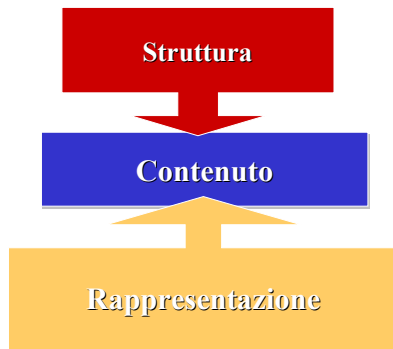


XML: componenti di un documento



Applicazioni di Rete - M. Ribaudò - DISI

XML Schema

- Meta linguaggio che usa XML per descrivere la struttura di un documento XML e offre una sintassi XML per descrivere i tipi di dati
- Marzo 2001: XML Schema specification
- Spesso si parla di XML Schema Definition (XSD)

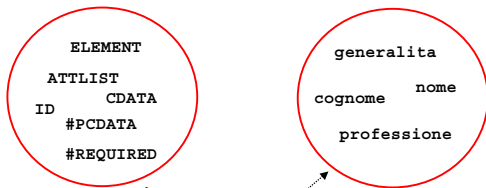
Applicazioni di Rete - M. Ribaudò - DISI

What are XML Schemas?

- Data Model
 - ✓ With XML Schemas you specify how your XML data will be organized, and the datatypes of your data. That is, with XML Schemas you model your XML data
- A contract
 - ✓ Organizations agree to structure their XML documents in conformance with the XML Schema. Thus, the XML Schema acts as a contract
- A rich source of metadata
 - ✓ An XML Schema document contains lots of data about the data in the XML documents, i.e., XML Schemas contain metadata

Applicazioni di Rete - M. Ribaudò - DISI

XML: DTD

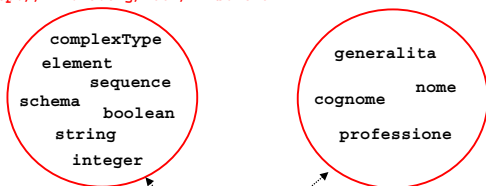


Con questo vocabolario la DTD permette di definire un nuovo linguaggio

Applicazioni di Rete - M. Ribaudò - DISI

XML Schema

<http://www.w3.org/2001/XMLSchema>



Con questo vocabolario uno schema XML permette di definire un nuovo linguaggio

Applicazioni di Rete - M. Ribaudò - DISI

XML Schema

▪ We think that very soon XML Schemas will be used in most Web applications as a replacement for DTDs. Here are some reasons

- ✓ XML Schemas are extensible to future additions
- ✓ XML Schemas are richer and more useful than DTDs
- ✓ XML Schemas are written in XML
- ✓ XML Schemas support data types
- ✓ XML Schemas support namespaces

Applicazioni di Rete - M. Ribaudò - DISI

Esempio: XML

```
<?xml version="1.0"?  
  encoding="ISO8859-1" ?>  
  
<note>  
  <to>Sara Longhi</to>  
  <from>Marina Ribaud</from>  
  <heading>Reminder</heading>  
  <body>Ricordati di mandarmi  
  l'ultima versione della tesi  
  entro venerdì.  
  </body>  
</note>
```

ISO8859-1 è l'encoding per usare le nostre lettere accentate senza problemi
Applicazioni di Rete - M. Ribaud - DISI

Esempio: DTD

```
<!ELEMENT note (to,from,heading,body)>  
<!ELEMENT to (#PCDATA)>  
<!ELEMENT from (#PCDATA)>  
<!ELEMENT heading (#PCDATA)>  
<!ELEMENT body (#PCDATA)>
```

Applicazioni di Rete - M. Ribaud - DISI

Esempio: XSD

```
<?xml version="1.0" ?>  
<xsd:schema>  
  <xsd:element name="note">  
    <xsd:complexType>  
      <xsd:sequence>  
        <xsd:element name="to" type="xsd:string"/>  
        <xsd:element name="from" type="xsd:string"/>  
        <xsd:element name="heading" type="xsd:string"/>  
        <xsd:element name="body" type="xsd:string"/>  
      </xsd:sequence>  
    </xsd:complexType>  
  </xsd:element>  
</xsd:schema>
```

