

Εισαγωγή στη γλώσσα XML

Μανόλης Γεργατσούλης

Χρήστος Παπαθεοδώρου

Ομάδα Βάσεων Δεδομένων και Πληροφοριακών
Συστημάτων, Τμήμα Αρχειονομίας – Βιβλιοθηκονομίας

Ιόνιο Πανεπιστήμιο

HTML

- Απλή γλώσσα σημειοθέτησης (*markup language*)
- Το κείμενο εμπλουτίζεται με “εντολές” της γλώσσας οι οποίες ονομάζονται ετικέτες (*tags*), οι οποίες συνήθως αποτελούνται από μια ετικέτα αρχής (*start tag*) και μια ετικέτα τέλους (*end tag*).
- Με την HTML περιγράφουμε πως θέλουμε να παρουσιάζεται η πληροφορία ενός κειμένου.

Παράδειγμα HTML: Λίστα Βιβλίων

<HTML>

<BODY>

Fiction:

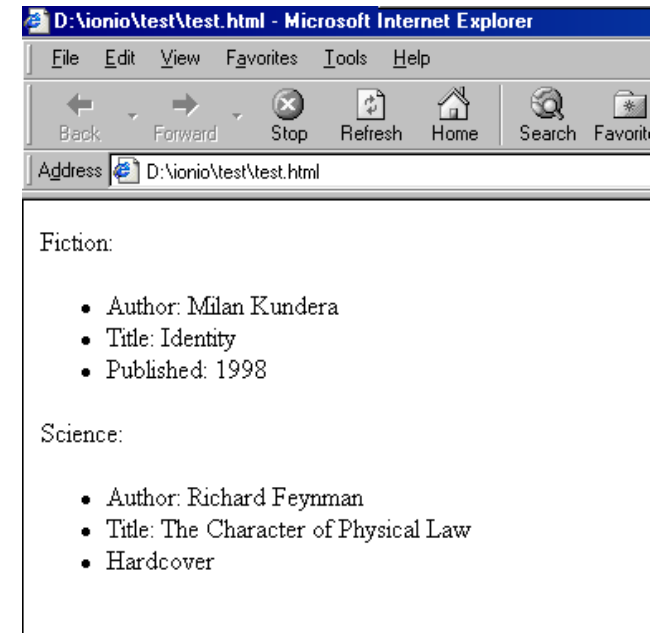
Author: Milan Kundera
 Title: Identity
 Published: 1998

Science:

Author: Richard Feynman
 Title: The Character of Physical Law
 Hardcover

</BODY>

</HTML>



Πέρα από την HTML: XML

- *e*X*tensible* M*arkup* L*anguage* (XML) αποτελεί μια εξαιρετικά απλή διάλεκτο της γλώσσας *S*andard *G*eneralized M*arkup* L*anguage* (SGML), η οποία αναπτύχθηκε με στόχο να διευκολύνει το χειρισμό, επεξεργασία, διακίνηση και αποθήκευση τεκμηρίων στον *Παγκόσμιο Ιστό* (web).
- Συνδυασμός SGML και HTML: Η ισχύς της SGML με την απλότητα της HTML.
- Επιτρέπει τον ορισμό νέων γλωσσών σημειοθέτησης, με τη βοήθεια δηλώσεων τύπων εγγράφων (*Document Type Declarations*) (DTDs).
- Τεχνικά εγχειρίδια:
 - «Extensible Markup Language (XML) 1.0 (Second Edition)» βρίσκεται στη διεύθυνση: <http://www.w3.org/TR/REC-xml>

Πως ξεκινά ένα XML τεκμήριο

Ένα απλό XML τεκμήριο:

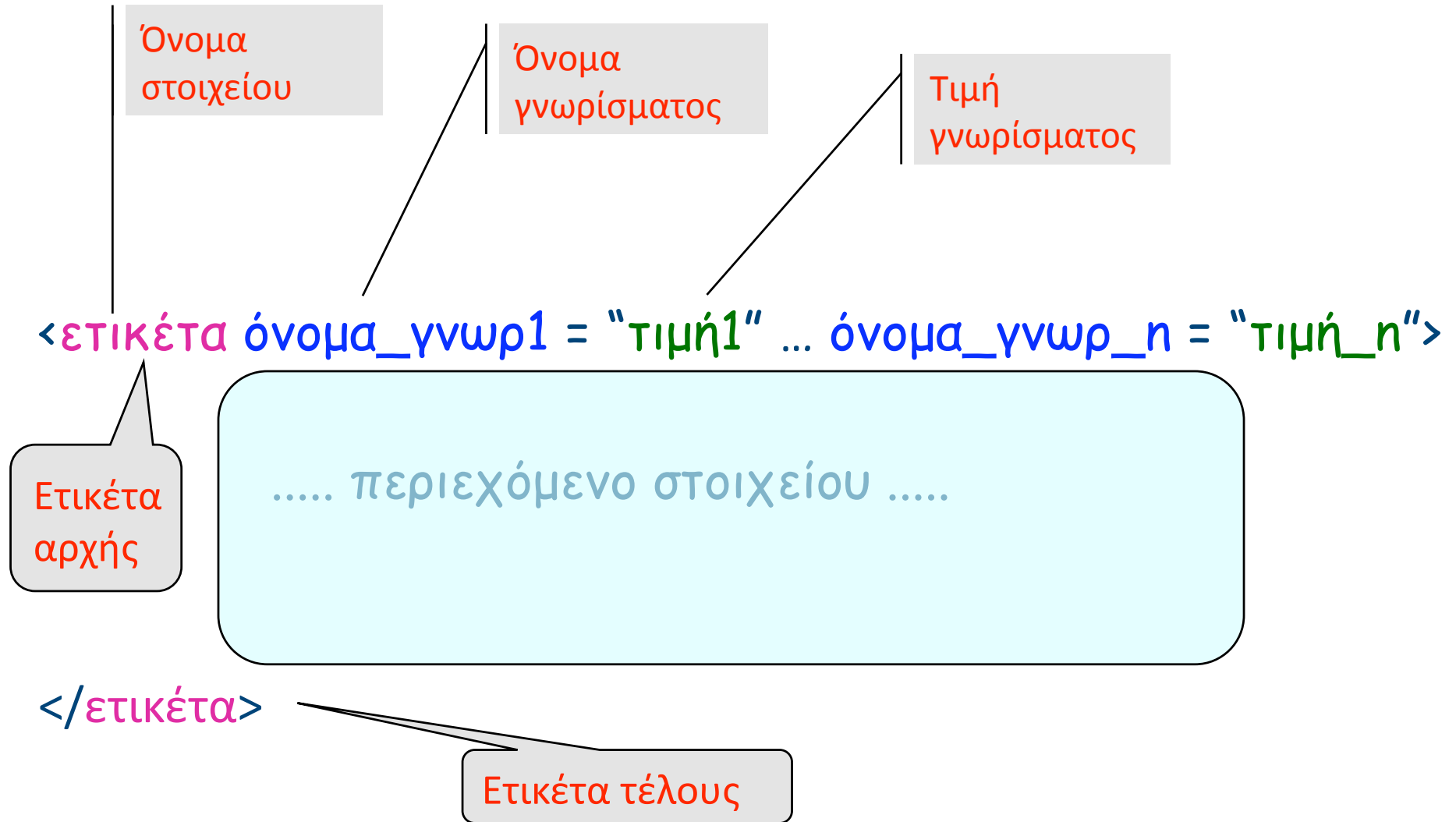
```
<?xml version="1.0"?>
```

Δήλωση XML

```
<greeting>Hello, world!</greeting>
```

Ένα απλό στοιχείο της XML

XML: Στοιχεία και γνωρίσματα



Παράδειγμα XML τεκμηρίου

Βιβλιογραφία

- S. Abiteboul, P. Buneman, D. Suciu “Data on the Web: From Relations to Semistructured Data and XML” Morgan Kaufmann Publishers, 2000.
- Norman Walsh “A Guide to XML” World Wide Web Journal, Vol. 2, Issue 4, 1997, pages 97-107.

<bibliography>

<book>

<author>S. Abiteboul</author>

<author>P. Buneman</author>

<author>D. Suciu</author>

<title>Data on the Web: From Relations to Semistructured Data and XML</title>

<publisher>Morgan Kaufmann Publishers</publisher>

<year>2000</year>

</book>

<article>

<author>Norman Walsh</author>

<title>A Guide to XML</title>

<journal>World Wide Web Journal</journal>

<volume>2</volume>

<issue>4</issue>

<year>1997</year>

<pages>97-107</pages>

</article>

</bibliography>

XML: Βασικά Δομικά Στοιχεία

- Στοιχεία (elements).
 - Οι βασικές δομικές μονάδες της XML.
 - Ετικέτα αρχής, ετικέτα τέλους.
 - Πρέπει να είναι κατάλληλα εμφωλευμένα.
- Τα στοιχεία μπορούν να διαθέτουν γνωρίσματα (attributes) τα οποία παρέχουν επιπλέον πληροφορία αναφορικά με τα στοιχεία.
- Οντότητες: όπως οι μακροεντολές, αναπαριστούν ένα συχνά εμφανιζόμενο κείμενο.
- Σχόλια.
- Οδηγίες επεξεργασίας (processing instructions): αναπαριστούν οδηγίες για εφαρμογές.
- Δηλώσεις τύπων εγγράφων (Document type declarations) (DTDs).

Απλά και Σύνθετα Στοιχεία

- Ένα **απλό στοιχείο** (έχει για περιεχόμενο απλό κείμενο):

<φοιτητής> Νίκος Νικολάου </φοιτητής>

- Ένα **σύνθετο στοιχείο** (περιλαμβάνει άλλα στοιχεία):

<φοιτητής>
<όνομα> Νίκος </όνομα>
<επώνυμο> Νικολάου </επώνυμο>
</φοιτητής>

Περιεχόμενο
στοιχείων

Σύνθετα Στοιχεία με Ανάμικτο Περιεχόμενο

- Στοιχείο με **ανάμικτο περιεχόμενο**:

<φοιτητής>

Το όνομα του φοιτητή είναι <όνομα>Νίκος</όνομα>
ενώ το επώνυμο του είναι <επώνυμο>Νικολάου</
επώνυμο>

</φοιτητής>

Ανάμικτο
περιεχόμενο

Καλά Διαμορφωμένο XML Τεκμήριο

- Για να είναι ένα XML τεκμήριο **καλά διαμορφωμένο** (well-formed) πρέπει να υπακούει στους κανόνες σύνταξης της XML:
 - Οι ετικέτες του τεκμηρίου πρέπει να είναι ισορροπημένες: σε κάθε ετικέτα αρχής πρέπει να αντιστοιχεί μια ετικέτα τέλους η οποία να βρίσκεται μετά την ετικέτα αρχής μέσα στο τεκμήριο.
 - Αν μια ετικέτα αρχής E1 εμφανίζεται νωρίτερα από μια ετικέτα αρχής E2, τότε η ετικέτα τέλους που αντιστοιχεί στην E1 εμφανίζεται αργότερα από την ετικέτα τέλους που αντιστοιχεί στην E2. Επομένως, οι ετικέτες τέλους πρέπει να εμφανίζονται με την ανάστροφη σειρά από αυτήν που εμφανίζονται οι αντίστοιχες ετικέτες αρχής.

