ΜΕΡΟΣ ΙΙ:

XML
DTD
XML Namespaces
XML Schema

Πίνακας Περιεχομένων

ΚΕΦΑΛΑΙΟ 3:
EXtensible Markup Language (XML)

3.1. Εισαγωγή

Η e***X***tensible ***M***arkup ***L***anguage (XML) αποτελεί μια εξαιρετικά απλή διάλεκτο της γλώσσας *Standard Generalized Markup Language* (SGML), η οποία αναπτύχθηκε με στόχο να διευκολύνει το χειρισμό, επεξεργασία, διακίνηση και αποθήκευση τεκμηρίων στον *Παγκόσμιο Ιστό* (web). Στα λιγότερα από 10 χρόνια που μεσολάβησαν από την δημιουργία της η XML καθιερώθηκε ως πρότυπο για την αναπαράσταση και ανταλλαγή δεδομένων στον Παγκόσμιο Ιστό και απετέλεσε την βάση για την ανάπτυξη μιας σειράς προτύπων στα οποία συμπεριλαμβάνονται πρότυπα μεταδεδομένων, γλώσσες οντολογιών, πρότυπα ηλεκτρονικής συγγραφής κ.α.
Στο κεφάλαιο αυτό περιγράφονται οι βασικοί κανόνες σύνταξης της γλώσσας XML, και παρέχονται απλά παραδείγματα που εισάγουν τον αναγνώστη στη χρήσης της γλώσσας για την αναπαράσταση δεδομένων. Η XML αποτελεί πρότυπο που προτείνεται από την κοινοπραξία *World Wide Web Consortium*[1] (W3C).

---

[1] Το *World Wide Web Consortium* (W3C) είναι μια διεθνής κοινοπραξία η οποία δημιουργήθηκε το 1994 από τον Tim Berners-Lee, τον δημιουργό του *Παγκόσμιου Ιστού* (Web). Σήμερα το W3C έχει ως μέλη περίπου 400 οργανισμούς από όλο τον κόσμο και έχει αποκτήσει τη διεθνή αναγνώριση για τη προσφορά του στην ανάπτυξη του Παγκόσμιου Ιστού. Στους μακροπρόθεσμους στόχους του W3C για τον Παγκόσμιο Ιστό αναφέρονται:

- «*Η προσπάθεια να γίνει ο Παγκόσμιος Ιστός προσπελάσιμος από όλους (Universal Access) μέσω της προώθησης τεχνολογιών οι οποίες λαμβάνουν υπ' όψιν τις τεράστιες διαφορές στην κουλτούρα, τη γλώσσα, τη μόρφωση, τις ικανότητες, τα υλικά μέσα, τις συσκευές προσπέλασης, και τους φυσικούς περιορισμούς των χρηστών σε όλες τις ηπείρους.*
- *Ο Σημασιολογικός Ιστός* (Semantic Web)*: Η ανάπτυξη ενός περιβάλλοντος λογισμικού που επιτρέπει σε κάθε χρήστη να χρησιμοποιεί με τον καλύτερο δυνατό τρόπο τους πόρους που είναι διαθέσιμοι στον Παγκόσμιο Ιστό.*
- *Να καθοδηγήσει την ανάπτυξη του Παγκόσμιου Ιστού λαμβάνοντας προσεκτικά υπ' όψιν τις νέες νομικές, εμπορικές και κοινωνικές πλευρές που προκύπτουν από αυτήν την τεχνολογία.*»

Το W3C επικεντρώνει τις προσπάθειες του σε τρεις βασικές κατευθύνσεις: Στην προώθηση και παραπέρα ανάπτυξη του οράματος του για το μέλλον του Παγκόσμιου Ιστού, στη σχεδίαση τεχνολογιών που προωθούν αυτό το όραμα και στην δημιουργία προτύπων για τις τεχνολογίες του Παγκόσμιου Ιστού με τη δημιουργία προδιαγραφών (οι οποίες ονομάζονται "Recommendations" που περιγράφουν τα δομικά στοιχεία του Παγκόσμιου Ιστού. Οι προδιαγραφές αυτές καθώς και άλλες πληροφορίες διατίθενται ελευθέρα από την ιστοσελίδα του W3C στη διεύθυνση http://www.w3.org. Επίσης πληροφορίες παρέχονται και από την ιστοσελίδα του Ελληνικού Γραφείου του W3C στην διεύθυνση http://www.w3c.gr.

3.2. Πώς ξεκινά ένα XML τεκμήριο

Ένα *XML τεκμήριο* (XML document) πρέπει να ξεκινά με μια *δήλωση XML* (XML declaration). Στη δήλωση αυτή, η οποία σηματοδοτεί την έναρξη του XML τεκμηρίου, περιλαμβάνονται πληροφορίες οι οποίες αφορούν, την *έκδοση* (version) της γλώσσας XML που χρησιμοποιείται.

Παράδειγμα 3.1. Το XML τεκμήριο που ακολουθεί είναι ένα πλήρες XML τεκμήριο:

```
<?xml version="1.0"?>
<greeting>Hello, world!</greeting>
```

❑

Ένα XML τεκμήριο θα λέμε ότι είναι *καλά διαμορφωμένο* (well-formed) όταν ικανοποιεί τους κανόνες σύνταξης των XML τεκμηρίων. Σε επόμενο κεφάλαιο θα αναφερθούμε στη δυνατότητα που δίνεται στον χρήστη να ορίζει επιπλέον περιορισμούς που θα πρέπει να πληροί ένα XML τεκμήριο με τη βοήθεια των **D**ocument **T**ype **D**eclarations (DTDs). Τα DTDs αποτελούν τμήμα της XML και παρέχουν στο χρήστη τη δυνατότητα να ορίζει δικούς του περιορισμούς σύνταξης οι οποίοι επιβάλλονται στα XML τεκμήρια.
Στην περίπτωση που ένα XML τεκμήριο συνοδεύεται και από ένα DTD, τότε εφόσον πληροί και τους επιπλέον περιορισμούς που τίθενται από το DTD, το XML τεκμήριο θα λέμε ότι είναι *έγκυρο* (valid).
Στις κεφάλαιο αυτό θα περιγράψουμε τους κανόνες σύνταξης της XML. Για τη σύνταξη και τη χρήση των DTD θα αναφερθούμε στο επόμενο κεφάλαιο.

3.3. Στοιχεία στην XML

Η βασική δομική μονάδα της XML είναι το *στοιχείο* (element), δηλαδή ένα κομμάτι κειμένου που περικλείεται ανάμεσα σε *ετικέτες* (tags) που ταιριάζουν μεταξύ τους όπως οι ετικέτες **<φοιτητής>** και **</φοιτητής>** ή οι ετικέτες **<όνομα>** και **</όνομα>** ή τέλος οι ετικέτες **<επώνυμο>** και **</επώνυμο>** στο Παράδειγμα 3.2 και το Παράδειγμα 3.3 που ακολουθούν:

Παράδειγμα 3.2. Ένα απλό στοιχείο:

```
<φοιτητής>Νίκος  Νικολάου</φοιτητής>
```

❑

Παράδειγμα 3.3. Ένα λίγο πιο «σύνθετο» στοιχείο:

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

❑

Στα πιο πάνω παραδείγματα μπορούμε να παρατηρήσουμε τα ακόλουθα: Ένα στοιχείο ξεκινά με μια ετικέτα, όπως η ετικέτα **<φοιτητής>**, η οποία ονομάζεται *ετικέτα αρχής* (start-tag), αφού υποδηλώνει την αρχή του στοιχείου, και τελειώνει με μια ετικέτα όπως η ετικέτα **</φοιτητής>**, η οποία ονομάζεται *ετικέτα τέλους* (end-tag), και υποδηλώνει το τέλος του στοιχείου. Το *περιεχόμενο* (element content), δηλαδή ότι βρίσκεται μεταξύ της ετικέτας αρχής και της αντίστοιχης ετικέτας τέλους., μπορεί να είναι είτε απλό κείμενο, όπως στην περίπτωση του στοιχείου **φοιτητής** στο Παράδειγμα 3.2, ένα η περισσότερα άλλα στοιχεία όπως στην περίπτωση του στοιχείου **φοιτητής** στο Παράδειγμα 3.3. Στην πρώτη περίπτωση θα λέμε ότι έχουμε ένα *απλό στοιχείο* ενώ στη δεύτερη ένα *σύνθετο στοιχείο*. Είναι επίσης επιτρεπτό το περιεχόμενο ενός στοιχείου να είναι *ανάμικτο* (mixed), δηλαδή να αποτελείται από απλό κείμενο αναμεμιγμένο με στοιχεία, όπως φαίνεται στο επόμενο παράδειγμα.