Applicazioni di Rete - M. Ribaud - DISI

<xsd:schema>

```
<?xml version="1.0" ?>
<xsd:schema>
...
...
</xsd:schema>
```

È la radice di ogni schema XML
e può contenere degli attributi

```
<?xml version="1.0" ?>
<xsd:schema
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.mydomain.com"
xmlns="http://www.mydomain.com"
elementFormDefault="qualified">
...
...
</xsd:schema>
```

Applicazioni di Rete - M. Ribaudo - DISI

Simple element <xsd:element>

Può contenere solo testo

```
<xsd:element name="xxx" type="yyy"/>
```

```
<xsd:element name="cognome" type="xsd:string"/>
<xsd:element name="eta" type="xsd:integer"/>
<xsd:element name="nascita" type="xsd:date"/>
```

xsd:string	xsd:decimal
xsd:integer	xsd:boolean
xsd:date	xsd:time

Applicazioni di Rete - M. Ribaudo - DISI

Simple element <xsd:element>

Può avere un valore di default
oppure un valore fissato

```
<xsd:element name="xxx" type="yyy"
default="valore"/>
```

```
<xsd:element name="xxx" type="yyy"
fixed="valore"/>
```

Applicazioni di Rete - M. Ribaudo - DISI

Facets

Si possono stabilire dei vincoli sui possibili valori di un elemento, vincoli che sono detti **facets**

```
<xsd:element name="eta">
  <xsd:simpleType>
    <xsd:restriction base="xsd:integer">
      <xsd:minInclusive value="0"/>
      <xsd:maxInclusive value="100"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

Applicazioni di Rete - M. Ribaudo - DISI

Facets

Si possono specificare dei **pattern** con le espressioni regolari ☺

```
<xsd:element name="password">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="[a-zA-Z0-9]{8,20}"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

Applicazioni di Rete - M. Ribaudo - DISI

Complex elements

- Si tratta di elementi che possono contenere altri elementi, del testo e/o degli attributi

```
<xsd:element name="dipendenti">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="nome" type="xsd:string"/>
      <xsd:element name="cognome" type="xsd:string"/>
      <xsd:element name="reparto" type="xsd:integer"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Applicazioni di Rete - M. Ribaudo - DISI

Attributi <xsd:attribute>

Solo gli elementi complessi possono avere degli attributi

```
<xsd:attribute name="xxx" type="yyy"/>
```

```
<xsd:attribute
  name="lang"
  type="xsd:string"
  default="EN"/>
```

Applicazioni di Rete - M. Ribaudo - DISI

Inclusione di uno schema in un file XML

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

```
<note
xmlns="http://www.mydomain.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.mydomain.com note.xsd">

  <to>Sara Longhi</to>
  <from>Marina Ribaudo</from>
  <heading>Reminder</heading>
  <body>Ricordati di mandarmi
l'ultima versione della tesi
entro venerdì.
  </body>
</note>
```

Applicazioni di Rete - M. Ribaudo - DISI

Inclusione di uno schema in un file XML

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

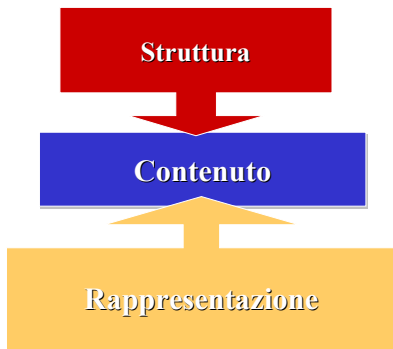
```
<note
xmlns="http://www.mydomain.com"
// La dichiarazione del namespace di default dice al validatore dello schema che gli
// elementi del documento saranno in questo spazio dei nomi

