

Δομημένα Έγγραφα Ιστού: *XML*

Εισαγωγή στο Σημασιολογικό Ιστό

Εισαγωγή (1/6)

- Σήμερα η **HTML** (hypertext markup language) είναι η **πρότυπη γλώσσα** με την οποία δημιουργούνται **ιστοσελίδες**
- Στο **χώρο του Παγκόσμιου Ιστού**, τα **πρότυπα καθορίζονται από τον οργανισμό W3C** (World Wide Web Consortium)
 - Και **αποκαλούνται συστάσεις** (*recommendations*), σε αναγνώριση του γεγονότος ότι η επιβολή προτύπων **δεν είναι δυνατή σε ένα κατανεμημένο περιβάλλον χωρίς κεντρική αρχή**.
- Η ανάπτυξη της XML (extensible markup language) οφείλεται στις ελλείψεις της HTML

Εισαγωγή (2/6) – Παράδειγμα 1

- Έστω μία ιστοσελίδα που περιέχει πληροφορίες για ένα συγκεκριμένο βιβλίο

```
<h2>Nonmonotonic Reasoning: Context-Dependent  
Reasoning</h2>  
<i>by <b>V. Marek</b> and <b>M. Truszczyński</b></i><br>  
Springer 1993<br>  
ISBN 0387976892
```

- Μία τυπική αναπαράσταση των παραπάνω πληροφοριών σε XML μπορεί να είναι η εξής:

```
<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>
```

Εισαγωγή (3/6)

- Και οι δύο μέθοδοι αναπαράστασης *HTML* και *XML* :
 - Χρησιμοποιούν ετικέτες (**tags**), όπως `<h2>` και `</year>`
 - Είναι **γλώσσες σήμανσης**
 - Δίνουν τη δυνατότητα σε κάποιον να δημιουργήσει περιεχόμενο και να παρέχει πληροφορίες σχετικά με το ρόλο του συγκεκριμένου περιεχομένου
 - Βασίζονται σε ετικέτες
 - Οι οποίες μπορεί να είναι **ένθετες**
 - Όλες οι ετικέτες στην *XML* πρέπει να κλείνουν, ενώ στην *HTML* υπάρχουν ορισμένες ετικέτες, όπως η `<br>`, οι οποίες μπορούν να παραμείνουν «ανοιχτές»
 - Το περικλειόμενο περιεχόμενο, μαζί με τις ετικέτες «ανοίγματος» και «κλεισίματος», αναφέρεται ως στοιχείο (**element**)
- Οι άνθρωποι χρήστες μπορούν να διαβάσουν τις αναπαραστάσεις *HTML* και *XML* αρκετά εύκολα

Εισαγωγή (4/6)

- Το έγγραφο HTML δεν περιέχει δομικές πληροφορίες
 - Δηλαδή πληροφορίες σχετικά με τμήματα του εγγράφου και τις σχέσεις τους
- Το έγγραφο XML είναι πολύ πιο εύκολα προσπελάσιμο για τους υπολογιστές, επειδή περιγράφονται όλες οι πληροφορίες
- Ορισμένες μορφές των σχέσεών τους ορίζονται μέσω της διαδικασίας ένθεσης
 - Π.χ. οι **ετικέτες <author>** εμφανίζονται μέσα στις ετικέτες <book>, άρα περιγράφουν ιδιότητες του συγκεκριμένου βιβλίου
 - Ένας υπολογιστής που επεξεργάζεται το έγγραφο XML θα μπορούσε να **συμπεράνει** ότι το **στοιχείο author αναφέρεται στο στοιχείο book** που το περικλείει, σε αντιθεση με την HTML
- Η XML επιτρέπει τον **ορισμό περιορισμών στις τιμές**
 - Π.χ. το έτος πρέπει να είναι τετραψήφιος αριθμός και ο αριθμός πρέπει να είναι μικρότερος από 3000
- Η XML επιτρέπει την αναπαράσταση πληροφοριών, οι οποίες είναι προσβάσιμες και από τους υπολογιστές

Εισαγωγή (5/6)