Καλά Διαμορφωμένο XML Τεκμήριο

- Καλά Διαμορφωμένο XML Τεκμήριο:

<φοιτητής>


<όνομα> Νίκος </όνομα>

<επώνυμο> Νικολάου </επώνυμο>

</φοιτητής>


- Σειρά εμφάνισης ετικετών:

<φοιτητής><όνομα></όνομα><επώνυμο></επώνυμο></φοιτητής>



- Λανθασμένη σειρά εμφάνισης ετικετών:

<φοιτητής><όνομα></όνομα><επώνυμο></φοιτητής></επώνυμο>



Ένα μεγαλύτερο παράδειγμα XML τεκμηρίου

- Αναπαράσταση λίστας φοιτητών του TAB σε μορφή XML τεκμηρίου:

<TAB>

<φοιτητής>

<όνομα> Νίκος </όνομα>

<επώνυμο> Νικολάου </επώνυμο>

</φοιτητής>

<φοιτητής>

<όνομα> Πέτρος </όνομα>

<επώνυμο> Πέτρου </επώνυμο>

</φοιτητής>

<φοιτητής>

<όνομα> Μίνα </όνομα>

<επώνυμο> Μίνου </επώνυμο>

</φοιτητής>

...

</TAB>

Κενά στοιχεία στην XML

- Η σύνταξη της XML επιτρέπει *κενά στοιχεία* (empty elements) δηλαδή στοιχεία με κενό περιεχόμενο.
- Τα κενά στοιχεία έχουν τη μορφή:
`<ετικέτα></ετικέτα>`
- Παρέχεται και η ακόλουθη συντομογραφία για τη σύνταξη κενών στοιχείων:
`<ετικέτα/>`

Γνωρίσματα στην XML

- Ένα στοιχείο της XML είναι δυνατό να διαθέτει ένα σύνολο από *γνωρίσματα* (attributes).
- Τα γνωρίσματα ορίζονται σαν ζεύγη *ονομάτων – τιμών*.
- Τα γνωρίσματα τοποθετούνται στην ετικέτα αρχής του στοιχείο στο οποίο αναφέρονται.
- Στο παρακάτω παράδειγμα το γνώρισμα με όνομα AM χρησιμοποιείται για να αποτυπωθεί ο αριθμός μητρώου του φοιτητή:

```
<φοιτητής AM = "12345">  
  <όνομα> Νίκος </όνομα>  
  <επώνυμο> Νικολάου </επώνυμο>  
</φοιτητής>
```

Γνώρισμα

Γνωρίσματα στην XML (συνέχεια)

- Οι τιμές των γνωρισμάτων περικλείονται ανάμεσα σε απλά ή διπλά εισαγωγικά.
- Ένα στοιχείο είναι δυνατόν να διαθέτει περισσότερα του ενός γνωρίσματα.

```
<book isbn="1-55860-622-X" language="English">  
  <title> Data on the Web </title>  
  <price currency = "USD"> 100 </price>  
</book>
```
- Ενώ ένα στοιχείο μπορεί να διαθέτει υποστοιχεία με το ίδιο όνομα, δεν επιτρέπεται σε περισσότερα του ενός γνωρίσματα του να έχουν το ίδιο όνομα.
- Η σειρά με εμφάνισης των γνωρισμάτων ενός στοιχείου δεν παίζει κανένα ρόλο σε αντίθεση με τη σειρά εμφάνισης των στοιχείων που είναι σημαντική.

Σχόλια

- Τα *σχόλια* (comments) επιτρέπονται οπουδήποτε εκτός από το εσωτερικό των ετικετών.
- Ένα σχόλιο ξεκινά με το `<!--` και τελειώνει με το `-->`.
 - Παράδειγμα:
`<!-- Αυτό είναι ένα σχόλιο -->`
- Τα σχόλια τοποθετούνται για να κάνουν το τεκμήριο ευανάγνωστο από τον άνθρωπο.

Οδηγίες Επεξεργασίας

- Οι *οδηγίες επεξεργασίας* (Processing Instructions) PI επιτρέπουν σε ένα XML τεκμήριο να περιέχει οδηγίες που απευθύνονται σε προγράμματα εφαρμογών.
- Μια οδηγία επεξεργασίας περιλαμβάνει το όνομα μιας εφαρμογής στην οποία απευθύνεται, ακολουθούμενο από πληροφορίες (οδηγίες επεξεργασίας, παραμέτρους κ.λ.π.) οι οποίες θέλουμε να περάσουν στην εφαρμογή.
- Παράδειγμα: Η παρακάτω οδηγία επεξεργασίας απευθύνεται στην εφαρμογή xml-stylesheet:
`<?xml-stylesheet href="book.css" type="text/css"?>`

CDATA

- Ένα τμήμα CDATA ξεκινά με `<![CDATA[` και τελειώνει με `]]>`.
- Το περιβάλλον CDATA χρησιμοποιείται όταν θέλουμε να συμπεριλαμβάνουμε ετικέτες και άλλα ειδικά σύμβολα τα οποία θέλουμε να εκληφθούν απλά σαν ακολουθίες χαρακτήρων (συμβολοσειρές).
- Παράδειγμα: Το ακόλουθο είναι ένα συντακτικά σωστό τμήμα XML τεκμηρίου:
`<![CDATA[<αρχή> αυτό είναι ένα λανθασμένο στοιχείο </τέλος>]]>`

Οντότητες και αναφορές οντοτήτων

- Ορισμένοι χαρακτήρες έχουν ειδική σημασία στην XML.
 - Ο χαρακτήρας < υποδηλώνει την έναρξη μιας ετικέτας ενώ ο χαρακτήρας > υποδηλώνει το τέλος της ετικέτας.
 - Οι χαρακτήρες &, ' και ", έχουν ειδική σημασία στην XML.
- Η απευθείας χρησιμοποίηση τέτοιων συμβόλων στο κείμενο που αποτελεί το περιεχόμενο ενός στοιχείου οδηγεί σε συντακτικά λανθασμένα XML τεκμήρια.
 - Παράδειγμα 1: Το παρακάτω στοιχείο είναι συντακτικά λανθασμένο:
<στοιχείο> Το σύμβολο < δεν μπορεί να εμφανίζεται έτσι </στοιχείο>
αφού το < στο περιεχόμενο του θα εκληφθεί ως έναρξη ετικέτας.
- Η XML παρέχει ενσωματωμένες οντότητες οι οποίες ονομάζονται **εσωτερικές οντότητες** (internal entities) για την αναπαράσταση τέτοιων συμβόλων σε ένα XML τεκμήριο αποφεύγοντας τα προβλήματα σύνταξης.

Οντότητες και αναφορές οντοτήτων

(συνέχεια)

- Η τοποθέτηση τέτοιων συμβόλων σε ένα XML τεκμήριο γίνεται μέσω αναφορών στις αντίστοιχες οντότητες. Μια **αναφορά οντότητας** (entity reference) ξεκινά με το σύμβολο **&**, ακολουθείται από το *όνομα της οντότητας*, και τελειώνει με το σύμβολο **;**.
 - Παράδειγμα 2: το **<** αποτελεί αναφορά στην οντότητα με όνομα **lt** που αναπαριστά το **<**, ενώ με τα **>**, **&**, **'**, **"**, αναφερόμαστε στις οντότητες που αντιστοιχούν στα **>**, **&**, **'** και **"** αντίστοιχα.
 - Παράδειγμα 3: Το στοιχείο στο Παράδειγμα 1 πρέπει να γραφτεί σαν:
<στοιχείο>Το σύμβολο < δεν μπορεί να εμφανίζεται έτσι</στοιχείο>

Οντότητες και αναφορές οντοτήτων

(συνέχεια)

- Οντότητες XML μπορούν επίσης να χρησιμοποιηθούν για να αναφερθούμε σε κείμενο που επαναλαμβάνεται συχνά. Στην περίπτωση αυτή οι οντότητες παίζουν το ρόλο συντομογραφιών.
- Επίσης, οντότητες μπορούν να χρησιμοποιηθούν για να ενσωματώσουμε το περιεχόμενο εξωτερικών αρχείων.
- Οι οντότητες της κατηγορίας αυτής ονομάζονται και **εξωτερικές οντότητες** (external entities), και θα πρέπει να δηλωθούν από το χρήστη στο DTD, με τον τρόπο που θα δούμε στην αντίστοιχη ενότητα.
- Κάθε οντότητα θα πρέπει να έχει ένα μοναδικό όνομα.
- Οι οντότητες της XML μοιάζουν με τις μακροεντολές των γλωσσών προγραμματισμού.

Αναφορές χαρακτήρων

- Οι *αναφορές χαρακτήρων* (character references) έχουν παρόμοια μορφή με τις αναφορές οντότητας.
- Χρησιμοποιούνται για την εισαγωγή οποιουδήποτε χαρακτήρα του συνόλου ISO/IEC 10646 σε ένα XML τεκμήριο. Αυτό γίνεται περικλείοντας το κωδικό του χαρακτήρα ανάμεσα σε & και ;.
- Αν η αναφορά χαρακτήρα ξεκινά με &#x τότε τα ψηφία που ακολουθούν μέχρι το σύμβολο τερματισμού ; παρέχουν τη δεκαεξαδική αναπαράσταση του χαρακτήρα στο ISO/IEC 10646.
 - Παράδειγμα: ℞
- Αν όμως ξεκινά απλά με &# τότε τα ψηφία που ακολουθούν μέχρι το σύμβολο τερματισμού ; παρέχουν τη δεκαδική αναπαράσταση του χαρακτήρα στο ISO/IEC 10646.
 - Παράδειγμα: ℞
- Μέσω των αναφορών χαρακτήρων μπορούμε να εισάγουμε χαρακτήρες οι οποίοι δεν είναι προσπελάσιμοι από τις διαθέσιμες συσκευές εισόδου.