Παράδειγμα 3.4. Ένα στοιχείο με ανάμικτο περιεχόμενο:

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

❑

Τα στοιχεία που αποτελούν το περιεχόμενο ενός άλλου στοιχείου μπορεί με τη σειρά τους να είναι απλά, σύνθετα ή στοιχεία με ανάμικτο περιεχόμενο.
Συχνά χρησιμοποιείται και ο όρος *υποστοιχείο* (subelement) προκειμένου να εκφράσει τη σχέση μεταξύ ενός στοιχείου και του στοιχείου το οποίο το περιλαμβάνει. Έτσι, το στοιχείο **<όνομα>** ... **</όνομα>** στο Παράδειγμα 3.3 είναι υποστοιχείο του στοιχείου **<φοιτητής>** ... **</φοιτητής>**.
Βασική προϋπόθεση ώστε ένα XML τεκμήριο να είναι καλά διαμορφωμένο είναι να ισχύουν για κάθε στοιχείο οι ακόλουθες συνθήκες:

1. Οι ετικέτες του τεκμηρίου πρέπει να είναι ισορροπημένες με την έννοια ότι σε κάθε ετικέτα αρχής πρέπει να αντιστοιχεί μια ετικέτα τέλους η οποία να βρίσκεται μετά την ετικέτα αρχής μέσα στο τεκμήριο.
2. Αν μια ετικέτα αρχής E1 εμφανίζεται νωρίτερα από μια ετικέτα αρχής E2 στο XML τεκμήριο, τότε η ετικέτα τέλους που αντιστοιχεί στην E1

εμφανίζεται αργότερα στο τεκμήριο από την ετικέτα τέλους που αντιστοιχεί στην Ε2. Επομένως, οι ετικέτες τέλους πρέπει να εμφανίζονται με την ανάστροφη σειρά από αυτήν που εμφανίζονται οι αντίστοιχες ετικέτες αρχής.

Παρατηρήστε στο Παράδειγμα 3.3 ότι η ετικέτα αρχής **<φοιτητής>** εμφανίζεται πριν από την ετικέτα αρχής **<όνομα>**, ενώ η ετικέτα τέλους **</φοιτητής>** εμφανίζεται μετά την ετικέτα τέλους **</όνομα>**.
Στην XML είναι επιτρεπτό να χρησιμοποιούμε επαναλαμβανόμενα στοιχεία με τις ίδιες ετικέτες για να αναπαραστήσουμε συλλογές ομοειδών αντικειμένων όπως φαίνεται στο παράδειγμα που ακολουθεί.

Παράδειγμα 3.5. Για να αναπαραστήσουμε το σύνολο των φοιτητών του ΤΑΒ θα μπορούσαμε να κατασκευάσουμε ένα XML τεκμήριο της μορφής:

```
<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>
```

❑

Σημειώστε ότι η XML επιτρέπει *κενά στοιχεία* (empty elements) δηλαδή στοιχεία με κενό περιεχόμενο. Τα στοιχεία αυτά έχουν τη μορφή:

```
<ετικέτα></ετικέτα>
```

Για τα στοιχεία αυτά παρέχεται και η συντομογραφία **<ετικέτα/>**.
Αξίζει να σημειώσουμε στο σημείο αυτό ότι μέσω των ετικετών εκφράζουμε τη λογική δομή ενός τεκμηρίου.

3.4. Γνωρίσματα στην XML

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

Παράδειγμα 3.6. Στοιχείο με γνώρισμα:

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

❑

Όπως και στα στοιχεία έτσι και στα γνωρίσματα ο χρήστης μπορεί να χρησιμοποιήσει ονόματα της επιλογής του σαν ονόματα γνωρισμάτων. Οι τιμές των γνωρισμάτων περικλείονται πάντα ανάμεσα σε απλά ή διπλά εισαγωγικά. Ένα στοιχείο είναι δυνατόν να διαθέτει περισσότερα του ενός γνωρίσματα όπως φαίνεται στο παράδειγμα που ακολουθεί:

Παράδειγμα 3.7. Στοιχεία με πολλαπλά γνωρίσματα:

```
<book isbn = "1-55860-622-X"  language = "English" >
      <title> Data on the Web </title>
      <price currency = "USD"> 100 </price>
</book>
```

❑

Παρατηρήστε στο Παράδειγμα 3.7 ότι το στοιχείο book διαθέτει τα γνωρίσματα isbn και language, ενώ στο στοιχείο price διαθέτει το γνώρισμα currency.
Πρέπει να τονιστεί ότι ενώ ένα στοιχείο μπορεί να διαθέτει υποστοιχεία με το ίδιο όνομα, εν τούτοις δεν επιτρέπεται σε ένα στοιχείο να επισυνάπτονται περισσότερα του ενός γνωρίσματα με το ίδιο όνομα. Θα πρέπει ακόμη να σημειώσουμε ότι η σειρά με την οποία παρατίθενται τα γνωρίσματα ενός στοιχείου δεν παίζει κανένα ρόλο. Αντίθετα, η σειρά με την οποία παρατίθενται τα υποστοιχεία ενός στοιχείου θεωρείται σημαντική.
Αξίζει επίσης να τονιστεί ότι μια δεδομένη πληροφορία μπορεί να κωδικοποιηθεί με πολλούς διαφορετικούς τρόπους σε XML.

Παράδειγμα 3.8. Η πληροφορία που περιλαμβάνεται στο στοιχείο:

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

θα μπορούσε επίσης να κωδικοποιηθεί με το στοιχείο:

```
<φοιτητής>
   <ΑΜ> 12345 </ΑΜ>
   <όνομα> Νίκος </όνομα>
   <επώνυμο> Νικολάου </επώνυμο>
</φοιτητής>
```

ή ακόμη και με το κενό στοιχείο:

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

Στην τελευταία περίπτωση έχουμε ένα κενό στοιχείο, ένα στοιχείο το οποίο δεν έχει περιεχόμενο έχει όμως γνωρίσματα.
Στη πράξη τα κενά στοιχεία κωδικοποιούν συνήθως κάποια πληροφορία μέσω γνωρισμάτων.

❑

3.5. Άλλα συντακτικά στοιχεία της XML

3.5.1. Σχόλια

Τα *σχόλια* (comments) επιτρέπονται οπουδήποτε εκτός από το εσωτερικό των ετικετών. Ένα σχόλιο ξεκινά με το <!--και τελειώνει με το -->.

Παράδειγμα 3.9. Το ακόλουθο είναι ένα σχόλιο:

```
<!-- Αυτό είναι ένα σχόλιο -->
```

❑

Τα σχόλια συνήθως τοποθετούνται για να κάνουν το τεκμήριο πιο ευανάγνωστο από τον χρήστη.

3.5.2. 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 επεξεργασίες, παραμέτρους κ.λ.π.) οι οποίες θέλουμε να περάσουν στην εφαρμογή.

Παράδειγμα 3.10. Η οδηγία επεξεργασίας που ακολουθεί απευθύνεται στην εφαρμογή xml-stylesheet:

```
<?xml-stylesheet href="book.css" type="text/css"?>
```

❑

Οι οδηγίες επεξεργασίας δεν έχουν κάποια άλλη σημασία εκτός από το ότι απευθύνονται και παρέχουν πληροφορίες σε συγκεκριμένες εφαρμογές.

