



InformaticaUmanistica

Biblioteche Digitali

Pasquale Savino

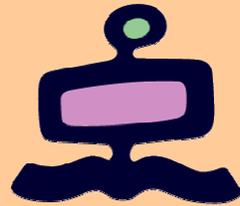
ISTI - CNR



UNIVERSITÀ DI PISA

Programma del corso

- ◆ Introduzione alle Biblioteche Digitali
- ◆ **Metadati**
- ◆ Esempi di Biblioteche Digitali (con esercitazioni)
- ◆ Architettura e tecnologie di base delle Biblioteche Digitali
- ◆ Progettazione di una Biblioteca Digitale (con esercitazioni)
- ◆ Nuove tendenze



InformaticaUmanistica

Metadati



UNIVERSITÀ DI PISA

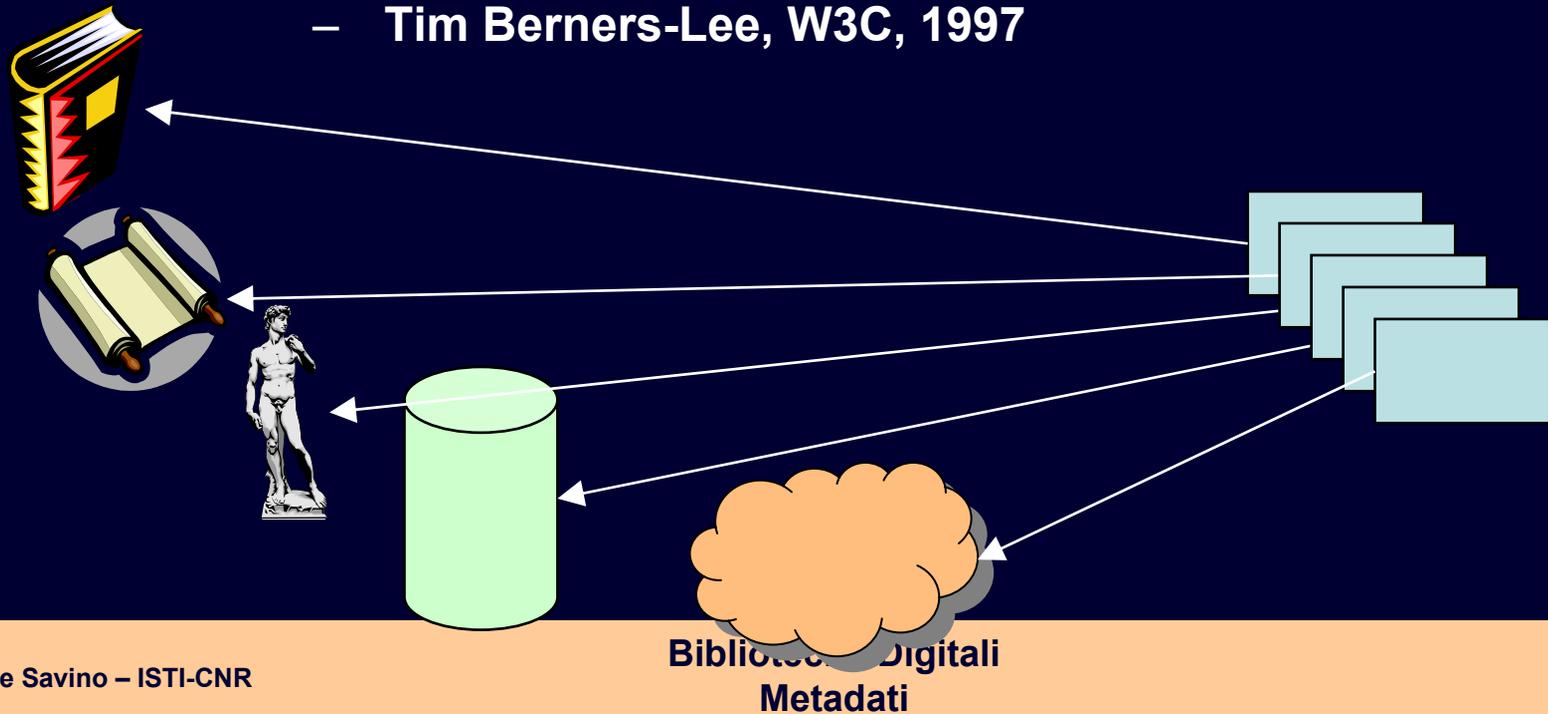
Sommario

- ◆ **Definizione**
- ◆ **XML**
- ◆ **I vari tipi di metadati**
- ◆ **Modello di riferimento – RDF**
- ◆ **Modelli di metadati**
 - MARC21
 - Dublin Core
 - MPEG-7
 - ECHO

Metadati - Definizione

Cosa sono i metadati?

- Alcune definizioni semplici...
- 'Structured data about data'.
 - **Dublin Core Metadata Initiative FAQ, 2003**
- Machine-understandable information about Web resources or other things.
 - **Tim Berners-Lee, W3C, 1997**



Una visione “funzionale” dei metadati

- Data associated with objects which relieves their potential users of having to have full advance knowledge of their existence or characteristics.
A user might be a program or a person.
 - **Lorcan Dempsey & Rachel Heery, 1998**
- Structured data about resources that can be used to help support a wide range of operations
 - **Michael Day, 2001**

A quali risorse possono essere associati dei metadati?

- I metadati possono essere associati a qualunque cosa – digitale, fisica, risorsa “astratta”
 - HTML documents
 - digital images
 - databases
 - books
 - museum objects
 - archival records
 - metadata records
 - Web sites
 - collections
 - services
 - physical places
 - people
 - institutions
 - abstract “works”
 - concepts
 - events

Che informazione contengono i metadati?

- I record di metadati includono
 - ➔ Record bibliografici appartenenti a cataloghi o ottenuti da servizi di indicizzazione e creazione di abstract
 - ➔ Descrizione del materiale archiviato
 - ➔ Record relativi ad oggetti contenuti nella documentazione del museo
 - ➔ Descrizione di oggetti digitali (documenti, immagini, video software, ...)
 - ➔ Descrizione di collezioni di oggetti digitali
 - ➔ Descrizione di servizi in rete
 - ➔ Descrizione di record di metadati

Quali operazioni possono essere effettuate sui metadati?

- Operazioni da parte degli utenti o da parte di tool software
- I metadati possono essere usati per supportare diverse funzioni, quali ad esempio
 - ➔ **resource disclosure & discovery**
 - ➔ **resource management, (incluso preservation)**
 - ➔ **intellectual property rights management**
 - ➔ **Autenticazione ed autorizzazione**
 - ➔ **Personalizzazione e localizzazione dei servizi**
- **Le diverse funzioni richiedono diversi tipi e classi di metadati**
 - ➔ **Non è possibile trovare una soluzione unica per tutti gli usi**
 - ➔ **È necessario specificare i requisiti funzionali per definire le caratteristiche dei metadati da utilizzare**

Elementi ed insiemi di elementi dei Metadati

- I Metadati descrivono gli attributi e le proprietà di una risorsa
- Ogni attributo o proprietà è descritto da un **metadata element**
 - ➔ Può essere identificato e definito in modo preciso
 - ➔ Ogni metadata element può essere rappresentato in diverse forme
- Un **metadata element set** è
 - ➔ Insieme coerente di elementi
 - ➔ Creato per uno scopo specifico, come una singola unità
- Schema
 - ➔ Rappresentazione strutturata di un element set

Autore = J Smith
Data = 2001-11-05
Titolo = Report

Resource1

I tipi di metadati

Tipi di metadati

- ◆ **Metadati descrittivi o per “resource discovery”**
 - Utilizzati per descrivere le risorse della Biblioteca Digitale
 - Utilizzati per facilitare la ricerca di documenti nella BD
 - Cataloghi, indici, ecc.
- ◆ **Metadati strutturali**
 - Descrivono la struttura degli oggetti della BD e le relazioni che esistono tra oggetti (o loro parti) ed altri oggetti
- ◆ **Metadati amministrativi**
 - Utilizzati per la gestione delle collezioni di documenti e per il controllo dell'accesso

Metadati per resource discovery

- Gli utenti desiderano
 1. Individuare le risorse in accordo a determinati criteri
 - Ad es. *Trova tutti i documenti scritti da ... che parlano di ...*
 2. (oppure) identificare una specifica risorsa
 - Essere in grado di distinguere tra diverse risorse simili tra loro
 3. Selezionare
 - **Scegliere una risorsa che sia appropriata ai bisogni**
 4. Localizzare la risorsa
 5. *Ottenere/accedere alla risorsa*
 6. *Utilizzare la risorsa*
 - *open, read, display, run, play, copy, unpackage/repackage*
 - *Interpretare il contenuto*
- I metadati per resource discovery sono utilizzati principalmente per le operazioni 1-4

Metadati per resource discovery

Continuum di complessità/functionalità

Indici full-text, feature per immagini	Generato con tool software in modo automatico. Secondo alcune definizioni potrebbero non essere considerati metadati	discovery per contenuto, localizzazione
Forme semanticamente semplici (e.g. Dublin Core)	Permettono la descrizione di un gran numero di risorse In alcuni casi possono essere generati in modo automatico, in altri sono prodotti manualmente	discovery, identificazione, selezione, localizzazione
Forme complesse più ricche (e.g. MARC, ECHO, etc)	Relativo a tipi di risorse specifiche Spesso associato con domini/comunità specifici La creazione può richiedere un significativo intervento umano	discovery, identificazione, selezione, localizzazione, accesso, uso

Associazione di una risorsa ai metadati [1/3]

Creator = J Smith

Date = 2001-11-05

Title = Report

I metadati possono essere contenuti all'interno della risorsa

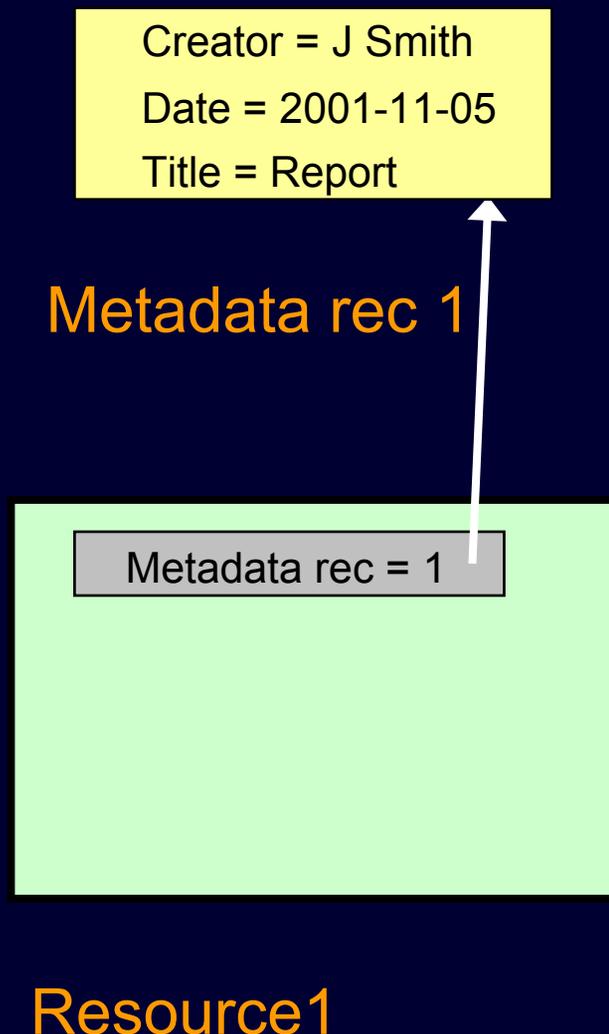
e.g. elementi `meta` in doc HTML;

Proprietà del documento nei word processor

- Le risorse possono permettere che i metadati siano contenuti al loro interno (embedding)?
- Il creatore dei metadati può avere i diritti di **scrittura** sulla risorsa?
- Può un servizio **estrarre** i metadati che sono contenuti nella risorsa?
- Possono esistere metadati per **aggregati di risorse**? Come vengono gestiti?

Resource 1

Associazione di una risorsa ai metadati [2/3]



*Il record dei metadati è un oggetto separato
L'identificatore del record e' contenuto
nella risorsa*

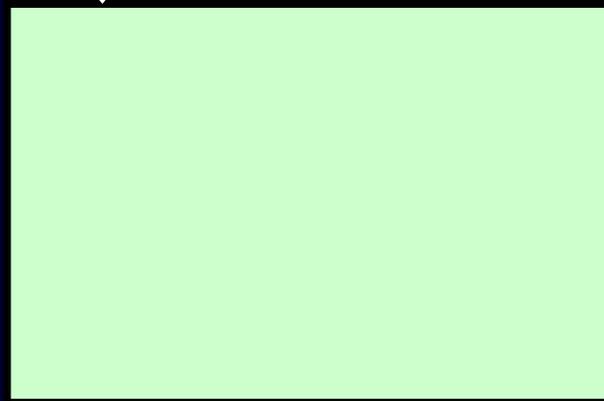
e.g. elementi `link` elements in doc HTML
I record di metadati possono essere remoti
rispetto alla risorsa

- Le risorse devono poter contenere i link
- Il creatore dei metadati può avere i diritti di **scrittura** sulla risorsa?
- Devono esistere dei meccanismi per seguire i link
- Cosa succede se una risorsa viene cancellata?
- Possono esistere metadati per **aggregati di risorse**? Come vengono gestiti?

