

Lo schema di metadati Dublin Core Il modello astratto

Editoria digitale

Maria A. Alberti

Informatica per la comunicazione digitale
AA 17/18

Il modello astratto di DCMI

- Un modello astratto consente di evidenziare tutti gli aspetti che devono essere inclusi nello schema dei metadati
- Un modello astratto è universale
- Un modello è indipendente dalla particolare sintassi di codifica
- Il modello che segue è stato adottato dalla Dublin Core Metadata Initiative

Il modello astratto di risorsa

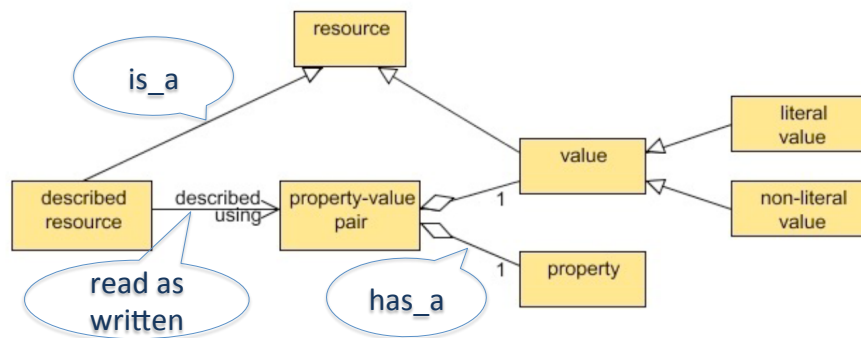


Diagramma rappresentato in UML, Unified Modelling Language: un modo standard di rappresentare graficamente object oriented software

M.A. Alberti. Editoria digitale, AA 17/18

3

Dublin Core

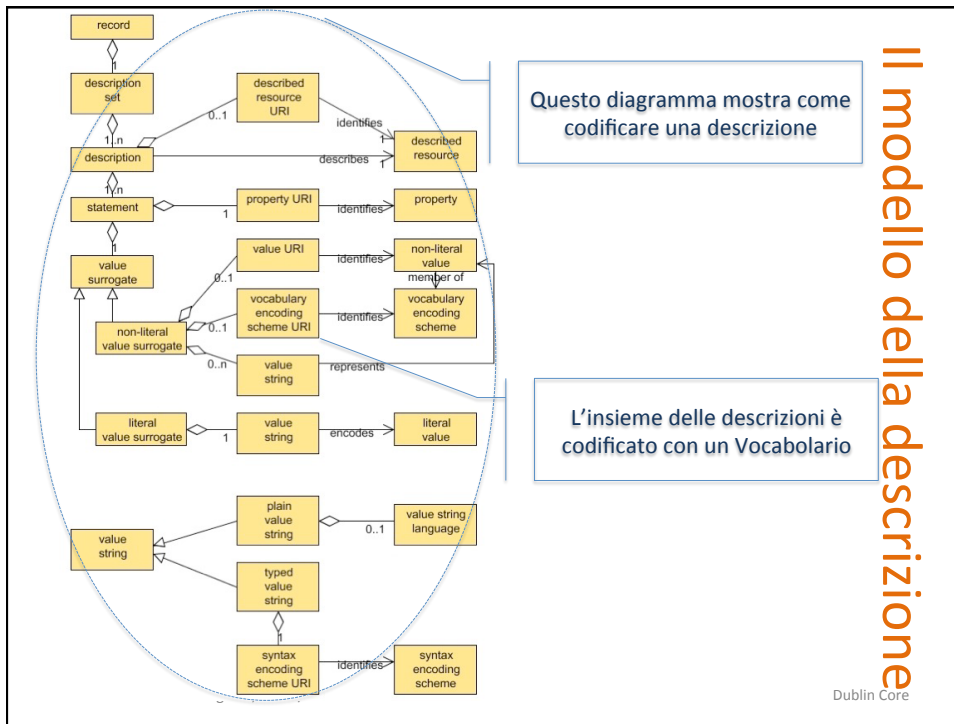
Il modello in sintesi

- Ogni risorsa che esiste nel mondo può essere descritta usando coppie proprietà-valore
- Anche ogni valore può essere una risorsa da descrivere
- Ogni valore è sia la stringa che indica il valore sia l'oggetto cui la stringa si riferisce, che a sua volta può essere una risorsa
creator: Leonardo da Vinci (l'artista)
creator: Leonardo da Vinci (il nome, riferimento all'artista)