- **Η αναπαράσταση σε HTML παρέχει περισσότερα από την αναπαράσταση σε XML:**
 - Περιγράφεται επίσης και η μορφοποίηση του εγγράφου
- **Ωστόσο, αυτή η δυνατότητα αποτελεί αδυναμία παρά πλεονέκτημα της HTML:**
 - Πρέπει να καθορίσει τη μορφοποίηση
 - Μάλιστα, η κύρια χρήση ενός εγγράφου HTML είναι η παρουσίαση πληροφοριών (πέρα από τη σύνδεση με άλλα έγγραφα)
- ***Η XML ξεχωρίζει το περιεχόμενο από τη μορφοποίηση***
 - Οι ίδιες πληροφορίες μπορούν να παρουσιαστούν με διαφορετικούς τρόπους, χωρίς να απαιτούνται πολλαπλά αντίγραφα του ίδιου περιεχομένου
 - Το περιεχόμενο μπορεί να χρησιμοποιηθεί και για σκοπούς διαφορετικούς από την παρουσίαση

Εισαγωγή (6/6) – Παράδειγμα 2

Έστω το κείμενο HTML:

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

Και η αναπαράσταση σε XML

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

- Αν συγκρίνουμε αυτό το έγγραφο **HTML** με το προηγούμενο, θα παρατηρήσουμε ότι και τα δύο χρησιμοποιούν βασικά τις ίδιες ετικέτες
 - Αυτό δεν αποτελεί έκπληξη, αφού οι ετικέτες είναι **προκαθορισμένες**
- Το δεύτερο έγγραφο **XML** χρησιμοποιεί **εντελώς διαφορετικές ετικέτες** από το πρώτο έγγραφο XML
- Οι αναπαραστάσεις σε HTML έχουν ως σκοπό την παρουσίαση πληροφοριών, άρα το σύνολο των ετικετών είναι σταθερό
- Στην XML μπορούμε να χρησιμοποιήσουμε τις πληροφορίες με διάφορους τρόπους, και ο ορισμός ενός κατάλληλου λεξιλογίου για την εφαρμογή εξαρτάται από το χρήστη
- **Η XML είναι μία μετα-γλώσσα για σήμανση**
 - **Δεν έχει σταθερό σύνολο ετικετών, αλλά επιτρέπει στους χρήστες να ορίσουν τις δικές τους ετικέτες**

Η γλώσσα XML

Ένα έγγραφο XML αποτελείται από έναν πρόλογο, έναν αριθμό στοιχείων και έναν προαιρετικό επίλογο

Πρόλογος

- Ο πρόλογος αποτελείται από μία δήλωση XML (*XML declaration*) και μία προαιρετική αναφορά σε εξωτερικά δομικά έγγραφα.
 - Ένα παράδειγμα μιας δήλωσης XML:
 - **<?xml version="1.0" encoding="UTF-16"?>**
 - Προσδιορίζει ότι το συγκεκριμένο έγγραφο είναι έγγραφο XML
 - Καθορίζει την έκδοση και την κωδικοποίηση χαρακτήρων που χρησιμοποιούνται στο συγκεκριμένο σύστημα (πχ UTF-16)
 - Καθορίζουμε, επίσης, αν το έγγραφο είναι αυτόνομο, δηλαδή αν δεν αναφέρεται σε εξωτερικά δομικά έγγραφα
 - **<?xml version="1.0" encoding="UTF-16" standalone="no"?>**
 - Μία αναφορά σε εξωτερικά δομικά έγγραφα είναι η εξής:
 - **<!DOCTYPE book SYSTEM "book.dtd">**
 - Εδώ, οι δομικές πληροφορίες βρίσκονται σε ένα τοπικό αρχείο (book.dtd)
 - Όμως, η αναφορά μπορεί να είναι και μία διεύθυνση URL

Στοιχεία

- Τα **στοιχεία XML** (XML elements) **αναπαριστούν τα «πράγματα»**
 - Όπως βιβλία, συγγραφείς και εκδότες
- Ένα **στοιχείο** αποτελείται από μία **ετικέτα ανοίγματος**, το **περιεχόμενο**, και μία **ετικέτα κλεισίματος**
 - Π.χ. **<lecturer>David Billington</lecturer>**
- Τα **ονόματα των ετικετών** μπορούν να επιλέγονται σχεδόν ελεύθερα
 - Αφού υπάρχουν ελάχιστοι περιορισμοί
- Το **περιεχόμενο** μπορεί να είναι **κείμενο ή άλλα στοιχεία ή τίποτα**

```
<lecturer>  
  <name>David </name>  
  <phone>Billington+61-3875507</phone>  
</lecturer>
```

- Αν δεν υπάρχει περιεχόμενο, τότε το στοιχείο αποκαλείται **κενό**
 - Ένα κενό στοιχείο όπως το **<lecturer></lecturer>** μπορεί να γραφεί συντετμημένα ως **<lecturer/>**