Associazione di una risorsa ai metadati [3/3]

Doc = 1
Creator = J Smith
Date = 2001-11-05
Title = Report

Metadata rec 1



Resource 1

***Il record dei metadati è un oggetto separato
L'identificatore della risorsa e' contenuto nel
record***

Il record dei metadati può essere remoto rispetto alla risorsa

Non richiede che nessuna informazione sui metadati sia contenuta nella risorsa Non richiede che il creatore dei metadati possa modificare la risorsa

Il record dei metadati può essere creato indipendentemente dalla risorsa – possibilità di record multipli

I servizi utilizzano i metadati indipendentemente dalla risorsa

Il record dei metadati può continuare ad esistere anche se la risorsa viene cancellata

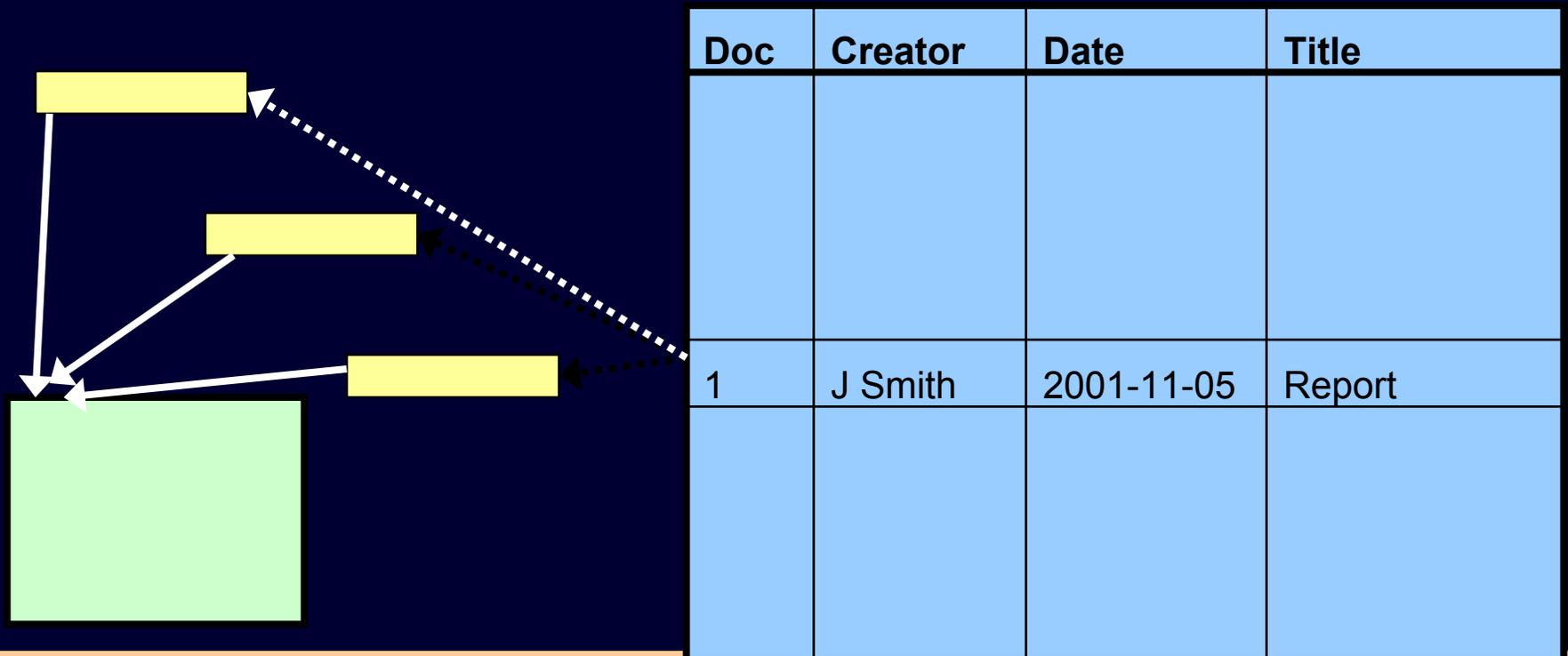
Il record dei metadati può descrivere qualunque oggetto che abbia un identificatore

Gestione dei metadati

Il record dei metadati viene usato separatamente dalla risorsa che viene descritta

I metadati sono essi stessi una risorsa

Il contenuto dei metadati può essere memorizzato nella forme più appropriate per il servizio: in un database, in forms, ecc.



XML

XML e Metadati

- ◆ XML (eXtensible Markup Language)
- ◆ XML viene usato per descrivere i dati
- ◆ I “tag” XML non sono predefiniti come in HTML
- ◆ XML viene spesso utilizzato come linguaggio per descrivere i metadati

Un semplice esempio

XML Declaration ("questo è XML")

Codifica dei caratteri usata

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

```
<book>
```

```
  <chapter>
```

```
    <title>Introduction</title>
```

```
  </chapter>
```

```
  <chapter>
```

```
    <title>Story</title>
```

```
    <Section>
```

```
      <title>Part 1</title>
```

```
    </Section>
```

```
    <Section>
```

```
      <title>Part 2</title>
```

```
    </Section>
```

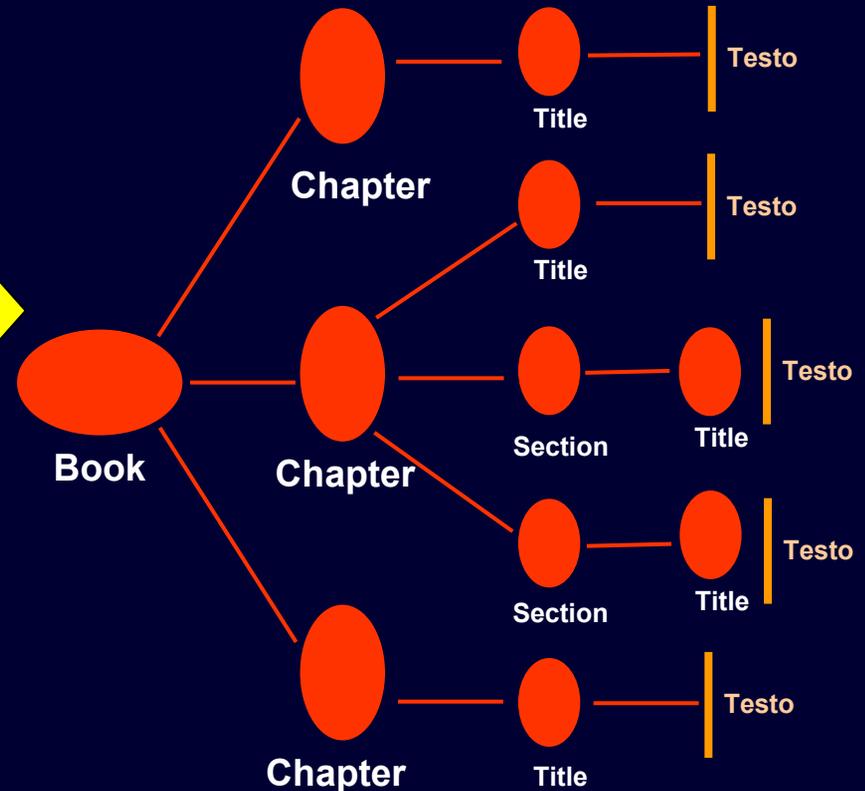
```
  </chapter>
```

```
  <chapter>
```

```
    <title>Index</title>
```

```
  </chapter>
```

```
</book>
```



Organizzazione gerarchica dei dati

XML elements

◆ Costituenti principali dei documenti XML

◆ Possono contenere testo, attributi o altri elementi

```
<name att1 = "val1" att2 = "val2">  
    contenuto  
</name>
```

◆ Regole sui nomi degli Elementi:

- I nomi possono contenere lettere, numeri ed altri caratteri
- Non possono iniziare con numeri o caratteri di punteggiatura
- Non possono iniziare con "XML"
- Non possono contenere spazi
→ `<first_name>`, `<last_name>`.

Esempio documento XML (Metadata)

tag

```
<?xml version="1.0"?>
<!DOCTYPE dlib-meta0.1 SYSTEM "http://www.dlib.org/dlib/dlib-meta01.dtd">
<dlib-meta0.1>
  <title>Digital Libraries and the Problem of Purpose</title>
  <creator>David M. Levy</creator>
  <publisher>Corporation for National Research Initiatives</publisher>
  <date date-type = "publication">January 2000</date>
  <type resource-type = "work">article</type>
  <identifier uri-type = "DOI">10.1045/january2000-levy</identifier>
  <identifier uri-type = "URL">http://www.dlib.org/dlib/jan00/01.html</identifier>
  <language>English</language>
  <relation rel-type = "InSerial">
    <serial-name>D-Lib Magazine</serial-name>
    <issn>1082-9873</issn>
    <volume>6</volume>
    <issue>1</issue>
  </relation>
  <rights>Copyright (c) David M. Levy</rights>
</dlib-meta0.1>
```

element

Attributi

◆ Gli attributi forniscono informazione aggiuntiva sugli elementi

```
<date date-type = "publication">January 2000</date>
```

- Ad es. gli attributi possono associare una label univoca all'elemento, oppure possono descrivere una proprietà dell'elemento.

◆ Sintassi

```
Name = "value"
```

◆ Un elemento può avere qualunque numero di attributi, ma ogni attributo ha una sola occorrenza

- `<team persona="Mario" persona="Anna">`
- `<team persone="Mario Anna">`
- `<team personal="Mario" persona2="Anna">`
- `<team>`
 `<persona>Mario<\persona>`
 `<persona>Anna<\persona>`
`<\team>`

Utilizzo elementi

Forma
errata

Forme
corrette

Esempio documento XML (Metadata)

Attributo dell'elemento date

```
<?xml version="1.0"?>
<!DOCTYPE dlib-meta0.1 SYSTEM "http://www.dlib.org/dlib/dlib-meta01.dtd">
<dlib-meta0.1>
  <title>Digital Libraries and the Problem of Purpose</title>
  <creator>David M. Levy</creator>
  <publisher>Corporation for National Research Initiatives</publisher>
  <date date-type = "publication">January 2000</date>
  <type resource-type = "work">article</type>
  <identifier uri-type = "DOI">10.1045/january2000-levy</identifier>
  <identifier uri-type = "URL">http://www.dlib.org/dlib/jan00/01.html</identifier>
  <language>English</language>
  <relation rel-type = "InSerial">
    <serial-name>D-Lib Magazine</serial-name>
    <issn>1082-9873</issn>
    <volume>6</volume>
    <issue>1</issue>
  </relation>
  <rights>Copyright (c) David M. Levy</rights>
</dlib-meta0.1>
```

Namespaces [1/2]

- ◆ **Meccanismo usato per identificare diversi “spazi” dei nomi in XML e combinarli nello stesso documento**
 - Nomi di *elementi* o *attributi*
- ◆ **È un modo per identificare diversi *dialetti*, che hanno una particolare semantica e/o modalità di elaborazione.**
 - Ad es. namespace dei termini matematici
 - Viene usato anche per inserire delle direttive di visualizzazione (XSLT) del documento XML
- ◆ **Ad esempio <title> può avere significati diversi in contesti diversi, oppure <key> può indicare una chiave di protezione (in un contesto di gestione della sicurezza) oppure una chiave d’accesso (nel contesto di un database)**
- ◆ **Un namespace consiste di un gruppo di elementi e di nomi di attributi. I nomi del namespace vengono identificati utilizzando un prefisso**
 - `ns-prefix:local-name`
- ◆ **Un namespace deve essere dichiarato prima di poterlo utilizzare**
 - `xmlns:name =“url”`

Namespaces [2/2]

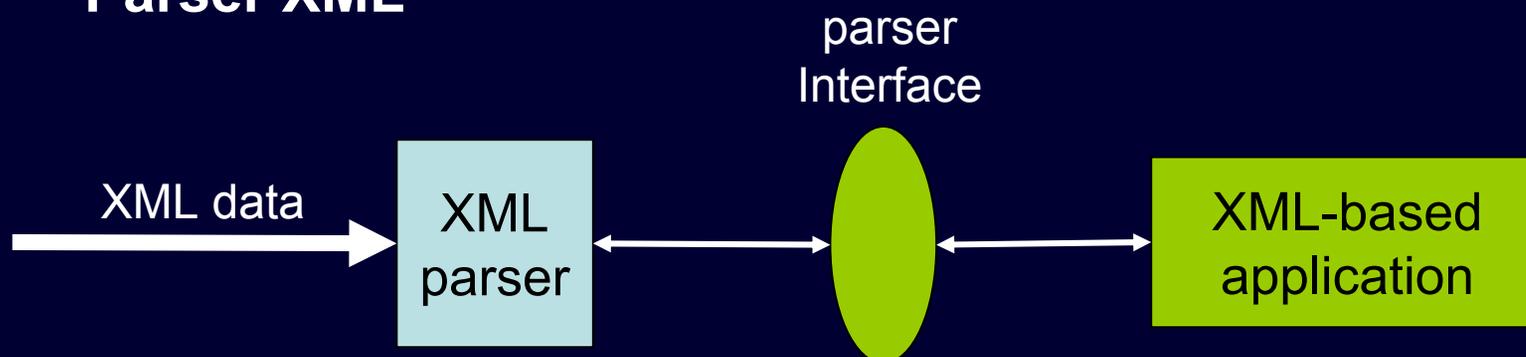
```
<?xml version= "1.0" encoding= "utf-8" ?>
<html xmlns="http://www.w3.org/1999/xhtml1"
      xmlns:mt="http://www.w3.org/1998/mathml" >
<head>
  <title> Title of XHTML Document </title>
</head><body>
<div class="myDiv">
  <h1> Heading of Page </h1>
  <mt:mathml>
    <mt:title> ... MathML markup . . .
  </mt:mathml>
  <p> more html stuff goes here </p>
</div>
</body>
</html>
```

Default 'space'
è *xhtml*

mt: prefisso che indica
lo 'spazio' *mathml*

Elaborazione di documenti XML

Parser XML



- ◆ **Il parser verifica che il file XML sia sintatticamente corretto**
- ◆ **Questi dati sono detti *well-formed***
 - È la condizione minima per essere considerati in formato XML
- ◆ **Il parser deve terminare l'elaborazione se i dati non sono *well-formed***
 - Per es. Termina l'elaborazione e lancia un'eccezione all'applicazione.