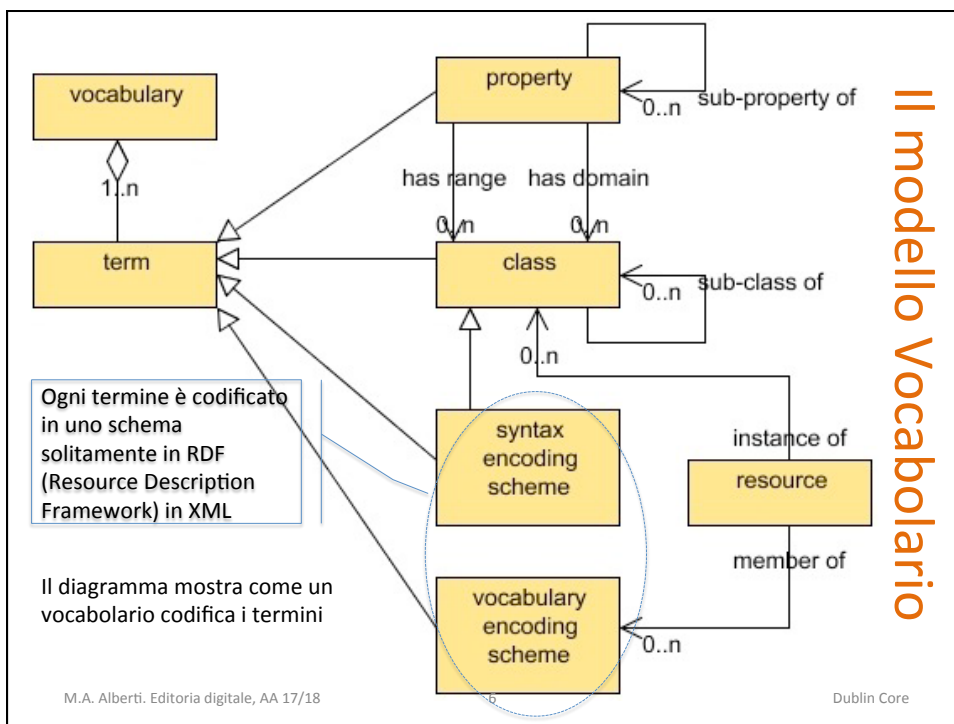
M.A. Alberti. Editoria digitale, AA 17/18

4

Dublin Core



Il modello della descrizione



Il modello Vocabolario

Il modello astratto generale

- Il modello astratto descrive dall'unità più piccola la *proprietà* o *valore* a quella più generale il *vocabolario* in 3 diagrammi che possono essere espansi: il diagramma della risorsa, della descrizione e del vocabolario
- Un modello ad alto livello per descrivere metadati in ogni schema
 - Dublin Core lo adotta come tutti gli altri schemi di metadattazione
- Facilita lo scambio di dati tra sistemi diversi

Lo spazio dei nomi

- Il cervello umano disambigua bene capendo il contesto le imprecisioni del linguaggio naturale
- Gli interpreti, i browser, i programmi sw no e hanno bisogno di definire precisamente la semantica dei nomi
- Le istruzioni per definire i termini devono essere accessibili al programma che li deve risolvere
- Spazio dei nomi è uno spazio virtuale in cui si definiscono identificatori e regole per formularli in modo che non si siano collisioni e ambiguità

Evitare collisioni

Nella creazione dei nomi di persona ci sono nelle diverse culture alcune regole

- Nome-luogo di nascita
 - Gesù di Nazareth
 - Leonardo da Vinci
- Nome-nome del padre
 - Lindon Johnson
 - Osama Bin Laden
- Nome-nome del figlio
 - Mohamed Abbas Abu Mazen

Esempio problematico: Corea, dove i nomi sono composti solo da 3 sillabe. Una definisce il cognome e due distinguono il nome proprio. Ma una sola sillaba per il cognome implica che ci siano pochi identificatori (circa 250) e che i primi 5 coprano circa metà popolazione.

Kim è il cognome di 11 mln di persone!

Risolvere le collisioni

- Come si costruiscono spazi di nomi che non producano collisioni di termini?
 - Il codice fiscale ne produce poche ma non le evita totalmente
 - Social Security Number *xxx-yy-zzzz* (10^9 combinazioni)
 - *xxx* area geografica, *yy* numero del gruppo nell'area geografica, *zzz* numeri casuali
 - Codice Fiscale *cccnnaamggllld*
 - *c* cognome, *n* nome, *a* anno di nascita, *m* mese, *g* giorno, *l* luogo di nascita, *d* un codice generato algoritmo da numeri e lettere precedenti.
 - I primi 6 caratteri sono consonanti presenti nel cognome e nome scelte in base ad alcune regole, il giorno di nascita distingue il genere

Gli indirizzi postali

- La strada, il numero civico, il codice postale, la città, la provincia identificano in modo univoco un edificio
- Modalità diverse
 - Via Comelico 35 I - 20135 Milano MI
 - 120 Massachusetts Ave., Cambridge, MA 02139
indirizzi iniziano dall'elemento meno significativo

Lo spazio dei nomi DC

<http://dublincore.org/documents/dcmi-namespace/>

- Spazio dei nomi URI, Uniform Resource Identifier mantenuto da purl-org (persistent url) <http://purl-org/dc/terms>
- Es: `contributor`, un'istanza di `property`, la sua definizione, il suo URI, la versione
- Embedded nella pagina il codice in RDF usato dai programmi per l'interpretazione del nome