Χαρακτηριστικά

- Ένα κενό στοιχείο δεν είναι απαραίτητα άσκοπο, επειδή μπορεί να περιέχει 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"/>`
- Παράδειγμα χαρακτηριστικών για ένα μη κενό στοιχείο:

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

Οι ίδιες πληροφορίες θα μπορούσαν να γραφτούν ως εξής, αντικαθιστώντας τα χαρακτηριστικά με ένθετα στοιχεία:

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

Η επιλογή μεταξύ στοιχείων και χαρακτηριστικών είναι συχνά θέμα γούστου

Σχόλια και Οδηγίες Επεξεργασίας

- Το σχόλιο (**comment**) είναι ένα κομμάτι κειμένου που **αγνοείται από τον συντακτικό αναλυτή (parser)**
 - It has the form **<!-- This is a comment -->**
- Οι **οδηγίες επεξεργασίας** (processing instructions, PI) **παρέχουν ένα μηχανισμό μεταβίβασης πληροφοριών** σε μία εφαρμογή
- Η γενική μορφή είναι **<?.....?>**
 - Π.χ. **<?stylesheet type="text/css" href="mystyle.css"?>**
- Η οδηγία είναι: "Συνέδεσε αυτό το έγγραφο με το αρχείο CSS με όνομα mystyle.css για σκοπούς μορφοποίησης"

Σωστά Δομημένα Έγγραφα XML

- Ένα έγγραφο XML είναι σωστά δομημένο (well-formed) αν είναι συντακτικά σωστό
- Μερικοί συντακτικοί κανόνες είναι οι εξής:
 - Υπάρχει **μόνο ένα εξωτερικό στοιχείο** στο έγγραφο
 - Καλείται **στοιχείο-ρίζα**
 - Κάθε **στοιχείο** περιέχει μία **ετικέτα ανοίγματος** και μία **αντίστοιχη ετικέτα κλεισίματος**
 - Οι **ετικέτες** δεν πρέπει να **αλληλεπικαλύπτονται**
 - Π.χ. **<author><name>Lee Hong</author></name> <!--lathos-->**
 - Τα **χαρακτηριστικά (attributes)** μέσα σ' ένα στοιχείο έχουν **μοναδικά ονόματα**
 - Τα **ονόματα των στοιχείων και των ετικετών** πρέπει να είναι **επιτρεπτά**

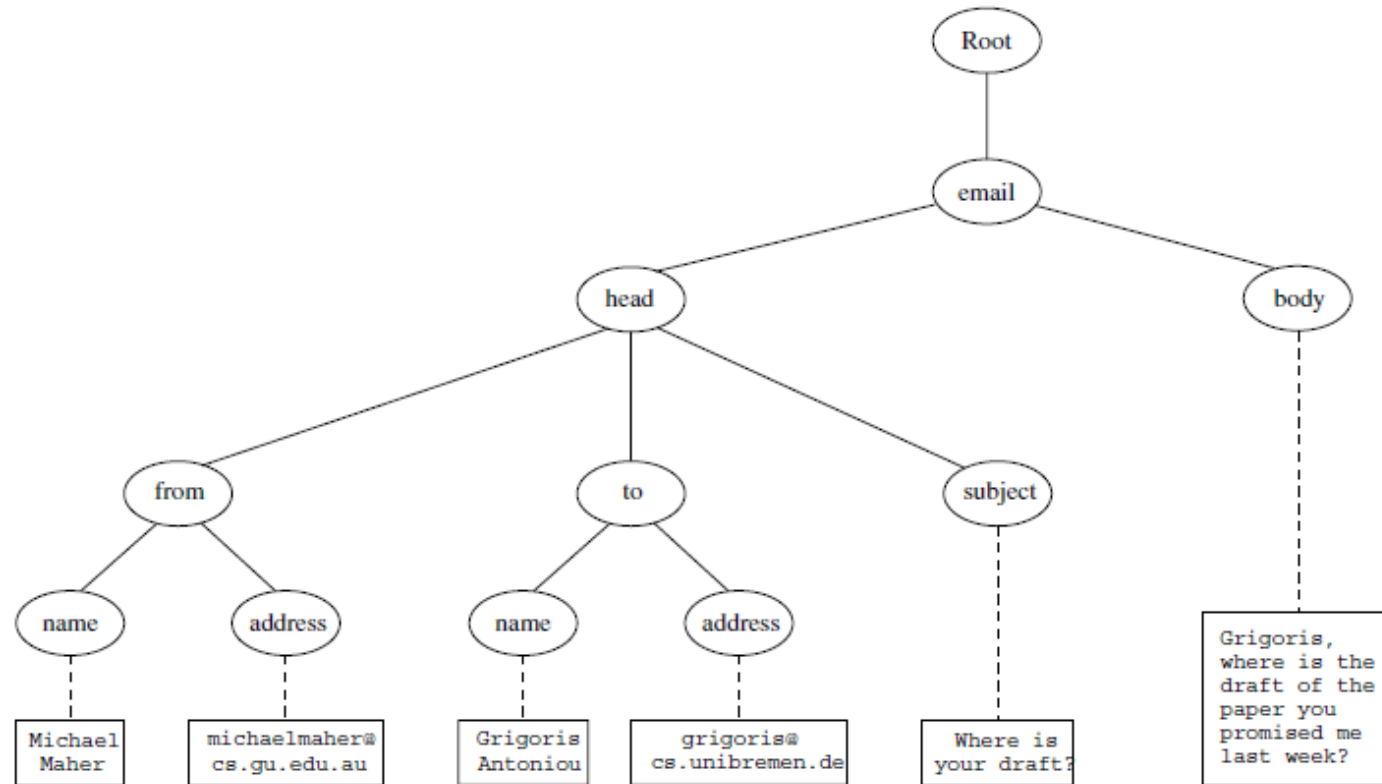
Δενδρικό Μοντέλο των Εγγράφων XML (1/3)