Definizione di modelli di documenti XML

- ◆ In XML è possibile creare il proprio modello di markup, specificando quali devono essere gli elementi e gli attributi dei documenti
- ◆ Vi sono due modi per definire un modello di documento XML
 - XML Document Type Declaration (DTD) – Fa parte della specifica di XML
 - XML Schema (anche detto XSD) – Permette di specificare vincoli più precisi sui documenti XML.
- ◆ Il DTD e XML schema permettono di specificare quali sono gli elementi permessi ed i nomi degli attributi, le regole di composizione gerarchica degli elementi, e le restrizioni di tipo o di contenuto degli elementi

Validazione dei documenti XML

- ◆ Documenti XML "Well Formed"
 - Un documento XML "Well Formed" ha una sintassi XML corretta
- ◆ Documenti XML "Validi"
 - Un documento XML "Valido" è un documento XML "Well Formed", che è anche conforme alle regole di un **Document Type Definition (DTD)** o di un **XML Schema Definition (XSD)**.
- ◆ **Il processo di elaborazione termina con un errore se il documento XML non è valido.**

XML DTD

- ◆ Il DTD definisce la struttura del documento con una lista di elementi legali
- ◆ Si può avere una dichiarazione del DTD (dichiarazione DOCTYPE)

Interna

```
<?xml version="1.0"?>
<!DOCTYPE note
[ <!ELEMENT note (to,from,heading,body)>
  <!ELEMENT to (#PCDATA)>
  <!ELEMENT from (#PCDATA)>
  <!ELEMENT heading (#PCDATA)>
  <!ELEMENT body (#PCDATA)>
]>
<note>
  <to>Jo</to>
  <from>Mary</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend</body>
</note>
```

Esterna

```
<?xml version="1.0"?>
<!DOCTYPE note SYSTEM "note.dtd">
<note>
  <to>Jo</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this
weekend!</body>
</note>
--- note.dtd ---
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
```

Perché usare un DTD

- ◆ Usando i DTD ogni documento contiene una descrizione del proprio formato.
- ◆ Gruppi di utenti possono accordarsi sull'uso di DTD comuni per facilitare lo scambio dei documenti.
- ◆ Le applicazioni possono usare un DTD standard per verificare che i dati ricevuti dall'esterno sono validi.
- ◆ Si possono utilizzare i DTD per verificare che i dati prodotti abbiano la struttura corretta.

Sintassi del DTD [1/8]

◆ Un DTD definisce

- L'insieme degli elementi permessi
 - ➔ Questo si può considerare come il “vocabolario” del linguaggio
- Il “modello del contenuto” di ogni elemento
 - ➔ Specifica quali elementi o dati possono essere inclusi in un elemento, in quale ordine, in quale numero, e se sono obbligatori o opzionali.
- Specifica l'insieme di attributi permessi per ogni elemento
 - ➔ Ogni dichiarazione definisce il nome, il tipo del dato, i valori di default (se ci sono) ed il comportamento dell'attributo

Sintassi del DTD [2/8]

- ◆ Un DTD è composto da un insieme di dichiarazioni
- ◆ Dichiarazioni di Elementi
 - Per ogni elemento che si userà nel documento si **deve** avere una dichiarazione
 - `<!ELEMENT name content-model>`
 - Name è il nome dell'elemento (ad es. titolo, persona, ecc.)
 - Il `content-model` permette di specificare quale tipo di contenuto può essere incluso nell'elemento, quanti possono essere i suoi elementi e in che ordine vanno inseriti



Sintassi del DTD [3/8]

◆ Vi sono 5 diversi tipi di content model

- Elementi EMPTY

→ `<!ELEMENT graphic EMPTY>`

- Elementi con nessuna restrizione sul contenuto

→ L'elemento può contenere qualunque altro elemento

→ Poco utile in un DTD reale, si usa nella fase di sviluppo

→ `<!ELEMENT esempio ALL>`

- Elementi che contengono solo dati di tipo carattere

→ `<!ELEMENT esempio (#PCDATA)>`

- Elementi che contengono solo altri elementi

→ Usa una sintassi particolare per specificare i vincoli sugli elementi

→ `<!ELEMENT articolo (para+)>`

→ `<!ELEMENT articolo (tit, (para | sez)+)>`

→ `<!ELEMENT articolo (tit, sottotit?, ((para+, sez*) | sez+))>`

+ una o più
ripetizioni

* Zero o più
ripetizioni

? opzionale

| richiede uno dei
due elementi

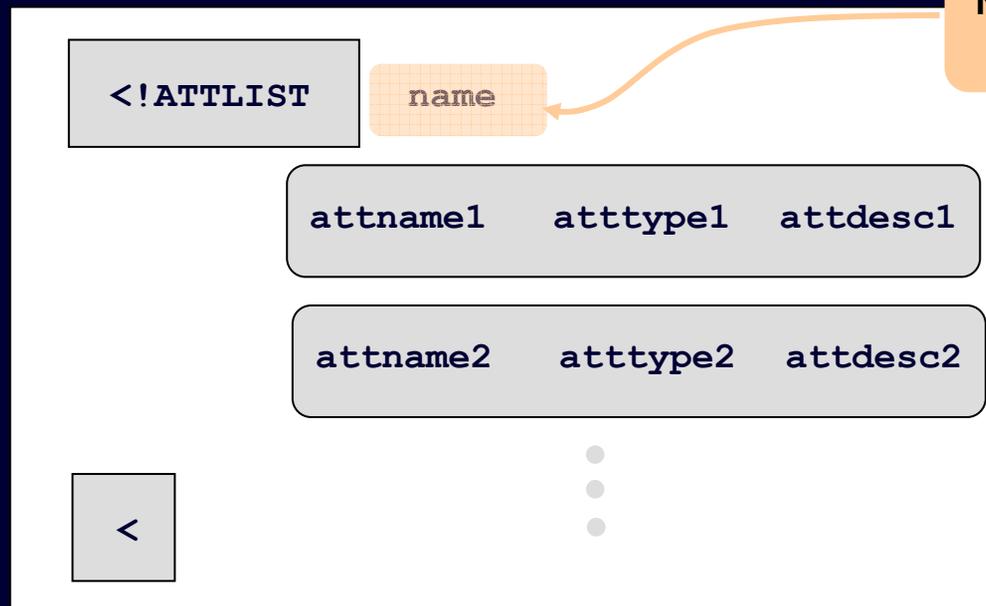
Sintassi del DTD [4/8]

- Elementi con contenuto misto
 - ➔ Possono contenere sia elementi che caratteri
 - ➔ `<!ELEMENT para (#PCDATA | subpara*)>`
 - ➔ `<!ELEMENT articolo (titolo, autore*, (para | tabella | lista)+, bibliografia?)>`
 - L'elemento articolo contiene
 - > Un titolo
 - > Zero o più autori
 - > Una o più occorrenze di para, tabella, lista
 - > Una bibliografia opzionale
 - > Ad es. titolo, autore, autore, para, list, bibliografia è un elemento articolo corretto
 - ➔ I simboli * + | ? hanno lo stesso significato del caso precedente

Sintassi del DTD [5/8]

◆ Dichiarazioni di attributi

- Per ogni elemento vengono dichiarati tutti i suoi attributi con la seguente sintassi



Sintassi del DTD [6/8]

◆ Una dichiarazioni di attributi

- Da un nome all'attributo
- Specifica il tipo di dato dell'attributo
- Descrive il comportamento dell'attributo (ad es. se ha un valore di default, o se l'autore deve specificare un valore)

```
<!ATTLIST memo
```

```
id          ID          #REQUIRED
security    (high | low) "high"
keywords    NMTOKENS    #IMPLIED
```

```
>
```

L'attributo id è di tipo ID ed è obbligatorio

Security può assumere due valori, "high" è di default

Keywords è opzionale e non ha valori di default

Sintassi del DTD [7/8]

◆ I datatype degli attributi sono

- CDATA: caratteri
- NMTOKEN (NMTOKENS): stringa di caratteri (lista di stringhe)
- ID: identificatore unico
- IDREF (IDREFS): riferimento ad un identificatore
- ENTITY (ENTITIES): accetta un nome di ENTITA' come valore
- Enumeration value list: lista di keywords

Sintassi del DTD [8/8]

◆ Comportamenti degli attributi

- Assegnamento di valori di default

→ `<!ATTLIST message`

`importance (high | medium | low) "medium" >`

- `#IMPLIED`: attributo opzionale e non vi è alcun valore di default
- `#REQUIRED`: deve essere fornito un valore
- `#FIXED`: il valore è prefissato e non può essere modificato

Esempio documento XML (Metadata)

```
<?xml version="1.0"?>
<!DOCTYPE dlib-meta0.1 SYSTEM "http://www.dlib.org/dlib/dlib-meta01.dtd">
<dlib-meta0.1>
  <title>Digital Libraries and the Problem of Purpose</title>
  <creator>David M. Levy</creator>
  <publisher>Corporation for National Research Initiatives</publisher>
  <date date-type = "publication">January 2000</date>
  <type resource-type = "work">article</type>
  <identifier uri-type = "DOI">10.1045/january2000-levy</identifier>
  <identifier uri-type = "URL">http://www.dlib.org/dlib/jan00/01.html</identifier>
  <language>English</language>
  <relation rel-type = "InSerial">
    <serial-name>D-Lib Magazine</serial-name>
    <issn>1082-9873</issn>
    <volume>6</volume>
    <issue>1</issue>
  </relation>
  <rights>Copyright (c) David M. Levy</rights>
</dlib-meta0.1>
```

Il DTD del D-Lib Magazine [1/6]

```
<!-- DTD to mark up the metadata elements in D-Lib Magazine -->  
<!-- William Y. Arms, Cathy Rey, March 8, 1999 Updated June 16,  
1999 -->
```

```
<!ELEMENT dlib-meta0.1 (title, creator+, publisher, date, type,  
identifier+, language*, relation, rights+)>
```

```
<!-- Element names are from the Dublin Core set of 15 names. -->  
<!-- Attributes are used to clarify the usage by D-Lib Magazine. -->
```

Continua nella prossima slide

Il DTD del D-Lib Magazine [2/6]

<!ELEMENT title (#PCDATA)>

<!-- Title as supplied with all punctuation -->

<!ELEMENT creator (#PCDATA)>

<!-- This element is repeated for each author or other creator -->

<!-- It contains the name of the author as provided, -->

<!-- without affiliation or contact information. -->

<!ELEMENT publisher (#PCDATA)>

<!-- Publisher is "Corporation for National Research Initiatives" -->

Continua nella prossima slide

Il DTD del D-Lib Magazine [3/6]

```
<!ELEMENT date (#PCDATA)>
<!ATTLIST date
    date-type CDATA #FIXED "publication">
<!-- Issue date, e.g., "July 1995", or "July/August 1998" -->
<!ELEMENT type (#PCDATA)>
<!ATTLIST type
    resource-type CDATA #FIXED "work">
<!-- D-Lib Magazine assigns metadata to works -->
<!-- The default type is an "article" -->
```

Continua nella prossima slide

Il DTD del D-Lib Magazine [4/6]

```
<!ELEMENT identifier (#PCDATA)>
```

```
<!ATTLIST identifier  
  uri-type (DOI | URL) #REQUIRED>
```

```
<!-- Every work should have a single DOI and one or more URLs. -->
```

Continua nella prossima slide

Il DTD del D-Lib Magazine [5/6]

```
<!ELEMENT relation (serial-name, (issn, volume, issue)*)>
```

```
<!ATTLIST relation
```

```
  rel-type CDATA #FIXED "InSerial">
```

```
  <!ELEMENT serial-name (#PCDATA)>
```

```
  <!ELEMENT issn (#PCDATA)>
```

```
  <!ELEMENT volume (#PCDATA)>
```

```
  <!ELEMENT issue (#PCDATA)>
```

```
<!-- The serial name is "D-Lib Magazine". -->
```

```
<!-- The ISSN is "1082-9873". -->
```

```
<!-- Volume corresponds to year of publication, 1995 is "1". -->
```

```
<!-- The issue is a count of the actual issues in the volume. -->
```

Continua nella prossima slide

Il DTD del D-Lib Magazine [6/6]

<!ELEMENT language (#PCDATA)>

<!-- The name of the language in English as: "English", "French", "Japanese" -->

<!ELEMENT rights (#PCDATA)>

<!-- The copyright statement as given on the work. -->

XML Schema

- ◆ **Un XML schema descrive la struttura di un documento XML.**
 - Definisce gli elementi che possono apparire nel documento
 - Definisce gli attributi che possono apparire nel documento
 - Definisce la struttura degli elementi
 - ➔ **Relazioni padre – figlio**
 - ➔ **Numero dei figli**
 - ➔ **Ordine dei figli**
 - Definisce se un elemento è vuoto oppure può includere del testo
 - Definisce il tipo dei dati degli elementi e degli attributi
 - Definisce i valori di default e i valori prefissati per gli elementi e gli attributi

Visualizzazione di documenti XML

◆ Si utilizzano gli stylesheet

- XML non usa tag predefiniti per cui il significato di ogni tag non è noto: un browser non sa come presentare un documento XML
- Il documento XML non include informazione di formattazione
- È necessario fornire informazione aggiuntiva che specifica come il documento deve essere visualizzato

◆ Il Cascading Style Sheets (CSS) è un semplice meccanismo per aggiungere gli stili (per es. Font, colori, spaziatura) ai documenti web.

◆ XSL - Extensible Stylesheet Language viene utilizzato per specificare gli stylesheet.