3.5.3. 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 3.11. Το ακόλουθο είναι ένα σωστό συντακτικά τμήμα XML τεκμηρίου που χρησιμοποιεί το περιβάλλον CDATA:

```
<![CDATA[ <αρχή> αυτό είναι ένα λανθασμένο στοιχείο </τέλος>]]>
```

❑

3.5.4. Αναφορές Οντοτήτων

Όπως είδαμε στις προηγούμενες παραγράφους ορισμένοι χαρακτήρες έχουν ειδική σημασία στην XML. Για παράδειγμα ο χαρακτήρας < υποδηλώνει την έναρξη μιας ετικέτας ενώ ο χαρακτήρας > υποδηλώνει το τέλος της ετικέτας. Εκτός από τους χαρακτήρες < και >, μερικοί ακόμη χαρακτήρες όπως οι &, ', ", που θα συναντήσουμε στη συνέχεια έχουν επίσης ειδική σημασία στην XML. Τίθεται επομένως το ερώτημα τι γίνεται αν εμείς θέλουμε να τοποθετήσουμε κάποιον από αυτούς τους χαρακτήρες στο τεκμήριο μας όχι με την ειδική σημασία που του αποδίδει η XML, αλλά απλά σαν ένα κοινό χαρακτήρα μέσα στο κείμενο; Η XML μας παρέχει έναν τρόπο για να γίνει αυτό δυνατό μέσω ενός εναλλακτικού συμβολισμού των χαρακτήρων αυτών με τη βοήθεια των *οντοτήτων* (entities). Προκειμένου να χρησιμοποιηθεί μια οντότητα σε ένα σημείο του XML τεκμηρίου θα πρέπει να γίνει αναφορά στο όνομα της. Η *αναφορά οντότητας* (entity reference) ξεκινά με το σύμβολο &, ακολουθείται από το όνομα της οντότητας, και τελειώνει με το σύμβολο ;. Για παράδειγμα το < αποτελεί αναφορά στην οντότητα με όνομα lt η οποία αναπαριστά το σύμβολο <. Επίσης με τα >, &, ', ", αναφερόμαστε στις

οντότητες που αναπαριστούν τα σύμβολα >, &, ', ", αντίστοιχα. Οι οντότητες αυτές ονομάζονται *εσωτερικές οντότητες* (internal entities) γιατί είναι ενσωματωμένες στην XML και μπορούμε να τις χρησιμοποιήσουμε (να αναφερθούμε σ' αυτές) χωρίς να χρειαστεί να τις δηλώσουμε προηγουμένως (στο DTD).
Εκτός από τη χρήση των οντοτήτων που περιγράψαμε πιο πριν, οι οντότητες μπορούν 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 αυτές, που ονομάζονται και *εξωτερικές οντότητες* (external entities), 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. Γενικά οι οντότητες στην XML μοιάζουν με τις μακροεντολές των γλωσσών προγραμματισμού[2].

3.5.5. Αναφορές Χαρακτήρων

Παρόμοια μορφή με τις αναφορές οντότητας έχουν και οι *αναφορές χαρακτήρων* (character references). Οι αναφορές χαρακτήρων χρησιμοποιούνται για την εισαγωγή οποιουδήποτε χαρακτήρα του ISO/IEC 10646 συνόλου χαρακτήρων 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. Αν όμως ξεκινά απλά με το &# τότε τα ψηφία που ακολουθούν μέχρι το σύμβολο τερματισμού ; παρέχουν τη δεκαδική αναπαράσταση του χαρακτήρα στο ISO/IEC 10646. Ένα παράδειγμα δεκαεξαδικής αναφοράς χαρακτήρα είναι το ℞, ενώ ένα παράδειγμα δεκαδικής αναφοράς χαρακτήρα είναι το ℞. Η δυνατότητα να έχουμε αναφορές χαρακτήρων είναι χρήσιμη αφού μέσω αυτής μπορούμε να εισάγουμε χαρακτήρες οι οποίοι δεν είναι προσπελάσιμοι από τις διαθέσιμες συσκευές εισόδου.

3.6. Δηλώσεις Τύπου Τεκμηρίων (DTD)

Βασικό χαρακτηριστικό που καθιστά την XML ιδιαίτερα ισχυρή γλώσσα αποτελεί το γεγονός ότι επιτρέπει στον χρήστη να ορίσει και να χρησιμοποιήσει ονόματα στοιχείων και γνωρισμάτων της αρεσκείας του

[2] Μια μακροεντολή σε μια γλώσσα προγραμματισμού αντιπροσωπεύει ένα κομμάτι κώδικα (ακολουθία εντολών της γλώσσας προγραμματισμού). Το κομμάτι του κώδικα που αντιστοιχεί στην μακροεντολή ενσωματώνεται στο πρόγραμμα (αντικαθιστά το όνομα της μακροεντολής) κατά τη μεταγλώττιση του προγράμματος.

καθώς και τις δικές του οντότητες. Όμως, για συγκεκριμένες κατηγορίες εφαρμογών, είναι χρήσιμο να τίθενται κάποιοι κοινά αποδεκτοί περιορισμοί οι οποίοι να προδιαγράφουν ένα συγκεκριμένο λεξιλόγιο από επιτρεπτά ονόματα στοιχείων και γνωρισμάτων, και να θέτουν συγκεκριμένους περιορισμούς ως προς την πολλαπλότητα εμφάνισης των στοιχείων, την μεταξύ τους σειρά κ.λ.π. Με αυτόν τον τρόπο κάθε κοινότητα ενδιαφερόντων μπορεί να προδιαγράψει τη δική της XML διάλεκτο με βάση τις ανάγκες των μελών της. Για την επιβολή τέτοιων περιορισμών απαιτείται ένας τρόπος περιγραφής των περιορισμών αυτών εκ μέρους του χρήστη, και αυτό μπορεί να γίνει με τη βοήθεια των *Δηλώσεων Τύπου Τεκμηρίων* (Document Type Definitions) (DTD). Οι δηλώσεις που διατυπώνονται σε ένα DTD παρέχουν (μετα)πληροφορία στα *προγράμματα συντακτικής ανάλυσης* (parsers) των XML τεκμηρίων, σχετική με τους περιορισμούς σύνταξης που πρέπει να πληρούν τα τεκμήρια ώστε να θεωρούνται έγκυρα ως προς το συγκεκριμένο DTD. Παρά τη σημαντική χρησιμότητα των DTD, αξίζει να σημειωθεί ότι ένα XML τεκμήριο δεν είναι υποχρεωτικό να περιλαμβάνει (ή να συνδέεται) με κάποιο DTD.

3.7. Η δήλωση "standalone"

Όπως είδαμε στις παραγράφους 3.5.4 και 3.6, συχνά ένα XML τεκμήριο επηρεάζεται από πληροφορία που βρίσκεται εκτός του τεκμηρίου αυτού (είτε πρόκειται για κάποιο DTD είτε για εξωτερικές οντότητες). Ένα τεκμήριο που δεν επηρεάζεται από εξωτερική πληροφορία θα λέμε ότι είναι *αυτόνομο* (standalone). Είναι φανερό ότι μια ένδειξη απευθυνόμενη στα προγράμματα που πρόκειται να χρησιμοποιήσουν ένα XML τεκμήριο και η οποία να τα πληροφορεί αν χρειάζονται ή όχι εξωτερική πληροφορία είναι ιδιαίτερα χρήσιμη. Η πληροφορία αυτή διατυπώνεται μέσω της δήλωσης `standalone` όπως φαίνεται στο παράδειγμα που ακολουθεί:

Παράδειγμα 3.12. Δίνοντας την τιμή "`yes`" στο `standalone` όπως στην παρακάτω δήλωση:

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

πληροφορούμε κάθε πρόγραμμα εφαρμογής που θα χρησιμοποιήσει το XML τεκμήριο που περιέχει τη δήλωση αυτή ότι δεν υπάρχουν εξωτερικές δηλώσεις οι οποίες να επηρεάζουν το τεκμήριο. Αν αντιθέτως δώσουμε την τιμή "`no`" στο `standalone` τότε αυτό σημαίνει ότι υπάρχουν ή πιθανώς να υπάρχουν τέτοιες δηλώσεις.

❑

Θα πρέπει να σημειωθεί ότι κάθε XML τεκμήριο για το οποίο ισχύει η δήλωση `standalone="no"` μπορεί να μετατραπεί σε ένα αυτόνομο τεκμήριο (με την

ενσωμάτωση όλης της απαραίτητης εξωτερικής πληροφορίας) κάτι το οποίο μπορεί να είναι επιθυμητό για κάποιες εφαρμογές.

3.8. Καλά διαμορφωμένα XML τεκμήρια

Στις προηγούμενες ενότητες παρουσιάσαμε τη (βασική) σύνταξη των XML τεκμηρίων. Ένα XML τεκμήριο το οποίο υπακούει στους κανόνες σύνταξης της XML θα λέμε ότι είναι ένα *καλά διαμορφωμένο XML τεκμήριο* (well-formed XML document).
Πρέπει να τονιστεί ότι οι κανόνες σύνταξης της XML μπορούν να διατυπωθούν με αυστηρό τρόπο χρησιμοποιώντας ένας από τους φορμαλισμούς διατύπωσης τέτοιων συντακτικών κανόνων όπως είναι για παράδειγμα η Extended BNF αναπαράσταση. Μια τέτοια αυστηρή περιγραφή είναι έξω από τους στόχους των σημειώσεων που έχετε στα χέρια σας. Ο ενδιαφερόμενος αναγνώστης μπορεί να ανατρέξει στα τεχνικά εγχειρίδια της γλώσσας [2] και σε άλλα σχετικά δημοσιεύματα [1], τα οποία περιλαμβάνονται στην προτεινόμενη βιβλιογραφία.