Source code: contributor

```
<h2>Section
  3:
  Properties in the <abbr title="http://purl.org/dc/elements/1.1/" class="xmlns"/>/elements/1.1/</abbr> namespace</h2>
<table cellpadding="0" cellspacing="0" class="border">
<tbody id="elements-contributor" class="term" resource="http://purl.org/dc/elements/1.1/contributor">
<tr>
<th colspan="2" scope="rowgroup">
Term Name:
<td colspan="2">
<span>contributor</span>
<link property="rdfs:isDefinedBy" href="http://purl.org/dc/elements/1.1/" />
<time property="dcterms:issued" datetime="1999-07-02"> </time>
<time property="dcterms:modified" datetime="2008-01-14"> </time>
</tr>
<tr class="attribute">
<th scope="row">URI:
<td axis="URI">
<a href="http://purl.org/dc/elements/1.1/contributor">http://purl.org/dc/elements/1.1/contributor</a>
</td>
</tr>
</tbody>
</table>
```

Dove trovare la definizione rdf di contributor che punta a questa pagina

Gli elementi DC

- Sono istanze del tipo `property` definite in RDF -Resource Description Framework
- RDF strumento del W3C per la codifica, lo scambio e il riutilizzo dei metadati, scritto in XML
- Gli elementi DC ereditano tutte le caratteristiche e la semantica del tipo `property`
- <http://www.w3.org/1999/02/22-rdf-syntax-ns>

Schemi di metadati

- Dublin Core non è l'unico schema di metadati, diventa quindi importante che sia possibile scambiare dati tra diversi schemi
- Si pone il problema della **interoperabilità** tra dati
- Il modello astratto è usato da tutti gli schemi
- RDF è lo standard per consentire interoperabilità tra applicazioni che condividono informazioni su web

Interoperabilità dei dati

Il framework per definire l'interoperabilità tra i dati a 4 livelli. Dal livello meno restrittivo a quello che stabilisce più formalmente i vincoli

- > 4: Description Set Profile Interoperability
 - Shared formal vocabularies and constraints in records
- > 3: Description Set syntactic interoperability
 - Shared formal vocabularies in exchangeable records
- > 2: Formal semantic interoperability
 - Shared vocabularies based on formal semantics
- > 1: Shared term definitions
 - Shared vocabularies defined in natural language

I livelli di interoperabilità

- Stabiliscono vincoli sempre più precisi cui attenersi per definire i metadati.
- Da livello 1 più orientato alla lettura da parte di un operatore, in linguaggio naturale
- Al livello 4 inteso per essere manipolato dagli algoritmi nei programmi
- Ogni livello si basa sempre più pesantemente sulle definizioni formali in RDF e quindi ogni livello superiore è più adatto per la gestione automatica con programmi sw e meno leggibile

Livello 1

Corrisponde ad usare le definizioni in linguaggio naturale dei nomi del Dublin Core

Implica usare gli elementi e il vocabolario DC

Esempio: a livello 1 si usa il termine **creator** per indicare autore, regista etc.

➤ 1: Shared term definitions

- Shared vocabularies defined in natural language

Livello 2

Corrisponde all'uso esplicito o almeno implicito della semantica RDF sottostante ai termini Dublin Core

Va oltre alla terminologia DC ma comincia a fare riferimento alle definizioni formale in RDF

> 2: Formal semantic interoperability

- Shared vocabularies based on formal semantics

Livello 3

Corrisponde all'uso esplicito del modello astratto dei dati adottato dalla Dublin Core Metadata Initiative

Quindi va fatto esplicito riferimento alla definizione RDF

> 3: Description Set syntactic interoperability

- Shared formal vocabularies in exchangeable records

Livello 4

Fornisce i vincoli formali alle descrizioni in modo da essere compatibili con le specifiche del Description Set Profile

Un Description Set è l'insieme delle affermazioni su una risorsa e un metadata record corrisponde ad un Description Set

> 4: Description Set Profile Interoperability

- Shared formal vocabularies and constraints in records

Dublin Core conclusioni

- Gli **elementi** di base DC sono 15, e molti altri sono **termini**
- Il DC può essere **esteso**: precisato e qualificato dalle diverse comunità d'interesse
- I metadata record composti da coppie **elemento-valore**
- I metadata sono essere implementati in HTML, usando il tag **meta**, con attributi **name** – **content**
- Adotta un **modello astratto** di riferimento generale
- Dichiarazioni formali esplicite della semantica nello **spazio dei nomi** DC

Pregi e limiti del DC

Dublin Core è generico e serve per descrivere ogni tipo di risorsa

- Per questo non è preciso
- Solo poche comunità hanno deciso di estenderlo
- DC è semplice, basso costo d'adozione
- È il punto di partenza e poi le cose si complicano con le definizioni RDF