XML: Ταυτότητες Αντικειμένων (Oids) και Αναφορές

```
<person id="o555"> <name> John </name>
                    <children idref="o123"/>
</person>
<person id="o456"> <name> Mary </name>
                    <children idref="o123"/>
</person>
<person id="o123" mother="o456" father="o555"
>
    <name>Jim</name>
</person>
```

Τα oids και οι αναφορές στην XML είναι απλά σύνταξη

Document Type Definition (DTD)

Μανόλης Γεργατσούλης

Χρήστος Παπαθεοδώρου

Ομάδα Βάσεων Δεδομένων και Πληροφοριακών
Συστημάτων, Τμήμα Αρχιονομίας – Βιβλιοθηκονομίας

Ιόνιο Πανεπιστήμιο

XML: Document Type Definitions

(DTDs)

- **Πλεονέκτημα της XML:** επιτρέπει να ορίσουμε και να χρησιμοποιήσουμε στοιχεία, γνωρίσματα και οντότητες της αρεσκείας μας.
- Ένα έγγραφο XML είναι **καλά διαμορφωμένο** (*well-formed*) αν:
 - Το έγγραφο ξεκινά με ένα δηλωτικό XML.
 - Διαθέτει στοιχείο ρίζα που περιέχει όλα τα υπόλοιπα στοιχεία.
 - Όλα τα στοιχεία του είναι κατάλληλα εμφωλευμένα.
- Είναι χρήσιμο να τίθενται κοινά αποδεκτοί κανόνες που προδιαγράφουν συγκεκριμένο λεξιλόγιο από επιτρεπτά ονόματα στοιχείων και γνωρισμάτων, και θέτουν περιορισμούς ως προς την πολλαπλότητα εμφάνισης των στοιχείων, την μεταξύ τους σειρά κ.λ.π.
- Κάθε κοινότητα χρηστών μπορεί να προδιαγράψει τη δική της XML διάλεκτο με βάση τις ανάγκες των μελών της.
- Για την επιβολή τέτοιων περιορισμών απαιτείται ένας τρόπος να περιγραφούν αυτοί. Αυτό μπορεί να γίνει με τη βοήθεια **Δηλώσεων Τύπου Τεκμηρίων** (Document Type Definitions) (DTD).
- **Δηλώσεις τύπου τεκμηρίων:** σύνολα κανόνων που ορίζουν τα στοιχεία, τα γνωρίσματα και τις οντότητες που επιτρέπεται να εμφανίζονται στα XML έγγραφα.

XML: Document Type Definitions (DTDs)

(συνέχεια)

- Το περιεχόμενο ενός DTD παρέχει (μετα)πληροφορία στα *προγράμματα συντακτικής ανάλυσης* (parsers) των XML τεκμηρίων. Η πληροφορία αφορά τους περιορισμούς σύνταξης που πρέπει να πληρούν τα τεκμήρια ώστε να θεωρούνται *έγκυρα* ως προς το συγκεκριμένο DTD.
- *Έγκυρο* (valid) XML τεκμήριο: αν συνοδεύεται από ένα DTD και είναι δομημένο σύμφωνα με τους κανόνες που ορίζει το DTD.
- Ένα DTD λειτουργεί ως *γραμματική* (grammar) για μια κατηγορία XML τεκμηρίων, αφού παρέχει ένα λεξιλόγιο (αποδεκτά ονόματα στοιχείων και γνωρισμάτων) καθώς και σύνολο από κανόνες που διέπουν τη σειρά εμφάνισης, το πλήθος των εμφανίσεων κ.λ.π. των στοιχείων σε ένα XML τεκμήριο προκειμένου αυτό να θεωρείται έγκυρο.
- Το DTD από την οπτική γωνία των βάσεων δεδομένων μπορεί να εκληφθεί σαν *σχήμα* (schema) για τα δεδομένα που αναπαριστά το XML τεκμήριο, με μια σημασία παρόμοια με αυτή των σχεσιακών βάσεων δεδομένων.
- Παρόλα αυτά ένα XML τεκμήριο δεν υποχρεούται να περιλαμβάνει (ή να συνδέεται) με κάποιο DTD.

Παράδειγμα DTD

- Παράδειγμα XML τεκμηρίου που κωδικοποιεί στοιχεία φοιτητών του TAB:
<TAB>

<φοιτητής>

<όνομα> Νίκος </όνομα>

<επώνυμο> Νικολάου </επώνυμο>

</φοιτητής>

<φοιτητής> ... </φοιτητής>

...

</TAB>

- Ένα DTD για το πιο πάνω τεκμήριο:

<!DOCTYPE TAB [

<!ELEMENT TAB (φοιτητής*)>

<!ELEMENT φοιτητής (όνομα, επώνυμο)>

<!ELEMENT όνομα (#PCDATA)>

<!ELEMENT επώνυμο (#PCDATA)>

-Το κεντρικό στοιχείο είναι το TAB...

-Αποτελείται από στοιχεία φοιτητής...

-Το στοιχείο φοιτητής περιλαμβάνει τα στοιχεία όνομα και επώνυμο...

-Το όνομα περιλαμβάνει χαρακτήρες

-Το επώνυμο περιλαμβάνει χαρακτήρες....

]>

DTD: Δηλώσεις Τύπου Στοιχείων

Κωδική λέξη
ELEMENT που
δηλώνει
έναρξη
δήλωσης
στοιχείου

Όνομα του
στοιχείου

Αυστηρή
περιγραφή του
περιεχομένου
του στοιχείου

<!ELEMENT όνομα_ στοιχείου τύπος_στοιχείου>

Δηλώσεις Τύπου Στοιχείων

Δήλωση	Σημασία
R?	0 ή 1 στιγμιότυπο του R
R+	1 ή περισσότερα στιγμιότυπα του R
R*	0 ή περισσότερα στιγμιότυπα του R
R_1, R_2, \dots, R_n	1 στιγμιότυπο του R_1 ακολουθούμενο από 1 στιγμ. του R_2, \dots , ακολουθούμενο από 1 στιγμ. του R_n
$R_1 \mid R_2 \mid \dots \mid R_n$	1 στιγμιότυπο του R_1 ή του R_2 ή ... ή του R_n
EMPTY	Κενό στοιχείο
ANY	Οτιδήποτε
#PCDATA	Χαρακτήρες που έχουν υποστεί συντακτική ανάλυση

DTD: Δηλώσεις Τύπου Στοιχείων:

Παράδειγμα

- Με την έκφραση:

`<!ELEMENT s (a, b?,c*)>`

δηλώνεται ότι: κάθε στοιχείο με ετικέτα **s** που εμφανίζεται σε ένα έγκυρο XML τεκμήριο, περιλαμβάνει ένα ακριβώς στοιχείο με ετικέτα **a** ακολουθούμενο προαιρετικά από ένα το πολύ στοιχείο με ετικέτα **b**, και στη συνέχεια από οποδήποτε μεγάλο πλήθος (μπορεί και μηδέν) στοιχείων με ετικέτα **c**.

DTD: Δηλώσεις Τύπου Στοιχείων:

Παράδειγμα

- Με την έκφραση:

$\langle !ELEMENT\ e\ ((c?,\ d)^* \mid (d,\ c)^*) \rangle$

δηλώνεται ότι κάθε στοιχείο με ετικέτα e που εμφανίζεται σε ένα έγκυρο XML τεκμήριο, περιλαμβάνει:

- μια ακολουθία από (μηδέν ή περισσότερα) ζεύγη στοιχείων c, d από τα οποία το c είναι προαιρετικό, είτε
- μια ακολουθία από (μηδέν ή περισσότερα) ζεύγη στοιχείων d, c .

DTD: Δηλώσεις Τύπου Στοιχείων:

Παράδειγμα (συνέχεια)

Στα XML τεκμήρια που ακολουθούν έχουμε για ευκολία θεωρήσει ότι τα στοιχεία **c** και **d** είναι κενά. Τα τεκμήρια εξετάζονται ως προς την εγκυρότητα τους με βάση το προηγούμενο DTD.

Έγκυρο XML τεκμήριο:

```
<e>  
  <d/>  
  <d/>  
  <c/>  
  <d/>  
  <d/>  
</e>
```

Μη έγκυρο XML τεκμήριο:

```
<e>  
  <d/>  
  <c/>  
  <c/>  
  <d/>  
</e>
```

Δηλώσεις Τύπου Στοιχείων (συνέχεια)

- Για να δηλώσουμε ότι το περιεχόμενο ενός στοιχείου είναι ακολουθία χαρακτήρων χρησιμοποιούμε δηλώσεις της μορφής:
`<!ELEMENT όνομα_στοιχείου (#PCDATA)>`
- Η παράσταση *τύπος_στοιχείου* είναι επίσης δυνατό να πάρει μια από τις τιμές **EMPTY** και **ANY** που σημαίνουν το *κενό στοιχείο*, και το *στοιχείο με οποιοδήποτε περιεχόμενο αντίστοιχα*.
- Αποδεκτές είναι επίσης τιμές που αποτελούν ανάμιξη **#PCDATA** και ονομάτων στοιχείων.
- **Προσοχή:** ένα στοιχείο δεν επιτρέπεται να δηλώνεται περισσότερο από μια φορά σε ένα DTD.

DTD: Δηλώσεις Λίστας Γνωρισμάτων

Κωδική λέξη
ATTLIST που
δηλώνει έναρξη
δήλωσης λίστας
γνωρισμάτων

Όνομα του
στοιχείου

Δηλώσεις
γνωρισμάτων

<!ATTLIST όνομα_στοιχείου λίστα_δηλώσεων_γνωρισμάτων>

Τριάδες της μορφής:

όνομα_γνωρίσματος τύπος_γνωρίσματος προκαθορισμός_τιμής

Δηλώσεις Λίστας Γνωρισμάτων:

Παράδειγμα

```
<!ATTLIST φοιτητής AM CDATA #REQUIRED  
AΔΤ CDATA #IMPLIED >
```

- Το στοιχείο **φοιτητής** έχει δύο γνωρίσματα με ονόματα **AM** και **AΔΤ**.
- Και τα δύο γνωρίσματα είναι του τύπου **CDATA**.
- Η παρουσία του γνωρίσματος **AM** είναι υποχρεωτική σε κάθε εμφάνιση του στοιχείου φοιτητής (λόγω του **#REQUIRED**).
- Η παρουσία του γνωρίσματος **AΔΤ** δεν είναι υποχρεωτική (λόγω του **#IMPLIED**).