ΚΕΦΑΛΑΙΟ 4:
Document Type Declarations (DTD)

4.1. Εισαγωγή

Ένα DTD λειτουργεί σαν μια *γραμματική* (grammar) για μια κατηγορία XML τεκμηρίων, αφού παρέχει ένα λεξιλόγιο (τα αποδεκτά ονόματα των στοιχείων και των γνωρισμάτων) καθώς και ένα 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. Από την άλλη μεριά, αν δούμε ένα DTD από μια οπτική γωνία προερχόμενη από την περιοχή των βάσεων δεδομένων, αυτό μπορεί να εκληφθεί σαν *σχήμα* (schema), με μια σημασία παρόμοια με αυτή των σχεσιακών βάσεων δεδομένων, για τα δεδομένα τα οποία αναπαριστά ένα XML τεκμήριο.

4.2. Ένα απλό DTD

Θα ξεκινήσουμε την παρουσίαση των DTD μέσα από ένα απλό παράδειγμα:

Παράδειγμα 4.1. Ας πάρουμε το ακόλουθο τμήμα ενός XML τεκμηρίου το οποίο κωδικοποιεί στοιχεία των φοιτητών του ΤΑΒ:

```
<TAB>
        <φοιτητής>
                <όνομα> Νίκος </όνομα>
                <επώνυμο> Νικολάου </επώνυμο>
        </φοιτητής>
        <φοιτητής> ... </φοιτητής>
                ...
</TAB>
```

Ένα DTD για το τεκμήριο αυτό θα μπορούσε να είναι:

```
<!DOCTYPE TAB [
  <!ELEMENT TAB (φοιτητής*)>
  <!ELEMENT φοιτητής (όνομα, επώνυμο)>
  <!ELEMENT όνομα (#PCDATA)>
  <!ELEMENT επώνυμο (#PCDATA)>
]>
```

❑

Η πρώτη γραμμή του DTD στο 0Παράδειγμα 4.1 δηλώνει ότι το στοιχείο της ρίζας είναι το **TAB**. Η επόμενη γραμμή δηλώνει ότι το στοιχείο **TAB** μπορεί να περιλαμβάνει (αποκλειστικά και μόνο) ένα οποιοδήποτε πλήθος στοιχείων με ετικέτα **φοιτητής**. Στην τρίτη γραμμή δηλώνεται ότι το στοιχείο **φοιτητής** περιλαμβάνει το στοιχείο **όνομα** ακολουθούμενο από το στοιχείο **επώνυμο**. Τέλος, στις επόμενες δύο γραμμές δηλώνεται ότι το περιεχόμενο καθενός από τα στοιχεία **όνομα** και **επώνυμο** είναι δεδομένα χαρακτήρων (δεν περιλαμβάνουν υποστοιχεία).

4.3. Το DTD σαν γραμματική

Ένα DTD αποτελεί στην πραγματικότητα μια *γραμματική* (grammar), δηλαδή ένα σύνολο κανόνων (περιορισμών) που αφορούν τη σύνταξη μιας κατηγορίας XML τεκμηρίων[3]. Έτσι το DTD στο Παράδειγμα 4.1 επιβάλλει σε ένα στοιχείο **TAB** να αποτελείται από μηδέν η περισσότερα στοιχεία **φοιτητής** ενώ κάθε ένα από τα στοιχεία **φοιτητής** περιλαμβάνει υποχρεωτικά ένα στοιχείο **όνομα** ακολουθούμενο από ένα στοιχείο **επώνυμο**. Σε ένα DTD ορίζονται τα ονόματα των ετικετών των στοιχείων καθώς και τα ονόματα των γνωρισμάτων τα οποία επιτρέπεται να εμφανίζονται στα XML τεκμήρια τα οποία είναι έγκυρα ως προς το συγκεκριμένο DTD. Ορίζεται επίσης το είδος του περιεχομένου κάθε στοιχείου, τα στοιχεία που μπορούν να εμφανίζονται σαν υποστοιχεία ενός στοιχείου, η σειρά εμφάνισης τους καθώς και το επιτρεπόμενο πλήθος εμφανίσεων κάθε στοιχείου. Ουσιαστικά ένα DTD περιλαμβάνει τέσσαρις κατηγορίες δηλώσεων: τις *δηλώσεις τύπου στοιχείων* (element type declarations), τις *δηλώσεις λίστας γνωρισμάτων* (attribute list declarations), τις *δηλώσεις οντοτήτων* (entity declarations) και τις *δηλώσεις σημειογραφίας* (notation declarations).

4.4. Δηλώσεις Τύπου Στοιχείων

Η δήλωση ενός στοιχείου σε ένα DTD γίνεται δια μέσου μιας δήλωσης της μορφής:

<!ELEMENT *όνομα_στοιχείου τύπος_στοιχείου*>

όπου:

[3] Η κατηγορία των γραμματικών που μπορεί να περιγράψει κανείς με τη βοήθεια των DTD είναι γνωστή στους ασχολούμενους με τη θεωρητική πληροφορική και την υπολογιστική γλωσσολογία με τον όρο *γραμματικές χωρίς συμφραζόμενα* (context-free grammars).

- ***όνομα_στοιχείου***: είναι το όνομα του στοιχείου, δηλαδή η ετικέτα με την οποία εμφανίζεται το στοιχείο στο XML τεκμήριο.
- ***τύπος_στοιχείου***: περιγράφει το περιεχόμενο του στοιχείου, δηλαδή αν το περιεχόμενο του στοιχείου είναι απλό κείμενο ή περιλαμβάνει άλλα (υπο)στοιχεία, τα ονόματα των υποστοιχείων από τα οποία αποτελείται, τη σειρά εμφάνισης τους, το αν είναι υποχρεωτική ή όχι η εμφάνιση τους, και το πόσες φορές επιτρέπεται να εμφανίζονται αυτά. Για την περιγραφή του περιεχομένου ενός στοιχείου χρησιμοποιούνται οι παρενθέσεις (και) καθώς και τα σύμβολα **?**, *****, **+**, **|**, τα οποία συνοδεύουν ονόματα στοιχείων η ομάδες στοιχείων (κλεισμένες σε παρενθέσεις) και έχουν την ακόλουθη σημασία:
  - **?**: υποδηλώνει προαιρετική εμφάνιση.
  - *****: υποδηλώνει εμφάνιση μηδέν ή περισσότερες φορές.
  - **+**: υποδηλώνει εμφάνιση μία ή περισσότερες φορές.
  - **|**: υποδηλώνει διάζευξη (ένα από τα δύο).

Παράδειγμα 4.2. 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 οσοδήποτε μεγάλο πλήθος (μπορεί και μηδέν) στοιχείων με ετικέτα **c**.

❑

Παράδειγμα 4.3. Με την έκφραση:

```
<!ELEMENT e ((c? , d)* | (d , c)*)>
```

δηλώνεται ότι κάθε στοιχείο με ετικέτα **e** που εμφανίζεται σε ένα έγκυρο XML τεκμήριο, περιλαμβάνει είτε 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**. Έτσι, το XML τεκμήριο (όπου για ευκολία έχουμε θεωρήσει ότι τα στοιχεία **c** και **d** είναι κενά):

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

είναι έγκυρο, ενώ αντίθετα το τεκμήριο:

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

δεν είναι έγκυρο.

❑

Για να δηλώσουμε ότι το περιεχόμενο ενός στοιχείου είναι ακολουθία χαρακτήρων χρησιμοποιούμε δηλώσεις της μορφής:

<!ELEMENT *όνομα_στοιχείου* (#PCDATA)>

Εκτός από τις παραπάνω πιθανές τιμές της παράστασης *τύπος_στοιχείου* η παράσταση αυτή είναι δυνατό να πάρει επίσης 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.
Θα πρέπει στο σημείο αυτό να τονιστεί ότι ένα στοιχείο δεν επιτρέπεται να δηλώνεται περισσότερο από μια φορά σε ένα DTD.

