



ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

Μηχανική Γνώσης και Συστήματα Γνώσης

Δομημένα Έγγραφα Ιστού: XML

Μανώλης Μαραγκουδάκης

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αιγαίου**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Ενότητα 2

Δομημένα Έγγραφα Ιστού:XML

Διαφάνειες από το βιβλίο **A Semantic
Web Primer**

των

- Grigoris Antoniou
- Frank van Harmelen

Παράδειγμα HTML

<h2>Nonmonotonic Reasoning: Context-
Dependent Reasoning</h2>

<i>by V. Marek and

M. Truszczyński</i>

Springer 1993

ISBN 0387976892

Το ίδιο παράδειγμα σε XML

```
<book>  
  <title>Nonmonotonic Reasoning: Context-  
    Dependent Reasoning</title>  
  <author>V. Marek</author>  
  <author>M. Truszczyński</author>  
  <publisher>Springer</publisher>  
  <year>1993</year>  
  <ISBN>0387976892</ISBN>  
</book>
```

HTML vs XML: Ομοιότητες

- Κοινή χρήση tags/ετικετών (π.χ. <h2> και </year>)
 - Εμφωλευμένες ετικέτες
 - Οι άνθρωποι μπορούν να διαβάζουν και να ερμηνεύουν τόσο HTML όσο και XML
- ... Τί γίνεται με τις μηχανές;

Προβλήματα στην Αυτοματοποιημένη Ερμηνεία HTML εγγράφων

Ένας ευφυής πράκτορας προσπαθεί να ανακτήσει τα ονόματα των συγγραφέων του βιβλίου

- Τα ονόματα θα μπορούσαν να εμφανίζονται αμέσως μετά τον τίτλο
- ή αμέσως μετά τη λέξη «από»
- Υπάρχουν 2 συγγραφείς ή ένας που λέγεται “V. Marek and M. Truszczyński”?

HTML vs. XML: Δομικές Πληροφορίες

- Τα έγγραφα σε HTML δεν περιέχουν **δομικές πληροφορίες**: σχετικά με τμήματα του εγγράφου και τις συσχετίσεις τους
- Η XML είναι περισσότερο προσβάσιμη στις μηχανές διότι
 - Κάθε τμήμα της πληροφορίας περιγράφεται
 - Οι μορφές των σχέσεων ορίζονται από την ένθεση (εμφώλιασμα)
 - Π.χ. Η ετικέτα **<author>** tags εμφανίζεται μέσα στην ετικέτα **<book>**, άρα ότι περικλείει το **<author>** αναφέρεται στο συγκεκριμένο βιβλίο.

HTML vs. XML: Δομικές Πληροφορίες (2)

- Η XML επιτρέπει επίσης περιορισμούς (constraints) σε τιμές
 - Π.χ. Το έτος πρέπει να είναι αριθμός 4 ψηφίων

HTML vs. XML: Μορφοποίηση

- Η μορφοποίηση σε HTML είναι πιο πλούσια από της XML
- Η κύρια χρήση της HTML είναι η παρουσίαση πληροφοριών: πρέπει να καθορίσει τη μορφοποίηση
- XML: διαχωρισμός περιεχομένου από την παρουσίαση
 - Η ίδια πληροφορία μπορεί να προβάλλεται με ποικίλους τρόπους

HTML vs. XML: Άλλο παράδειγμα

- Σε HTML

```
<h2>Relationship force-mass</h2>  
<i> F = M × a </i>
```

- Σε XML

```
<equation>  
  <meaning>Relationship force-  
mass</meaning>  
  <leftside> F </leftside>  
  <rightside> M × a </rightside>  
</equation>
```

HTML vs. XML: Διαφορετική χρήση ετικετών

- Στην HTML οι ετικέτες είναι κοινές
- Στην XML εντελώς διαφορετικές
- Στην HTML οι ετικέτες προσδιορίζουν την παρουσίαση: χρώμα, γραμματοσειρά, λίστες...
- Στην XML οι ετικέτες ποικίλουν: **τις ορίζει ο χρήστης**
- XML **meta markup language**: language for defining markup languages

XML Λεξικά

- Οι διαδικτυακές εφαρμογές πρέπει να συμφωνούν σε κοινά αποδεκτά λεξικά για να επικοινωνήσουν
- Οι κοινότητες και οι επιχειρηματικοί τομείς δημιουργούν επεκτάσεις (εξού και το X στην XML) για διάφορους τομείς
 - μαθηματικά (MathML)
 - βιοπληροφορική (BSML)
 - human resources (HRML)
- Το W3C έχει ορίσει διάφορες γλώσσες πάνω από την XML
 - SVG
 - SMIL

Η γλώσσα XML

Ένα έγγραφο XML αποτελείται από:

- πρόλογο
- Ένα αριθμό **στοιχείων**
- Ένα προαιρετικό **επίλογο** (δεν θα συζητηθεί)

Πρόλογος σε ένα XML αρχείο

Αποτελείται από:

- Μια δήλωση XML
- Μια προαιρετική αναφορά σε εξωτερικά δομικά έγγραφα

```
<?xml version="1.0" encoding="UTF-16"?>
```

```
<!DOCTYPE book SYSTEM "book.dtd">
```

XML Στοιχεία

- Τα “πράγματα” στα οποία η XML αναφέρεται
 - Π.χ. Βιβλία, συγγραφείς, εκδότες, κτλ.
- Ένα στοιχείο αποτελείται από:
 - *Μια ετικέτα ανοίγματος*
 - *Το περιεχόμενο*
 - *Μια ετικέτα κλεισίματος*

<lecturer>David Billington</lecturer>

XML Στοιχεία (2)

- Τα ονόματα μπορούν να επιλέγονται σχεδόν ελεύθερα
- Ο πρώτος χαρακτήρας μπορεί να είναι γράμμα, ή κάτω παύλα ή ανώ κάτω τελεία
- Κανένα όνομα δεν μπορεί να ξεκινάει με το “xml” σε οποιονδήποτε συνδυασμό κεφαλαίων-μικρών
 - Π.χ. “Xml”, “xML”

XML Στοιχεία (3)

- Το περιεχόμενο μπορεί να είναι κείμενο, άλλα στοιχεία ή το κενό

<lecturer>

<name>David Billington</name>

<phone> +61 – 7 – 3875 507 </phone>

</lecturer>

- Εάν δεν υπάρχει περιεχόμενο το στοιχείο αποκαλείται κενό και μπορεί να γραφεί **<lecturer/>** αντί για **<lecturer></lecturer>**

XML Χαρακτηριστικά

- Ένα κενό στοιχείο δεν είναι αναγκαστικά άσκοπο
 - Μπορεί να έχει ορισμένες ιδιότητες (properties) με τη μορφή των χαρακτηριστικών (attributes)
- Ένα χαρακτηριστικό είναι ένα ζεύγος ονόματος-τιμής μέσα στην ετικέτα ανοίγματος κάποιου στοιχείου
`<lecturer name="David Billington" phone="+61 - 7 - 3875 507"/>`

XML Χαρακτηριστικά (2)

```
<order orderNo="23456" customer="John  
  Smith"  
  date="October 15, 2002">  
  <item itemNo="a528" quantity="1"/>  
  <item itemNo="c817" quantity="3"/>  
</order>
```

Το προηγούμενο χωρίς χαρακτηριστικά

```
<order>
  <orderNo>23456</orderNo>
  <customer>John Smith</customer>
  <date>October 15, 2002</date>
  <item>
    <itemNo>a528</itemNo>
    <quantity>1</quantity>
  </item>
  <item>
    <itemNo>c817</itemNo>
    <quantity>3</quantity>
  </item>
</order>
```

XML Στοιχεία vs. Χαρακτηριστικών

- Τα χαρακτηριστικά μπορούν να αντικατασταθούν από στοιχεία
- Είναι θέμα γούστου να χρησιμοποιούνται στοιχεία ή χαρακτηριστικά
- Ομως τα χαρακτηριστικά ΔΕΝ μπορούν να εμφωλεύονται

Άλλα συστατικά της XML

- Σχόλια
 - Αγνοείται από τον συντακτικό αναλυτή (parser)
 - **<!-- This is a comment -->**
- Οδηγίες επεξεργασίας (PIs)
 - Παρέχουν ένα μηχανισμό πληροφορίας
 - **<?stylesheet type="text/css" href="mystyle.css"?>**