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
// Namespace dello XML Schema Instance

xsi:schemaLocation="http://www.mydomain.com note.xsd">
// File XML che contiene lo schema da usare per il namespace www.mydomain.com
...
...
...
</note>
```

Applicazioni di Rete - M. Ribaudo - DISI

XML: componenti di un documento



Applicazioni di Rete - M. Ribaudò - DISI

XML: rappresentazione

**What You See
Is NOT
What You Get**

Applicazioni di Rete - M. Ribaudò - DISI

XML: rappresentazione

- È possibile trasformare il markup che descrive il contenuto di un documento in markup di presentazione tramite programmi oppure mediante fogli di stile
- Per trasformare un documento XML in un documento interpretabile da un browser si possono usare i CSS (anche se si riesce a fare ben poco ...)

Applicazioni di Rete - M. Ribaudò - DISI

XML: esempio (spesa CLICK)

- Si deve creare un file .css che descrive come i vari tag devono essere **visualizzati** da un browser
- Il file .css viene **incluso** nel file .xml

```
<?xml version="1.0" encoding="UTF-8"?>  
<?xml-stylesheet type="text/css" href="spesaclick.css" ?>
```

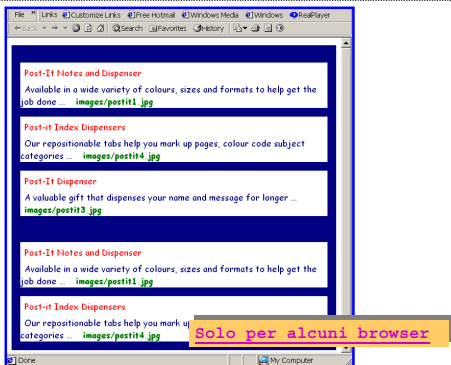
Applicazioni di Rete - M. Ribaudo - DISI

XML: esempio (spesa CLICK)

```
order {  
display:block; border-width:10;  
background-color:#000080; padding:1em; }  
  
item {  
display:block; font-family:"Comic Sans MS";  
background-color:#ffffff; }  
  
nameprod {  
display:block; color:red; padding:0.5em;  
margin-top:1em; }  
  
...
```

Applicazioni di Rete - M. Ribaudo - DISI

XML: esempio (spesa CLICK)



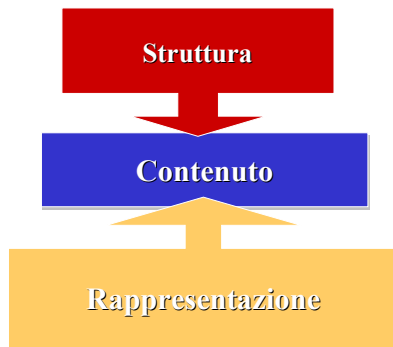
Applicazioni di Rete - M. Ribaudo - DISI

Trasformazioni

XSLT

Applicazioni di Rete - M. Ribaudò - DISI

XML: componenti di un documento



Applicazioni di Rete - M. Ribaudò - DISI

XLST

" ... XSLT is an XML application that specifies rules by which one XML document is transformed into another. An XSLT document - that is, an XSLT stylesheet - contains templates. An XSLT processor compares the elements in an input XML document to the templates in a stylesheet. When it finds a matching template, it writes the template's contents into an output tree. When it completes this step, it may serialize the output tree further into an XML document or another format, such as plain text or HTML ..."

Applicazioni di Rete - M. Ribaudò - DISI

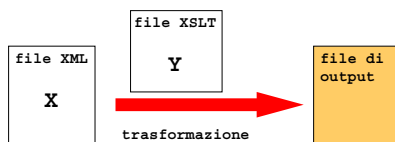
XSLT

- XSLT è un'applicazione XML che permette di specificare delle **regole** per trasformare un documento XML in un altro tipo di documento
- È una raccomandazione del W3C dal novembre 1999
- Scrivere una trasformazione XSLT vuol dire scrivere un **foglio di stile** che viene interpretato da un **processore XSLT**

Applicazioni di Rete - M. Ribaudo - DISI

XSLT

- Un documento XSLT è un file XML con estensione **.xsl**
- Contiene dei modelli (**template**) che il processore XSLT usa per "tradurre" un documento XML sorgente in un documento di output



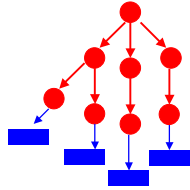
Applicazioni di Rete - M. Ribaudo - DISI

XSLT

- Esistono tre **strategie per trasformare** il documento XML (X) applicandovi una trasformazione XSLT (Y)
 - ✓ X e Y vengono forniti al client (di solito un browser) che contiene un interprete XSLT capace di eseguire la trasformazione
 - ✓ Si usa un **programma ad hoc** (ad esempio XML Spy)
 - ✓ Il **web server attua la trasformazione** e spedisce il file di output al client

Applicazioni di Rete - M. Ribaudo - DISI

XSLT



Albero associato ad un documento XML

Il processore XSLT visita l'albero del documento XML e, nodo per nodo, applica le trasformazioni specificate nei template

Applicazioni di Rete - M. Ribaudò - DISI

XSLT: struttura di un file .xsl

- È un file XML, quindi ha un elemento radice `<xsl:stylesheet>` che appartiene ad un preciso namespace