Δηλώσεις Λίστας Γνωρισμάτων

(συνέχεια)

- Ανάμεσα στις πιθανές τιμές που μπορεί να πάρει η παράμετρος *τύπος_γνωρίσματος* περιλαμβάνονται και οι ακόλουθες: **CDATA**, **ID**, **IDREF**, **IDREFS**, **ENTITY**, **ENTITIES**, **NMTOKEN**, **NMTOKENS**.
 - Τα γνωρίσματα του τύπου **CDATA** μπορούν να πάρουν για τιμή οποιοδήποτε κείμενο (ακολουθία χαρακτήρων).
 - Τα γνωρίσματα του τύπου **NMTOKEN** είναι ειδική περίπτωση των γνωρισμάτων τύπου **CDATA** και μπορούν να πάρουν για τιμή μια απλή λέξη.
 - Τα γνωρίσματα του τύπου **NMTOKENS** μπορούν να πάρουν για τιμή πολλαπλές τιμές τύπου **NMTOKEN** που χωρίζονται μεταξύ τους με κενά.
 - Τα γνωρίσματα του τύπου **ENTITY** μπορεί να πάρουν για τιμή το όνομα μιας απλής οντότητας ενώ τα γνωρίσματα του τύπου **ENTITIES** μπορούν να πάρουν για τιμή πολλαπλές τιμές του τύπου **ENTITY** οι οποίες χωρίζονται μεταξύ τους με κενά.
- Πέρα από τις παραπάνω τιμές, η παράμετρος *τύπος_γνωρίσματος* είναι δυνατό να πάρει σαν τιμή τη λίστα των πιθανών τιμών του γνωρίσματος. Στην περίπτωση αυτή μιλάμε για **απαριθμήσιμους τύπους γνωρισμάτων** (enumerated attribute types).

Δηλώσεις Λίστας Γνωρισμάτων: Παράδειγμα

- Στη δήλωση λίστας γνωρισμάτων που ακολουθεί:
`<!ATTLIST book color (red|green|blue) "blue">`
ορίζεται ότι:
 - Το στοιχείο `book` έχει ένα γνώρισμα με όνομα `color`.
 - Το γνώρισμα αυτό μπορεί να πάρει μια από τις τιμές `red`, `green`, `blue`.
 - Σε περίπτωση που το γνώρισμα απουσιάζει από ένα στοιχείο `book` θεωρείται ως εάν να είναι παρών και η τιμή του να είναι η `blue` (προκαθορισμένη τιμή).

Δηλώσεις Λίστας Γνωρισμάτων

(συνέχεια)

- Τιμές της παράμετρου *προκαθορισμός_τιμής*:
 - Μπορεί να πάρει σαν τιμή μια από τις πιθανές τιμές του γνωρίσματος, με τη σημασία που αναφέραμε προηγούμενα.
 - Η τιμή **#REQUIRED** η οποία επιβάλλει την υποχρεωτική εμφάνιση του γνωρίσματος στο αντίστοιχο στοιχείο.
 - Η τιμή **#IMPLIED** η οποία υποδηλώνει ότι δεν παρέχεται κάποια προκαθορισμένη τιμή (και δεν είναι υποχρεωτική η εμφάνιση του συγκεκριμένου γνωρίσματος)
 - Η τιμή **#FIXED** ακολουθούμενη από μια συγκεκριμένη τιμή. Στην περίπτωση αυτή όλες οι εμφανίσεις του αντίστοιχου γνωρίσματος στο XML τεκμήριο θα πρέπει να έχουν για τιμή τη συγκεκριμένη τιμή που έχει δηλωθεί μετά από το **#FIXED**.

Δηλώσεις Λίστας Γνωρισμάτων: Παράδειγμα

- Με τη δήλωση:
 <!ATTLIST form method CDATA #FIXED "POST">
- ορίζεται ότι:
 - το στοιχείο **form** διαθέτει το γνώρισμα **method** το οποίο είναι τύπου **CDATA** και έχει πάντα τη τιμή **POST**.

Οι τύποι γνωρίσματος **ID**, **IDREF**, και **IDREFS**

- Ο τύπος **ID** δηλώνει ότι το γνώρισμα παίζει ρόλο ταυτότητας για το στοιχείο, παίρνει δηλαδή μοναδική τιμή που προσδιορίζει μονοσήμαντα το κάθε στοιχείο.
 - Δεν επιτρέπεται να υπάρχουν περισσότερα του ενός στοιχεία σε ένα έγκυρο XML τεκμήριο τα οποία να διαθέτουν γνώρισμα τύπου **ID** και να έχουν την ίδια τιμή στο γνώρισμα αυτό.
- Ο τύπος **IDREF** δηλώνει γνωρίσματα που παίρνουν σαν τιμή τη τιμή του γνωρίσματος τύπου **ID** κάποιου άλλου στοιχείου.
- Ένα γνώρισμα του τύπου **ID** συμπεριφέρεται όπως ένα “*κλειδί*” σε μια σχεσιακή βάση δεδομένων, ενώ ένα γνώρισμα του τύπου **IDREF** σαν ένα “*ξένο κλειδί*” σε μια σχεσιακή βάση δεδομένων.
- Ως **IDREFS** δηλώνονται γνωρίσματα τα οποία παίρνουν σαν τιμή μια λίστα τιμών του τύπου **IDREF** οι οποίες χωρίζονται μεταξύ τους με κενά.
- Τα γνωρίσματα του τύπου **ID** πρέπει να έχουν ως προκαθορισμό τιμής είτε **#REQUIRED** είτε **#IMPLIED**.
- Κάθε τιμή που παίρνει ένα γνώρισμα τύπου **IDREF**, ή **IDREFS** πρέπει να υπάρχει στο τεκμήριο ως τιμή ενός γνωρίσματος του τύπου **ID**.

Οι τύποι γνωρίσματος ID, IDREF, και IDREFS: Παράδειγμα DTD

- Ένα DTD:

```
<!DOCTYPE οικογένεια [  
<!ELEMENT οικογένεια (πρόσωπο)*>  
<!ELEMENT πρόσωπο (όνομα, επώνυμο)>  
<!ATTLIST πρόσωπο ΑΔΤ ID #REQUIRED  
                  μητέρα IDREF #IMPLIED  
                  πατέρας IDREF #IMPLIED  
                  παιδιά IDREFS #IMPLIED>  
<!ELEMENT όνομα (#PCDATA)>  
<!ELEMENT επώνυμο (#PCDATA)>  
>
```

Οι τύποι γνωρίσματος ID, IDREF, και IDREFS: Παράδειγμα (συνέχεια)

- Ένα XML τεκμήριο:

<οικογένεια>

<πρόσωπο ΑΔΤ = "Κ123456" παιδιά = "Μ345678 Ν456789">

<όνομα> Πέτρος </όνομα>

<επώνυμο> Πέτρου </επώνυμο>

</πρόσωπο>

<πρόσωπο ΑΔΤ = "Λ234567" παιδιά = "Μ345678 Ν456789">

<όνομα> Μαρία </όνομα>

<επώνυμο> Πέτρου </επώνυμο>

</πρόσωπο>

<πρόσωπο ΑΔΤ = "Μ345678" πατέρας = "Κ123456" μητέρα = "Λ234567" >

<όνομα> Γιώργος </όνομα>

<επώνυμο> Πέτρου </επώνυμο>

</πρόσωπο>

<πρόσωπο ΑΔΤ = "Ν456789" πατέρας = "Κ123456" μητέρα = "Λ234567" >

<όνομα> Άννα </όνομα>

<επώνυμο> Πέτρου-Ιωάννου </επώνυμο>

</πρόσωπο>

....

</οικογένεια>

Δηλώσεις Οντοτήτων

- **Βασική ιδέα:** οι **οντότητες** αποτελούν συντομογραφίες σε ένα XML τεκμήριο και επομένως οι δηλώσεις των οντοτήτων στην πραγματικότητα αποτελούν ορισμό των συντομογραφιών αυτών.
- Διακρίνουμε τρία είδη οντοτήτων.
 - Οι **εσωτερικές οντότητες** (internal entities),
 - Οι **εξωτερικές οντότητες** (external entities),
 - Οι **οντότητες παραμέτρων** (parameter entities).
- Η **δήλωση μιας εσωτερικής οντότητας** γίνεται μέσω μιας έκφρασης της μορφής:
`<!ENTITY όνομα_οντότητας τιμή_οντότητας>`
 - Στην περίπτωση αυτήν, το κείμενο με το οποίο αντικαθίσταται η εμφάνιση της οντότητας αποθηκεύεται μέσα στη δήλωση της οντότητας.

Δηλώσεις Οντοτήτων

- **Δηλώσεις εξωτερικών οντοτήτων:** μαζί με το όνομα της οντότητας δηλώνεται και ένα URI το οποίο θα πρέπει να ακολουθηθεί προκειμένου να βρεθεί το κείμενο το οποίο θα δοθεί σαν τιμή στην οντότητα.
- **Παράδειγμα:** Με τη δήλωση:

```
<!ENTITY TABinfo  
        SYSTEM "/ionio/information/info.xml">
```

ορίζεται η οντότητα **TABinfo**. Κατά την επεξεργασία του τεκμηρίου κάθε εμφάνιση του **&TABinfo;** θα αντικαθίσταται με το περιεχόμενο του αρχείου **"/ionio/information/info.xml"**.

Οι οντότητες παραμέτρων

- Οι *οντότητες παραμέτρων* εμφανίζονται μόνο στα πλαίσια ενός DTD.
- Στη *δήλωση μιας οντότητας παραμέτρου* πριν από το όνομα της οντότητας πρέπει να τοποθετείται ο χαρακτήρας %.
- Το ίδιο σύμβολο % εμφανίζεται και στις *αναφορές σε οντότητες παραμέτρου* αντί για το σύμβολο &.
- Οι αναφορές σε οντότητες παραμέτρου αντικαθίσταται άμεσα με το κείμενο που αποτελεί την τιμή της οντότητας το οποίο αποτελεί πλέον κομμάτι του DTD.

