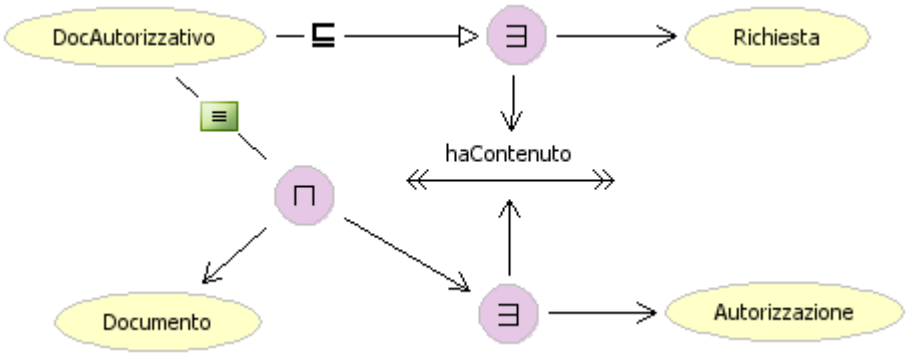


non posso indicare che Persona ed Ente esauriscono Autore, eliminabile come

non posso dire che i documenti hanno un solo titolo ed un solo autore e che deve avere almeno un contenuto, eliminabili come



non posso esprimere definizione e vincolo dei documenti autorizzativi, eliminabile come



Le due query sono semanticamente equivalenti, tuttavia la prima richiede di conoscere l'autorizzazione contenuta nel documento (perché deve unificare con la variabile ?a), mentre la seconda richiede solo di conoscerne l'esistenza.
Quindi la prima query restituisce un sottoinsieme della seconda.