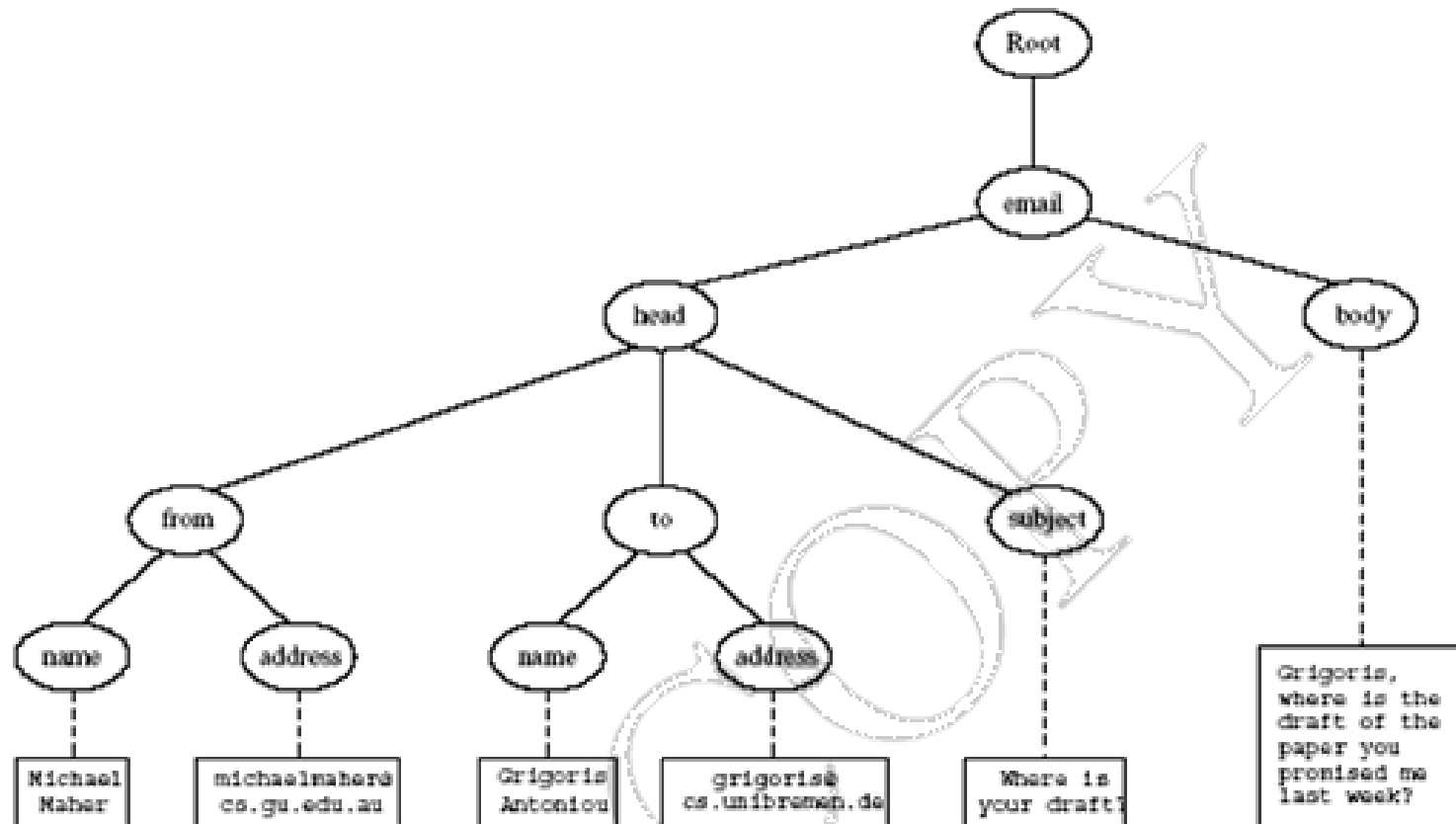
Σωστά δομημένα XML έγγραφα

- Αυτά που είναι συντακτικά σωστά
- Μερικοί συντακτικοί κανόνες:
 - Μόνο ένα εξωτερικό στοιχείο (**root element**)
 - Κάθε στοιχείο περιέχει μια ετικέτα ανοίγματος και κλεισίματος
 - Δεν επιτρέπεται η αλληλοκάλυψη ετικετών
 - **<author><name>Lee Hong</author></name>**
 - Τα χαρακτηριστικά μέσα σε ένα στοιχείο έχουν μοναδικά ονόματα
 - Τα ονόματα ετικετών και στοιχείων να είναι επιτρεπτά

Δενδρικό μοντέλο της XML

```
<email>
  <head>
    <from name="Michael Maher"
          address="michaelmaher@cs.gu.edu.au"/>
    <to name="Grigoris Antoniou"
        address="grigoris@cs.unibremen.de"/>
    <subject>Where is your draft?</subject>
  </head>
  <body>
    Grigoris, where is the draft of the paper you promised me
    last week?
  </body>
</email>
```

Δενδρικό μοντέλο της XML



Δενδρικό μοντέλο της XML

- Το διατεταγμένο αυτό δέντρο έχει:
 - Μια ακριβώς ρίζα
 - Δεν υπάρχουν κύκλοι
 - Κάθε κόμβος έχει ακριβώς ένα γονιό, εκτός από τη ρίζα
 - Κάθε κόμβος έχει μια ετικέτα
 - Η σειρά των στοιχείων είναι σημαντική
 - Όχι και των χαρακτηριστικών

Δόμηση XML εγγράφων

- Ένα έγγραφο XML είναι **έγκυρο** αν:
 - Είναι καλά δομημένο
 - Χρησιμοποιεί τις δομικές πληροφορίες
- Υπάρχουν 2 τρόποι για να ορίσουμε τη δομή ενός XML εγγράφου:
 - DTDs (παλαιότερος και πιο περιοριστικός τρόπος)
 - XML Schema (προσφέρει εκτεταμένες δυνατότητες)

DTD: Τύποι Στοιχείων

```
<lecturer>  
  <name>David Billington</name>  
  <phone> +61 – 7 – 3875 507 </phone>  
</lecturer>
```

DTD για το παραπάνω στοιχείο (και όλα τα στοιχεία τύπου **lecturer**):

```
<!ELEMENT lecturer (name,phone)>  
<!ELEMENT name (#PCDATA)>  
<!ELEMENT phone (#PCDATA)>
```

Το νόημα του DTD

- Οι τύποι στοιχείων **lecturer**, **name**, και **phone** μπορούν να χρησιμοποιηθούν μέσα στο έγγραφο
- Ένα στοιχείο **lecturer** περιέχει ένα **name** και ένα **phone**, σε αυτή την ακολουθία
- Ένα στοιχείο **name** και ένα **phone** μπορούν να έχουν οποιοδήποτε περιεχόμενο
- Στα DTDs, ο τύπος **#PCDATA** είναι ο μόνος ατομικός τύπος για στοιχεία

DTD: Διαζεύξεις στα στοιχεία

- Μπορούμε να πούμε ότι ένα **lecturer** στοιχείο περιέχει είτε ένα **name** ή ένα **phone** ως εξής:

```
<!ELEMENT lecturer (name|phone)>
```

- Ένα **lecturer** περιέχει ένα **name** και ένα **phone** σε οποιαδήποτε σειρά.

```
<!ELEMENT lecturer((name,phone)|(phone,name))>
```

- δεν είναι αποδοτική για πολλά στοιχεία

Παράδειγμα XML Στοιχείου

```
<order orderNo="23456"  
        customer="John Smith"  
        date="October 15, 2002">  
  <item itemNo="a528" quantity="1"/>  
  <item itemNo="c817" quantity="3"/>  
</order>
```


Το αντίστοιχο DTD

<!ELEMENT order (item+)>

<!ATTLIST order	orderNo	ID	#REQUIRED
	customer	CDATA	#REQUIRED
	date	CDATA	#REQUIRED

>

<!ELEMENT item EMPTY>

<!ATTLIST item	itemNo	ID	#REQUIRED
	quantity	CDATA	#REQUIRED
	comments	CDATA	#IMPLIED

>

Σχόλια για το DTD

- Ο τύπος στοιχείου **item** ορίζεται ως κενός
- **+** (μετά το **item**) είναι **τελεστής πληθικότητας (cardinality operator)**:
 - **?**: εμφανίζεται μια ή καμία φορά
 - *****: εμφανίζεται καμία ή περισσότερες φορές
 - **+**: εμφανίζεται μια ή περισσότερες φορές
 - Εάν δεν υπάρχει τελεστής σημαίνει ακριβώς μια φορά

Σχόλια για το DTD (2)

- Όπως ορίζουμε στοιχεία ορίζουμε και χαρακτηριστικά (attributes)
- `<!ATTLIST` που περιέχει:
 - Το όνομα του στοιχείου για το οποίο ισχύει η λίστα
 - Μια σειρά απο τριάδες που έχουν το όνομα, τον τύπο και τον τύπο της τιμής του χαρακτηριστικού
- **Όνομα χαρακτηριστικού**: ένα όνομα που μπορεί να χρησιμοποιηθει σε κάποιο έγγραφο XML με DTD

DTD: Τύποι χαρακτηριστικών

- Οι πιο σημαντικοί
 - **CDATA**, αλφαριθμητικό
 - **ID**, όνομα μοναδικό σε ολόκληρο το έγγραφο
 - **IDREF**, μια αναφορά σε κάποιο άλλο στοιχείο με χαρακτηριστικό ID, το οποίο έχει την ίδια τιμή με το χαρακτηριστικό IDREF
 - **IDREFS**, μια σειρά από IDREF
 - **(v1|...|vn)**, απαρίθμηση όλων των δυνατών τιμών
- Περιορισμοί: όχι ημερομηνίες, εύρη αριθμών, κτλ.