Riferimenti

- ◆ W3C <http://www.w3.org/XML/>
- ◆ Erik T. Ray, Learning XML, O'Reilly, 2001
- ◆ XML Tutorial
 - <http://www.w3schools.com/xml/default.asp>
 - <http://www.zvon.org/xxl/XMLTutorial/General/contents.html>

Il modello RDF

Introduzione al modello RDF [1/6]

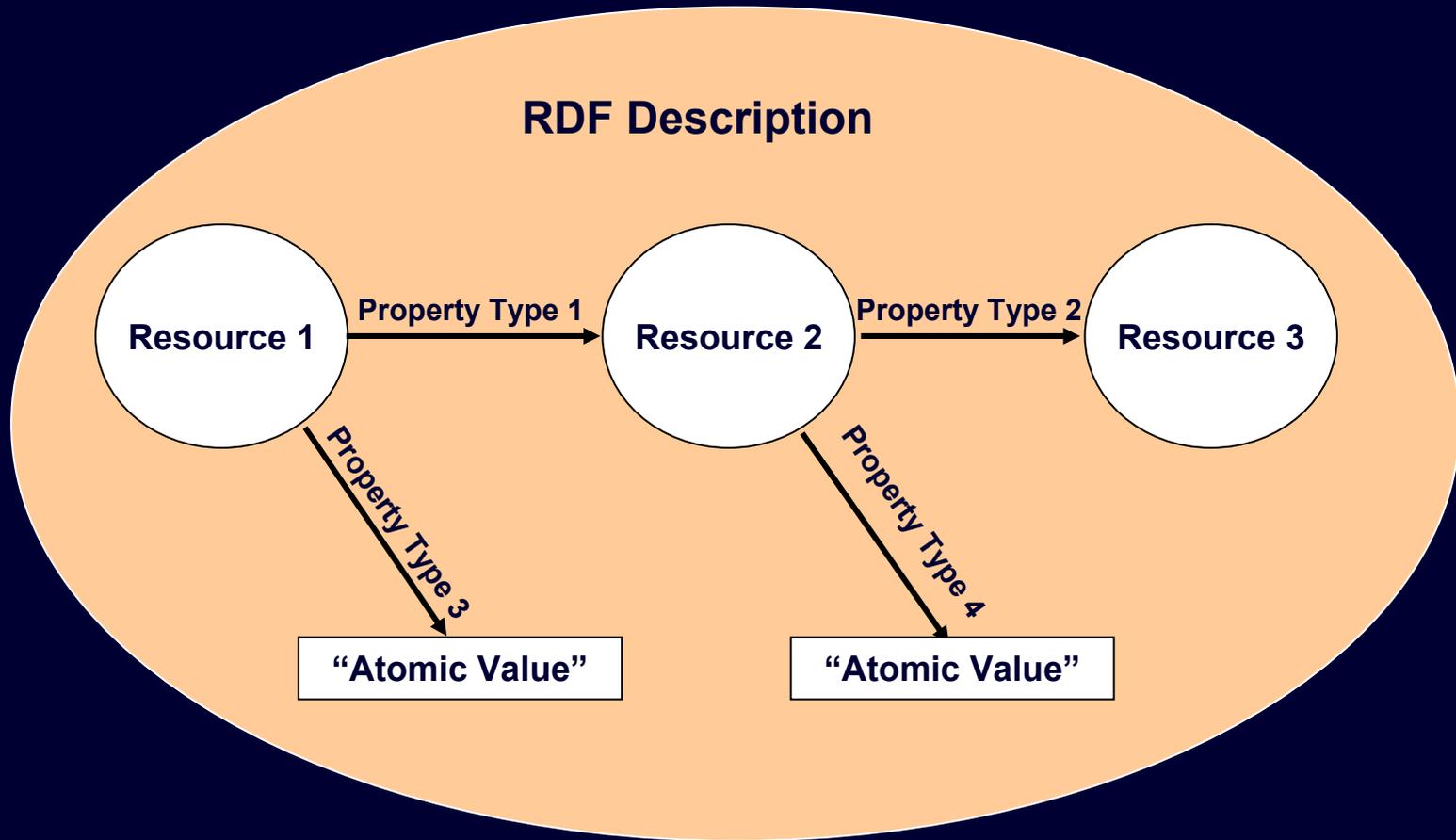
- ◆ **RDF (Resource Description Framework) è un modello di riferimento per descrivere metadati a supporto del riuso e dello scambio tra applicazioni diverse**
- ◆ **Resource Description Framework Model & Syntax - Recommendation of W3C, 1999**
- ◆ **Fornisce un' "architettura" generica per descrivere i metadati**
 - Insieme di convenzioni per applicazioni che scambiano metadati
 - Fornisce un mezzo per pubblicare vocabolari di termini specifici per una comunità di utenti; in tal modo permette che la semantica dei documenti sia definita da diverse comunità
 - ➔ **Ad es. Dublin Core utilizza RDF (RDFDC) e Educom IMS (Instructional Metadata System) è basato su DC esteso con una serie di termini *domain specific*.**

Introduzione al modello RDF [2/6]

◆ Definisce

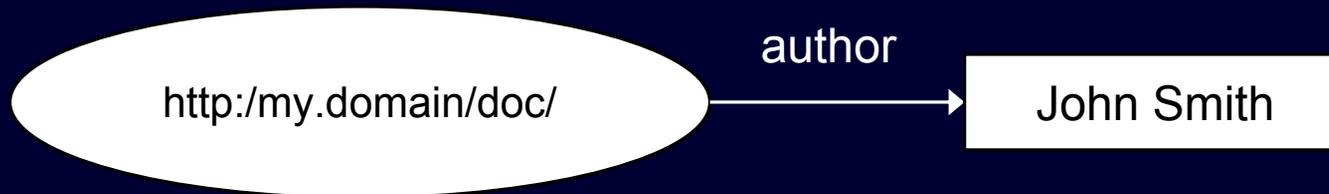
- Un modello per descrivere delle risorse (*resources*)
 - ➔ **Resource : un qualunque oggetto identificato da un identificatore unico (URI – Uniform Resource Identifier)**
- Ogni risorsa ha un insieme di proprietà (*property*)
 - ➔ **Anche le proprietà sono univocamente identificate da un URI**
- Una collezione di proprietà che si riferisce alla stessa risorsa è detta *description*
- Convenzioni per codificare le descrizioni usando una sintassi XML

Introduzione al modello RDF [3/6]



Introduzione al modello RDF [4/6]

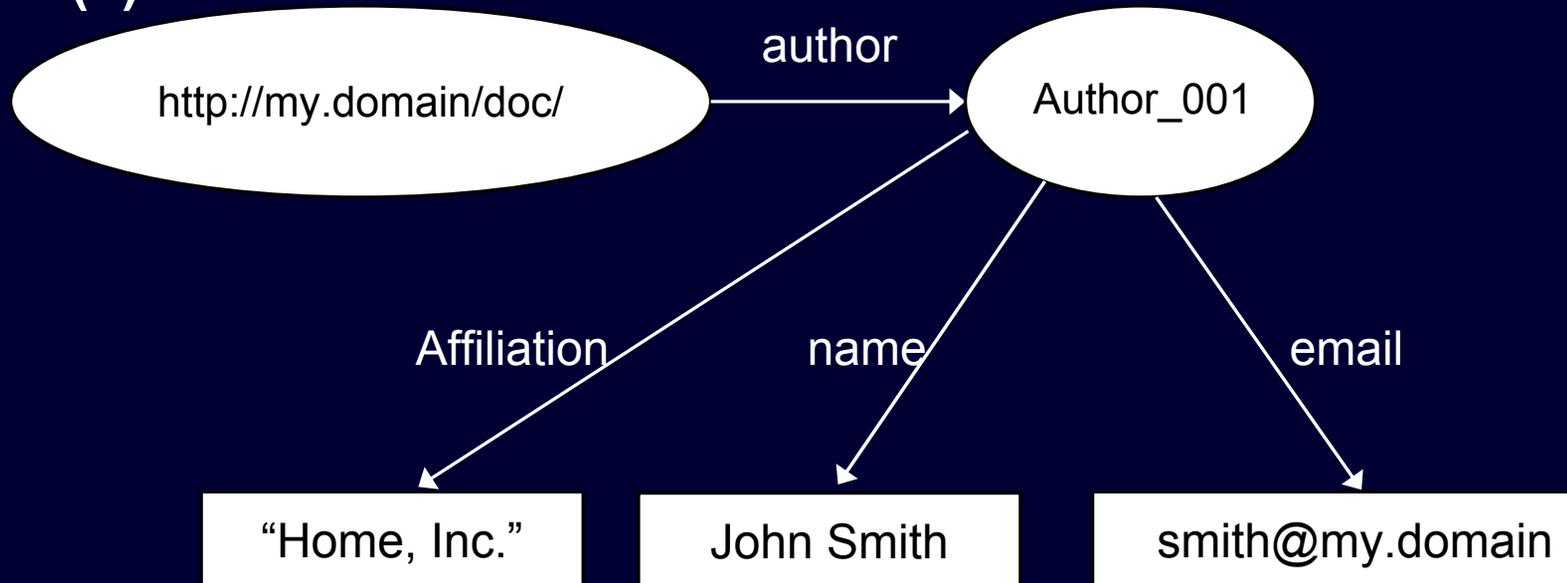
- ◆ Una *risorsa* ha una *proprietà* il cui *valore* è o (i) un valore stringa (letterale)....



- La *risorsa* identificata dall'URI `http://my.domain/doc/` ha una *proprietà* di tipo autore il cui *valore* è "John Smith"
- Oppure si può dire che "John Smith" è l'autore della risorsa identificata da `http://my.domain/doc/`

Introduzione al modello RDF [5/6]

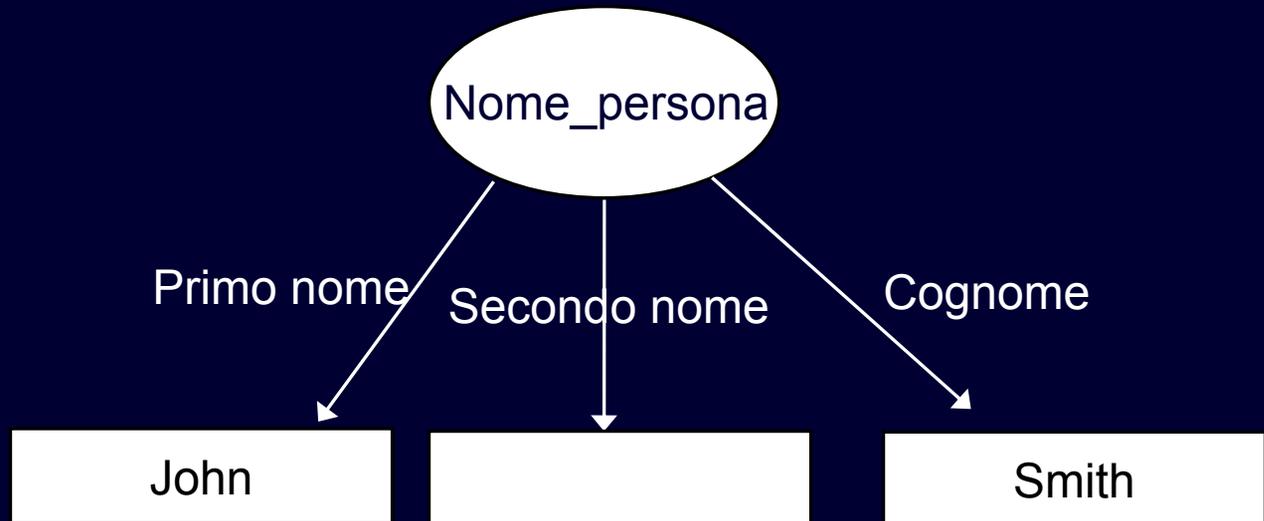
◆ ... o (ii) un'altra *risorsa*



➔ Il *valore* della *proprietà* “author” è un'altra *risorsa* (con URI Author_001) che ha una *proprietà* “affiliation” con *valore* “Home, Inc.”, una *proprietà* “name” con *valore* “John Smith” ed una *proprietà* “email” con *valore* “john@my.domain”

Introduzione al modello RDF [6/6]

- ◆ L'uso di identificatori unici per le risorse permette il loro *riuso* in modo non ambiguo, per cui `persona1` può essere l'autore del documento ma può anche essere l'impiegato di una ditta
- ◆ Avremmo potuto creare altri livelli, per es.



La sintassi di RDF XML [1/4]

◆ Rappresentazione in XML del modello

- Memorizzazione/scambio delle descrizioni

◆ I nomi delle proprietà sono resi unici attraverso l'uso dei namespace XML.

- Semantica unica associata ad ogni property-type
- Ad es. "author" viene specificato in Dublin Core come elemento CREATOR



- Identifica la risorsa con URI `http://my.domain/doc/` con la semantica del property-type CREATOR definita nel contesto di DC (vocabolario Dublin Core)

La sintassi di RDF XML [2/4]

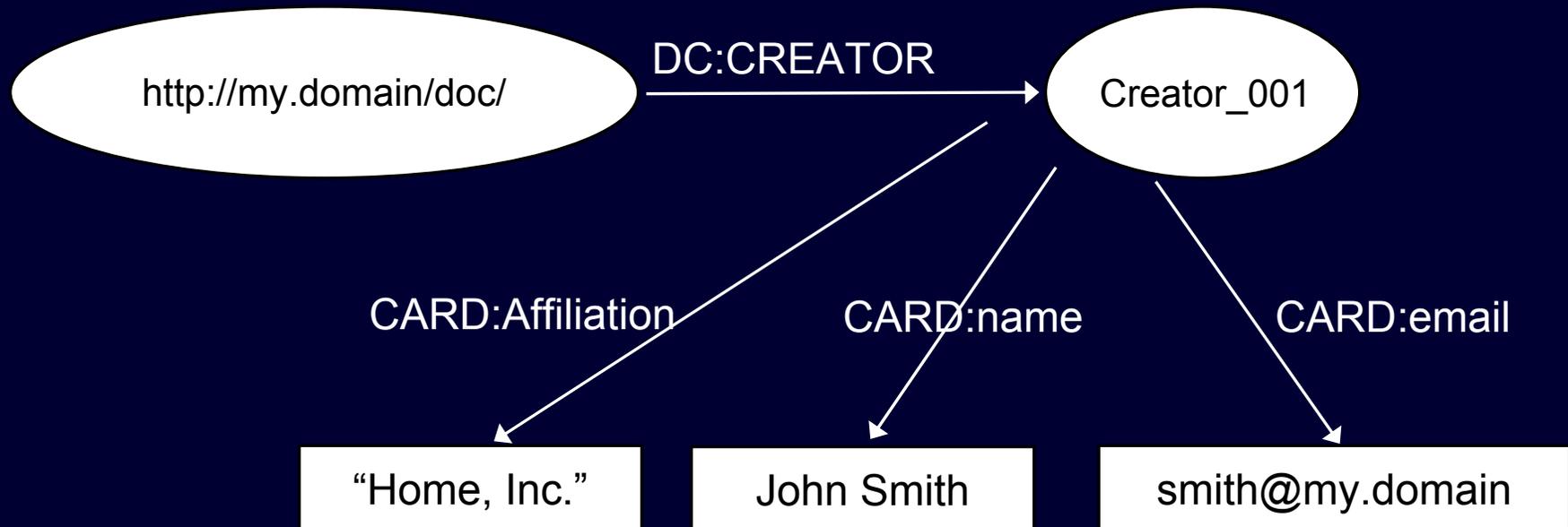
◆ Rappresentazione XML

```
<?xml:namespace ns = "http://www.w3.org/RDF/RDF/" prefix = "RDF" ?>
<?xml:namespace ns = "http://purl.oclc.org/DC/" prefix = "DC" ?>
<RDF:RDF>
  <RDF:Description RDF:HREF = "http://my.domain/doc/">
    <DC:Creator>John Smith</DC:Creator> </RDF:Description>
</RDF:RDF>
```

- Vengono specificati i namespace RDF e Dublin Core (DC) ed utilizzati con i prefissi RDF e DC
- L'elemento `<RDF:Description>` viene usato per denotare una risorsa che URI `"http://my.domain/doc/"`
- L'elemento `<DC:Creator>` nel contesto dell' `<RDF:Description>` rappresenta un property-type CREATOR di Dublin Core con valore `"John Smith"`