```
<?xml version="1.0" ?> ← È un documento XML!  
<xsl:stylesheet version="1.0"  
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">  
  
  template di visualizzazione qui  
  
</xsl:stylesheet>
```

Applicazioni di Rete - M. Ribaudò - DISI

XSLT: struttura di un file .xsl

- Per controllare la formattazione del documento di output si devono aggiungere dei **template** nel foglio di stile, dopo la radice
- Si usano l'elemento `xsl:template` e l'attributo `match`

Applicazioni di Rete - M. Ribaudò - DISI

XSLT: struttura di un file .xsl

```
<?xml version="1.0" ?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:template match="...">

    azioni che il processore XSLT
    svolge quando incontra l'elemento
    specificato nell'attributo match

  </xsl:template>

</xsl:stylesheet>
```

Applicazioni di Rete - M. Ribaudò - DISI

XSLT: struttura di un file .xsl

```
<?xml version="1.0" ?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:template match="/">

    <html>
      <head></head>
      <body></body>
    </html>

  </xsl:template>

</xsl:stylesheet>
```



```
<html>
<head></head>
<body></body>
</html>
```

Applicazioni di Rete - M. Ribaudò - DISI

XSLT: struttura di un file .xsl

- Elemento radice
`<xsl:stylesheet ...> ... </xsl:stylesheet>`
- Primo livello
`<xsl:template match="..."> ... </xsl:template>`
- Secondo livello
 - ✓ `xsl:apply-templates`
 - ✓ `xsl:value-of`
 - ✓ `xsl:for-each`

Applicazioni di Rete - M. Ribaudò - DISI

XSLT: struttura di un file .xsl

```
...  
<xsl:template match="/">  
  <html>  
  <head></head>  
  <body>  
    <xsl:apply-templates />  
  </body>  
</html>  
</xsl:template>  
...
```



Discesa ricorsiva ai nodi figli del nodo selezionato dall'attributo match

Applicazioni di Rete - M. Ribaudo - DISI

XSLT: struttura di un file .xsl

```
...  
<xsl:template match="/">  
  <html>  
  <head></head>  
  <body>  
    <xsl:apply-templates />  
  </body>  
</html>  
</xsl:template>  
<xsl:template match="nome1">  
  trasformazioni per nome1 qui  
</xsl:template>
```



Si scende sul nodo nome1 e si applicano le trasformazioni

Applicazioni di Rete - M. Ribaudo - DISI

XSLT: struttura di un file .xsl

```
...  
<xsl:template match="/">  
  <html>  
  <head></head>  
  <body>  
    <xsl:apply-templates />  
  </body>  
</html>  
</xsl:template>  
<xsl:template match="nome1">  
  <b><xsl:value-of select="nome2" /></b><br/>  
</xsl:template>
```



Si scende sul nodo nome1 e si restituisce il valore del suo nodo figlio nome2

Applicazioni di Rete - M. Ribaudo - DISI

XSLT: struttura di un file .xsl