DTD: Τυποί τιμών ενός χαρακτηριστικού

- **#REQUIRED**
 - υποχρεωτικό
- **#IMPLIED**
 - προαιρετικό
- **#FIXED "value"**
 - Κάθε στοιχείο πρέπει να έχει αυτό το χαρακτηριστικό, το οποίο έχει πάντα την τιμή "value"
- **"value"**
 - Ορίζει την προεπιλεγμένη τιμή
 - Αν οριστεί μια τιμή μέσα στο έγγραφο XML υπερισχύει έναντι της προεπιλεγμένης

Αναφορές με IDREF και IDREFS (πρώτα το DTD)

```
<!ELEMENT family (person*)>
```

```
<!ELEMENT person (name)>
```

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

```
<!ATTLIST person      id      ID      #REQUIRED  
                    mother IDREF #IMPLIED  
                    father IDREF #IMPLIED  
                    children IDREFS #IMPLIED>
```

Το XML που ακολουθεί το DTD

```
<family>
  <person id="bob" mother="mary" father="peter">
    <name>Bob Marley</name>
  </person>
  <person id="bridget" mother="mary">
    <name>Bridget Jones</name>
  </person>
  <person id="mary" children="bob bridget">
    <name>Mary Poppins</name>
  </person>
  <person id="peter" children="bob">
    <name>Peter Marley</name>
  </person>
</family>
```

XML Οντότητες

- Μια οντότητα XML παίζει πολλούς ρόλους:
 - Δεσμευτικό θέσης (placeholder) για επαναλαμβανόμενους χαρακτήρες
 - Τμήμα εξωτερικών δεδομένων
 - Μέρος μιας δήλωσης στοιχείων
- Μπορούμε να χρησιμοποιήσουμε αναφορά στην οντότητα *&thisyear* αντί για την τιμή "2010"
<!ENTITY thisyear " 2010" >

DTD για ένα στοιχείο Email (2)

```
<!ELEMENT cc EMPTY>
```

```
<!ATTLIST cc  
           name CDATA #IMPLIED  
           address CDATA #REQUIRED>
```

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

```
<!ELEMENT body (text,attachment*)>
```

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

```
<!ELEMENT attachment EMPTY>
```

```
<!ATTLIST attachment  
           encoding (mime|binhex) "mime"  
           file CDATA #REQUIRED>
```

Το προηγούμενο DTD σε τμήματα

- Ένα στοιχείο **head** περιέχει τα ακόλουθα (με αυτή τη σειρά):
 - Ένα στοιχείο **from**
 - Τουλάχιστον ένα στοιχείο **to**
 - Μηδέν ή περισσότερα στοιχεία **cc**
 - Ένα στοιχείο **subject**
- Στα **from**, **to**, και **cc**
 - Το χαρακτηριστικό **name** δεν είναι αναγκαίο
 - Το **address** είναι

Το προηγούμενο DTD σε τμήματα (2)

- Ένα στοιχείο **body**
 - ένα **text**
 - Πιθανά να ακολουθείται από έναν αριθμό στοιχείων **attachment**
- Το χαρακτηριστικό **encoding** ενός στοιχείου **attachment** πρέπει να έχει είτε την τιμή “**mime**” ή “**binhex**”
 - Η “**mime**” είναι η προκαθορισμένη τιμή

Αναδρομικοί τύποι σε DTDs

- **<!ELEMENT bintree
((bintree root bintree)|emptytree)>**

XML Schema

- Σαφώς πλουσιότερη γλώσσα για να ορίζουμε δομές εγγράφων
- Η σύνταξη της βασίζεται στην ίδια την XML
 - Επαναχρησιμοποίηση σχημάτων (schemas)
- Νέοι, πλουσιότεροι τύποι δεδομένων σε σχέση με τα DTD που υποστηρίζουν μόνο αλφαριθμητικά

XML Schema (2)

- Ένα XML schema είναι ένα στοιχείο με ετικέτα ανοίγματος `<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema" version="1.0">`
- Δομή
 - Τύποι στοιχείων και χαρακτηριστικών με τύπους δεδομένων

Element Types

```
<element name="email"/>
```

```
<element name="head" minOccurs="1"  
  maxOccurs="1"/>
```

```
<element name="to" minOccurs="1"/>
```

Περιορισμοί πληθικότητας

- **minOccurs="x"** (προκαθορισμένη τιμή το 1)
- **maxOccurs="x"** (προκαθορισμένη τιμή το 1)

Τύποι Χαρακτηριστικών

```
<attribute name="id" type="ID" use="required"/>
```

```
< attribute name="speaks" type="Language"  
  use="default" value="en"/>
```

- Περιλαμβάνουν τύπους
 - Type="..."
- ύπαρξη: **use="x"**, όπου το **x** είναι **optional** ή **required**
- προκαθορισμένη τιμή: **use="x" value="..."**, όπου το **x** είναι **default** ή **fixed**

Τύποι Δεδομένων

- Υπάρχει ποικιλία ενσωματωμένων τύπων δεδομένων
 - Αριθμητικοί: **integer**, **Short** etc.
 - Αλφαριθμητικά: **string**, **ID**, **IDREF**, **CDATA** etc.
 - Ημερομηνία και ώρα: **time**, **Month** etc.
- Υπάρχουν επίσης και τύποι ορισμένοι από το χρήστη
 - Απλοί τύποι δεδομένων, που δεν μπορούν να χρησιμοποιήσουν στοιχεία ή χαρακτηριστικά
 - σύνθετοι, που μπορούν

Τύποι Δεδομένων (2)

- Οι **Σύνθετοι** προκύπτουν από ήδη υπάρχοντες τύπους δεδομένων, με τον ορισμό κάποιων χαρακτηριστικών (εάν υπάρχουν) και τη χρήση των:
 - **sequence**, μια ακολουθία υπαρχόντων τύπων δεδομένων(η σειρά μετράει)
 - **all**, μια συλλογή από στοιχεία που πρέπει να υπάρχουν (η σειρά δε μετράει)
 - **choice**, μια συλλογή από στοιχεία, ένα από τα οποία πρέπει να επιλεγεί

Παράδειγμα σύνθετου τύπου

```
<complexType name="lecturerType">  
  <sequence>  
    <element name="firstname" type="string"  
      minOccurs="0" maxOccurs="unbounded"/>  
    <element name="lastname" type="string"/>  
  </sequence>  
  <attribute name="title" type="string" use="optional"/>  
</complexType>
```

Επέκταση τύπων δεδομένων

- Ήδη υπάρχοντες τύποι δεδομένων μπορούν να επεκταθούν με νέα στοιχεία ή χαρακτηριστικά

```
<complexType name="extendedLecturerType">  
  <extension base="lecturerType">  
    <sequence>  
      <element name="email" type="string"  
        minOccurs="0" maxOccurs="1"/>  
    </sequence>  
    <attribute name="rank" type="string" use="required"/>  
  </extension>  
</complexType>
```

Προκύπτει ο εξής τύπος

```
<complexType name="extendedLecturerType">  
  <sequence>  
    <element name="firstname" type="string"  
      minOccurs="0" maxOccurs="unbounded"/>  
    <element name="lastname" type="string"/>  
    <element name="email" type="string"  
      minOccurs="0" maxOccurs="1"/>  
  </sequence>  
  <attribute name="title" type="string" use="optional"/>  
  <attribute name="rank" type="string" use="required"/>  
</complexType>
```

Επέκταση τύπων δεδομένων(2)