Σύνδεση XML με DTD

- Προκειμένου να εξεταστεί η εγκυρότητα ενός XML τεκμηρίου ως προς ένα DTD, θα πρέπει τα δύο αυτά να συσχετιστούν.
- Γενικά έχουμε δύο επιλογές.
 - Πρώτη επιλογή: να συμπεριλάβουμε το DTD στο ίδιο αρχείο με αυτό που βρίσκεται το XML τεκμήριο.
 - Δεύτερη επιλογή: να τοποθετήσουμε τις δηλώσεις του DTD σε ξεχωριστό αρχείο και στη συνέχεια να συσχετίσουμε κατάλληλα τα δύο αρχεία.

Σύνδεση XML με DTD: Παράδειγμα 1

- DTD ενσωματωμένο στο XML τεκμήριο:

DTD

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

```
<!DOCTYPE greeting [  
  <!ELEMENT greeting (#PCDATA)>  
>
```

```
<greeting>Hello, world!</greeting>
```


Σύνδεση XML με DTD: Παράδειγμα 2

- Σύνδεση με εξωτερικό αρχείο στο οποίο έχει αποθηκευτεί το DTD:

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

```
<!DOCTYPE greeting SYSTEM "hello.dtd">
```

```
<greeting>Hello, world!</greeting>
```

Σύνδεση με
DTD

Υπό συνθήκη τμήματα

- Ένα **υπό συνθήκη τμήμα** (conditional section) είναι τμήμα ενός εξωτερικού DTD το οποίο συμπεριλαμβάνεται ή δεν συμπεριλαμβάνεται στη λογική δομή του DTD με βάση κάποια **λέξη κλειδί** (keyword) η οποία συνδέεται με αυτό.
- Αν η τιμή της λέξης κλειδί είναι **INCLUDE**, τότε το περιεχόμενο του υπό συνθήκη τμήματος συμπεριλαμβάνεται στο DTD.
- Αν η τιμή της λέξης κλειδί είναι **IGNORE**, τότε το περιεχόμενο του υπό συνθήκη τμήματος δεν συμπεριλαμβάνεται στο DTD.
- Η λέξη κλειδί σε ένα υπό συνθήκη τμήμα μπορεί να είναι αναφορά σε οντότητα παραμέτρου. Τότε η οντότητα παραμέτρου πρέπει να αντικατασταθεί με την τιμή της για να μπορέσει ο επεξεργαστής να αποφασίσει αν θα συμπεριλάβει το υπό συνθήκη τμήμα.

Υπό συνθήκη τμήματα: Παράδειγμα

- Παράδειγμα:

```
<!ENTITY % draft 'INCLUDE' >
```

```
<!ENTITY % final 'IGNORE' >
```

```
<![%draft; [
```

```
<!ELEMENT article (comments*, title, body, supplements?)>
```

```
]]>
```

```
<![%final; [
```

```
<!ELEMENT article (title, body, supplements?)>
```

```
]]>
```

- Η συμμετοχή ή όχι των υπό συνθήκη τμημάτων καθορίζεται από την τιμή των παραμέτρων **draft** και **final**. Μεταβάλλοντας τις τιμές των παραμέτρων αυτών μπορούμε εύκολα να εναλλάξουμε τα υπό συνθήκη τμήματα τα οποία θα συμπεριληφθούν στο DTD.

DTD για Εξειδικευμένα Πεδία

- Η ανάπτυξη τυποποιημένων DTD για εξειδικευμένα πεδία επιτρέπει την ανταλλαγή δεδομένων ανάμεσα σε ετερογενείς πηγές.
- Παράδειγμα: *Μαθηματική Γλώσσα Σημειοθέτησης (Mathematical Markup Language) (MathML)*.
 - Τεχνικό εγχειρίδιο «*Mathematical Markup Language (MathML) Version 2.0*» από τη διεύθυνση <http://www.w3.org/TR/MathML2>
 - Κωδικοποίηση μαθηματικού υλικού στο web.
 - Η παράσταση $(x+y)^2$ κωδικοποιείται στην MathML ως:

```
<apply> <power/>  
  <apply> <plus/> <ci>x</ci> <ci>y</ci> </apply>  
  <cn>2</cn>  
</apply>
```

XML Namespaces

Μανόλης Γεργατσούλης

Χρήστος Παπαθεοδώρου

Ομάδα Βάσεων Δεδομένων και Πληροφοριακών
Συστημάτων, Τμήμα Αρχειονομίας – Βιβλιοθηκονομίας

Ιόνιο Πανεπιστήμιο

Χώροι Ονομάτων XML

- Οι *Χώροι Ονομάτων XML* (XML Namespaces) είναι συλλογές ονομάτων που προσδιορίζονται από μια αναφορά URI (**U**niform **R**esource **I**dentifier) και χρησιμοποιούνται για την ονομασία στοιχείων και γνωρισμάτων της XML.
- Όταν χρησιμοποιούνται χώροι ονομάτων, τα ονόματα στοιχείων και γνωρισμάτων εμφανίζονται με δύο συνθετικά, το *πρόθεμα* (χώρου ονομάτων) (prefix) και το *τοπικό τμήμα* (local part), τα οποία χωρίζονται μεταξύ τους με **:**.
- Ο μηχανισμός αυτός παράγει ονόματα τα οποία είναι παγκοσμίως μοναδικά.

Γιατί χρησιμοποιούμε χώρους ονομάτων

- Οι σχεδιαστές της XML προσβλέπουν σε εφαρμογές στις οποίες ένα XML τεκμήριο μπορεί να περιλαμβάνει στοιχεία και γνωρίσματα που έχουν οριστεί και απευθύνονται σε διαφορετικές εφαρμογές λογισμικού.
- Στόχο της ύπαρξης καλά τεκμηριωμένων και κατανοητών “λεξιλογίων ετικετών” για το οποίο υπάρχει διαθέσιμο χρήσιμο λογισμικό, αποτελεί η επιθυμία για **επαναχρησιμοποίηση** των τεκμηρίων από πολλές διαφορετικές εφαρμογές.
- Τα τεκμήρια που περιέχουν πολλαπλά “λεξιλόγια ετικετών” εμφανίζουν προβλήματα αναγνώρισης (σε ποια εφαρμογή απευθύνεται η κάθε ετικέτα) και “σύγκρουσης” (ίδιες ετικέτες που απευθύνονται σε διαφορετικές εφαρμογές).
- Οι εφαρμογές λογισμικού πρέπει να μπορούν να αναγνωρίζουν τα στοιχεία και τα γνωρίσματα για την επεξεργασία των οποίων έχουν σχεδιαστεί, ακόμη και στην περίπτωση που εμφανίζεται “σύγκρουση” ονομάτων στοιχείων ή γνωρισμάτων. Τα παραπάνω απαιτούν συντακτικές δομές που επιτρέπουν ονόματα διαμορφωμένα κατά τρόπον ώστε να αποφεύγεται η σύγκρουση μεταξύ ονομάτων από διαφορετικά “λεξιλόγια ετικετών”.
- Η λύση είναι η χρησιμοποίηση του μηχανισμού που ονομάζεται **χώρος ονομάτων XML**, ο οποίος πετυχαίνει τον στόχο αυτόν επιτρέποντας επεκταμένα ονόματα στοιχείων και γνωρισμάτων.

Δηλώσεις χώρων ονομάτων

- Ένας χώρος ονομάτων δηλώνεται μέσω μιας οικογένειας από προκαθορισμένα γνώρισμα όπως το γνώρισμα **xmlns** ή οποιοδήποτε γνώρισμα έχει το **xmlns:** σαν πρόθεμα. Η τιμή του γνωρίσματος, που είναι μια αναφορά URI, είναι το **όνομα του χώρου ονομάτων** (namespace name).
- **Παράδειγμα:** Στο παρακάτω τμήμα XML τεκμηρίου βλέπουμε μια δήλωση χώρου ονομάτων η οποία συσχετίζει το πρόθεμα **bk** με το όνομα χώρου ονομάτων **www.books.org/book:**

```
<book xmlns:bk="www.books.org/book" >  
.....  
</book>
```

▪ Δήλωση
χώρου
ονομάτων

Πολλαπλές δηλώσεις χώρων

ονομάτων

- Είναι δυνατό να δηλωθούν πολλοί χώροι ονομάτων στο ίδιο στοιχείο.
- **Παράδειγμα:** Στο τμήμα XML τεκμηρίου που ακολουθεί:

```
<book xmlns:bk="www.books.org/book"  
      xmlns:isbn="www.isbn.org/def" >
```

.....

```
</book>
```

δηλώνονται δύο χώροι ονομάτων:

- Ο πρώτος συνδέει το πρόθεμα **bk** με το χώρο ονομάτων με όνομα www.books.org/book.
- Ο δεύτερος συνδέει το πρόθεμα **isbn** με το όνομα χώρου ονομάτων www.isbn.org/def.

Εμβέλεια χώρων ονομάτων

- Μια δήλωση χώρου ονομάτων εφαρμόζεται στο στοιχείο που γίνεται καθώς και σε όλα τα στοιχεία που περικλείονται μέσα σ' αυτό, εκτός εάν αντικατασταθεί από μια νέα δήλωση που συνδέει το ίδιο πρόθεμα με άλλο όνομα χώρου ονομάτων (διαφορετικό URI).
- Δεν είναι υποχρεωτικό όλα τα στοιχεία που αποτελούν το περιεχόμενο του στοιχείου στο οποίο έχει οριστεί ένας χώρος ονομάτων, να χρησιμοποιούν το συγκεκριμένο χώρο ονομάτων.

Η χρήση των χώρων ονομάτων

- Σε ένα XML τεκμήριο που χρησιμοποιεί χώρους ονομάτων, μερικά (ή και όλα) από τα ονόματα των στοιχείων και των γνωρισμάτων που περιλαμβάνει μπορεί να αποτελούνται από δύο συνθετικά. Το πρώτο ονομάζεται **πρόθεμα χώρου ονομάτων** (namespace prefix) ενώ το δεύτερο ονομάζεται **τοπικό τμήμα** (local part).
- Τα δύο αυτά συνθετικά χωρίζονται μεταξύ τους με **:**.
- Το πρόθεμα πρέπει να έχει δηλωθεί σε μια δήλωση χώρου ονομάτων και να έχει συνδεθεί με ένα όνομα χώρου ονομάτων (μια αναφορά URI).
- Η σύνταξη των ονομάτων στοιχείων και γνωρισμάτων τα οποία χρησιμοποιούν χώρους ονομάτων έχουν την μορφή:
prefix:localpart
- Ο συνδυασμός των δύο συνθετικών παράγει ονόματα τα οποία είναι μοναδικά.