4.5. Δηλώσεις Λίστας Γνωρισμάτων

Με τις δηλώσεις λίστας γνωρισμάτων μπορούμε να δηλώσουμε ποια στοιχεία έχουν γνωρίσματα, ποια είναι τα ονόματα των γνωρισμάτων αυτών, ποια από τα γνωρίσματα πρέπει να εμφανίζονται υποχρεωτικά, τι είδους τιμές παίρνουν τα γνωρίσματα, και ποιες είναι οι *προκαθορισμένες τιμές* (default values) (αν υπάρχουν) των γνωρισμάτων αυτών. Οι δηλώσεις λίστας γνωρισμάτων έχουν τη μορφή:

<!ATTLIST *όνομα_στοιχείου λίστα_δηλώσεων_γνωρισμάτων*>

Το *όνομα_στοιχείου* είναι το όνομα του στοιχείου το οποίο διαθέτει τα γνωρίσματα που δηλώνονται στη *λίστα_δηλώσεων_γνωρισμάτων*. Η *λίστα_δηλώσεων_γνωρισμάτων* περιλαμβάνει τριάδες της μορφής:

όνομα_γνωρίσματος τύπος_γνωρίσματος προκαθορισμός_τιμής

Παράδειγμα 4.4. Με τη δήλωση:

```
<!ATTLIST φοιτητής ΑΜ CDATA #REQUIRED
                   ΑΔΤ CDATA #IMPLIED >
```

δηλώνεται ότι το στοιχείο **φοιτητής** έχει δύο γνωρίσματα με ονόματα **ΑΜ** και **ΑΔΤ**. Και τα δύο γνωρίσματα είναι του τύπου **CDATA**. Η παρουσία του γνωρίσματος **ΑΜ** είναι υποχρεωτική σε κάθε εμφάνιση του στοιχείου **φοιτητής** (λόγω του **#REQUIRED**), ενώ αντίθετα, η παρουσία του γνωρίσματος **ΑΔΤ** δεν είναι υποχρεωτική (λόγω του **#IMPLIED**).

❑

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** αποτελούν ειδική περίπτωση των γνωρισμάτων του τύπου **CDATA** και μπορούν να πάρουν για τιμή μια απλή λέξη, ενώ τα γνωρίσματα του τύπου **NMTOKENS** μπορούν να πάρουν για τιμή πολλαπλές τιμές του τύπου **NMTOKEN** οι οποίες χωρίζονται μεταξύ τους με κενά. Τέλος 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 της παραμέτρου *τύπος_γνωρίσματος*, η παράμετρος αυτή είναι δυνατό να πάρει σαν τιμή τη λίστα των πιθανών τιμών του γνωρίσματος. Στην περίπτωση αυτή μιλάμε για *απαριθμήσιμους τύπους γνωρισμάτων* (enumerated attribute types).

Παράδειγμα 4.5. 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**, ενώ σε περίπτωση που απουσιάζει από ένα στοιχείο **book** θεωρείται ως εάν να είναι παρών και η τιμή του να είναι η **blue** (προκαθορισμένη τιμή).

❑

Σχετικά με την παράμετρο *προκαθορισμός_τιμής*, όπως είδαμε στα δύο τελευταία παραδείγματα, η παράμετρος αυτή μπορεί να πάρει σαν τιμή είτε μια από τις πιθανές τιμές του γνωρίσματος, με τη σημασία που αναφέραμε

προηγούμενα, είτε την τιμή #REQUIRED η οποία επιβάλει την υποχρεωτική εμφάνιση του γνωρίσματος στο αντίστοιχο στοιχείο. Μπορεί ακόμη να πάρει την τιμή #IMPLIED η οποία υποδηλώνει ότι δεν παρέχεται κάποια προκαθορισμένη τιμή (και δεν είναι υποχρεωτική η εμφάνιση του συγκεκριμένου γνωρίσματος), ή την τιμή #FIXED ακολουθούμενη από μια συγκεκριμένη τιμή. Στην τελευταία περίπτωση όλες οι εμφανίσεις του αντίστοιχου ορίσματος στο XML τεκμήριο θα πρέπει να έχουν για τιμή τη συγκεκριμένη τιμή που έχει δηλωθεί μετά από το #FIXED.

Παράδειγμα 4.6. Με τη δήλωση:

```
<!ATTLIST form method CDATA #FIXED "POST">
```

ορίζεται ότι το στοιχείο form διαθέτει το γνώρισμα method το οποίο έχει πάντα τη τιμή POST.

❑

Ιδιαίτερο ενδιαφέρον παρουσιάζουν οι τύποι γνωρίσματος ID, IDREF, και IDREFS. Ο τύπος ID δηλώνει ότι το συγκεκριμένο γνώρισμα παίζει ρόλο ταυτότητας για το αντικείμενο, παίρνει δηλαδή μοναδική τιμή που προσδιορίζει μονοσήμαντα το κάθε στοιχείο. Δεν επιτρέπεται επομένως 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. Αντίστοιχα, ο τύπος IDREF δηλώνει ότι τα γνωρίσματα αυτού του τύπου παίρνουν σαν τιμή τη τιμή του γνωρίσματος του τύπου ID κάποιου άλλου στοιχείου. Με άλλα λόγια ένα γνώρισμα του τύπου ID συμπεριφέρεται όπως ένα "*κλειδί*" σε μια σχεσιακή βάση δεδομένων, ενώ ένα γνώρισμα του τύπου IDREF συμπεριφέρεται όπως ένα "*ξένο κλειδί*" σε μια σχεσιακή βάση δεδομένων. Τέλος με τον τύπο IDREFS δηλώνονται γνωρίσματα τα οποία παίρνουν σαν τιμή μια λίστα από τιμές του τύπου IDREF οι οποίες χωρίζονται μεταξύ τους με κενά.

Πρέπει να σημειωθεί ότι τα γνωρίσματα του τύπου ID πρέπει να έχουν ως προκαθορισμό τιμής είτε #REQUIRED είτε #IMPLIED. Επίσης πρέπει να τονίσουμε ότι κάθε τιμή που παίρνει ένα γνώρισμα τύπου IDREF, ή IDREFS πρέπει να υπάρχει στο τεκμήριο ως τιμή ενός γνωρίσματος του τύπου ID.

Σαν παράδειγμα χρήσης των γνωρισμάτων τύπου ID, IDREF, και IDREFS θεωρείστε το DTD του Παραδείγματος 4.7. το οποίο δίνει τις προδιαγραφές (απλοποιημένων) XML τεκμηρίων τα οποία αναπαριστούν γενεαλογικά δέντρα:

Παράδειγμα 4.7. Θεωρείστε το ακόλουθο 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)>
]>
```

Ένα XML τεκμήριο το οποίο είναι έγκυρο ως προς το παραπάνω DTD είναι το ακόλουθο:

```
<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>
```

❑

Θα πρέπει τέλος να σημειώσουμε ότι είναι δυνατό να έχουμε περισσότερες από μια δηλώσεις ATTLIST οι οποίες αφορούν ένα συγκεκριμένο στοιχείο. Στην περίπτωση αυτή τα περιεχόμενα όλων αυτών των δηλώσεων

συγχωνεύονται. Αν οι δηλώσεις αυτές περιλαμβάνουν το ίδιο γνώρισμα περισσότερο από μια φορές τότε λαμβάνεται υπ' όψιν μόνο η πρώτη δήλωση του γνωρίσματος. Πάντως είναι καλό (για λόγους συμβατότητας) να δηλώνονται όλα τα γνωρίσματα ενός στοιχείου σε μια μόνο δήλωση λίστας γνωρισμάτων.

4.6. Δηλώσεις Οντοτήτων

Στο σημείο αυτό θα αναφερθούμε με συντομία στις δηλώσεις οντοτήτων στα DTD. Θα πρέπει να τονιστεί ότι η βασική ιδέα είναι ότι οι οντότητες αποτελούν συντομογραφίες σε ένα XML τεκμήριο και επομένως οι δηλώσεις των οντοτήτων στην πραγματικότητα αποτελούν ορισμό των συντομογραφιών αυτών. Γενικά διακρίνουμε τρία είδη οντοτήτων. Τις *εσωτερικές οντότητες* (internal entities), τις *εξωτερικές οντότητες* (external entities) και τις *οντότητες παραμέτρων* (parameter entities). Η δήλωση μιας εσωτερικής οντότητας γίνεται μέσω μιας έκφρασης της μορφής:

<!ENTITY *όνομα_οντότητας τιμή_οντότητας*>

Δηλαδή, το κείμενο με το οποίο αντικαθίσταται η εμφάνιση της οντότητας αποθηκεύεται μέσα στη δήλωση της οντότητας.

Παράδειγμα 4.8. Με τη δήλωση:

```
<!ENTITY TABENT "Τμήμα Αρχειονομίας και Βιβλιοθηκονομίας">
```

ορίζεται η οντότητα TABENT με τιμή τη συμβολοσειρά "Τμήμα Αρχειονομίας και Βιβλιοθηκονομίας". Έτσι κάθε φορά που εμφανίζεται το &TABENT; σε ένα XML τεκμήριο αυτό αντικαθίσταται (από το συντακτικό αναλυτή) με το "Τμήμα Αρχειονομίας και Βιβλιοθηκονομίας".

❑

Σχετικά με τις εξωτερικές οντότητες, η βασική ιδέα είναι ότι 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 4.9. Με τη δήλωση:

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

ορίζεται η οντότητα TABinfo. Έτσι, κάθε εμφάνιση του &TABinfo; στο τεκμήριο θα αντικαθίσταται κατά την επεξεργασία του τεκμηρίου με το περιεχόμενο του αρχείου "/ionio/information/info.xml".

❑

Τέλος, 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 στις αναφορές σε οντότητες παραμέτρου αντί για το σύμβολο **&**. Οι αναφορές σε οντότητες παραμέτρου αντικαθίστανται άμεσα με το κείμενο που αποτελεί την τιμή της οντότητας το οποίο αποτελεί πλέον κομμάτι του DTD.

4.7. Σύνδεση XML με DTD

Είναι φανερό ότι, προκειμένου να εξεταστεί η εγκυρότητα ενός XML τεκμηρίου ως προς ένα DTD, θα πρέπει τα δύο αυτά να συσχετιστούν με κάποιο τρόπο. Γενικά έχουμε δύο επιλογές. Η πρώτη επιλογή είναι να συμπεριλάβουμε το DTD στο ίδιο αρχείο με αυτό που βρίσκεται το XML τεκμήριο, ενώ η δεύτερη επιλογή είναι να τοποθετήσουμε τις δηλώσεις του DTD σε ξεχωριστό αρχείο και στη συνέχεια να συσχετίσουμε κατάλληλα τα δύο αρχεία. Στο απλό παράδειγμα που ακολουθεί το DTD έχει ενσωματωθεί στο XML τεκμήριο:

Παράδειγμα 4.10. Στο τεκμήριο:

```
<?xml version="1.0"?>
<!DOCTYPE greeting [
<!ELEMENT greeting (#PCDATA)>
]>
<greeting>Hello, world!</greeting>
```

το DTD ξεκινά με τη δήλωση `<!DOCTYPE` και τελειώνει με το `]>`. Είναι φανερό ότι το XML τεκμήριο το οποίο περιλαμβάνει σαν μοναδικό στοιχείο το στοιχείο `greeting` είναι έγκυρο με βάση το DTD που το συνοδεύει.

❑

Ο δεύτερος τρόπος συσχέτισης ενός XML τεκμηρίου με το αντίστοιχο DTD είναι να αποθηκεύσουμε το DTD σε ένα ξεχωριστό αρχείο και να τοποθετήσουμε μια αναφορά στο αρχείο αυτό στο XML τεκμήριο όπως στο παράδειγμα που ακολουθεί.

Παράδειγμα 4.11. Παράδειγμα σύνδεσης του XML τεκμηρίου με το αντίστοιχο DTD μέσω αναφοράς στο αρχείο που είναι αποθηκευμένο το DTD:

```
<?xml version="1.0"?>
<!DOCTYPE greeting SYSTEM "hello.dtd">
<greeting>Hello, world!</greeting>
```

❑

Αξίζει τέλος να σημειωθεί ότι μπορούμε να αναμίξουμε τους δύο αυτούς τρόπους συσχέτισης XML τεκμηρίου με το αντίστοιχο DTD. Θα μπορούσαμε για παράδειγμα να συμπεριλάβουμε μέρος των δηλώσεων στοιχείων στο αρχείο του XML τεκμηρίου, ενώ ένα άλλο μέρος των δηλώσεων να τοποθετηθεί σε ένα εξωτερικό αρχείο.

4.8. 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. Έτσι, αν η τιμή της λέξης κλειδί είναι `INCLUDE`, τότε το περιεχόμενο του υπό συνθήκη τμήματος συμπεριλαμβάνεται στο DTD. Αντίθετα, 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, όπως φαίνεται και στο Παράδειγμα 4.12. Στην περίπτωση αυτή η οντότητα παραμέτρου πρέπει να αντικατασταθεί με την τιμή της προκειμένου να μπορέσει να αποφασίσει ο επεξεργαστής αν θα συμπεριλάβει το υπό συνθήκη τμήμα ή όχι.

Παράδειγμα 4.12. Στο ακόλουθο τμήμα DTD:

```
<!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.

❑

4.9. Η φυσική δομή ενός XML τεκμηρίου

Ένα XML τεκμήριο μπορεί να χωρίζεται σε μια ή περισσότερες φυσικές μονάδες (διαφορετικά αρχεία). Οι μονάδες αυτές λέγονται *οντότητες* (entities).

Κάθε XML τεκμήριο περιλαμβάνει μια κεντρική οντότητα που λέγεται *οντότητα τεκμηρίου* (document entity). Η οντότητα τεκμηρίου είναι το κεντρικό αρχείο και θεωρείται το σημείο από το οποίο ξεκινά την εργασία του ο *επεξεργαστής της XML* (XML processor). Οι οντότητες έχουν *περιεχόμενο* (content) και κάθε μια προσδιορίζεται από ένα όνομα οντότητας, εκτός από την οντότητα τεκμηρίου και το εξωτερικό DTD (αν υπάρχει).
Οι οντότητες μπορεί να είναι *συντακτικά αναλύσιμες* (parsed entities) ή *μη αναλύσιμες συντακτικά* (unparsed entities). Οι οντότητες της πρώτης κατηγορίας περιλαμβάνουν κείμενο το οποίο αντικαθιστά την αναφορά της οντότητας στο XML τεκμήριο και θεωρείται συστατικό κομμάτι του XML τεκμηρίου. Το περιεχόμενο των οντοτήτων της δεύτερης κατηγορίας ανεξάρτητα είναι ή όχι κείμενο (XML ή οτιδήποτε άλλο), δεν υπόκειται σε κάποιους περιορισμούς εκ μέρους τις XML αλλά απλά καθίσταται διαθέσιμο στον επεξεργαστή XML και τις υπόλοιπες εφαρμογές που θα χειριστούν το XML τεκμήριο.
Η επίκληση των συντακτικά αναλύσιμων οντοτήτων γίνεται μέσω των αναφορών οντοτήτων. Η επίκληση των συντακτικά μη αναλύσιμων οντοτήτων γίνεται μέσα από γνωρίσματα του τύπου ENTITY ή ENTITIES.
Περισσότερες τεχνικές λεπτομέρειες σχετικά με τον τρόπο ορισμού και χρήσης των οντοτήτων μπορεί να βρει ο ενδιαφερόμενος αναγνώστης στο τεχνικό εγχειρίδιο ορισμού της XML [2].

ΚΕΦΑΛΑΙΟ 5: XML Namespaces

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 να έχουν δύο συνθετικά, το *πρόθεμα* (χώρου ονομάτων) (prefix) και το *τοπικό τμήμα* (local part), τα οποία χωρίζονται μεταξύ τους με :. Ο συνδυασμός των δύο αυτών παράγει ονόματα τα οποία είναι παγκοσμίως μοναδικά.

5.1. Η ανάγκη για τη χρήση χώρων ονομάτων

Οι σχεδιαστές της XML προσβλέπουν σε εφαρμογές στις οποίες ένα XML τεκμήριο είναι δυνατό να περιλαμβάνει στοιχεία και γνωρίσματα τα οποία έχουν οριστεί και απευθύνονται σε διαφορετικές εφαρμογές λογισμικού. Στόχο της ύπαρξης ενός τέτοιου καλά τεκμηριωμένου και κατανοητού "λεξιλογίου ετικετών" για το οποίο υπάρχει διαθέσιμο χρήσιμο λογισμικό, αποτελεί η επιθυμία για επαναχρησιμοποίηση των τεκμηρίων από πολλές διαφορετικές εφαρμογές, παρά η επανασύνταξη τους για κάθε εφαρμογή χωριστά. Τέτοια τεκμήρια τα οποία 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 είναι σε θέση 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 δομές σύνταξης τεκμηρίων οι οποίες θα επιτρέπουν ονόματα διαμορφωμένα κατά τρόπο ώστε θα αποφεύγεται η σύγκρουση μεταξύ ονομάτων από διαφορετικά "λεξιλόγια ετικετών". Ο τρόπος με τον οποίο προτείνεται να ξεπεραστούν τα προβλήματα αυτά είναι η χρησιμοποίηση του μηχανισμού που ονομάζεται *χώρος ονομάτων XML*, ο οποίος πετυχαίνει τον στόχο αυτόν επιτρέποντας επεκταμένα ονόματα στοιχείων και γνωρισμάτων.