- Είναι εφικτό να αναπαραστήσουμε σωστά δομημένα έγγραφα XML ως δένδρα
 - Συνεπώς, τα δένδρα παρέχουν ένα τυπικό μοντέλο δεδομένων για την XML
- Για παράδειγμα, θεωρίστε το παρακάτω έγγραφο:

```
<?xml version="1.0" encoding="UTF-16"?>
<!DOCTYPE email SYSTEM "email.dtd">
<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 των Εγγράφων XML (2/3)

- Η δενδρική αναπαράσταση του παραπάνω εγγράφου XML:



Δενδρικό Μοντέλο των Εγγράφων XML (3/3)

- **Είναι ένα διατεταγμένο δένδρο με ετικέτες:**
 - Υπάρχει ακριβώς μία ρίζα
 - Δεν υπάρχουν κύκλοι
 - Κάθε κόμβος έχει ακριβώς ένα γονιό, εκτός από τη ρίζα
 - Κάθε κόμβος περιλαμβάνει μια ετικέτα
 - Η σειρά των στοιχείων είναι σημαντική

Δόμηση (DTD)

Ένα έγγραφο XML είναι σωστά δομημένο όταν ακολουθεί ορισμένους συντακτικούς κανόνες

DTD (1/9)

- Υπάρχουν δύο τρόποι ορισμού της δομής των εγγράφων XML
 - **DTD**: που αποτελούν τον παλαιότερο και πιο περιορισμένο τρόπο
 - **Γλώσσα XML Schema**: που προσφέρει εκτεταμένες δυνατότητες, κυρίως όσον αφορά τον ορισμό τύπων δεδομένων
- **Εξωτερικά και Εσωτερικά DTD**
 - Τα **συστατικά στοιχεία ενός DTD** μπορούν να οριστούν
 - Σε ξεχωριστό αρχείο (*εξωτερικό DTD*)
 - Ή μέσα στο ίδιο το έγγραφο XML (*εσωτερικό DTD*)
 - Συνήθως είναι προτιμότερο να **χρησιμοποιούμε εξωτερικά DTD**
 - Επειδή οι ορισμοί τους μπορούν να χρησιμοποιηθούν **σε πολλά έγγραφα**

DTD (2/9)

- **Στοιχεία**
- Θεωρείστε το στοιχείο:
- Ένα **DTD** γι' αυτόν τον τύπο στοιχείων θα ήταν:

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

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

- Το νόημα αυτού του DTD είναι το εξής:
 - Ένα **στοιχείο τύπου lecturer** **περιέχει** ένα στοιχείο **name** και ένα στοιχείο **phone**, με αυτήν τη σειρά
 - Τα στοιχεία τύπου **name** και **phone** μπορούν να **έχουν οποιοδήποτε περιεχόμενο**
 - Στα DTD, ο τύπος **#PCDATA** είναι ο μοναδικός ατομικός τύπος για στοιχεία

DTD (3/9)

- Χαρακτηριστικά
- Θεωρείστε το
- στοιχείο:

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

- Ένα DTD γι' αυτό το στοιχείο θα μπορούσε να είναι το ακόλουθο:

```
<!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>
```

DTD (4/9)

- Τελεστές πληθικότητας (*cardinality operators*):
 - **?**: εμφανίζεται καμία ή μία φορά
 - *****: εμφανίζεται καμία ή περισσότερες φορές
 - **+**: εμφανίζεται μία ή περισσότερες φορές
 - Η μη ύπαρξη πληθικότητας σημαίνει ακριβώς μία εμφάνιση
- Ορισμός χαρακτηριστικών:
 - **Λίστα χαρακτηριστικών** (*ATTLIST=attribute list*)
 - Το **!ATTLIST** συνδέει ένα όνομα element με μια λίστα από ορισμούς χαρακτηριστικών, όπου κάθε ορισμός είναι μια τριάδα (Όνομα - Τύπος – Περιορισμός , π.χ. **customer** CDATA **#REQUIRED**).

DTD (5/9)

```
<organization>
  <employee id="E101" name="Dimitris Papas"/>
  <employee id="E102" name="Maria Karra"/>
  <project name="New Website" team_members="E101 E102"/>
  <project name="Mobile App" team_members="E101"/>
</organization>
```

- **Τύποι χαρακτηριστικών**
- Οι **τύποι χαρακτηριστικών** (attribute types) **είναι παρόμοιοι** με τους **προκαθορισμένους τύπους δεδομένων**, αλλά ο αριθμός τους είναι πολύ περιορισμένος
- Οι σημαντικότεροι τύποι χαρακτηριστικών είναι:
 - **CDATA**, ένα **αλφαριθμητικό** (ακολουθία χαρακτήρων)
 - **ID**, ένα όνομα που είναι **μοναδικό** σε ολόκληρο το έγγραφο XML
 - **IDREF**, μια **αναφορά** σε κάποιο άλλο στοιχείο με χαρακτηριστικό ID, το οποίο έχει την ίδια τιμή με το χαρακτηριστικό IDREF
 - **IDREFS**, μία σειρά από IDREFs
 - **(v1 / . . . / vn)**, μία **απαρίθμηση όλων των δυνατών τιμών**

```
<!ELEMENT product EMPTY> <!ATTLIST product itemNo ID #REQUIRED status (InStock | OutOfStock ) "InStock">
```

DTD (7/9)

- Αναφορές
- Παράδειγμα της χρήσης των τύπων **IDREF** και **IDREFS**
- Πρώτα βλέπουμε 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>
```

- Ένα στοιχείο XML που ακολουθεί το παραπάνω DTD είναι το εξής:

```
<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>
```

DTD (6/9)

- Τύποι τιμών
- Υπάρχουν τέσσερις τύποι τιμών:
 - **#REQUIRED**
 - Το χαρακτηριστικό **πρέπει να υπάρχει σε κάθε εμφάνιση του τύπου** του στοιχείου στο έγγραφο XML
 - Στο προηγούμενο παράδειγμα, τα *itemNo* (κωδ. προϊόντος) και *quantity* (ποσότητα) πρέπει πάντα να εμφανίζονται μέσα σε ένα στοιχείο τύπου *item*
 - **#IMPLIED**
 - Η εμφάνισή του **είναι προαιρετική** (π.χ. τα σχόλια)
 - **#FIXED "value"**
 - Κάθε στοιχείο πρέπει να έχει αυτό το χαρακτηριστικό, το οποίο έχει πάντα την τιμή που δίνεται μετά από τον τύπο #FIXED στο DTD
 - **"value " π.χ "Greece "**
 - Ορίζει την προεπιλεγμένη τιμή του χαρακτηριστικού
 - Αν εμφανιστεί μία συγκεκριμένη τιμή στο έγγραφο XML, τότε υπερισχύει έναντι της προεπιλεγμένης τιμής

```
<!ATTLIST order orderNo CDATA #REQUIRED protocol_version CDATA #FIXED "1.0" >  
<order orderNo="23456" protocol_version="1.0">
```

```
<!ATTLIST item itemNo CDATA #REQUIRED shipping_country CDATA "Greece">  
<item itemNo="c817" shipping_country="Germany"/>
```


DTD (8/9)