Η χρήση των χώρων ονομάτων

- Παράδειγμα: Στο το παρακάτω τμήμα XML τεκμηρίου:

```
<book xmlns:isbn="www.isbn.org/def">
```

```
<title> ... </title>
```

```
<authors> ... </authors>
```

```
<isbn:number> ... </isbn:number>
```

```
</book>
```

Έχει οριστεί εδώ

στο στοιχείο **book** έχει δηλωθεί χώρος ονομάτων ο οποίος συνδέει το πρόθεμα **isbn** με το όνομα χώρου ονομάτων **www.isbn.org/def**. Στη συνέχεια, στο περιεχόμενο του στοιχείου **book** συμπεριλαμβάνεται και το στοιχείο **isbn:number** το όνομα του οποίου περιλαμβάνει το πρόθεμα **isbn**.

Η χρήση των χώρων ονομάτων

(συνέχεια)

- **Παράδειγμα:** Στο τεκμήριο που ακολουθεί έχουμε δήλωση και χρήση δύο χώρων ονομάτων που συσχετίζονται με τα προθέματα **bk** και **isbn:**

```
<book xmlns:isbn="www.isbn.org/def"
      xmlns:bk="www.books.org/book" >
  <bk:title> ... </bk:title>
  <bk:authors> ... </bk:authors>
  <isbn:number> ... </isbn:number>
</book>
```

Χρήση χώρων ονομάτων σε γνωρίσματα

- Οι χώροι ονομάτων μπορούν να χρησιμοποιηθούν και για τον σχηματισμό ονομάτων γνωρισμάτων.
- **Παράδειγμα:** Στο παρακάτω τμήμα XML τεκμηρίου:

```
<tab xmlns:sns="www.ionio.gr/sdef" >  
  <sns:student sns:scode = "12345">  
    <sns:name> ... </sns:name>  
  </sns:student>  
</tab>
```

έχουμε δήλωση χώρου ονομάτων, ο οποίος χρησιμοποιείται για την παράγωγή ονομάτων των στοιχείων **sns:student** και **sns:name** και του γνωρίσματος **sns:scode**.

Χρήση χώρων ονομάτων

- Κάθε πρόθεμα που χρησιμοποιείται σε ένα όνομα στοιχείου ή γνωρίσματος, εκτός από τα **xm:ns** και **xm:l**, πρέπει να έχει δηλωθεί σε μια δήλωση χώρου ονομάτων, είτε στην ετικέτα αρχής του στοιχείου μέσα στο οποίο χρησιμοποιείται, είτε σε κάποιο στοιχείο “πρόγονο” του στοιχείου αυτού.
- Τα ονόματα στοιχείων και γνωρισμάτων συνοδεύονται από το πρόθεμα τους όταν αυτά δηλώνονται σε ένα DTD.

Χώροι ονομάτων χωρίς προθέματα

- Συχνά είναι χρήσιμο να δηλωθεί ένας χώρος ονομάτων ο οποίος να μην διαθέτει πρόθεμα. Ένας τέτοιος χώρος ονομάτων ονομάζεται **προκαθορισμένος χώρος ονομάτων** (default namespace). Η χρησιμότητα του έγκειται στο ότι τα ονόματα στοιχείων που ανήκουν στο χώρο αυτό δεν χρειάζεται να συνοδεύονται από συγκεκριμένο πρόθεμα.
- Ο χώρος ονομάτων χωρίς πρόθεμα εφαρμόζεται στο στοιχείο στο οποίο δηλώνεται (αν αυτό δεν έχει πρόθεμα χώρου ονομάτων) καθώς και σε όλα τα στοιχεία χωρίς πρόθεμα που περιλαμβάνονται στο περιεχόμενο του στοιχείου αυτού.
- Η αναφορά URI στη δήλωση του χώρου ονομάτων χωρίς πρόθεμα μπορεί να είναι κενή. Τα στοιχεία χωρίς πρόθεμα και βρίσκονται στην εμβέλεια μιας τέτοιας δήλωσης θεωρείται ότι δεν ανήκουν σε κανένα χώρο ονομάτων.
- Ο χώρος ονομάτων χωρίς πρόθεμα δεν εφαρμόζεται απευθείας σε γνωρίσματα.

Χώροι ονομάτων χωρίς προθέματα:

Παράδειγμα 1

- **Παράδειγμα:** Στο παρακάτω XML τεκμήριο φαίνεται δήλωση και χρήση του χώρου ονομάτων χωρίς πρόθεμα (όλα τα στοιχεία του τεκμηρίου ανήκουν σε αυτόν τον χώρο ονομάτων):

```
<?xml version="1.0"?>
<!-- τα στοιχεία ανήκουν στον default χώρο ονομάτων "HTML" -->
<html xmlns='http://www.w3.org/TR/REC-html40'>
  <head>
    <title>Frobnostication</title>
  </head>
  <body>
    <p>Moved to <a href='http://frob.com'>here</a>.</p>
  </body>
</html>
```

Χώροι ονομάτων χωρίς προθέματα:

Παράδειγμα 2

- **Παράδειγμα:** Στο ακόλουθο XML τεκμήριο έχουμε δήλωση και χρήση χώρου ονομάτων χωρίς πρόθεμα καθώς και ενός χώρου ονομάτων που συνδέεται με το πρόθεμα **isbn**:

```
<?xml version="1.0"?>
<!-- τα στοιχεία χωρίς πρόθεμα ανήκουν στον "books" -->
<book xmlns='urn:loc.gov:books'
      xmlns:isbn='urn:ISBN:0-395-36341-6'>
  <title>Cheaper by the Dozen</title>
  <isbn:number>1568491379</isbn:number>
</book>
```

Έτσι, εκτός από το στοιχείο **isbn:number**, το οποίο ανήκει στο χώρο ονομάτων με πρόθεμα **isbn**, όλα τα άλλα ανήκουν στον χώρο ονομάτων **'urn:loc.gov:books'**.

Επαναορισμός χώρου ονομάτων χωρίς πρόθεμα

- **Παράδειγμα:** Στο ακόλουθο XML τεκμήριο φαίνεται η αλλαγή του χώρου ονομάτων χωρίς όνομα λόγω επαναορισμού του σε ένα εσωτερικό στοιχείο:

```
<?xml version="1.0"?>
  <!-- αρχικά, ο default χώρος ονομάτων είναι ο "books" -->
  <book xmlns='urn:loc.gov:books'
        xmlns:isbn='urn:ISBN:0-395-36341-6'>
    <title>Cheaper by the Dozen</title>
    <isbn:number>1568491379</isbn:number>
    <notes>
      <!-- κάνει τον "HTML" default χώρο ονομάτων -->
      <p xmlns='urn:w3-org-ns:HTML'>
        This is a <i>funny</i> book!
      </p>
    </notes>
  </book>
```

Μοναδικότητα γνωρισμάτων

- Στα XML τεκμήρια τα οποία υπακούουν στις προδιαγραφές που περιγράψαμε, καμιά ετικέτα στοιχείου δεν μπορεί να περιέχει δύο γνωρίσματα τα οποία:
 - έχουν ταυτόσημα ονόματα, ή
 - έχουν ονόματα με το ίδιο τοπικό τμήμα και τα προθέματα των οποίων έχουν συνδεθεί με ταυτόσημα ονόματα χώρων ονομάτων.
- **Παράδειγμα:** Οι ετικέτες αρχής **bad** παρακάτω δεν είναι έγκυρες λόγω της ταύτισης των ονομάτων των γνωρισμάτων τους:

```
<!-- http://www.w3.org is bound to n1 and n2 -->  
<x xmlns:n1="http://www.w3.org"  
  xmlns:n2="http://www.w3.org" >  
  <bad a="1"    a="2" />  
  <bad n1:a="1" n2:a="2" />  
  
</x>
```

Μοναδικότητα γνωρισμάτων

(συνέχεια)

- **Παράδειγμα:** Η σύνταξη των στοιχείων του XML τεκμηρίου που ακολουθεί είναι έγκυρη. Το δεύτερο στοιχείο **good** είναι έγκυρο επειδή ο χώρος ονομάτων χωρίς πρόθεμα δεν εφαρμόζεται στα ονόματα γνωρισμάτων.

```
<!-- http://www.w3.org is bound to n1 and is the default -->  
<x xmlns:n1="http://www.w3.org"  
  xmlns="http://www.w3.org" >  
  <good a="1" b="2" />  
  <good a="1" n1:a="2" />  
</x>
```

XML Schema

Μανόλης Γεργατσούλης

Χρήστος Παπαθεοδώρου

Ομάδα Βάσεων Δεδομένων και Πληροφοριακών
Συστημάτων, Τμήμα Αρχιονομίας – Βιβλιοθηκονομίας

Ιόνιο Πανεπιστήμιο

Η γλώσσα *XML Schema*

- Η γλώσσα *XML Schema* είναι μια γλώσσα XML κατάλληλη για την περιγραφή της δομής XML τεκμηρίων.
- Η XML Schema (όπως και τα DTD) είναι γλώσσα περιγραφής σχήματος.
- Η XML Schema προσφέρει χαρακτηριστικά και δυνατότητες, ισχυρότερα αυτών που παρέχονται από τα DTD.

XML Schema: Παράδειγμα

- Ζητάμε περιγραφή σε XML Schema της δομής των τεκμηρίων της μορφής:

<TAB>

<φοιτητής>

<όνομα> Νίκος </όνομα>

<επώνυμο> Νικολάου </επώνυμο>

</φοιτητής>

<φοιτητής> ... </φοιτητής>

...