La sintassi di RDF XML [3/4]

- ◆ **Nell'esempio più complesso abbiamo proprietà non specificate in Dublin Core**
 - In questo caso si può usare un'ulteriore standard di descrizione delle risorse che le contenga



La sintassi di RDF XML [4/4]

◆ Nell'esempio più complesso abbiamo proprietà non specificate in Dublin Core

- In questo caso si può usare un'ulteriore standard di descrizione delle risorse che le contenga

```
<?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://my.domain/doc/">
    <DC:Creator RDF:HREF = "#Creator_001"/>
  </RDF:Description>
  <RDF:Description ID="Creator_001">
    <CARD:Name>John Smith</CARD:Name>
    <CARD:Email>smith@my.domain</CARD:Email>
    <CARD:Affiliation>Home, Inc.</CARD:Affiliation>
  </RDF:Description>
</RDF:RDF>
```

La potenza di RDF

- ◆ **Modello estendibile**
- ◆ **Supporta descrizioni di complessità arbitraria**
- ◆ **Gli URI sono gli elementi unici per identificare**
 - risorse
 - proprietà
- ◆ **Descrizioni create indipendentemente possono essere “fuse” usando gli URI come “ancore”**

RDF Schema [1/2]

- ◆ **Resource Description Framework Schema**
- ◆ **Candidate Recommendation of W3C, 2000**
- ◆ **Fornisce dei meccanismi per definire vocabolari usati negli statement RDF**
 - e.g. Dublin Core metadata element set definito usando RDF

RDF Schema [2/2]

◆ **Gli schemi RDF sono usati per**

- Dichiarare i vocabolari
- Dichiarare l'insieme dei property-types di una determinata comunità di utenti

◆ **Forniscono meccanismi per**

- Raggruppare le risorse in classi
- Organizzare le classi gerarchicamente (subClassOf)
- Organizzare le proprietà in modo gerarchico (subPropertyOf)
- Imporre dei vincoli alle proprietà (domain, range)

RDF, XML ed interoperabilità [1/2]

◆ Cosa manca ad XML

- XML consente varianti diverse per esprimere gli stessi concetti
- Mentre gli esseri umani riescono ad interpretare queste variazioni, i programmi applicativi hanno bisogno di una conoscenza a priori dello schema/DTD
- RDF impone degli ulteriori vincoli sintattici su come i concetti devono essere espressi
- con RDF/XML, sia gli esseri umani che i programmi possono interpretare le descrizioni in modo consistente

◆ Minore flessibilità, maggiore interoperabilità

RDF, XML ed interoperabilità [2/2]

- ◆ **È possibile usare solo XML per lo scambio quando**
 - Entrambe le applicazioni “conoscono” la semantica dei metadati
- ◆ **È necessario usare RDF/XML per lo scambio quando**
 - I (meta)dati possono essere utilizzati dalle applicazioni senza che vi sia una conoscenza dello schema specifico
 - I (meta)dati incorporano strutture che derivano da domini diversi

Riferimenti

- ◆ Eric Miller, An introduction to the Resource Description Framework, D-Lib Magazine, May 1998, <http://www.dlib.org/dlib/may98/miller/05miller.html>
- ◆ Frank Manola, Eric Miller, RDF Primer, W3C Recommendation 10 February 2004, <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>

Modelli di metadati

Modelli di metadati

- ◆ **MARC – MACHine Readable Cataloguing record**
- ◆ **Dublin Core**
- ◆ **MPEG7**
- ◆ **ECHO**

MARC

MARC – Introduzione

- ◆ **Sviluppato alla fine degli anni 60 alla Library of Congress (USA) per promuovere la condivisione di cataloghi tra biblioteche**
- ◆ **È diventato un formato quasi standard**
- ◆ **Utilizza un sistema di numeri, lettere e simboli all'interno del record per individuare i diversi tipi di informazione**
- ◆ **Fornisce**
 - Descrizione dell'oggetto (titolo, edizione, pubblicazione, ecc.)
 - Soggetto
 - Numero di catalogo
- ◆ **Molti formati simili a MARC sono stati sviluppati da organismi bibliotecari nazionali (USMARC, UKMARC, CANMARC, AUSMARC, DANMARC, ANNAMARC, INTERMARC, ...)**
- ◆ **UNIMARC (Universal MARC) come formato standard e come formato di scambio tra i diversi formati**
 - Ad es. USMARC → UNIMARC → INTERMARC
 - Problemi di degrado dell'informazione nei vari passaggi

Record della Library of Congress

LC Control Number:68093427

Type of Material:Book (Print, Microform, Electronic, etc.)

Personal Name:[Manzoni, Alessandro, 1785-1873.](#)

Main Title:I promessi sposi.

Edition Information:[2. ed.]

Published/Created:Firenze, La nuova Italia, 1967.

Related Names:[Russo, Luigi, 1892-1961. ed.](#)

Description:xxii, 716 p. 21 cm.

LC Classification:PQ4713 .P3 1967

Other System No.:(OCoLC)23497169

CALL NUMBER:

[PQ4713 .P3 1967](#)

Copy 1-

Request in:Jefferson or Adams Bldg General or Area Studies Reading Rms-

Status:Not Charged

Record della Library of Congress MARC

LC Control Number:68093427
000 00678nam 22002291 450
001 8495557
005 19980421195046.0
008 890330s1967 it 000 0 ita
035 __ |9 (DLC) 68093427
906__ |a 7 |b cbc |c oclcrpl |d u |e ncip |f 19 |g y-gencatlg
010 __ |a 68093427
035 __ |a (OCoLC)23497169
040 __ |a DLC |c MiU |d DLC
050 00 |a PQ4713 |b .P3 1967
100 1_ |a Manzoni, Alessandro, |d 1785-1873.
245 12 |a I promessi sposi.
250 __ |a [2. ed.]
260 __ |a Firenze, |b La nuova Italia, |c 1967.
300 __ |a xxii, 716 p. |c 21 cm.
700 1_ |a Russo, Luigi, |d 1892-1961. |e ed.
985 __ |e OCLC REPLACEMENT
991 __ |b c-GenColl |h PQ4713 |i .P3 1967 |t Copy 1 |w OCLCREP

Struttura di MARC

◆ Ogni record bibliografico è suddiviso in campi (fields)

- Campo autore, campo titolo, ecc.
- Ogni campo può essere suddiviso in sottocampi
- I campi sono rappresentati da un codice di 3 cifre (tag)

◆ Esempi di tag

010 tag **marks the** Library of Congress Control Number (**LCCN**)

020 tag **marks the** International Standard Book Number (**ISBN**)

100 tag **marks a** personal name main entry (**author**)

245 tag **marks the** title information

(which includes the title, other title information, and the statement of responsibility)

250 tag **marks the** edition

260 tag **marks the** publication information

300 tag **marks the** physical description

(often referred to as the "collation" when describing books)

440 tag **marks the** series statement/added entry

520 tag **marks the** annotation or summary note

650 tag **marks a** topical subject heading

700 tag **marks a** personal name added entry (**joint author, editor, or illustrator**)

Struttura di MARC

◆ Indicatori

- Ogni campo (con alcune eccezioni) può essere seguito da due caratteri usati come indicatori.
- Gli indicatori specificano come deve essere interpretato il valore associato al campo
- Gli indicatori possono assumere valori da 0 a 9 (# se l'indicatore è vuoto)
- Ad esempio
245 14 \$a The emperor's new clothes /
 \$c adapted from Hans Christian Andersen
 and illustrated by Janet Stevens.
→ **245 è il campo per Titolo**
→ **Il primo indicatore con valore 1 specifica che nel catalogo esiste una diversa scheda per il titolo**
→ **Il secondo indicatore specifica il numero di caratteri all'inizio del campo che devono essere trascurati (nell'esempio sono 4)**

Struttura di MARC

◆ Sottocampi

- Diversi campi sono costituiti da diverse parti di informazione. Ogni parte è specificata da un sottocampo, con uno specifico codice
- Ogni sottocampo è preceduto da un delimitatore
- Ad esempio

300 ## \$a 675 p. : \$b ill. ; \$c 24 cm.

300 è il campo “descrizione fisica di un libro”

Il delimitatore del sottocampo è \$

Vi sono 3 sottocampi

Numero di pagine (\$a)

Informazioni sulle illustrazioni (\$b)

Dimensioni (\$c)

Regole generali

- ◆ **La divisione dei tag di MARC21 è la seguente**
 - **0XX** Informazione di controllo, numeri, codici
 - **1XX** Parte principale
 - **2XX** Titoli, edizione, statement di responsabilità, informazione sulla pubblicazione
 - **3XX** Descrizione fisica, etc.
 - **4XX** Serie
 - **5XX** Note
 - **6XX** Soggetto
 - **7XX** Altre entries diverse dal soggetto o dalla serie
 - **8XX** Altro
 - **9XX** Lasciato per uso locale

Punti d'accesso

- ◆ I punti d'accesso costituiscono le parti più importanti del record
- ◆ Vengono utilizzati per effettuare le ricerche dei record
 - 1XX fields (main entries)
 - 4XX fields (series statements)
 - 6XX fields (subject headings)
 - 7XX fields (added entries other than subject or series)
 - 8XX fields (series added entries)
- ◆ Questi campi devono rispettare dei formati prestabiliti (Authority control)

Authority control

- ◆ **Permette la scelta di forme codificate ed uguali per tutti. Un catalogatore sceglie i soggetti ed i nomi da una lista utilizzata da tutti**
 - Ad esempio, se in una intestazione parliamo alle volte di “Getty Museum” ed altre di “J. Paul Getty Museum“, avremo problemi in fase di ricerca
 - Per i nomi si utilizzano il Library of Congress Name Authority file
 - Per gli argomenti ed i nomi geografici si usano il *Library of Congress Subject Headings* o il *Sears List of Subject Headings*.

Parallel tag construction

- ◆ Ogni tipo di tag (da 0XX a 9XX) può specificare valori che appartengono a diverse tipologie di dati (Nomi di persona, Nomi di aziende, Nomi geografici, ecc.)
- ◆ I campi che richiedono un “authority control” avranno tag specifici per valori di tipologie specifiche
 - X00 Nomi di persona
 - X10 Nomi di aziende
 - X11 Nomi di meeting
 - X30 Titoli
 - X40 Titoli bibliografici
 - X50 Termini relativi ad argomenti
 - X51 Nomi geografici
- ◆ Ad es., se il soggetto di un libro (6XX) è una persona, il tag sarà 600, se è un’azienda il tag sarà 610

Intestazione

- ◆ **L'intestazione contiene informazione che normalmente viene utilizzata dal sistema**
 - **Leader**
 - ➔ **24 caratteri utilizzati dal sistema**
 - ➔ **Contiene informazioni quali la lunghezza del record, il tipo, ecc.**
 - **Directory**
 - ➔ **Specifica quali tag ci sono nel record e dove si trovano**
 - **Il campo 008**
 - ➔ **Contiene informazione importante ma in forma sintetica**
 - ➔ **Ad es. può contenere un codice per il paese di pubblicazione, un codice per indicare la lingua del testo, ecc.**

Riferimenti

- ◆ **Understanding MARC**, Betty Furrie in conjunction with the Data Base Development Department of The Follett Software Company, Published by the Cataloging Distribution Service, Library of Congress, in collaboration with The Follett Software Company, <http://www.loc.gov/marc/umb/>
- ◆ *Anglo-American Cataloguing Rules*. 2nd ed., 2002 revision. Chicago: American Library Association, 2002.
- ◆ Dewey, Melvil. *Dewey Decimal Classification and Relative Index*. 22nd. 4 vols. Albany, N.Y.: Forest Press, 2003.

Dublin Core

Dublin Core: background

- ◆ Necessità di dare una rappresentazione formalizzata alle risorse di informazione accessibili in rete via WEB
- ◆ Varietà delle risorse: testi, suoni, immagini tridimensionali, video, fotografie, film ...
- ◆ Necessità di una rappresentazione capace di comprendere i vari tipi di risorse

Dublin Core: background

- ◆ Esigenze accolte da un gruppo di bibliotecari, e di informatici....
- ◆ “Finding relevant information on the World Wide Web has become increasingly problematic due to the explosive growth of networked resources. Current Web indexing evolved rapidly to fill the demand for resource discovery tools, but that indexing, while useful, is a poor substitute for richer varieties of resource description.”
- ◆ Dublin Core Metadata Initiative <http://www.ietf.org/rfc/rfc2413.txt>
- ◆ **Dublin Core Metadata Set. Un insieme di elementi descrittivi capaci di rappresentare qualsiasi risorsa di informazione accessibile in rete**

Dublin Core: background

- ◆ **Il *D.C. metadata set* e' il formato di metadati piu' diffuso oggi, risultato di un progetto iniziato nel 1995 [Dublin Core Metadata Workshop Series] con l'obiettivo di affrontare il problema della ricerca di risorse di informazione disponibili sulla rete**
 - documenti testuali, programmi, video....ma anche basi di dati, archivi, etc..
- ◆ **La sua caratteristica - frutto di una scelta ben precisa - è la generalità e la semplicità. Esso cioè è stato pensato come un insieme minimo di elementi descrittivi capaci di rappresentare qualsiasi risorsa di informazione -pagina web, un audio, un video etc. - che abbia un indirizzo WEB**

Caratteristiche di Dublin Core

- ◆ *Si applicano a tutti i tipi di risorse di informazione*
- ◆ *Sono facili da creare, anche da parte di personale non specializzato (a differenza di quanto succede per i record catalografici di tipo MARC)*
- ◆ *Sono di uso generale per la “scoperta” delle risorse di informazione (a differenza delle descrizioni catalografiche che in genere seguono normative specifiche del settore di applicazione: biblioteche, musei, videoteche...)*
- ◆ *Il modello Dublin Core è “piatto”, cioè non permette di descrivere metadati strutturati, ed ha un numero di elementi limitato (15)*

Caratteristiche di Dublin Core

- ◆ **Vi sono diverse ragioni per adottare Dublin Core:**
 - I metadati Dublin Core sono facili da creare
 - Possono essere facilmente indicizzati
 - Permettono una precisione maggiore rispetto ai sistemi che utilizzano full text
 - Permettono l'interoperabilità
- ◆ **Dublin Core fornisce gli strumenti per**
 - Descrivere una risorsa (location, form etc.)
 - Fornire semplici descrizioni e permetterne l'indicizzazione
 - Individuare una risorsa
 - Ottenere ed accedere una risorsa
 - Migliorare la qualità della gestione delle risorse

Dublin Core metadata set [1/18]

- ◆ L'insieme degli elementi definiti nel Dublin Core Metadata Element Set, Version 1.1: Reference Description è presentato nella traduzione italiana, reperibile all'indirizzo <http://www.iccu.sbn.it/metadati.htm>
- ◆ Esiste un mapping tra UNIMARC e Dublin Core
 - ◆ Title
 - ◆ Creator
 - ◆ Subject
 - ◆ Description
 - ◆ Publisher
 - ◆ Contributor
 - ◆ Date
 - ◆ Type
 - ◆ Format
 - ◆ Identifier
 - ◆ Source
 - ◆ Language
 - ◆ Relation
 - ◆ Coverage
 - ◆ Rights

Dublin Core metadata set [2/18]

◆ Elemento: Titolo

Nome: Titolo

Identificatore: Title

Definizione: Un nome dato alla risorsa. Tipicamente, Title è il nome con il quale la risorsa è conosciuta

*Esempi: Title = "La Divina Commedia"
Title = "La Nona Sinfonia"
Title = "Pietà"*

Dublin Core metadata set [3/18]

◆ Elemento: **Creatore**

Nome: *Creatore*

Identificatore: *Creator*

Definizione: *Un'entità che ha la responsabilità principale della produzione del contenuto della risorsa. Esempi di creator: una persona, un'organizzazione, un servizio.*

Regole generali: *I vari creator vanno riportati separatamente.*

Il cognome deve precedere il nome.