- Προκύπτει μια **ιεραρχική συσχέτιση** μεταξύ του αρχικού και του εκτεταμένου τύπου
 - Τα στιγμιότυπα του εκτεταμένου είναι και στιγμιότυπα του αρχικού
 - Μπορούν να περιέχουν και άλλες πληροφορίες αλλά όχι λιγότερες ή να έχουν πληροφορίες με λάθος τύπου

Περιορισμοί τύπων δεδομένων

- Δεν είναι το αντίθετο της επέκτασης
- Δεν διαγράφονται στοιχεία ή χαρακτηριστικά
- Τα στιγμιότυπα του περιορισμένου τύπου είναι και στιγμιότυπα του αρχικού
- Ικανοποιούν **τουλάχιστο** τους περιορισμούς του αρχικού καθώς επίσης και νέους περιορισμούς

Περιορισμοί τύπων δεδομένων (παράδειγμα)

```
<complexType name="restrictedLecturerType">
  <restriction base="lecturerType">
    <sequence>
      <element name="firstname" type="string"
        minOccurs="1" maxOccurs="2"/>
    </sequence>
    <attribute name="title" type="string"
      use="required"/>
    </restriction>
  </complexType>
```

Περιορισμοί απλών τύπων δεδομένων

```
<simpleType name="dayOfMonth">  
  <restriction base="integer">  
    <minInclusive value="1"/>  
    <maxInclusive value="31"/>  
  </restriction>  
</simpleType>
```

Περιορισμοί τύπων δεδομένων (απαρίθμηση)

```
<simpleType name="dayOfWeek">  
  <restriction base="string">  
    <enumeration value="Mon"/>  
    <enumeration value="Tue"/>  
    <enumeration value="Wed"/>  
    <enumeration value="Thu"/>  
    <enumeration value="Fri"/>  
    <enumeration value="Sat"/>  
    <enumeration value="Sun"/>  
  </restriction>  
</simpleType>
```

XML Schema: Το παράδειγμα με το email

```
<element name="email" type="emailType"/>
```

```
<complexType name="emailType">
```

```
  <sequence>
```

```
    <element name="head" type="headType"/>
```

```
    <element name="body" type="bodyType"/>
```

```
  </sequence>
```

```
</complexType>
```

XML Schema: Το παράδειγμα με το email(2)

```
<complexType name="headType">  
  <sequence>  
    <element name="from" type="nameAddress"/>  
    <element name="to" type="nameAddress"  
      minOccurs="1" maxOccurs="unbounded"/>  
    <element name="cc" type="nameAddress"  
      minOccurs="0" maxOccurs="unbounded"/>  
    <element name="subject" type="string"/>  
  </sequence>  
</complexType>
```

XML Schema: Το παράδειγμα με το email (3)

```
<complexType name="nameAddress">  
  <attribute name="name" type="string"  
    use="optional"/>  
  <attribute name="address"  
    type="string" use="required"/>  
</complexType>
```

- Παρόμοιο για **bodyType**

Χώροι Ονομάτων-Namespaces

- Ένα έγγραφο σε XML μπορεί να χρησιμοποιεί περισσότερα από ένα DTD σχήματα
- Αν όμως 2 έγγραφα DTD A και B ορίζουν ένα στοιχείο με διαφορετικούς τρόπους, θα έχουμε συγκρούσεις ονομάτων
- Η λύση είναι απλή:
 - Διαφορετικό πρόθεμα για κάθε σχήμα
 - `prefix:name`

Παράδειγμα

```
<vu:instructors xmlns:vu="http://www.vu.com/empDTD"
  xmlns:gu="http://www.gu.au/empDTD"
  xmlns:uky="http://www.uky.edu/empDTD">

  <uky:faculty uky:title="assistant professor"
    uky:name="John Smith"
    uky:department="Computer Science" />

  <gu:academicStaff
    gu:title="lecturer"
    gu:name="Mate Jones"
    gu:school="Information Technology" />

</vu:instructors>
```


Ορισμοί χώρων ονομάτων

- Δηλώνονται μέσα σε ένα στοιχείο και μπορούν να χρησιμοποιηθούν στο στοιχείο και σε οποιοδήποτε από τα παιδιά του (στοιχεία και χαρακτηριστικά)
- Μια δήλωση namespace έχει τη μορφή:
 - **xmlns:prefix="location"**
 - **location** η διεύθυνση του DTD ή του σχήματος
- Αν δεν μπει πρόθεμα, **xmlns="location"** τότε χρησιμοποιείται εκ των προτέρων η **location**

Παράδειγμα

```
<vu:instructors xmlns:vu="http://www.vu.com/empDTD"
  xmlns:"http://www.gu.au/empDTD"
  xmlns:uky="http://www.uky.edu/empDTD">

  <uky:faculty uky:title="assistant professor"
    uky:name="John Smith"
    uky:department="Computer Science"/>

  <academicStaff title="lecturer"
    name="Mate Jones"
    school="Information Technology"/>

</vu:instructors>
```

Αναφορά και ερώτημα σε XML έγγραφα

- Στις σχεσιακές βάσεις δεδομένων, τμήματα μιας βάσης μπορούν να αναζητηθούν με χρήση SQL
 - Ίδια αναγκαιότητα για XML έγγραφα
 - Γλώσσες επερώτησης: XQuery, XQL, XML-QL
- Η κεντρική ιδέα μιας XML γλώσσας επερώτησης είναι η παράσταση διαδρομής (path expression)
 - Ορίζει τον τρόπο με τον οποίο μπορούμε να φτάσουμε σε ένα κόμβο ενός δέντρου

XPath

- Είναι η κύρια γλώσσα επερώτησης XML
- Αναφερόμαστε σε τμήματα ενός εγγράφου XML
 - Λειτουργεί στο επίπεδο του δενδρικού μοντέλου της XML
 - Δεν έχει συντακτικό σαν της XML

Ορισμοί

- Έστω το ακόλουθο έγγραφο:
 - `<bookstore>`
 - `<book>`
 - `<title>Harry Potter</title>`
 - `<author>J K. Rowling</author>`
 - `<year>2005</year>`
 - `<price>29.99</price>`
 - `</book>`
 - `</bookstore>`

Ορισμοί

- Κόμβοι στοιχείων και ιδιοτήτων
 - `<bookstore>` (root element node)
 - `<author>J K. Rowling</author>` (element node)
 - `lang="en"` (attribute node)
- Ατομικές τιμές
 - J K. Rowling
 - "en«

Ορισμοί

- Parent
 - Book=parent(title)
- Children
 - Author=child(book)
- Siblings
 - title,author,year=siblings
- Ancestors
 - book,bookstore=ancestors(title)
- Decendants
 - book,title,author=decendants(bookstore)

```
<book>  
  <title>Harry Potter</title>  
  <author>J K. Rowling</author>  
  <year>2005</year>  
  <price>29.99</price>  
</book>
```

XPATH

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

<bookstore>

<book>
  <title lang="eng">Harry Potter</title>
  <price>29.99</price>
</book>

<book>
  <title lang="eng">Learning XML</title>
  <price>39.95</price>
</book>

</bookstore>
```


Βασική χρήση ΧΡΑΤΗ

Έκφραση	Περιγραφή
nodename	Επιλέγει όλα τα παιδιά του Nodename
/	Επιλέγει αρχίζοντας από τον κόμβο ρίζα
//	Επιλέγει κόμβους του εγγράφου από τον τρέχοντα κόμβο που ταιριάζουν στην επιλογή, όπου και να βρίσκονται
.	Επιλέγει τον τρέχοντα κόμβο
..	Επιλέγει τον γονέα του τρέχοντα κόμβου
@	Επιλέγει ιδιότητες

Έκφραση	Αποτέλεσμα
bookstore	Επιλέγει όλα τα παιδιά του Bookstore
/bookstore	Επιλέγει το στοιχείο ρίζα bookstore
bookstore/book	Επιλέγει όλα τα στοιχεία book που είναι παιδιά του bookstore
//book	Επιλέγει όλα τα στοιχεία book όπου και αν είναι στο έγγραφο
bookstore//book	Επιλέγει όλα τα στοιχεία book που είναι απόγονοι του bookstore, αναξάρτητα από αν είναι κάτω από το στοιχείο bookstore
//@lang	Επιλέγει άλλες τις ιδιότητες που ονομάζονται lang

Κατηγορήματα στην ΧΡΑΤΗ

Έκφραση

/bookstore/book[1]

/bookstore/book[last()]

/bookstore/book[last()-1]

/bookstore/book[position()<3]

//title[@lang]

//title[@lang='eng']

/bookstore/book[price>35.00]

/bookstore/book[price>35.00]/title

Αποτέλεσμα

Επιλέγει το πρώτο στοιχείο book που είναι παιδί του στοιχείου Bookstore

Επιλέγει το τελευταίο στοιχείο book που είναι παιδί του στοιχείου Bookstore

Επιλέγει το προτελευταίο στοιχείο book που είναι παιδί του στοιχείου Bookstore

Επιλέγει τα πρώτα 2 στοιχεία book που είναι παιδιά του στοιχείου Bookstore

Επιλέγει όλα τα στοιχεία title που έχουν ιδιότητα **lang**

Επιλέγει όλα τα στοιχεία title που έχουν ιδιότητα **lang** με τιμή 'eng'

Επιλέγει όλα τα στοιχεία book που είναι παιδιά του στοιχείου Bookstore που έχουν price >35.00

Επιλέγει όλα τα στοιχεία title που είναι παιδιά του στοιχείου book που είναι παιδιά του στοιχείου Bookstore που έχουν price >35.00

Wildcards

Έκφραση

/bookstore/*

//*

//title[@*]

Αποτέλεσμα

Επιλέγει όλα τα παιδιά του στοιχείου bookstore

Επιλέγει όλα τα παιδιά του εγγράφου

Επιλέγει όλα τα στοιχεία title που έχουν οποιαδήποτε ιδιότητα

Χρήση XPATH σε browser