5.2. Δηλώσεις και εμβέλεια χώρων ονομάτων

Ένας χώρος ονομάτων δηλώνεται χρησιμοποιώντας μια οικογένεια από προκαθορισμένα γνωρίσματα. Ένα τέτοιο όνομα γνωρίσματος είναι το `xmlns`. Μπορεί επίσης να είναι 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 5.1. Στο τμήμα XML τεκμηρίου που ακολουθεί 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>
```

❑

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).
Είναι δυνατό να δηλωθούν πολλοί χώροι ονομάτων στο ίδιο στοιχείο όπως στο παράδειγμα που ακολουθεί:

Παράδειγμα 5.2. Στο τμήμα XML τεκμηρίου που ακολουθεί:

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

όπου δηλώνονται δύο χώροι ονομάτων. Ο πρώτος συνδέει το πρόθεμα bk με το χώρο ονομάτων με όνομα "www.books.org/book", ενώ ο δεύτερος συνδέει το πρόθεμα isbn με το όνομα χώρου ονομάτων "www.isbn.org/def".

❑

Οι αναφορές URI οι οποίες προσδίδουν ταυτότητα σε χώρους ονομάτων θεωρούνται *ταυτόσημοι* όταν είναι ακριβώς οι ίδιοι χαρακτήρα-προς-χαρακτήρα. Το URI που αποτελούν ονόματα χώρων ονομάτων δεν υποχρεούνται να αντιστοιχούν κατ' ανάγκη σε μια διεύθυνση από την οποία μπορεί να ανακτήσει κανείς πληροφορίες για τον χώρο ονομάτων. Αντίθετα, ο ρόλος του ονόματος του χώρου ονομάτων είναι η εξασφάλιση της μοναδικότητας των στοιχείων και γνωρισμάτων που χρησιμοποιούν το αντίστοιχο πρόθεμα.
Θα πρέπει να σημειώσουμε επίσης ότι η ακολουθία των τριών χαρακτήρων 'x', 'm', 'l' με οποιονδήποτε συνδυασμό κεφαλαίων/μικρών, όπως για παράδειγμα οι XML, Xml, xml, κ.λ.π., αποτελούν δεσμευμένες λέξεις για προθέματα που χρησιμοποιούνται από την XML και τις εφαρμογές που σχετίζονται με την XML.

5.3. Πως χρησιμοποιούνται οι χώροι 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 από αυτά 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). Ο συνδυασμός των δύο συνθετικών παράγει ονόματα τα οποία είναι μοναδικά. Με βάση τα παραπάνω, η σύνταξη των ονομάτων στοιχείων και γνωρισμάτων τα οποία χρησιμοποιούν χώρους ονομάτων έχουν την μορφή:

prefix:localpart

Παράδειγμα 5.3. Θεωρείστε το παρακάτω τμήμα XML τεκμηρίου:

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

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

❑

Όπως φαίνεται από το Παράδειγμα 5.3., 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.

Παράδειγμα 5.4. Στο παράδειγμα αυτό, το οποίο αποτελεί μια παραλλαγή του Παραδείγματος 5.3, έχουμε δήλωση 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>
```

Τα ονόματα δύο εκ των στοιχείων που αποτελούν το περιεχόμενο του στοιχείου **book**, συγκεκριμένα των στοιχείων **bk:title**, και **bk:authors**, προκύπτουν με τη χρήση του ενός χώρου ονομάτων (αυτού που αναφέρεται με το πρόθεμα **bk**) ενώ το όνομα του στοιχείου **isbn:number** προκύπτει με τη χρήση του άλλου χώρου ονομάτων (πρόθεμα **isbn**).

❑

Όπως έχει ήδη αναφερθεί, οι χώροι ονομάτων μπορούν να χρησιμοποιηθούν και για τον σχηματισμών ονομάτων γνωρισμάτων όπως φαίνεται στο παράδειγμα που ακολουθεί:

Παράδειγμα 5.5. Στο τμήμα του XML τεκμηρίου που ακολουθεί:

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

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

❑

Αξίζει επίσης να σημειωθεί ότι, μια αναφορά URI μπορεί να περιέχει χαρακτήρες οι οποίοι δεν μπορούν να χρησιμοποιηθούν απευθείας σαν προθέματα χώρων ονομάτων. Επομένως, το πρόθεμα στους χώρους ονομάτων παίζει το ρόλο αντιπροσώπου μιας αναφοράς URI.
Σημειώστε ακόμη ότι κάθε πρόθεμα που χρησιμοποιείται σε ένα όνομα στοιχείου ή γνωρίσματος, εκτός από τα προθέματα **xmlns** και **xml**, 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.
Τέλος, σημειώστε ότι τα ονόματα στοιχείων και γνωρισμάτων δίνονται συνοδευόμενα με το πρόθεμα τους όταν αυτά δηλώνονται σε ένα DTD.

5.4. **Χώροι ονομάτων χωρίς προθέματα**

Πολλές φορές 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 ενός χώρου ονομάτων χωρίς πρόθεμα έγκειται στο γεγονός ότι δεν χρειάζεται να συνοδεύονται τα ονόματα στοιχείων που ανήκουν στο χώρο αυτό από συγκεκριμένο πρόθεμα.

Όπως και στους χώρους ονομάτων με πρόθεμα, έτσι και ο χώρος ονομάτων χωρίς πρόθεμα θεωρείται ότι εφαρμόζεται στο στοιχείο στο οποίο δηλώνεται (αν το στοιχείο 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 είναι δυνατό να είναι κενή. Στην περίπτωση αυτή, τα στοιχεία που δεν έχουν πρόθεμα και βρίσκονται μέσα στην εμβέλεια της συγκεκριμένης δήλωσης θεωρείται ότι δεν ανήκουν σε κανένα χώρο ονομάτων. Σημειώστε τέλος ότι ο χώρος ονομάτων χωρίς πρόθεμα δεν εφαρμόζεται απευθείας σε γνωρίσματα.

Παράδειγμα 5.6. Στο XML τεκμήριο που ακολουθεί φαίνεται δήλωση και χρήση του χώρου ονομάτων χωρίς πρόθεμα (όλα τα στοιχεία του τεκμηρίου αυτού ανήκουν στο χώρο ονομάτων που δηλώνεται στο παράδειγμα αυτό):

```
<?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>
```

❑

Παράδειγμα 5.7. Στο ακόλουθο XML τεκμήριο έχουμε τη δήλωση και χρήση ενός 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>
```

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

❑

Παράδειγμα 5.8. 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>
```

❑

5.5. Μοναδικότητα γνωρισμάτων

Στα XML τεκμήρια τα οποία υπακούουν στις προδιαγραφές που περιγράψαμε, καμιά ετικέτα στοιχείου δεν μπορεί να περιέχει δύο γνωρίσματα τα οποία:

1. έχουν ταυτόσημα ονόματα, ή
2. έχουν ονόματα με το ίδιο τοπικό τμήμα και τα προθέματα των οποίων έχουν συνδεθεί με ταυτόσημα ονόματα χώρων ονομάτων.