</TAB>

XML Schema: Παράδειγμα (συνέχεια)

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="TAB">
    <xs:complexType>
      <xs:element name="φοιτητής" minOccurs=0
        maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="όνομα" type="xs:string"/>
            <xs:element name="επώνυμο" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

<!ELEMENT TAB (φοιτητής*)>

<!ELEMENT φοιτητής (όνομα. επώνυμο)>

<!ELEMENT όνομα (#PCDATA)>

<!ELEMENT επώνυμο (#PCDATA)>

Χαρακτηριστικά της XML Schema

- Ένα τεκμήριο σε XML Schema είναι ένα XML τεκμήριο.
- Τα στοιχεία στο XML Schema του παραδείγματος έχουν το πρόθεμα **xs** το οποίο συνδέεται με το χώρο ονομάτων της XML Schema μέσω της δήλωσης:
 - `xmlns:xs="http://www.w3.org/2001/XMLSchema"`.
- Η XML Schema παρέχει στοιχεία (όπως τα **element**, **sequence**, **complexType**) με συγκεκριμένη σημασία, τα οποία μαζί με αντίστοιχα γνωρίσματα (όπως τα **name**, **type**, **minOccurs**, **maxOccurs**) επιτρέπουν την περιγραφή της δομής XML τεκμηρίων.
- Μια περιγραφή σχήματος στην XML Schema έχει ως ετικέτα αρχής του πιο εξωτερικού στοιχείου (της ρίζας) την ετικέτα **xs:schema**.
- Η XML Schema παρέχει αξιολογούμενη εκφραστικότητα η οποία επιτρέπει την περιγραφή της δομής XML τεκμηρίων με πολλούς διαφορετικούς τρόπους.

Παράδειγμα (μια εναλλακτική περιγραφή του προηγούμενου)

- Η ακόλουθη περιγραφή σε XML Schema περιγράφει ακριβώς την ίδια κατηγορία τεκμηρίων με την προηγούμενη περιγραφή:

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

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

```
<xs:element name="TAB" type="TABtype" />
```

```
<xs:complexType name="TABtype">
```

```
<xs:element name="φοιτητής" type="studentType"
  minOccurs=0 maxOccurs="unbounded" />
```

```
</xs:complexType>
```

```
<xs:complexType name="studentType">
```

```
<xs:sequence>
```

```
<xs:element name="όνομα" type="xs:string" />
```

```
<xs:element name="επώνυμο" type="xs:string" />
```

```
</xs:sequence>
```

```
</xs:complexType>
```

```
</xs:schema>
```

Δηλώσεις στοιχείων

- Στοιχεία της XML:
 - **Σύνθετου τύπου**: περιέχουν υποστοιχεία ή διαθέτουν γνωρίσματα.
 - **Απλού τύπου**: δεν έχουν υποστοιχεία, ούτε διαθέτουν γνωρίσματα.
- Τα γνωρίσματα θεωρούνται ότι είναι απλού τύπου.
- Η XML Schema παρέχει το στοιχείο **complexType** για τη δημιουργία νέων σύνθετων τύπων. Οι σύνθετοι τύποι που δημιουργούνται αποκτούν όνομα μέσω του προαιρετικού γνωρίσματος **name**.
 - Οι τύποι **TABtype** και **studentType** είναι σύνθετοι.

- Με την ακόλουθη έκφραση ορίζεται ο σύνθετος τύπος **TABtype** :
`<xs:complexType name="TABtype">`

...
`</xs:complexType>`

Το **TABtype** χρησιμοποιείται σε άλλο σημείο της περιγραφής σχήματος για να δηλώσουμε ότι ένα στοιχείο είναι τύπου **TABtype**.

- **Παράδειγμα**: Στη δήλωση:

```
<xs:element name="TAB" type="TABtype"/>
```

δηλώνεται ότι το στοιχείο **TAB** είναι τύπου **TABtype**.

- Το όνομα ενός σύνθετου τύπου μπορεί χρησιμοποιηθεί στη δήλωση πολλών στοιχείων μειώνοντας έτσι το μέγεθος των περιγραφών.

Δηλώσεις γνωρισμάτων

- Για τη δήλωση γνωρισμάτων χρησιμοποιείται το στοιχείο **attribute**. Ένα γνώρισμα δηλώνεται ως εξής:

```
<xs:attribute name=" ... " type="..." .... />
```

 - **Παράδειγμα:** Με την έκφραση

```
<xs:attribute name="ηλικία" type="xs:positiveInteger" use="required"/>
```

δηλώνεται το γνώρισμα **ηλικία** το οποίο παίρνει τιμές του τύπου **positiveInteger**.
- Ο τύπος **positiveInteger**, είναι απλός τύπος και αντιπροσωπεύει τους θετικούς ακέραιους αριθμούς.
- Η παράσταση **use="required"**, δηλώνει ότι η εμφάνιση του συγκεκριμένου γνωρίσματος είναι υποχρεωτική.
- Ένα στοιχείο που διαθέτει γνωρίσματα είναι πάντα σύνθετου τύπου και επομένως δηλώνεται μέσω του **complexType**.
- Οι δηλώσεις γνωρισμάτων τοποθετούνται στο τέλος της δήλωσης του σύνθετου τύπου του αντίστοιχου στοιχείου, ακριβώς πριν από το **</xs:complexType>**.

Δηλώσεις γνωρισμάτων (συνέχεια)

- **Παράδειγμα:** Το στοιχείο **φοιτητής** στην παρακάτω δήλωση εμφανίζεται να διαθέτει το γνώρισμα **ηλικία**:

```
<xs:element name="φοιτητής" minOccurs=0 maxOccurs="unbounded">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element name="όνομα" type="xs:string"/>  
      <xs:element name="επώνυμο" type="xs:string"/>  
    </xs:sequence>  
    <xs:attribute name="ηλικία" type="xs:positiveInteger"  
      use="required"/>  
  </xs:complexType>  
</xs:element>
```

Παρατηρήστε ότι το στοιχείο **φοιτητής** περιλαμβάνει τα υποστοιχεία **όνομα** και **επώνυμο**, καθώς και το γνώρισμα **ηλικία**, το οποίο εμφανίζεται υποχρεωτικά σε κάθε στοιχείο **φοιτητής**, και παίρνει ως τιμή έναν θετικό ακέραιο.

Συχνότητα εμφάνισης στοιχείων

- Η XML Schema επιτρέπει να προδιαγράψουμε το πλήθος των εμφανίσεων των στοιχείων με εξαιρετική ακρίβεια. Αυτό γίνεται μέσω δύο γνωρισμάτων του **element**:
 - Του **minOccurs** μέσω του οποίου δηλώνουμε τον ελάχιστο αριθμό εμφανίσεων του στοιχείου,
 - Του **maxOccurs** μέσω του οποίου δηλώνουμε τον μέγιστο αριθμό εμφανίσεων του στοιχείου.
 - Οι τιμές των γνωρισμάτων αυτών είναι μη αρνητικοί ακέραιοι.
 - Όταν κάποιο από τα **minOccurs** ή **maxOccurs** παραλείπεται (και τα δύο ή ένα από αυτά) τότε θεωρείται ότι έχει σαν τιμή το **1**.
 - **Παράδειγμα.** Στη δήλωση:

```
<xs:element name="φοιτητής" type="studentType"
              minOccurs=0 maxOccurs="unbounded"/>
```

η τιμή **unbounded** του **maxOccurs** υποδηλώνει απεριόριστο αριθμό εμφανίσεων του **φοιτητής**, (είναι δυνατό να μην έχουμε καμιά εμφάνιση λόγω της τιμής **0** του γνωρίσματος **minOccurs**).

Περιορισμοί συχνότητας εμφάνισης στοιχείων και γνωρισμάτων

- Ένα γνώρισμα μπορεί να εμφανίζεται σε ένα στιγμιότυπο στοιχείου μία φορά ή να μην εμφανίζεται καθόλου. Δεν μπορεί όμως το ίδιο γνώρισμα να εμφανίζεται στο ίδιο στοιχείο περισσότερες από μια φορές (με τη ίδια ή διαφορετική τιμή).
- Η XML Schema επιτρέπει να προσδιορίσουμε την υποχρεωτικότητα ή μη της εμφάνισης ενός γνωρίσματος μέσω του γνωρίσματος **use** του στοιχείου **attribute** (μέσω του οποίου δηλώνουμε το γνώρισμα).
- Η τιμή **required** στο γνώρισμα **use** υποδηλώνει υποχρεωτική εμφάνιση του γνωρίσματος ενώ τιμή **optional** υποδηλώνει προαιρετική εμφάνιση του συγκεκριμένου γνωρίσματος.

Απλοί τύποι

- Η XML Schema διαθέτη πλούσια συλλογή ενσωματωμένων απλών τύπων όπως οι `byte`, `integer`, `positiveInteger`, `negativeInteger`, `int`, `decimal`, `long`, `float`, `double`, `boolean`, `date`, `dateTime`, `ID`, `string`, `IDREF`, `IDREFS`, `ENTITY`, `ENTITIES`, `NMTOKEN`, `NMTOKENS`.
- Η XML Schema επιτρέπει να ορίσουμε νέους απλούς τύπους μέσω του στοιχείου `simpleType`. Οι δηλώσεις περιγράφουν τον τρόπο που παράγονται οι νέοι τύποι από άλλους απλούς (ενσωματωμένους ή παραγόμενους) τύπους.

- **Παράδειγμα.** Με τη δήλωση:

```
<xs:simpleType name="myInteger">  
  <xs:restriction base="xs:integer">  
    <xs:minInclusive value="0"/>  
    <xs:maxInclusive value="100"/>  
  </xs:restriction>  
</xs:simpleType>
```

ορίζεται ο τύπος `myInteger`, μέσω επιβολής περιορισμών στον τύπο `integer`. Το όνομα του τύπου βάσης δίνεται ως τιμή στο γνώρισμα `base` του στοιχείου `restriction`. Οι περιορισμοί ορίζονται μέσω των στοιχείων `minInclusive` και `maxInclusive` και ορίζουν ως ελάχιστη επιτρεπτή τιμή το `0`, και ως μέγιστη επιτρεπτή τιμή το `100`.

Απλοί τύποι (συνέχεια)

- Η XML Schema παρέχει πλούσια ποικιλία «όψεων» (facets) (όπως οι `minInclusive` και `maxInclusive`) για την επιβολή περιορισμών κατά τον ορισμό νέων τύπων.
- Μια χρήσιμη όψη είναι το στοιχείο `enumeration` που περιορίζει έναν απλό τύπο σε ένα σύνολο διακριτών τιμών:

- **Παράδειγμα.** Στην περιγραφή:

```
<xs:simpleType name="νόμισμα">  
  <xs:restriction base="xs:string">  
    <xs:enumeration value="GRD"/>  
    <xs:enumeration value="EURO"/>  
    <xs:enumeration value="USD"/>  
    <!-- κ.λ.π. ... -->  
  </xs:restriction>  
</xs:simpleType>
```

ορίζεται ο απλός τύπος `νόμισμα`, με τιμές τα (διεθνή) σύμβολα των νομισμάτων όπως τα `GRD`, `EURO`, `USD`, κ.λ.π..