```
...
<xsl:template match="/">
  <html>
    <head></head>
    <body>
      <xsl:apply-templates />
    </body>
  </html>
</xsl:template>
<xsl:template match="nome1">
  <xsl:for-each select="nome2">
    <b><xsl:value-of select="."/></b><br/>
  </xsl:for-each>
</xsl:template>
```

Si scende sul nodo nome1 e si restituiscono i valori di tutti i figli nome2

Applicazioni di Rete - M. Ribaudò - DISI

XSLT: struttura di un file .xsl

Filtro sull'output

```
<xsl:for-each
  select="nome1[nome2='valore']">
  trasformazioni qui
</xsl:for-each>
```

nome2 != valore
nome2 < valore
nome2 > valore

Ordinamento

```
<xsl:for-each select="nome1">
  <xsl:sort select="nome2"/>
  trasformazioni qui
</xsl:for-each>
```

Applicazioni di Rete - M. Ribaudò - DISI

XSLT: struttura di un file .xsl

Istruzione IF

```
<xsl:for-each select="nome1">
  <xsl:if test="condizione">
    trasformazioni qui
  </xsl:if>
</xsl:for-each>
```

nome2 < 10

Applicazioni di Rete - M. Ribaudò - DISI

XPath

- XSLT usa XPath per **identificare parti di un documento XML**
- XPath permette di far riferimento al **nodo radice**, ai **nodi intermedi**, agli **attributi** di un nodo
- Permette di **identificare un nodo**
 - ✓ **rispetto alla sua posizione**
(es. Il terzo nodo figlio di un certo elemento)
 - ✓ **rispetto al suo contenuto**
(es. Il nodo il cui valore di nome è Rossi)

Applicazioni di Rete - M. Ribaud - DISI

XPath

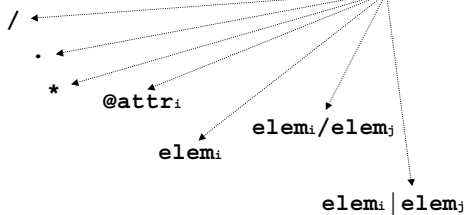
- Con XPath si definiscono le parti del documento XML che verranno trasformate nel documento di output
- Xpath distingue tra
 - ✓ nodo radice
 - ✓ nodo di tipo elemento
 - ✓ nodo di tipo testo
 - ✓ nodo di tipo attributo
 - ✓ nodo di tipo commento
 - ✓ nodo di tipo processing instruction
 - ✓ nodo di tipo namespace

Applicazioni di Rete - M. Ribaud - DISI

Xpath

Nel caso più semplice si scriverà

```
<xsl:apply-templates select="*">
```



Applicazioni di Rete - M. Ribaud - DISI

Inclusione di un file .xsl

- Se il processore XSLT è incorporato nel browser il documento XML deve avere un'istruzione che specifica quale foglio di stile deve essere applicato

```
<?xml version="1.0" ?>
<?xml-stylesheet
  type="text/xsl"
  href="nomefile.xsl"
?>
```

NB: solo per alcuni browser!

Applicazioni di Rete - M. Ribaudo - DISI

Inclusione: attenzione !!!

- Namespace MSIE (non è lo standard)

URI: "http://www.w3.org/TR/WD-xsl"

- Namespace proposto dal W3C
(valido anche su Netscape 7.x)

URI: "http://www.w3.org/1999/XSL/Transform"

Applicazioni di Rete - M. Ribaudo - DISI

XSLT: esempio (spesa CLICK)

spesa CLICK			
Articoli per Ufficio >>> Post-it			
Prodotto	Descrizione	Prezzo	Quantità
	Post-It Notes and Dispenser Available in a wide variety of colours, sizes and formats to help get the job done.	2 euro	<input type="checkbox"/>
	Post-it Index Dispensers Our repositionable tabs help you mark up pages, colour code subject categories and highlight salient points. Available in a choice of colours, sizes and a range of convenient dispensers.	3 euro	<input type="checkbox"/>
	Post-It Dispenser A valuable gift that dispenses your name and message for longer.	2 euro	<input type="checkbox"/>
	Post-It Notes A variety of colours and sizes to suit your needs - you can imprint your company logo or any message you choose.	2 euro	<input type="checkbox"/>

File HTML

Applicazioni di Rete - M. Ribaudo - DISI

XSLT: esempio (spesa CLICK)