Nel caso di organizzazioni strutturate, elencare le varie componenti, a partire da quella principale.

Esempi: *Creator = "Alighieri, Dante"*

Creator = "Internal Revenue Service. Customer Complaints Unit"

Dublin Core metadata set [4/18]

◆ Elemento: **Soggetto**

Nome: Soggetto e Parole chiave

Identificatore: Subject

Definizione: L'argomento della risorsa. Tipicamente un Subject viene espresso da parole chiave o frasi o codici di classificazione che descrivono l'argomento della risorsa. È buona norma selezionare il valore da un vocabolario controllato o da uno schema di classificazione.

*Esempio: Subject = "Poesia"
Subject = "Sinfonia"*

Dublin Core Metadata Set [5/18]

◆ Elemento:	Descrizione
<i>Nome:</i>	<i>Descrizione</i>
<i>Identificatore:</i>	<i>Description</i>
<i>Definizione:</i>	<i>Una spiegazione del contenuto della risorsa</i>
<i>Commento:</i>	<i>Una descrizione può includere ma non è limitata solo a: un riassunto analitico, un indice, un riferimento al contenuto di una rappresentazione grafica o un testo libero del contenuto.</i>
<i>Regole generali:</i>	<i>Molto utile in fase di ricerca. Normalmente vengono utilizzate frasi complete.</i>

Dublin Core Metadata Set [6/18]

- ◆ **Elemento:** Editore
 - Nome:* Editore
 - Identificatore:* Publisher
 - Definizione:* Un'entità responsabile della produzione della risorsa, disponibile nella sua forma presente
 - Commento:* Esempi di un Publisher includono una persona, un'organizzazione o un servizio. Normalmente il nome di un Publisher è usato per indicare l'entità stessa
Questo campo viene specificato per identificare l'entità che fornisce accesso alla risorsa.
 - Esempi:* Publisher="Mondadori"

Dublin Core Metadata Set [7/18]

◆ Elemento: Autore di contributo subordinato

Nome: Autore di contributo subordinato

Identificatore: Contributor

Definizione: Un'entità responsabile della produzione di un contributo al contenuto della risorsa

Commento: Esempi di un Contributor includono una persona, un'organizzazione o un servizio. Normalmente il nome di un Contributor è usato per indicare l'entità stessa

Dublin Core Metadata Set [8/18]

◆ Elemento: Data

Nome: Data

Identificatore: Date

Definizione: Una data associata a un evento del ciclo di vita della risorsa. Normalmente viene associata con la data di creazione o di disponibilità della risorsa. Si raccomanda l'uso di codifiche standard per le date. Se non si conosce la data completa, si può usare solo il mese-anno o solo l'anno.

Dublin Core Metadata Set [9/18]

◆ Elemento: Tipo

Nome: Tipo di risorsa

Identificatore: Type

Definizione: La natura o il genere del contenuto della risorsa

Commento: Tipo include termini che descrivono categorie generali, funzioni, generi o livelli di aggregazione per contenuto. Si raccomanda di selezionare un valore da un vocabolario controllato (ad esempio, la lista dei “Dublin Core Types”). Per descrivere la manifestazione fisica o digitale della risorsa, si usa l’elemento Format.

Se una risorsa ha più tipi, bisogna utilizzare più campi Type.

Esempi: Type=“Text”

Type=“Sound”

Type=“Image”

Dublin Core Metadata Set [10/18]

- ◆ <http://dublincore.org/usage/terms/dcmitype/>
- ◆ **Dublin Core Types** - Dizionario dei termini usabili quali “valore” dell’elemento “TYPE” per identificare il genere di una risorsa:
 - Collection
 - Dataset
 - Event
 - Image
 - Interactive Resource
 - Service
 - Software
 - Sound
 - Text

Dublin Core Metadata Set [11/18]

◆ Elemento: Formato

Nome: Formato

Identificatore: Format

Definizione: La manifestazione fisica o digitale della risorsa.

Commento: Normalmente Format può includere il tipo di supporto o le dimensioni della risorsa. Format può essere usato per determinare il software, l'hardware o altro apparato necessario alla visualizzazione o all'elaborazione della risorsa. Esempi di dimensioni includono grandezza e durata. Si raccomanda di selezionare un valore da un vocabolario controllato (ad esempio la lista di Internet Media Types [MIME] che definisce i formati dei supporti elettronici. Se vi sono più formati per una stessa risorsa, il campo Format deve essere ripetuto.

Dublin Core Metadata Set [12/18]

INTERNET MEDIA TYPES

<http://www.isi.edu/in-notes/iana/assignments/media-types/media-types>

Testo

- *Plain*
- *Richtext*
- *Sgml*
-

Immagine

- *Jpeg*
- *Gif*
- *Tiff*
-

Video

- *Mpeg*
-

Dublin Core Metadata Set [13/18]

◆ Elemento: Identificatore

Nome: Identificatore della risorsa

Identificatore: Identifier

Definizione: Un riferimento univoco alla risorsa nell'ambito di un dato contesto

Commento: Si raccomanda di identificare la risorsa utilizzando un sistema formale di identificazione (ad.es. URL, URI, ISBN). Si può anche utilizzare un identificatore locale.

*Esempi: Identifier="http://www.intratext.com/X/ITA0191.htm"
Identifier="ISBN:0382364759"*

Dublin Core Metadata Set [14/18]

◆ Elemento: Fonte

Nome: Fonte

Identificatore: Source

Definizione: Un riferimento a una risorsa dalla quale è derivata la risorsa in oggetto

Commento: La risorsa può essere derivata dalla risorsa Source in tutto o in parte. Si raccomanda di identificare la risorsa per mezzo di una sequenza di caratteri alfabetici o numerici secondo un sistema di identificazione formalmente definito

Esempi: Source="RC607.A26W574 1996" (Call Number della versione a stampa)

Source="Immagine da pagina 32 dell'edizione pubblicata nel 1922 della Divina Commedia"

Dublin Core Metadata Set [15/18]

◆ Elemento: **Lingua**

Nome: Lingua

Identificatore: Language

Definizione: La lingua del contenuto intellettuale della risorsa. Si raccomanda l'uso di valori presi da una lista standard (RCF 3066)

Esempi: Language="it"

Language="Primarily English, with some abstracts also in French"

Dublin Core Metadata Set [16/18]

◆ Elemento: **Relazione**

Nome: Relazione

Identificatore: Relation

Definizione: Un riferimento alla risorsa correlata. Si raccomanda l'uso di un sistema di identificazione. Può indicare relazioni di tipo Parte/Tutto, Versioni, Formato derivato, citazioni, ecc.

*Esempio: Title="La Divina Commedia. Inferno"
Relation="Parte 1/3"*

Dublin Core Metadata Set [17/18]

- ◆ **Elemento:** **Copertura**
 - Nome:* *Copertura*
 - Identificatore:* *Coverage*
 - Definizione:* *L'estensione o scopo del contenuto della risorsa*
 - Commento:* *Normalmente Coverage include la localizzazione spaziale (il nome di un luogo o le coordinate geografiche), il periodo temporale (l'indicazione di un periodo, una data o un range di date) o una giurisdizione (ad esempio il nome di un'entità amministrativa). Si raccomanda di selezionare un valore da un vocabolario controllato (ad esempio il Thesaurus of Geographic Names [TGN]) e, se possibile, di utilizzare i nomi di luogo o i periodi di tempo piuttosto che identificatori numerici come serie di coordinate o range di date*

Dublin Core Metadata Set [18/18]

◆ Elemento: **Gestione dei diritti**

Nome: Gestione dei diritti

Identificatore: Rights

Definizione: Informazione sui diritti esercitati sulla risorsa

Commento: Normalmente un elemento Rights contiene un'indicazione di gestione dei diritti sulla risorsa, o un riferimento ad un servizio che fornisce questa informazione. L'informazione sui diritti spesso comprende i diritti di proprietà intellettuale Intellectual Property Rights (IPR), Copyright e vari diritti di proprietà. Se l'elemento Rights è assente, non si può fare alcuna ipotesi sullo stato di questi o altri diritti in riferimento alla risorsa

Esempio: Rights="Accesso limitato ai membri dell'associazione"

Uso di Dublin Core

- ◆ Le risorse web vengono descritte indicando il valore di uno o più degli elementi descrittivi (attributi) scelti fra l'insieme che costituisce il DC Metadata Set
- ◆ I metadati assumono perciò la forma di dichiarazioni relative agli attributi delle risorse, ad esempio:
 - ***L'attributo TITOLO ha il valore LA BIBLIOTECA E I SUOI DOCUMENTI***
 - ***L'attributo DATA ha il valore 1998***
 - ***L'attributo LINGUA ha il valore ITALIANA***
 - ***.....***

Qualifiers [1/2]

- ◆ **Gli elementi di Dublin Core possono essere poco specifici ed ambigui in molti casi, per cui è in corso un processo di estensione basato sull'uso di *qualifiers* da associare ad ogni Dublin Core element**
- ◆ **Ogni programma che utilizza una descrizione Dublin Core deve essere in grado di ignorare i qualifiers, oppure di interpretarli**
- ◆ **Vi sono attualmente due classi di qualifiers**
 - *Element refinement*
 - ➔ *Rendono il significato di un elemento più specifico*
 - *Encoding Scheme*
 - ➔ *Semplificano l'interpretazione dell'elemento.*
 - ➔ *Possono includere vocabolari controllati o regole di elaborazione (ad es. per le date)*

Qualifiers [1/2]

◆ Esempi

- **Subject** può avere diversi encoding schema, quali LCSH (Library of Congress Subject Headings), MeSH (Medical Subject Headings), ecc.
- **Date** può avere
 - ➔ **Refinements**
 - Ad es. **created** (data di creazione di una risorsa), **valid** (data di validità di una risorsa), ecc.
 - ➔ **Encoding schema**
- **Type** può avere un valore tra quelli specificati nella lista dei “DC types”
- **Format** può avere due diversi refinement
 - ➔ **Extent**: la dimensione o la durata di una risorsa
 - ➔ **Medium**: il materiale o l’elemento fisico che contiene la risorsa (ad es. carta, DVD, ecc.)

Rappresentazione di Dublin Core

- ◆ **Dublin Core può essere rappresentato utilizzando diversi formati**
- ◆ **Sono state sviluppate rappresentazioni sia di “simple DC” che di “qualified DC” in**
 - XML: <http://dublincore.org/documents/2003/04/02/dc-xml-guidelines/>
 - RDF: <http://dublincore.org/documents/2002/07/31/dcmes-xml/> e <http://dublincore.org/documents/dcq-rdf-xml/>

Riferimenti

- ◆ **Informazione aggiornata su Dublin Core**
<http://dublincore.org/index.shtml>
- ◆ **La documentazione aggiornata si trova a partire dall'indirizzo:** <http://dublincore.org/usage/>
- ◆ **Diane Hillman, Using Dublin Core,**
<http://dublincore.org/documents/usageguide/>