```
<html>
<body>
<script type="text/javascript">
function loadXMLDoc(dname)
{
if (window.XMLHttpRequest)
{
xhttp=new XMLHttpRequest();
}
else
{
xhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
xhttp.open("GET",dname,false);
xhttp.send("");
return xhttp.responseXML;
}
```

```
xml=loadXMLDoc("books.xml");
path="/bookstore/book[price>35]/title";
// code for IE
if (window.ActiveXObject)
{
var nodes=xml.selectNodes(path);

for (i=0;i<nodes.length;i++)
{
document.write(nodes[i].childNodes[0].nodeValue);
document.write("<br />");
}
// code for Mozilla, Firefox, Opera, etc.
else if (document.implementation &&
document.implementation.createDocument)
{
var nodes=xml.evaluate(path, xml, null, XPathResult.ANY_TYPE,null);
var result=nodes.iterateNext();

while (result)
{
document.write(result.childNodes[0].nodeValue);
document.write("<br />");
result=nodes.iterateNext();
}
}
}
</script>
</body>
</html>
```

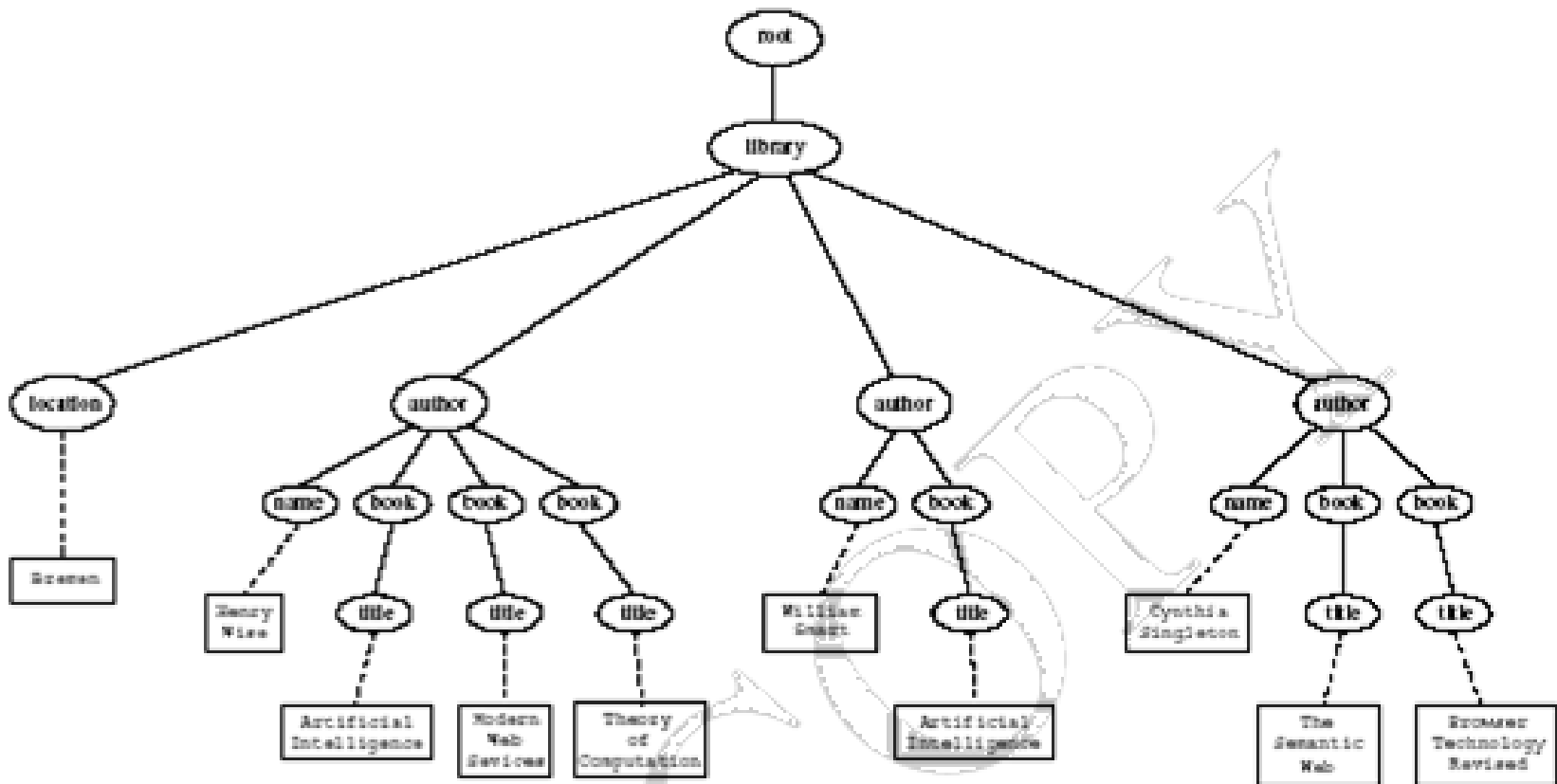
Παραστάσεις Διαδρομής

- **Απόλυτες** (ξεκινώντας από τη ρίζα)
 - Αρχίζουν με το σύμβολο /
 - Αναφέρεται στη ρίζα του εγγράφου (ένα επίπεδο πάνω από το στοιχείο – ρίζα του εγγράφου)
- **Σχετικές** με ένα κόμβο αναφοράς

Παράδειγμα XML