```
<?xml version="1.0" encoding="UTF-8" ?>
<?xml-stylesheet type="text/xml"
  href="spesaclick.xsl" ?>

<order num="234" date="2004-12-15">
  <item idprod="1" quantity="10" price="2">
    <nameprod>Post-It Notes and Dispenser</nameprod>
    <description>Available in a wide variety of colours,
      sizes and formats to help get the job done ...
    </description>
    <photo>postit1.jpg</photo>
  </item>
  <item idprod="2" quantity="10" price="3">
    ...
  </item>
  ...
</order>
```

File XML (minimale) per descrivere un ordine

Applicazioni di Rete - M. Ribaud - DISI

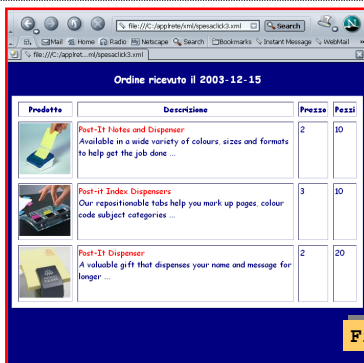
XSLT: esempio (spesa CLICK)




```
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="order">
  <html>
  <head>
  <link rel="stylesheet" type="text/css" href="stile.css" />
  </head>
  <body bgcolor="#000080" text="white">
  <center>
  <h3>Ordine ricevuto il
    <xsl:apply-templates select="@date"/></h3>
  ...
</template>
```

Applicazioni di Rete - M. Ribaud - DISI

XSLT: esempio (spesa CLICK)



Prodotto	Descrizione	Prezzo	Pezzi
	Post-It Notes and Dispenser Available in a wide variety of colours, sizes and formats to help get the job done ...	2	10
	Post-It Index Dispensers Our responsible tabs help you mark up pages, colour code subject categories ...	3	10
	Post-It Dispenser A valuable gift that dispenses your name and message for longer ...	2	20

File XML!

Applicazioni di Rete - M. Ribaud - DISI

XSLT e PHP: esempio

```
<?php
// Creo un processore XSLT
$xmlproc = xslt_create();

// Eseguo la trasformazione;
// il risultato viene copiato nella variabile $html
$html = xslt_process($xmlproc, 'file.xml', 'file.xsl');

// Se ci sono errori scrivo un messaggio di errore
if (!$html)
    die('XSLT processing error: '. xslt_error($xmlproc));

// Rilascio il processore XSLT
xslt_free($xmlproc);

// Invio al client il risultato
echo $html;
?>
```

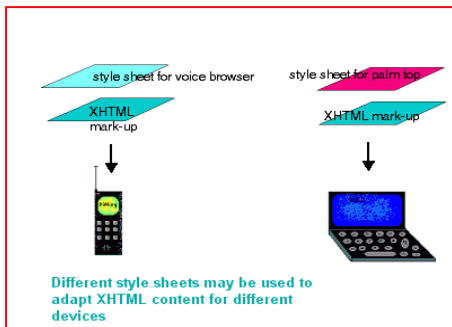
Applicazioni di Rete - M. Ribaudò - DISI

XSLT e PHP: esempio

- Naturalmente ... il file XML dovrebbe essere generato dinamicamente, mediante un'interrogazione al database MySQL
- Inoltre, con PHP siamo in grado di riconoscere il tipo di user agent da cui arriva una richiesta e si può pensare di applicare un foglio di stile appropriato a seconda del tipo di client
- Il tutto "on the fly" per arrivare alla ...

Applicazioni di Rete - M. Ribaudò - DISI

... Comunicazione multicanale



Lo vedremo in laboratorio ...

Applicazioni di Rete - M. Ribaudò - DISI