Ορισμός γνωρισμάτων σε στοιχεία απλού τύπου

- Στοιχεία απλού τύπου δηλώνονται όπως στο παράδειγμα:
`<xs:element name="ονοματεπώνυμο" type="xs:string"/>`
- Οι απλοί τύποι (στοιχεία απλού τύπου) δεν επιτρέπεται να έχουν γνωρίσματα. Η επισύναψη γνωρισμάτων σε στοιχεία απλού τύπου (π.χ. το γνώρισμα **AΔΤ**-αριθμός δελτίου ταυτότητας στο στοιχείο **ονοματεπώνυμο**) γίνεται με τον ορισμό σύνθετου τύπου όπως στο επόμενο παράδειγμα. Το περιεχόμενο του στοιχείου παραμένει απλού τύπου (του τύπου **string** στην περίπτωση μας).

```
<xs:element name="ονοματεπώνυμο">  
  <xs:complexType>  
    <xs:simpleContent>  
      <xs:extension base="xs:string">  
        <xs:attribute name="AΔΤ" type="xs:string"/>  
      </xs:extension>  
    </xs:simpleContent>  
  </xs:complexType>  
</xs:element>
```

Το **simpleContent** υποδηλώνει στοιχεία με απλό περιεχόμενο χωρίς υπο-στοιχεία (στο παράδειγμα ακολουθία απλών χαρακτήρων). Για τον ορισμό του νέου τύπου, επεκτείνουμε τύπο **string**. Η επέκταση συνίσταται στην προσθήκη του γνωρίσματος **AΔΤ** μέσω μιας συνήθους δήλωσης γνωρίσματος.

Ομαδοποίηση στοιχείων

- Η XML Schema παρέχει τη δυνατότητα να ορίζουμε ομάδες στοιχείων (στις οποίες μπορούμε να δίνουμε και ονόματα). Στόχος η χρήση αυτών των ομάδων για τη δόμηση του περιεχομένου των συνθέτων τύπων.

Ομαδοποίηση στοιχείων με το *sequence*

- Το στοιχείο *sequence* χρησιμοποιείται για να δηλώσει (διατεταγμένη) ακολουθία (υπο)στοιχείων.
- **Παράδειγμα.** Στο παρακάτω:

```
<xs:complexType name="studentType">  
  <xs:sequence>  
    <xs:element name="όνομα" type="xs:string"/>  
    <xs:element name="επώνυμο" type="xs:string"/>  
    <xs:element name="πατρώνυμο" type="xs:string"/>  
  </xs:sequence>  
</xs:complexType>
```

δηλώνεται ότι κάθε στοιχείο τύπου *studentType* πρέπει να περιλαμβάνει ένα υποστοιχείο *όνομα* ακολουθούμενο από ένα υποστοιχείο *επώνυμο* και από ένα υποστοιχείο *πατρώνυμο*. Και τα τρία (υπο)στοιχεία είναι τύπου *string*.

Ομαδοποίηση στοιχείων με το **choice**

- Το στοιχείο **choice** επιτρέπει την εμφάνιση κάθε φορά ενός μόνο από τα (υπο)στοιχεία που περιλαμβάνει, ως περιεχόμενο του στοιχείου που ανήκει στον συγκεκριμένο τύπο.
- **Παράδειγμα.** Στο παρακάτω τμήμα περιγραφής:

```
<xs:complexType name="studentType">
  <xs:choice>
    <xs:element name="ονοματεπώνυμο" type="xs:string"/>
    <xs:sequence>
      <xs:element name="όνομα" type="xs:string"/>
      <xs:element name="επώνυμο" type="xs:string"/>
    </xs:sequence>
  </xs:choice>
</xs:complexType>
```

δηλώνεται ότι κάθε στοιχείο του τύπου **studentType** έχει ως περιεχόμενο είτε το στοιχείο **ονοματεπώνυμο** είτε ένα στοιχείο **όνομα** ακολουθούμενο από ένα στοιχείο **επώνυμο**.

Ομαδοποίηση στοιχείων με το **all**

- Με το στοιχείο **all** δηλώνεται ότι κάθε στοιχείο της ομάδας εμφανίζεται υποχρεωτικά (ή προαιρετικά αν συνοδεύεται με το γνώρισμα **minOccurs=0**). Η σειρά εμφάνισης δεν παίζει ρόλο. Επιτρεπτές τιμές των **minOccurs** και **maxOccurs** στις δηλώσεις των στοιχείων του **all** είναι οι **0** και **1**.
 - **Παράδειγμα.** Στο παρακάτω τμήμα περιγραφής:

```
<xs:complexType name="studentType">  
  <xs:all>  
    <xs:element name="όνομα" type="xs:string"/>  
    <xs:element name="επώνυμο" type="xs:string"/>  
    <xs:element name="πατρώνυμο" type="xs:string" minOccurs=0/>  
  </xs:all>  
</xs:complexType>
```

δηλώνεται ότι κάθε στοιχείο του τύπου **studentType** περιλαμβάνει από μια εμφάνιση καθενός από τα στοιχεία **όνομα**, **επώνυμο** και μια ή καμία εμφάνιση του στοιχείου **πατρώνυμο** αλλά με οποιαδήποτε σειρά.

Ομαδοποίηση στοιχείων με το **all**

(συνέχεια)

- Το **all** πρέπει να εμφανίζεται ως το μοναδικό υποστοιχείο του **complexType**.
 - **Παράδειγμα.** Δεν είναι επιτρέπονται δηλώσεις της μορφής:

```
<xs:complexType name="studentType">  
  <xs:all>  
    <xs:element name="όνομα" type="xs:string"/>  
    <xs:element name="επώνυμο" type="xs:string"/>  
    <xs:element name="πατρώνυμο" type="xs:string"/>  
  </xs:all>  
  <xs:sequence>  
    <xs:element name="ηλικία" type="xs:positiveInteger"/>  
    <xs:element name="διεύθυνση" type="xs:string"/>  
  </xs:sequence>  
</xs:complexType>
```
- Το περιεχόμενο του **all** μπορεί να είναι μόνο δηλώσεις ξεχωριστών στοιχείων και όχι ομάδων στοιχείων.

Ομαδοποίηση γνωρισμάτων

- Το στοιχείο `attributeGroup` επιτρέπει ομαδοποίηση γνωρισμάτων.
- **Παράδειγμα.** Εδώ ορίζεται ομάδα γνωρισμάτων με όνομα `personAttributes`. Περιλαμβάνει το γνώρισμα `AΔΤ` που είναι τύπου `string`, το `AΦΜ` που είναι τύπου `positiveInteger`, και το `φύλο` για το οποίο ορίζεται ανώνυμος τύπος που περιλαμβάνει τις τιμές `άνδρας` και `γυναίκα`:

```
<xs:attributeGroup name="personAttributes">
  <xs:attribute name="AΔΤ" type="xs:string" use="required"/>
  <xs:attribute name="AΦΜ" type="xs:positiveInteger"/>
  <xs:attribute name="φύλο">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="άνδρας"/>
        <xs:enumeration value="γυναίκα"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:attributeGroup>
```

Ομαδοποίηση γνωρισμάτων (συνέχεια)

- Το όνομα ομάδας γνωρισμάτων μπορεί να χρησιμοποιηθεί για να δηλωθεί ότι ένα στοιχείο διαθέτει τα γνωρίσματα που περιλαμβάνει η συγκεκριμένη ομάδα.
- **Παράδειγμα.** Στην παρακάτω περιγραφή γίνεται επίκληση της δήλωσης του συνόλου γνωρισμάτων `personAttributes`, μέσω του γνωρίσματος `ref`, προκειμένου να δηλώσουμε ότι το στοιχείο `φοιτητής` διαθέτει τα γνωρίσματα του συνόλου `personAttributes`:

```
<xs:element name="φοιτητής" minOccurs=0 maxOccurs="unbounded" >
  <xs:complexType>
    <xs:sequence>
      <xs:element name="όνομα" type="xs:string"/>
      <xs:element name="επώνυμο" type="xs:string"/>
    </xs:sequence>
    <xs:attributeGroup ref="personAttributes"/>
  </xs:complexType>
</xs:element>
```

Ομαδοποίηση γνωρισμάτων

(συνέχεια)

- Για να ορίσουμε και άλλα στοιχεία που διαθέτουν την ίδια ομάδα γνωρισμάτων (π.χ. τα στοιχεία **μέλος Δεπ, εργαζόμενος** κ.λ.π.), χρησιμοποιούμε δηλώσεις ορισμού στοιχείων παρόμοιες με αυτήν του στοιχείου **φοιτητής**, στις οποίες θα υπάρχει απλή αναφορά στην ίδια ομάδα γνωρισμάτων, χωρίς να χρειάζεται να οριστεί ξανά αυτή.
- Χρησιμοποιώντας ομάδες γνωρισμάτων διαμορφώνουμε πιο ευανάγνωστες περιγραφές σχήματος και διευκολύνουμε τη μελλοντική ενημέρωση του σχήματος, αφού μια ομάδα γνωρισμάτων μπορεί να οριστεί (και να τροποποιείται) σε ένα σημείο και να χρησιμοποιείται με επίκληση του ονόματος της σε πολλούς ορισμούς στοιχείων.
- Μια ομάδα γνωρισμάτων μπορεί να περιλαμβάνει με τη σειρά της άλλες ομάδες γνωρισμάτων.
- Οι δηλώσεις γνωρισμάτων και οι αναφορές σε ομάδες γνωρισμάτων πρέπει να τοποθετούνται στο τέλος των ορισμών συνθέτων τύπων.

Περίληψη

- Το XML Σχήμα (*XML Schema*) ενοποιεί προηγούμενες προτάσεις για περιγραφή σχήματος.
- Γενικεύει τα DTDs.
- Χρησιμοποιεί τη σύνταξη της XML.
- Τεχνικά εγχειρίδια:
 - «XML Schema Part 0: Primer» βρίσκεται στη διεύθυνση: <http://www.w3.org/TR/xmlschema-0>
 - «XML Schema Part 1: Structures» βρίσκεται στη διεύθυνση: <http://www.w3.org/TR/xmlschema-1>
 - «XML Schema Part 2: Datatypes» βρίσκεται στη διεύθυνση: <http://www.w3.org/TR/xmlschema-2>