```
<library location="Bremen">
  <author name="Henry Wise">
    <book title="Artificial Intelligence"/>
    <book title="Modern Web Services"/>
    <book title="Theory of Computation"/>
  </author>
  <author name="William Smart">
    <book title="Artificial Intelligence"/>
  </author>
  <author name="Cynthia Singleton">
    <book title="The Semantic Web"/>
    <book title="Browser Technology Revised"/>
  </author>
</library>
```

Δενδρική αναπαράσταση



Δυνατότητες της Xpath

- Διεύθυνση όλων των στοιχείων **author**
/library/author
- Δείχνει όλα τα στοιχεία **author** που είναι παιδιά του στοιχείου **library** που είναι αμέσως κάτω από τη ρίζα
- **/t1/.../tn**, όπου κάθε **ti+1** είναι παιδί του **ti**

Δυνατότητες της Xpath(2)

- Δείχνει όλα τα στοιχεία **author**
//author
- Το // λέει πως πρέπει να δούμε όλα τα στοιχεία του εγγράφου και να δούμε εάν είναι τύπου **author**
- Το μονοπάτι αυτό δείχνει όλα τα στοιχεία **author** οπουδήποτε στο έγγραφο

Δυνατότητες της Xpath (3)

- Δείχνει το χαρακτηριστικό location μέσα σε κόμβους του library

`/library/@location`

- Το σύμβολο @ χρησιμοποιείται να δείξει κόμβους χαρακτηριστικών

Δυνατότητες της Xpath (4)

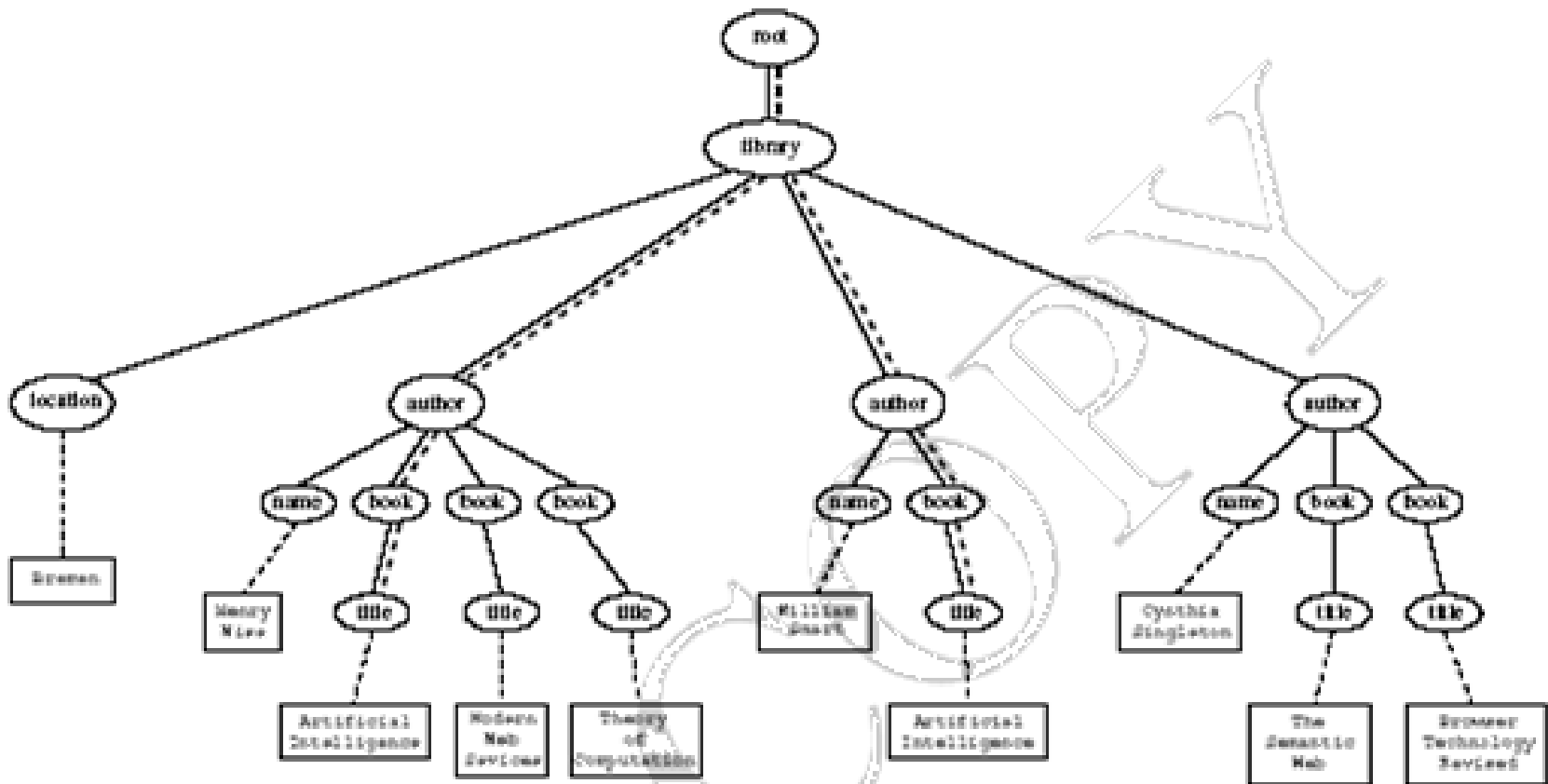
- Δείχνει όλα τα χαρακτηριστικά title μέσα στο book, οπουδήποτε στο έγγραφο, με τιμή "Artificial Intelligence"

`//book/@title="Artificial Intelligence"`

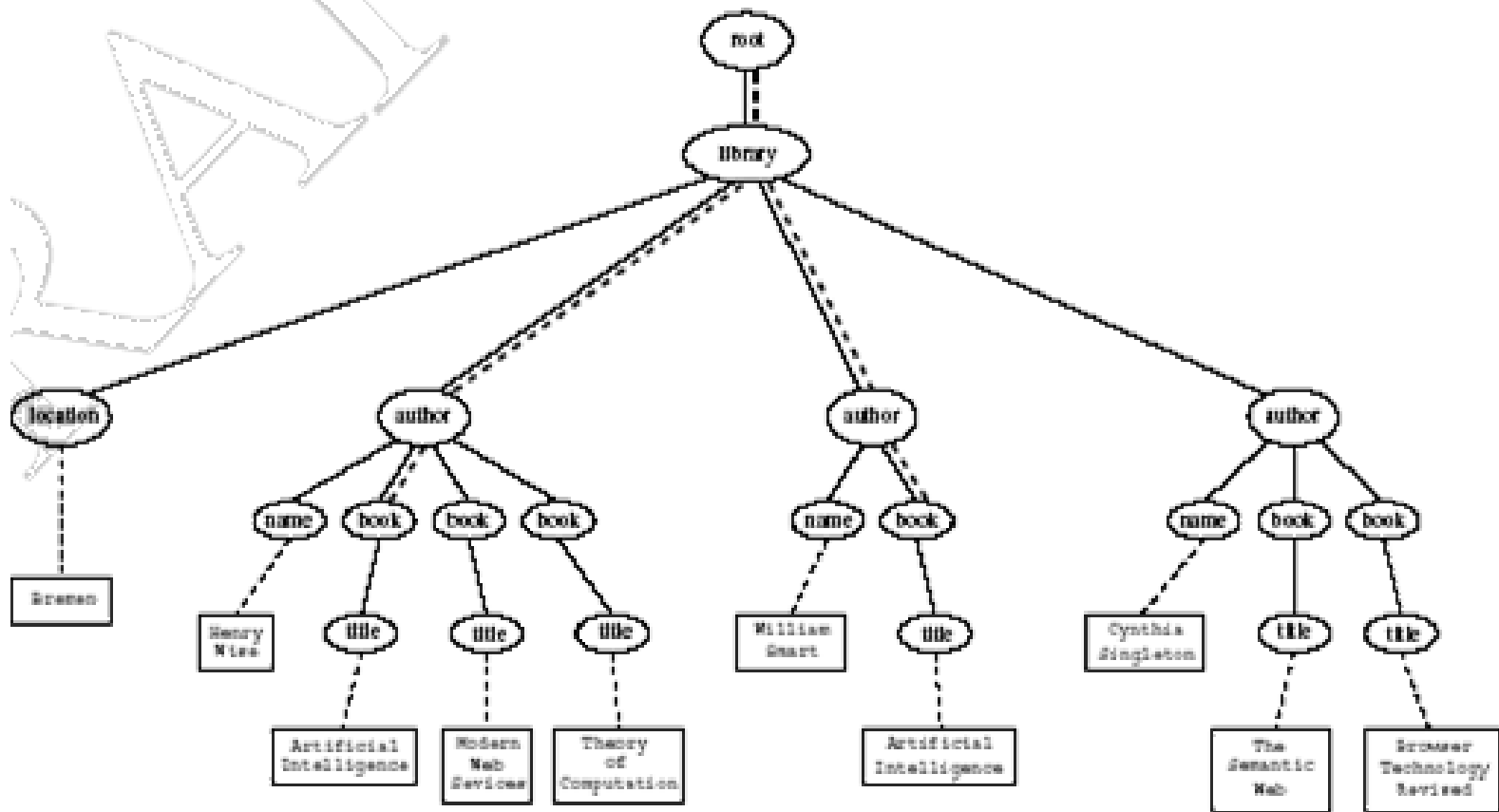
Δυνατότητες της Xpath (5)

- Δείχνει όλα τα βιβλία με τίτλο “Artificial Intelligence”
`/book[@title="Artificial Intelligence"]`
- Έλεγχος μέσα στις αγγύλες: μια παράσταση φίλτρου (filter expression)
- Διαφορά με την ερώτηση 4.
 - Η ερώτηση 5 δείχνει όλα τα στοιχεία book για τα οποία ο title έχει μια συγκεκριμένη συνθήκη
 - Η ερώτηση 4 συλλέγει όλους τους κόμβους title των στοιχείων book

Δενδρική αναπαράσταση Ερ 4



Δενδρική αναπαράσταση Ερ 5



Δυνατότητες της Xpath (6)

- Αναφορά στον πρώτο κόμβο του στοιχείου author
//author[1]
- Αναφορά στο τελευταίο στοιχείο book μέσα στον πρώτο author
//author[1]/book[last()]
- Αναφορά σε όλα τα στοιχεία book χωρίς χαρακτηριστικό title
//book[not @title]

Xpath

- Περισσότερα θα βρείτε στο
 - <http://www.w3.org/TR/xpath/>

Επεξεργασία XML εγγράφων

<author>

<name>Grigoris Antoniou</name>

<affiliation>University of Bremen</affiliation>

<email>ga@tzi.de</email>

</author>

Μπορεί να παρουσιαστεί με πολλούς τρόπους:

Grigoris Antoniou
University of Bremen
ga@tzi.de

Grigoris Antoniou
University of Bremen
ga@tzi.de

Style Sheets

- Τα φύλλα στυλ μπορούν να γραφούν σε πολλές γλώσσες
 - CSS2 (cascading style sheets level 2)
 - XSL (extensible stylesheet language)
- Η XSL περιλαμβάνει
 - Μια γλώσσα μετασχηματισμού XSLT
 - Μια γλώσσα μορφοποίησης

XSL μετασχηματισμοί (XSLT)

- Οι μετασχηματισμοί XSLT ορίζουν κανόνες με τους οποίους ένα έγγραφο XML μετατρέπεται σε
 - Ένα άλλο έγγραφο XML
 - Ένα HTML έγγραφο
 - plain text
- Το έγγραφο εξόδου μπορεί να χρησιμοποιεί το ίδιο DTD ή σχήμα ή ένα εντελώς διαφορετικό λεξικό

XSLT για προβολή σε HTML