- **Οντότητες XML**
- Μία οντότητα XML (XML entity) μπορεί να παίξει πολλούς ρόλους, όπως να χρησιμεύσει ως
 - Δεσμευτικό θέσης (**placeholder** © -> ©) επαναλαμβανόμενων χαρακτήρων
 - Παραπέμπει σε εξωτερικά δεδομένα(π.χ. **σε αλλα XML αρχεία**)
- Π.χ. έστω ότι κάποιο έγγραφο έχει αρκετές επισημάνσεις για τα πνευματικά δικαιώματα (copyright notices) που αναφέρονται στο τρέχον έτος
 - Σε αυτήν την περίπτωση έχει νόημα η δήλωση της οντότητας **<!ENTITY thisyear "2025">**
 - Στη συνέχεια μπορούμε να χρησιμοποιήσουμε την αναφορά στην οντότητα **&this year**
 - Σε οποιοδήποτε σημείο χρειάζεται να περιληφθεί το τρέχον έτος
 - **Με αυτόν τον τρόπο, η ενημέρωση της τιμής του έτους σε «2024» για ολόκληρο το έγγραφο θα απαιτεί μόνο την αλλαγή της δήλωσης της αντίστοιχης οντότητας**

DTD (9/9)

```
<!ELEMENT email (head,body)>
<!ELEMENT head (from,to+,cc*,subject)>
<!ELEMENT from EMPTY>
<!ATTLIST from name CDATA #IMPLIED address CDATA #REQUIRED>
<!ELEMENT to EMPTY>
<!ATTLIST to name CDATA #IMPLIED address CDATA #REQUIRED>
<!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>
```

- Τελικό Παράδειγμα
- Στο τελευταίο παράδειγμα παρέχουμε το DTD για το στοιχείο email:

Ας δούμε κάποια ενδιαφέροντα τμήματα του DTD:

- Ένα στοιχείο **head (κεφαλίδα)** περιέχει ένα στοιχείο **from (από)**, τουλάχιστον ένα στοιχείο **to (προς)**, κανένα ή περισσότερα στοιχεία **cc (κοινοποίηση)**, και ένα στοιχείο **subject (θέμα)** με τη συγκεκριμένη σειρά
- Στα στοιχεία **from, to, και cc** δεν απαιτείται το χαρακτηριστικό **name**, όμως το χαρακτηριστικό **address** είναι πάντα απαραίτητο
- Ένα στοιχείο **body (σώμα)** περιέχει ένα στοιχείο text (κείμενο), το οποίο πιθανότατα ακολουθείται από έναν αριθμό στοιχείων attachment (συννημμένο)
- Το χαρακτηριστικό **encoding (κωδικοποίηση)** ενός στοιχείου **attachment** πρέπει να έχει είτε την τιμή "mime" ή την τιμή "binhex", με την **πρώτη** να είναι **προεπιλεγμένη**

XML Schema

XML Schema (1/10)

- Η γλώσσα **XML Schema** αποτελεί μία σαφώς **πλουσιότερη γλώσσα** για τον **ορισμό της δομής των εγγράφων XML**
- Ένα από τα χαρακτηριστικά της είναι ότι η **σύνταξή της βασίζεται στην ίδια την XML**
 - Αυτό βελτιώνει σημαντικά την **αναγνωσιμότητα**
 - Επίσης επιτρέπει την **επαναχρησιμοποίηση** της τεχνολογίας σε μεγάλο βαθμό
- **Δεν είναι πλέον απαραίτητη η δημιουργία ξεχωριστών συντακτικών αναλυτών, επεξεργαστών, εργαλείων στοίχισης κώδικα, κλπ** προκειμένου να επιτευχθεί μία ξεχωριστή σύνταξη
 - Η XML είναι αρκετή
- Επίσης υπάρχει η δυνατότητα **επαναχρησιμοποίησης και βελτίωσης των σχημάτων (schemas)**
 - Η XML Schema επιτρέπει τον **ορισμό νέων τύπων με την επέκταση ή τον περιορισμό των ήδη υπαρχόντων τύπων**
 - Σε συνδυασμό με τη βασισμένη σε XML σύνταξη, αυτή η δυνατότητα **επιτρέπει τη δημιουργία σχημάτων από άλλα σχήματα, μειώνοντας έτσι το φόρτο εργασίας**
- Η **XML Schema** παρέχει **ένα εξελιγμένο σύνολο τύπων δεδομένων** που μπορούν να χρησιμοποιούνται σε έγγραφα XML