MPEG-7

MPEG-7

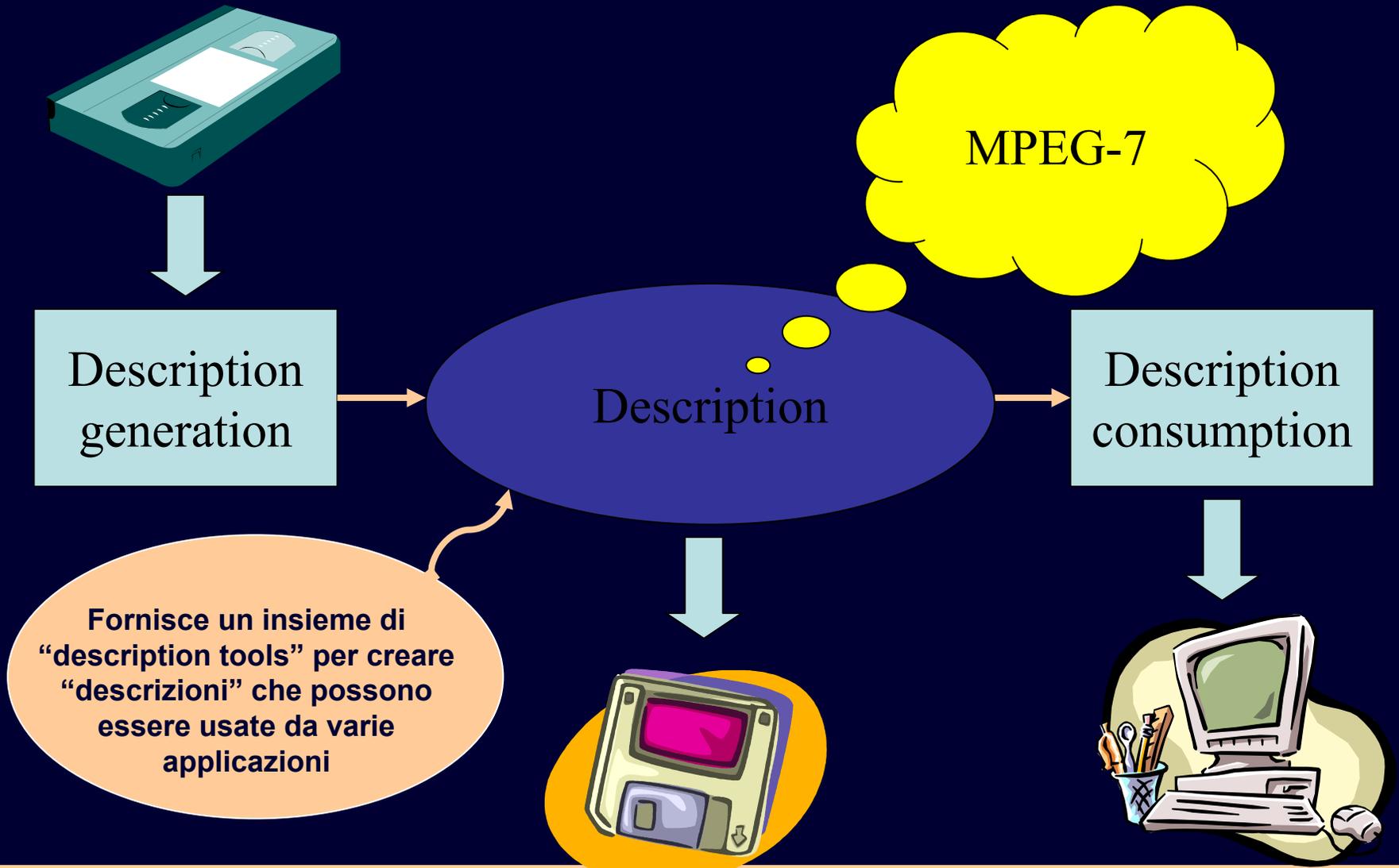
- ◆ **MPEG-7: standard sviluppato dall'MPEG (Moving Picture Expert Group)**
- ◆ **Viene denominato “Multimedia content description interface”**
- ◆ **Fornisce una descrizione del contenuto di dati multimediali, estraibile attraverso la sua analisi ed elaborazione**
- ◆ **Strumento generico non finalizzato ad una applicazione specifica, ma adatto a supportare un ampio numero di applicazioni**
- ◆ **È stato sviluppato da esperti in diversi settori applicativi**
 - Broadcasters, electronic manufacturers, content creators, publishers, right managers, telecommunication service providers, and academia

MPEG-7

◆ Application scenarios:

- Image understanding
- Intelligent vision
- Smart cameras/VCRs
- Information retrieval
- Information filtering
- Digital libraries
- Computer based training

MPEG-7



MPEG-7

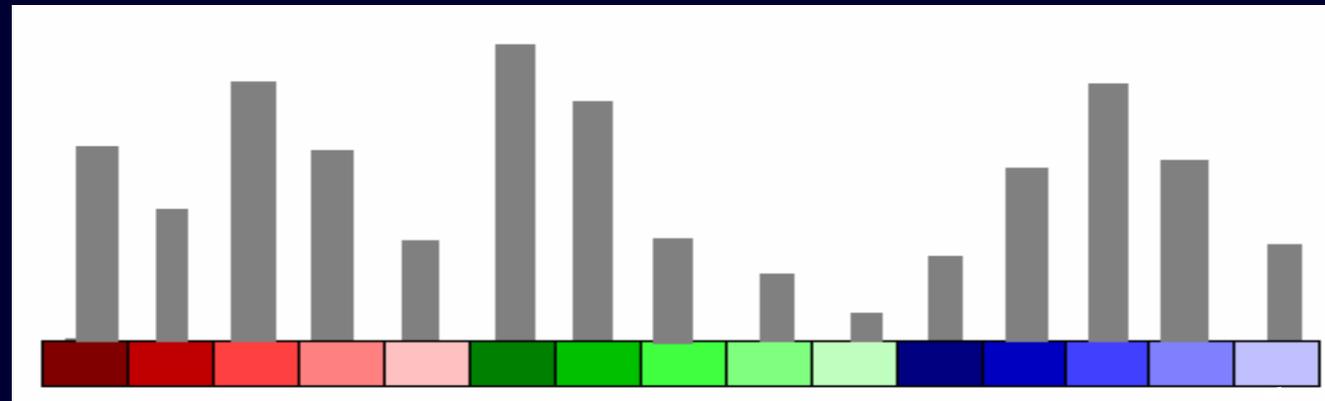
- ◆ **I dati multimediali possono essere descritti in molti modi diversi a seconda dell'uso che ne vogliamo fare**
 - Ad es. un video può essere descritto attraverso un indice delle scene, se vogliamo supportare il browsing da una scena all'altra
 - Oppure può essere descritto attraverso un insieme di keyframes (immagini significative estratte dal video) se vogliamo supportare il retrieval per contenuto sui frames.

- ◆ **MPEG-7 supporta diversi livelli di astrazione nella descrizione degli oggetti multimediali**
 - Da caratteristiche (features) di basso livello dei dati (ad es. la distribuzione dei colori, la forma degli oggetti, la tessitura in un key frame, la traiettoria di un oggetto, ecc.)
 - Ad informazioni semantiche di alto livello (ad es. “questa scena illustra l'inaugurazione di xx da parte di yy. Sullo sfondo si vedono ...”
...)

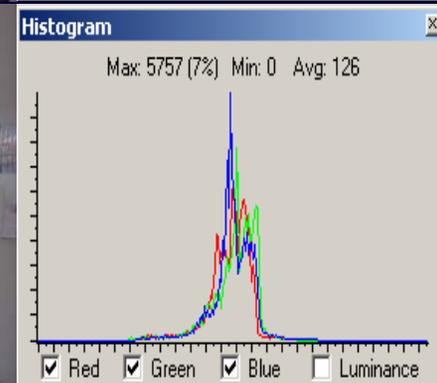
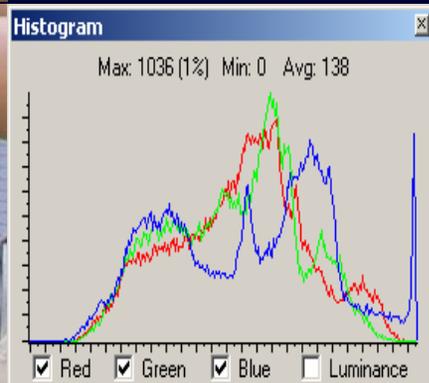
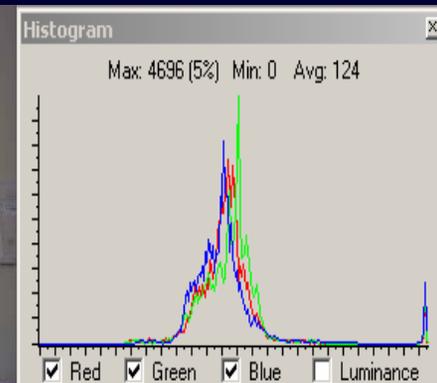
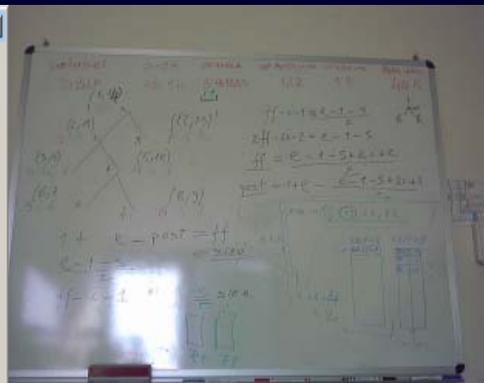
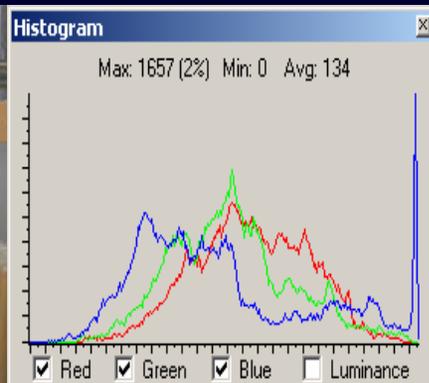
Image indexing

◆ Istogrammi di colori

- Lo spettro dei colori viene diviso in n parti
- Il valore di ogni “parte” è proporzionale al numero di pixel che hanno quel colore

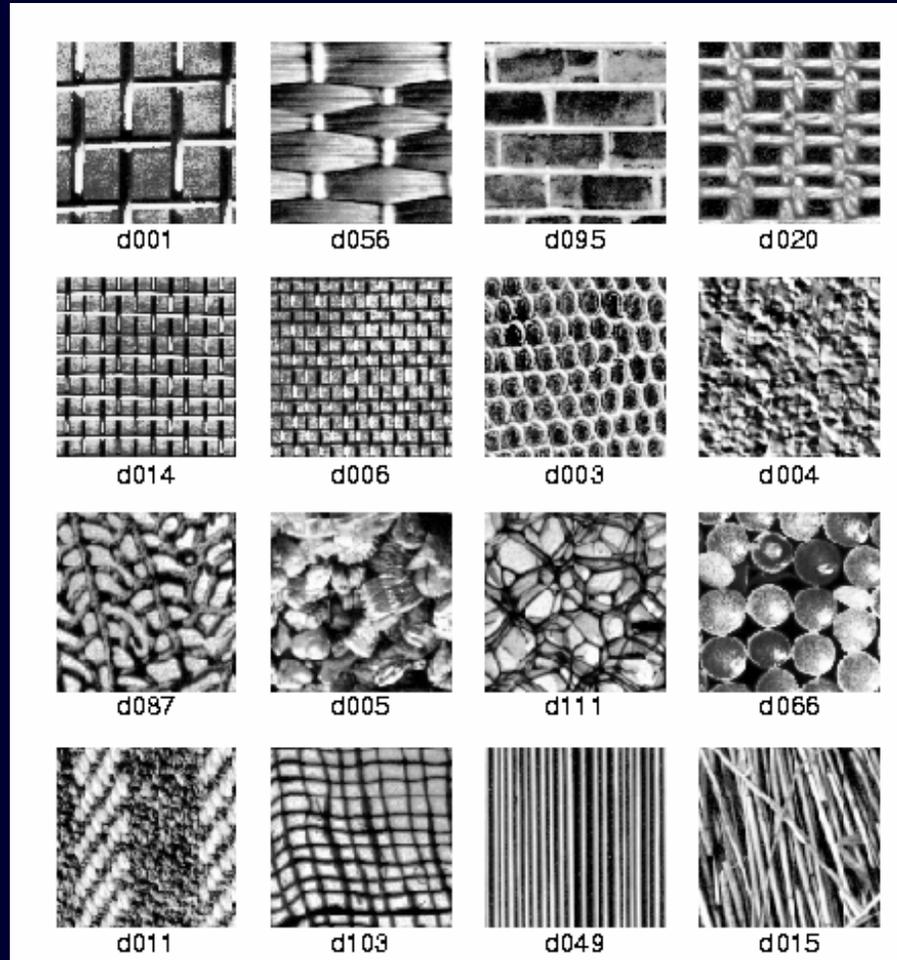


Indexing images



Indexing images

◆ Textures:



MPEG-7

- ◆ **Queste caratteristiche possono essere estratte automaticamente (tipicamente le features di basso livello) o richiedono l'intervento umano (tipicamente le feature di alto livello)**
- ◆ **MPEG-7 definisce solo il formato da usare per descrivere queste features e non le metodologie usate per estrarle.**
 - Per lo scambio, ricerca, ecc. è necessario conoscere solo il formato delle descrizioni, non come sono state ottenute
 - Questo permette di avere uno standard che si “adatta” alle evoluzioni tecnologiche (nuovi algoritmi di estrazione di feature possono essere utilizzati senza modificare lo standard)

Componenti di MPEG-7 [1/2]

◆ Descriptors (Ds)

- Rappresentazione di una feature. Definisce la sintassi, il tipo di dato, i valori permessi e la semantica della rappresentazione di una feature (ad es. RGB-Color:[integer, integer, integer])

◆ Description schemas (DSs)

- Insieme di Descriptors e di altri Description Schemas. Definisce anche la struttura e la semantica delle relazioni tra i vari Ds e DSs

◆ Description Tools

- Insieme di Ds e DSs

◆ Description Definition Language (DDL)

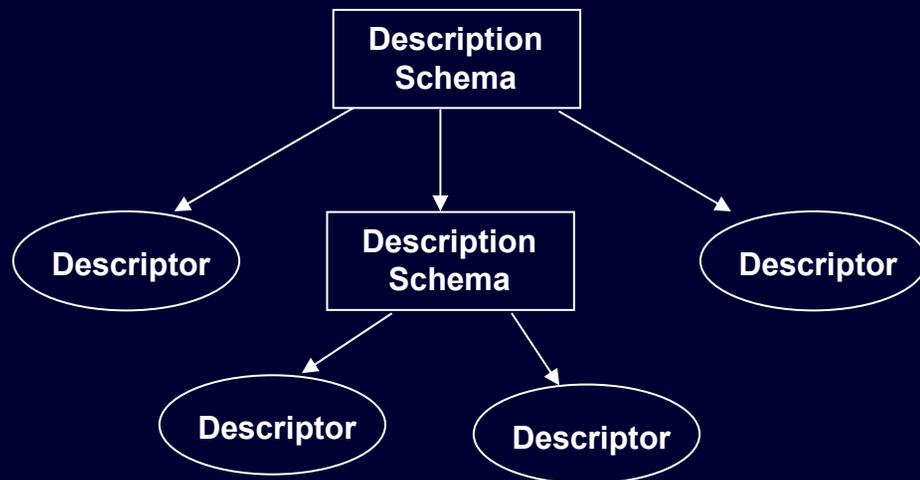
- Permette di definire nuovi Ds e DSs e di estendere quelli esistenti.
- Meccanismo utilizzato per l'estensione del modello.