```
<xsl:template match="/author">
  <html>
    <head><title>An author</title></head>
    <body bgcolor="white">
      <b><xsl:value-of select="name" /></b>
      <br />
      <xsl:value-of select="affiliation" />
      <br />
      <i><xsl:value-of select="email" /></i>
    </body>
  </html>
</xsl:template>
```

Style Sheet έξοδος

```
<html>
  <head><title>An author</title></head>
  <body bgcolor="white">
    <b>Grigoris Antoniou</b><br>
    University of Bremen<br>
    <i>ga@tzi.de</i>
  </body>
</html>
```

Παρατηρήσεις για της XSLT

- Είναι πάνω από την XML
- Ορίζουν ένα **template** (πρότυπο)
 - Στο προηγούμενο παράδειγμα ένα HTML έγγραφο με μερικούς placeholders για να τοποθετούμε περιεχόμενο στο έγγραφο
- Το **xsl:value-of** ανακτά την τιμή ενός στοιχείου και την αντιγράφει στην έξοδο

Παράδειγμα HTML πίνακα (static)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">
<html>
<body>
<h1>Music Collection:</h1>
<table border="1">
<tr bgcolor="#9acd32">
<th>Title</th>
<th>Artist</th>
</tr>
<tr>
<td><xsl:value-of select="catalog/cd/title" /></td>
<td><xsl:value-of select="catalog/cd/artist" /></td>
</tr>
</table>
</body>
</html>
</xsl:template>

</xsl:stylesheet>
```

Music Collection:

Title	Artist
Empire Burlesque	Bob Dylan

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Edited by XMLSpy -->
<catalog>
<cd>
<title>Empire Burlesque</title>
<artist>Bob Dylan</artist>
<country>USA</country>
<company>Columbia</company>
<price>10.90</price>
<year>1985</year>
</cd>
<cd>
<title>Hide your heart</title>
<artist>Bonnie Tyler</artist>
<country>UK</country>
<company>CBS
Records</company>
<price>9.90</price>
<year>1988</year>
</cd>
<cd>
<title>Greatest Hits</title>
<artist>Dolly Parton</artist>
<country>USA</country>
<company>RCA</company>
<price>9.90</price>
<year>1982</year>
</cd>
</catalog>
```

Παράδειγμα HTML πίνακα (dynamic)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">
<html>
<body>
<h1>Music Collection:</h1>
<table border="1">
<tr bgcolor="#9acd32">
<th>Title</th>
<th>Artist</th>
</tr>
<xsl:for-each select="catalog/cd">
<xsl:sort select="artist"/>
<tr>
<td><xsl:value-of select="title" /></td>
<td><xsl:value-of select="artist" /></td>
</tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>

</xsl:stylesheet>
```

Music Collection:

Title	Artist
Empire Burlesque	Bob Dylan
Hide your heart	Bonnie Tyler
Greatest Hits	Dolly Parton

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Edited by XMLSpy -->
<catalog>
<cd>
<title>Empire Burlesque</title>
<artist>Bob Dylan</artist>
<country>USA</country>
<company>Columbia</company>
<price>10.90</price>
<year>1985</year>
</cd>
<cd>
<title>Hide your heart</title>
<artist>Bonnie Tyler</artist>
<country>UK</country>
<company>CBS
Records</company>
<price>9.90</price>
<year>1988</year>
</cd>
<cd>
<title>Greatest Hits</title>
<artist>Dolly Parton</artist>
<country>USA</country>
<company>RCA</company>
<price>9.90</price>
<year>1982</year>
</cd>
</catalog>
```


Παράδειγμα με if

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Edited by XMLSpy® -->
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
  <body>
    <h2>My CD Collection</h2>
    <table border="1">
      <tr bgcolor="#9acd32">
        <th>Title</th>
        <th>Artist</th>
      </tr>
      <xsl:for-each select="catalog/cd">
        <xsl:if test="price>10">
          <tr>
            <td><xsl:value-of select="title"/></td>
            <td><xsl:value-of select="artist"/></td>
          </tr>
        </xsl:if>
      </xsl:for-each>
    </table>
  </body>
</html>
</xsl:template>
</xsl:stylesheet>
```

My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Edited by XMLSpy -->
<catalog>
  <cd>
    <title>Empire Burlesque</title>
    <artist>Bob Dylan</artist>
    <country>USA</country>
    <company>Columbia</company>
    <price>10.90</price>
    <year>1985</year>
  </cd>
  <cd>
    <title>Hide your heart</title>
    <artist>Bonnie Tyler</artist>
    <country>UK</country>
    <company>CBS
Records</company>
    <price>9.90</price>
    <year>1988</year>
  </cd>
  <cd>
    <title>Greatest Hits</title>
    <artist>Dolly Parton</artist>
    <country>USA</country>
    <company>RCA</company>
    <price>9.90</price>
    <year>1982</year>
  </cd>
</catalog>
```

Παράδειγμα με choose

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Edited by XMLSpy@ -->
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<body>
<h2>My CD Collection</h2>
<table border="1">
<tr bgcolor="#9acd32">
<th>Title</th>
<th>Artist</th>
</tr>
<xsl:for-each select="catalog/cd">
<tr>
<td><xsl:value-of select="title"/></td>
<xsl:choose>
<xsl:when test="price > 10">
<td bgcolor="#ff00ff">
<xsl:value-of select="artist"/>
</td>
</xsl:when>
<xsl:otherwise>
<td><xsl:value-of select="artist"/></td>
</xsl:otherwise>
</xsl:choose>
</tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan
Hide your heart	Bonnie Tyler
Greatest Hits	Dolly Parton

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Edited by XMLSpy -->
<catalog>
<cd>
<title>Empire Burlesque</title>
<artist>Bob Dylan</artist>
<country>USA</country>
<company>Columbia</company>
<price>10.90</price>
<year>1985</year>
</cd>
<cd>
<title>Hide your heart</title>
<artist>Bonnie Tyler</artist>
<country>UK</country>
<company>CBS
Records</company>
<price>9.90</price>
<year>1988</year>
</cd>
<cd>
<title>Greatest Hits</title>
<artist>Dolly Parton</artist>
<country>USA</country>
<company>RCA</company>
<price>9.90</price>
<year>1982</year>
</cd>
</catalog>
```

Παράδειγμα με apply-templates

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Edited by XMLSpy® -->
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">
  <html>
  <body>
  <h2>My CD Collection</h2>
  <xsl:apply-templates/>
  </body>
  </html>
</xsl:template>

<xsl:template match="cd">
  <p>
    <xsl:apply-templates select="title"/>
    <xsl:apply-templates select="artist"/>
  </p>
</xsl:template>

<xsl:template match="title">
  Title: <span style="color:#ff0000">
<xsl:value-of select="."/></span>
  <br />
</xsl:template>>
```

My CD Collection

Title: **Empire Burlesque**
Artist: **Bob Dylan**

Title: **Hide your heart**
Artist: **Bonnie Tyler**

Title: **Greatest Hits**
Artist: **Dolly Parton**

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Edited by XMLSpy -->
<catalog>
  <cd>
    <title>Empire Burlesque</title>
    <artist>Bob Dylan</artist>
    <country>USA</country>
    <company>Columbia</company>
    <price>10.90</price>
    <year>1985</year>
  </cd>
  <cd>
    <title>Hide your heart</title>
    <artist>Bonnie Tyler</artist>
    <country>UK</country>
    <company>CBS
Records</company>
    <price>9.90</price>
    <year>1988</year>
  </cd>
  <cd>
    <title>Greatest Hits</title>
    <artist>Dolly Parton</artist>
    <country>USA</country>
    <company>RCA</company>
    <price>9.90</price>
    <year>1982</year>
  </cd>
</catalog>
```