Επέκταση ενός συνθετικού τύπου

Περιορισμός ενός απλού τύπου (simple type)

```
<xsd:complexType name="Person">
  <xsd:sequence>
    <xsd:element name="firstName" type="xsd:string"/>
    <xsd:element name="lastName" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Employee">
  <xsd:complexContent>
    <xsd:extension base="Person">
      <xsd:sequence>
        <xsd:element name="employeeId" type="xsd:string"/>
        <xsd:element name="title" type="xsd:string"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

```
<employeeData >
  <firstName>John</firstName>
  <lastName>Smith</lastName>

  <employeeId>E452</employeeId>
  <title>SeniorDeveloper</title>
</employeeData>
```

```
<xsd:simpleType name="SmallInteger">
  <xsd:restriction base="xsd:integer">
    <xsd:minInclusive value="1"/>
    <xsd:maxInclusive value="100"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:element name="Person">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Name" type="xsd:string"/>
      <xsd:element name="Age" type="SmallInteger"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

```
<person>
  <Name>Maria</Name>
  <Age>27</Age>
</person>
```

XML Schema (2/10)

- Ένα σχήμα XML είναι ένα στοιχείο με ετικέτα ανοίγματος:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema" version="1.0">
```

- Το στοιχείο χρησιμοποιεί το σχήμα της γλώσσας XML Schema που βρίσκεται στον ιστότοπο του οργανισμού W3C
 - Αποτελεί τη βάση επάνω στην οποία μπορούν να χτιστούν νέα σχήματα
 - Το πρόθεμα *xsd* δηλώνει το χώρο ονομάτων αυτού του σχήματος
 - Αν το πρόθεμα μέσα στο χαρακτηριστικό *xmlns* παραλειφθεί, τότε χρησιμοποιούνται εξ' ορισμού στοιχεία από τον ίδιο χώρο ονομάτων:
- ```
<schema xmlns="http://www.w3.org/2000/10/XMLSchema" version="1.0">
```
- Τα πιο σημαντικά περιεχόμενα των στοιχείων τύπου *schema* είναι οι **ορισμοί των τύπων για στοιχεία και χαρακτηριστικά** (simple ή complex types), που ορίζονται με τη **χρήση τύπων δεδομένων**

# XML Schema (3/10)

- Τύποι στοιχείων
- Η σύνταξη των τύπων στοιχείων (**element types**) είναι  
***<element name=". . ." />***
- Μπορούν να έχουν ένα αριθμό προαιρετικών χαρακτηριστικών, όπως τύπους, ***type=". . ."*** ή περιορισμούς πληθικότητας (***type="PCDATA", type="integer", type=complexType, simpleType***)
  - ***minOccurs="1"***, όπου μπορεί να είναι οποιοσδήποτε φυσικός αριθμός (συμπεριλαμβανομένου και του μηδενός)
  - ***maxOccurs="12"***, μπορεί να είναι οποιοσδήποτε φυσικός αριθμός (συμπεριλαμβανομένου και του μηδενός) ή να είναι unbounded
- Οι περιορισμοί ***minOccurs*** και ***maxOccurs*** αποτελούν γενικεύσεις των τελεστών πληθικότητας ***?, \*, +***, που παρέχονται από τα DTD
- Όταν δεν δίνονται ρητά περιορισμοί πληθικότητας, παίρνουν την προεπιλεγμένη τιμή **1**

# XML Schema (4/10)

- Τύποι χαρακτηριστικών
- Οι τύποι χαρακτηριστικών (**attribute types**) έχουν τη σύνταξη  
***<attribute name="..." />***
- Μπορούν να περιλαμβάνουν έναν αριθμό προαιρετικών χαρακτηριστικών, όπως τύπους ***type="..."*** (***type="CDATA"***, ***type="integer"***, ***type="date"***, ***type="ID"***)
- Το use (αντιστοιχεί στους τύπους #OPTIONAL ή #IMPLIED των DTD)
  - ***use="optional"***, όπου αντί για ***optional*** (προαιρετικό) μπορεί να είναι ***required*** (απαραίτητο) ή ***prohibited*** (απαγορευμένο)
- Ή μια προεπιλεγμένη (***default***) τιμή (αντιστοιχεί στον τύπο #FIXED και τις προεπιλεγμένες τιμές των DTD)
- Ακολουθούν μερικά παραδείγματα:
  - ***<attribute name="id" type="ID" use="required" />***
  - ***<attribute name="speaks" type="Language" use="optional" default="en" />*** ή ***default="1.0"***



# XML Schema (5/10)

- Τύποι δεδομένων
- Η XML Schema παρέχει μεγάλες δυνατότητες για τον ορισμό τύπων δεδομένων
- Ακολουθούν μερικοί από αυτούς:
  - Αριθμητικοί τύποι δεδομένων, στους οποίους περιλαμβάνονται οι **integer, short, Byte, long, float, ktl**
  - Αλφαριθμητικοί τύποι δεδομένων, συμπεριλαμβανομένων των **string, ID, IDREF, CDATA, language**
  - Τύποι δεδομένων ημερομηνίας και ώρας, στους οποίους περιλαμβάνονται οι **time, date, gMonth, gYear**

# XML Schema (6/10)

- ... Τύποι δεδομένων
- **Υπάρχουν επίσης τύποι δεδομένων ορισμένοι από το χρήστη**, οι οποίοι περιλαμβάνουν
  - **Απλούς τύπους δεδομένων**, που δεν μπορούν να χρησιμοποιήσουν στοιχεία ή χαρακτηριστικά
  - **Σύνθετους τύπους δεδομένων**, που μπορούν να κάνουν χρήση στοιχείων και χαρακτηριστικών
    - Οι σύνθετοι τύποι δεδομένων προκύπτουν από ήδη υπάρχοντες τύπους δεδομένων, με τον ορισμό κάποιων χαρακτηριστικών (αν υπάρχουν) και με τη χρήση των
      - **sequence**, μία ακολουθία από υπάρχοντα στοιχεία τύπων δεδομένων, η παρουσία των οποίων είναι σημαντική με προκαθορισμένη σειρά **<xsd:sequence>**
      - **all**, μία συλλογή στοιχείων που πρέπει να εμφανίζονται, αλλά η σειρά των οποίων δεν είναι σημαντική **<xsd:all>**
      - **choice**, μία συλλογή από στοιχεία, ένα από τα οποία θα επιλεγεί **<xsd:choice>**

# XML Schema (7/10)

- Ακολουθεί ένα παράδειγμα:

```
<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>
```

- Το νόημα είναι ότι ένα **στοιχείο** σε ένα έγγραφο XML, το οποίο δηλώνεται ότι είναι **τύπου *lecturerType*** μπορεί να έχει ένα χαρακτηριστικό ***title***
- Μπορεί επίσης να **περιέχει** **οποιονδήποτε αριθμό στοιχείων *firstname***
- και πρέπει να περιέχει **ακριβώς ένα στοιχείο *lastname***

# XML Schema (8/10)

- Επέκταση τύπων δεδομένων
- Οι ήδη υπάρχοντες τύποι δεδομένων μπορούν να επεκταθούν με νέα στοιχεία ή χαρακτηριστικά (π.χ. επεκτείνουμε τον τύπο δεδομένων *lecturer*)

```
<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>
```

- Στο παράδειγμά μας, ο τύπος *lecturerType* επεκτείνεται με ένα στοιχείο *email* και ένα χαρακτηριστικό *rank*.

```
<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>
```

```
<Lecturer rank="Professor" title="Dr.">
 <firstname>Anna</firstname>
 <firstname>Maria</firstname>
 <lastname>Petridou</lastname>
 <email>anna.p@university.edu</email>
</Lecturer>
```

# XML Schema (9/10)

- Περιορισμός τύπων δεδομένων
- Ένας τύπος δεδομένων μπορεί να περιοριστεί με την προσθήκη περιορισμών σε συγκεκριμένες τιμές
- Π.χ. μπορούν να προστεθούν νέα χαρακτηριστικά *type* και *use* ή να γίνουν πιο αυστηροί οι αριθμητικοί περιορισμοί των *minOccurs* και *maxOccurs*
- Η ακόλουθη σχέση είναι ιεραρχική:
  - Ικανοποιούν τουλάχιστον τους περιορισμούς του αρχικού τύπου καθώς και ορισμένους νέους περιορισμούς
- Π.χ. περιορίζουμε τον τύπο δεδομένων *lecturer* ως εξής:

```
<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>
```

# XML Schema (10/10)

- Οι απλοί τύποι δεδομένων μπορούν ακόμη να οριστούν με τον περιορισμό
- Π.χ. μπορούμε να ορίσουμε έναν τύπο **dayOfMonth** ο οποίος παίρνει τιμές **1 έως 31** όπως φαίνεται παρακάτω:

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

- Επίσης, είναι δυνατός ο ορισμός ενός τύπου δεδομένων με την **απαρίθμηση όλων των δυνατών τύπων**
- Π.χ. μπορούμε να ορίσουμε έναν τύπο δεδομένων **dayOfWeek** ως εξής:

```
<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>
```