◆ Description

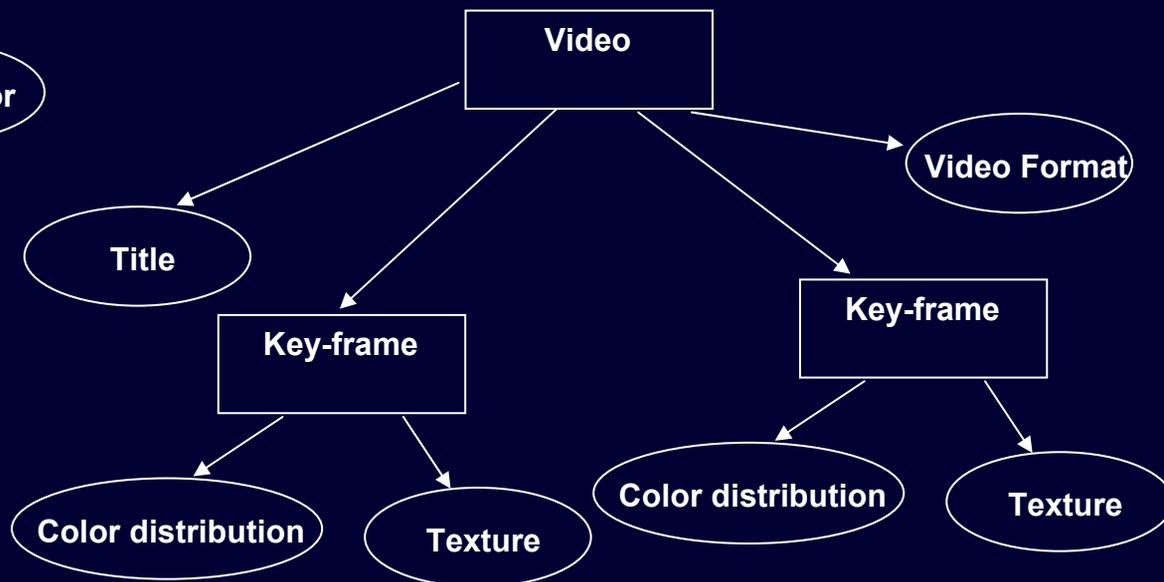
- Consiste di un Description Schema (struttura) e di un insieme di Descriptor Values (istanziamenti) che descrivono il dato

Esempio di Description Schema

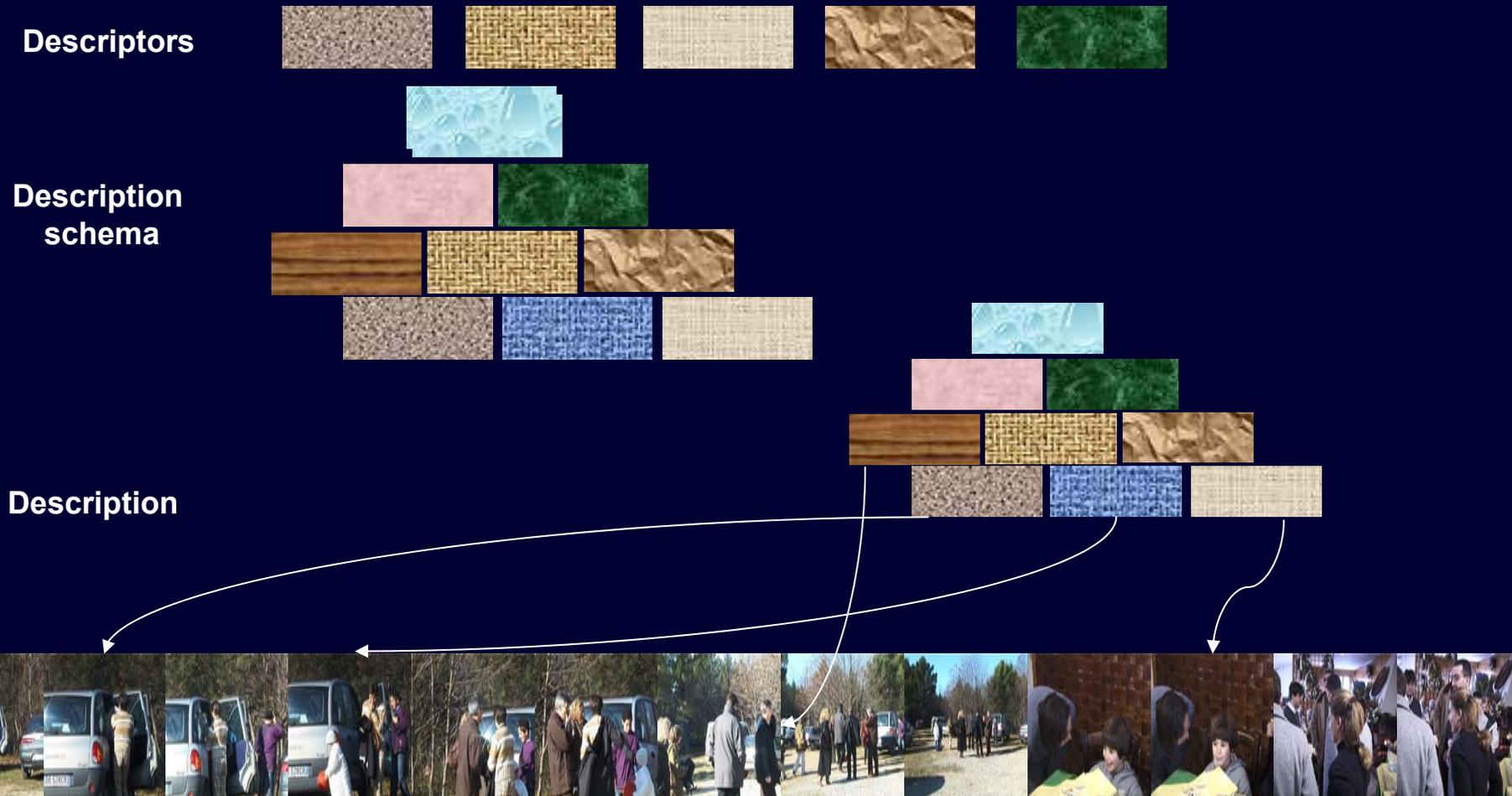
Schema



Istanza



Componenti di MPEG-7 [2/2]



MPEG-7 descriptions [1/3]

◆ Con i description tools di MPEG-7 possiamo creare diversi tipi di Descriptions

- Descriptions tradizionali orientate all'archiviazione
 - ➔ Informazione relativa alla creazione del contenuto ed al processo di produzione (ad es. regista, titolo, attori, ...)
 - ➔ Informazione legata all'uso del contenuto (ad es. pianificazione di broadcasting, copyright, ...)
 - ➔ Informazione relativa al metodo di storage e di rappresentazione del contenuto (ad es. formato AVI, MPEG2, ...)
- Descriptions che specificano come viene percepito il dato multimediale
- Informazione aggiuntiva per l'organizzazione, la gestione e l'accesso al contenuto

MPEG-7 descriptions [2/3]

- Descriptions che specificano come viene percepito il dato multimediale
 - ➔ **Informazione relativa alla struttura spaziale, temporale o spazio-temporale del contenuto (ed es. suddivisione in scene, segmentazione in regioni, tracking del moto di oggetti o regioni)**
 - ➔ **Informazione relativa alle features di basso livello (ad es. colore, tessitura, timbri sonori, descrizioni della melodia, ...)**
 - ➔ **Informazione semantica collegata alla realtà descritta dal dato multimediale (ad es. oggetti, eventi, interazioni tra gli oggetti, ...)**
- **Informazione aggiuntiva per l'organizzazione, la gestione e l'accesso al contenuto**

MPEG-7 descriptions [3/3]

- **Informazione aggiuntiva per l'organizzazione, la gestione e l'accesso al contenuto**
 - ➔ **Informazione su quali relazioni esistono tra gli oggetti e come sono raggruppati in collezioni**
 - ➔ **Informazione per gestire efficientemente il browsing del contenuto (ad es. sommari, indici, ...)**
 - ➔ **Informazione relativa all'interazione dell'utente (ad es. preferenze dell'utente, history dell'uso del contenuto)**

MPEG-7 description tools [1/3]

- ◆ **Insieme dei Descriptors e dei Description Schemas**

- ◆ **Standard description tools**
 - **Basic Elements**
 - ➔ **Entità generiche utilizzate come elementi costitutivi dei vari description tools (ad es. time, places, people, annotazione testuale, vocabolari controllati, ...)**
 - **Schema Tools**
 - ➔ **Usati per creare strutture complesse**
 - **Content Description Tools**
 - ➔ **Aspetti strutturali**
 - ➔ **MPEG-7 Visual Description Tools**
 - ➔ **MPEG-7 Audio Description Tools**

MPEG-7 Content Description Tools [1/2]

◆ **Aspetti strutturali**

- Permettono di descrivere il contenuto in termini di segmenti spazio-temporali organizzati in una struttura gerarchica (ad es. indice, tabella dei contenuti, composizione del video, ...)

◆ **MPEG-7 Visual:**

- Permettono di descrivere le seguenti feature visuali:
 - **Colour,**
 - **Texture,**
 - **Shape,**
 - **Motion,**
 - **Localisation,**
 - **Faces**
 - **....**

MPEG-7 Content Description Tools [2/2]

◆ MPEG-7 Audio:

- Audio description tools covering the following:
 - ➔ **Descriptors:**
 - spectral, parametric, temporal features
 - ➔ **Description Tools:**
 - sound recognition, instrumental timber, spoken content, audio signature, melody

MPEG-7 description tools [2/3]

◆ Standard description tools

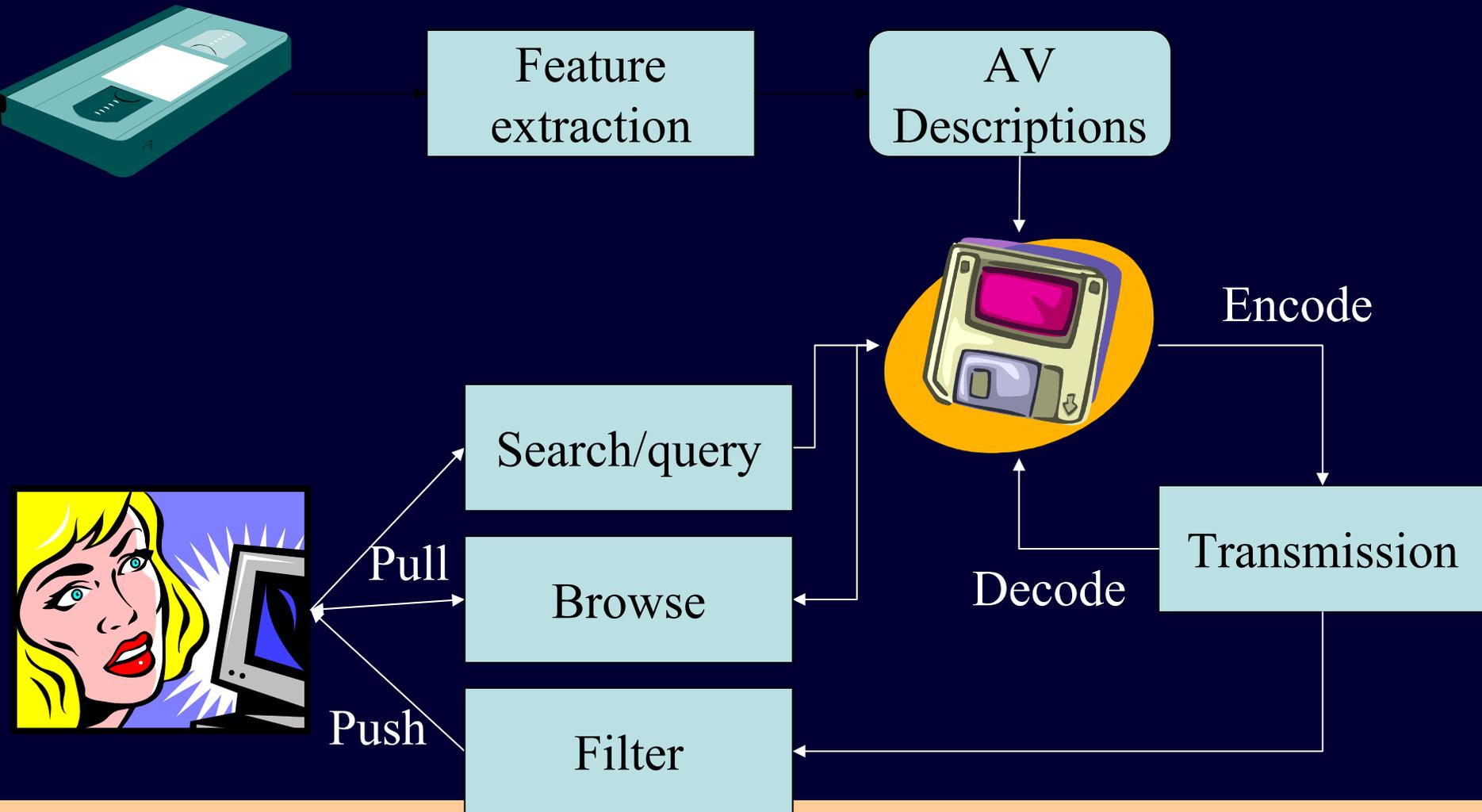
- Content Management Tools
 - ➔ Permettono di specificare informazione sulle caratteristiche del media, e sulla creazione ed uso dei dati multimediali
 - ➔ Media Description Tools
 - Storage media, formato di codifica, qualità, ...
 - ➔ Creation Description Tools
 - Titolo, luoghi, date, genere, soggetto, lingua, ...
 - ➔ Usage Description Tools
 - Condizioni per l'utilizzo del materiale (ad es. diritti, disponibilità)

MPEG-7 description tools [3/3]

◆ Standard description tools

- Navigation and Access tools
 - ➔ Permettono di specificare sommari, partizioni e decomposizioni del dato multimediale al fine di facilitare il browsing e la ricerca
 - ➔ Summary Description Tools
 - Forniscono modalità di navigazione sia sequenziale che gerarchica, al fine di permettere una efficace visione sintetica del materiale multimediale

MPEG-7



Riferimenti

- ◆ MPEG home page: <http://www.chiariglione.org/mpeg/>
- ◆ <http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>
- ◆ MPEG-7: behind the scenes, Jane Hunter, DLIB Magazine, settembre 1999, <http://www.dlib.org/dlib/september99/hunter/09hunter.html>
- ◆ Introduction to MPEG-7: Multimedia Content Description Interface, edited by B.S. Manjunath, P. Salebier, T. Sikora, Wiley, 2002
- ◆ MPEG-7 The Generic Multimedia Content Description Standard, Part 1 and Part 2, J. Martinez ed al. IEEE Multimedia, April-June 2002 e July-September 2002