Παράδειγμα (έστω ότι έχουμε 2 authors)

```
<authors>
  <author>
    <name>Grigoris Antoniou</name>
    <affiliation>University of Bremen</affiliation>
    <email>ga@tzi.de</email>
  </author>
  <author>
    <name>David Billington</name>
    <affiliation>Griffith University</affiliation>
    <email>david@gu.edu.net</email>
  </author>
</authors>
```

Ορίζουμε το ακόλουθο XSLT

```
<xsl:template match="/">
  <html>
    <head><title>Authors</title></head>
    <body bgcolor="white">
      <xsl:apply-templates select="authors"/>
      <!-- Apply templates for AUTHORS
      children -->
    </body>
  </html>
</xsl:template>
```

Ορίζουμε το ακόλουθο XSLT (2)

```
<xsl:template match="authors">
  <xsl:apply-templates select="author"/>
</xsl:template>
<xsl:template match="author">
  <h2><xsl:value-of select="name"/></h2>
  Affiliation:<xsl:value-of
    select="affiliation"/><br>
  Email: <xsl:value-of select="email"/>
  <p>
</xsl:template>
```

Παραγόμενη έξοδος

```
<html>
  <head><title>Authors</title></head>
  <body bgcolor="white">
    <h2>Grigoris Antoniou</h2>
    Affiliation: University of Bremen<br>
    Email: ga@tzi.de
    <p>
    <h2>David Billington</h2>
    Affiliation: Griffith University<br>
    Email: david@gu.edu.net
    <p>
  </body>
</html>
```

Ερμηνεία του παραδείγματος

- Το **xsl:apply-templates** προκαλεί όλα τα παιδιά του κόμβου να ταιριάζουν με την έκφραση του επιλεγμένου μονοπατιού
 - Π.χ. Αν το τρέχον template εφαρμόζεται στο /,
 - Τότε το στοιχείο **xsl:apply-templates** εφαρμόζεται στο /
 - Δηλ στο στοιχείο **authors** (το / είναι πάνω από το στοιχείο-ρίζα)
 - Αν ο τρέχον κόμβος είναι το στοιχείο **authors**, τότε το στοιχείο **xsl:apply-templates select="author"** προκαλεί τα template για τα στοιχεία **author** να εφαρμόζονται σε όλα τα παιδιά **author** του στοιχείου **authors**

Ερμηνεία του παραδείγματος

- Αποτελεί καλή πρακτική να ορίζεται `template` για κάθε τύπο στοιχείου του εγγράφου
 - Ακόμη και αν δεν εφαρμόζεται κάποιο στύλ σε ορισμένα στοιχεία το **`xsl:apply-templates`** καλό είναι να χρησιμοποιείται

Template για XML χαρακτηριστικά

- Έστω η μετατροπή του στοιχείου στον εαυτό του (η πιο απλή περίπτωση)

```
<person firstname="John" lastname="Woo"/>
```

- Λάθος λύση:

```
<xsl:template match="person">  
  <person firstname="<xsl:value-of  
    select="@firstname">"  
    lastname="<xsl:value-of select="@lastname">" />  
</xsl:template>
```

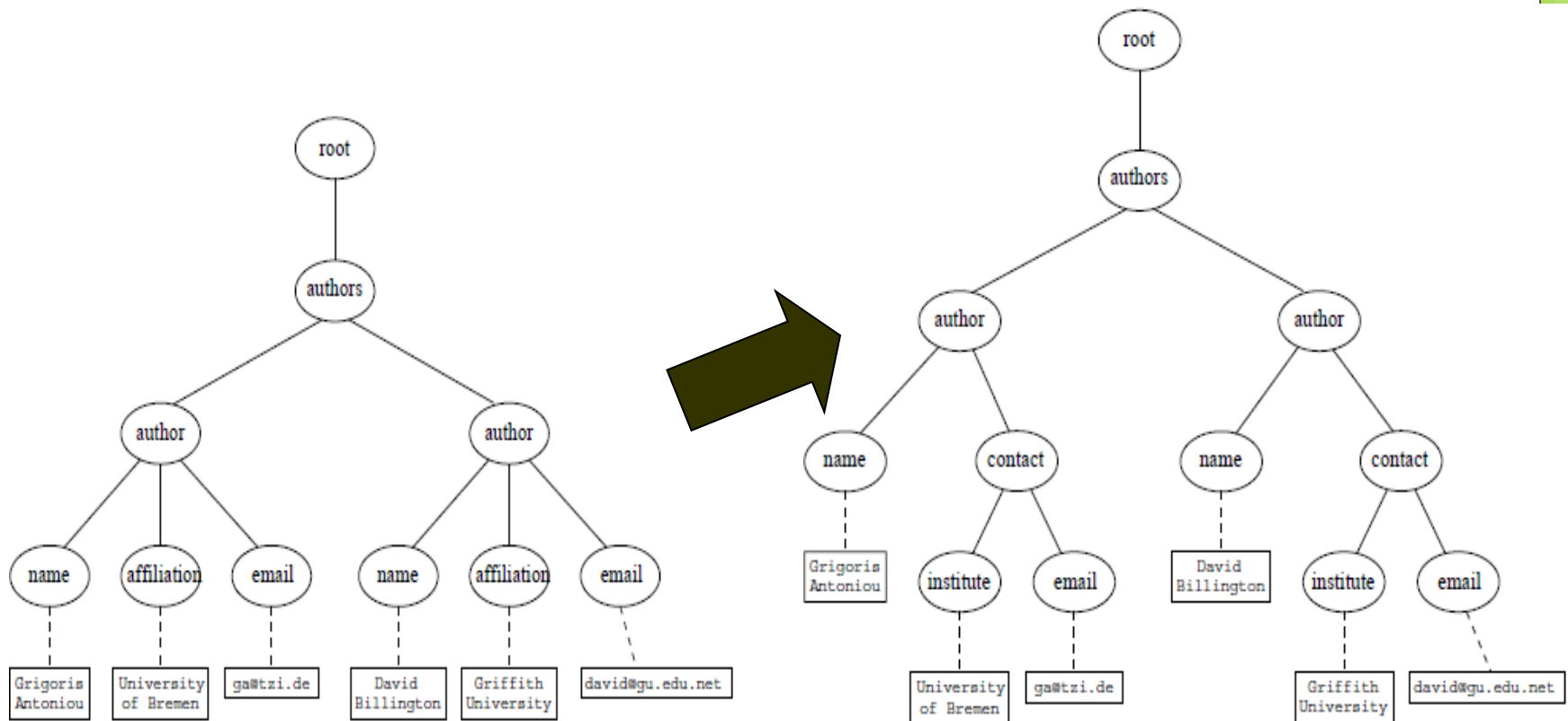
Template για XML χαρακτηριστικά (2)

- Όχι σωστά δομημένο
 - Οι ετικέτες δεν επιτρέπονται μέσα σε τιμές χαρακτηριστικών

- Σωστό παράδειγμα

```
<xsl:template match="person">  
  <person      firstname="{@firstname}"  
              lastname="{@lastname}"/>  
</xsl:template>
```

Μετασχηματισμός ενός XML σε ένα άλλο



Μετασχηματισμός ενός XML σε ένα άλλο(2)

```
<xsl:template match="/">  
  <?xml version="1.0" encoding="UTF-16"?>  
  <authors>  
    <xsl:apply-templates select="authors"/>  
  </authors>  
</xsl:template>
```

```
<xsl:template match="authors">  
  <author>  
    <xsl:apply-templates select="author"/>  
  </author>  
</xsl:template>
```

Μετασχηματισμός ενός XML σε ένα άλλο (3)

```
<xsl:template match="author">
  <name><xsl:value-of select="name" /></name>
  <contact>
    <institution>
      <xsl:value-of select="affiliation" />
    </institution>
    <email><xsl:value-of select="email" /></email>
  </contact>
</xsl:template>
```

Σύνοψη

- Η XML είναι μια μετα-γλώσσα που επιτρέπει σε χρήστες να ορίζουν σήμανση με χρήση ετικετών
- Διαχωρίζει το περιεχόμενο και τη δομή από τη μορφοποίηση
- Είναι το ντε φάκτο πρότυπο για αναπαράσταση και ανταλλαγή πληροφοριών στον Ιστό
- Υποστηρίζεται από γλώσσες επερώτησης