Παράδειγμα 5.9. Καμιά από τις ετικέτες αρχής bad στο παρακάτω δεν είναι έγκυρη λόγω της ταυτότητας των ονομάτων των γνωρισμάτων που διαθέτουν:

```
<!-- 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>
```

❑

Παράδειγμα 5.9. Η σύνταξη καθενός από τα στοιχεία 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>
```

❑

ΚΕΦΑΛΑΙΟ 6: XML Schema

6.1. Εισαγωγή

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

6.2. Η γλώσσα XML Schema με ένα παράδειγμα

Στόχος ενός τεκμηρίου κωδικοποιημένου σε XML Schema, όπως και ενός DTD, είναι να περιγράψει μια κατηγορία XML τεκμηρίων, να λειτουργήσει δηλαδή σαν μια γραμματική που παρέχει τους κανόνες σύνταξης που διέπουν τη συγκεκριμένη κατηγορία XML τεκμηρίων. Για να πάρουμε μια πρώτη γεύση του τρόπου με τον οποίο περιγράφεται η δομή μιας κατηγορίας XML τεκμηρίων με την XML Schema ας δούμε ένα απλό παράδειγμα:

Παράδειγμα 6.1. Ας υποθέσουμε ότι έχουμε XML τεκμήρια όπως το παρακάτω[4]:

```
<TAB>
        <φοιτητής>
                <όνομα> Νίκος </όνομα>
                <επώνυμο> Νικολάου </επώνυμο>
        </φοιτητής>
        <φοιτητής> ... </φοιτητής>
                ...
</TAB>
```

και θέλουμε να αναπτύξουμε ένα XML Schema που περιγράφει τη δομή τεκμηρίων αυτής της μορφής. Ένα τέτοιο XML Schema είναι το ακόλουθο[5]:

---

[4] Το τεκμήριο αυτό το συναντήσαμε ήδη στο Παράδειγμα 4.1 του Κεφαλαίου 4. Εκεί είδαμε πως μπορούμε να περιγράψουμε την κατηγορία αυτήν των XML τεκμηρίων με τη χρήση των DTD.

[5] Για λόγους σύγκρισης υπενθυμίζουμε ότι στο Παράδειγμα 4.1 παρουσιάσαμε το ακόλουθο DTD που περιγράφει τη δομή των τεκμηρίων αυτής της κατηγορίας.

```
<!DOCTYPE TAB [
  <!ELEMENT TAB (φοιτητής*)>
  <!ELEMENT φοιτητής (όνομα, επώνυμο)>
  <!ELEMENT όνομα (#PCDATA)>
```

```
<?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>
```

❑

Όπως φαίνεται στο Παράδειγμα 6.1.:

- Ένα τεκμήριο σε XML Schema είναι ένα XML τεκμήριο.
- Κάθε στοιχείο στο XML Schema του παραδείγματος έχει το πρόθεμα **xs** το οποίο συνδέεται με το χώρο ονομάτων της XML Schema μέσω της δήλωσης:

  xmlns:xs="http://www.w3.org/2001/XMLSchema".

  Το **xs** θα χρησιμοποιηθεί στη συνέχεια σαν πρόθεμα σε κάθε στοιχείο στο σχήμα (σε όλα τα παραδείγματα) για να δηλώσει το χώρο ονομάτων της XML Schema. Το ίδιο πρόθεμα, όπως θα δούμε στη συνέχεια, εμφανίζεται και πριν από τα ονόματα των ενσωματωμένων απλών τύπων της XML Schema.
- Η XML Schema παρέχει ένα σύνολο από στοιχεία (όπως τα στοιχεία **element**, **sequence**, **complexType**) με συγκεκριμένη σημασία, τα οποία μαζί με τα γνωρίσματα τα οποία μπορεί να τα συνοδεύουν (όπως τα **name**, **type**, **minOccurs**, **maxOccurs**) μας επιτρέπουν να περιγράψουμε τη δομή XML τεκμηρίων.
- Μια περιγραφή σχήματος με την XML Schema έχει σαν ετικέτα αρχής του πιο εξωτερικού στοιχείου (της ρίζας) την ετικέτα **xs:schema**.

Η XML Schema παρέχει αξιοσημείωτη εκφραστικότητα η οποία επιτρέπει να περιγράψουμε τη δομή μιας κατηγορίας XML τεκμηρίων με πολλούς διαφορετικούς τρόπους.

```
   <!ELEMENT επώνυμο (#PCDATA)>
]>
```

Παράδειγμα 6.2. Παρατηρήστε το ακόλουθο τεκμήριο (σε XML Schema), το οποίο περιγράφει ακριβώς την ίδια κατηγορία XML τεκμηρίων με το σχήμα που παρουσιάσαμε στο Παράδειγμα 6.1.:

```
<?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>
```

❑

6.3. Δηλώσεις στοιχείων και γνωρισμάτων στην XML Schema

6.3.1. Δηλώσεις στοιχείων

Σε κάθε XML τεκμήριο συναντάμε δύο κατηγορίες στοιχείων:

1. Στοιχεία που περιέχουν υποστοιχεία ή διαθέτουν γνωρίσματα τα οποία θα λέμε ότι έχουν *σύνθετο τύπο* (complex type).
2. Στοιχεία που δεν έχουν υποστοιχεία, ούτε διαθέτουν γνωρίσματα. Τα στοιχεία αυτά θα λέμε ότι έχουν *απλό τύπο* (simple type).

Τα γνωρίσματα θα θεωρούμε πάντα ότι είναι απλού τύπου.
Τα στοιχεία TAB και φοιτητής που συναντήσαμε στα Παραδείγματα 6.1 και 6.2 είναι σύνθετου τύπου, ενώ τα στοιχεία όνομα και επώνυμο είναι απλού τύπου. Για τους απλούς τύπους θα μιλήσουμε σε επόμενη ενότητα.
Για τη δημιουργία νέων σύνθετων τύπων από τον χρήστη η XML Schema παρέχει το στοιχείο complexType. Οι σύνθετοι τύποι που δημιουργούνται

μέσω του στοιχείου complexType μπορούν, εφόσον το επιθυμούμε, να αποκτούν όνομα μέσω της τιμής του γνωρίσματος name. Στο Παράδειγμα 6.2 έχουμε ορίσει δύο σύνθετους τύπους, τους τύπους TABtype και studentType. Έτσι, με την έκφραση:

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

ορίζουμε το σύνθετο τύπο με όνομα TABtype. Το όνομα αυτό χρησιμοποιείται σε άλλο σημείο της περιγραφής σχήματος για να δηλώσουμε ότι ένα στοιχείο του τεκμηρίου έχει τη δομή που περιγράφεται από το συγκεκριμένο σύνθετο τύπο. Στο Παράδειγμα 6.2 αυτό γίνεται με τη δήλωση:

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

στην οποία δηλώνεται ότι στα XML τεκμήρια που διέπονται από το συγκεκριμένο XML Schema περιλαμβάνεται το στοιχείο με όνομα TAB η δομή του οποίου είναι αυτή που περιγράφεται στο σύνθετο τύπο με όνομα TABtype.

Αξίζει να παρατηρήσουμε ότι στο Παράδείγμα 6.1 δεν είχαμε δώσει όνομα στους σύνθετους τύπους που χρησιμοποιήσαμε. Αντίθετα οι σύνθετοι τύποι τοποθετήθηκαν μέσα στον ορισμό του στοιχείου το οποίο αφορούν, με αποτέλεσμα να μην χρειάζεται η επίκληση τους από άλλο σημείο, της περιγραφής σε XML Schema, και κατά συνέπεια να μην είναι απαραίτητο να διαθέτουν όνομα. Η επισύναψη ονόματος σε ένα σύνθετο τύπο διευκολύνει ιδιαίτερα όταν περισσότερα του ενός στοιχεία του τεκμηρίου έχουν τον ίδιο σύνθετο τύπο. Στις περιπτώσεις αυτές με τον ορισμό του τύπου μια μόνο φορά και την επίκληση στη συνέχεια του ονόματος του πετυχαίνουμε σημαντική οικονομία.

6.3.2. Δηλώσεις γνωρισμάτων

Για να δηλώσουμε τα γνωρίσματα ενός στοιχείου χρησιμοποιούμε το στοιχείο attribute που μας παρέχει η XML Schema. Έτσι ένα γνώρισμα δηλώνεται με μια έκφραση της μορφής:

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

Παράδειγμα 6.3. Με την έκφραση:

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

δηλώνεται το γνώρισμα με όνομα **ηλικία** το οποίο παίρνει τιμές του τύπου **positiveInteger**.

❑

Σημειώνεται ότι ο τύπος **positiveInteger**, είναι ένας απλός τύπος της XML Schema και αντιπροσωπεύει τους θετικούς ακέραιους αριθμούς.
Επίσης, μέσω της παράστασης **use="required"**, δηλώνεται ότι η εμφάνιση του συγκεκριμένου γνωρίσματος είναι υποχρεωτική.
Τίθεται τώρα το ερώτημα: Σε ποιο σημείο της περιγραφής τοποθετείται μια δήλωση γνωρίσματος; Θα πρέπει κατ' αρχήν να υπενθυμίσουμε ότι ένα στοιχείο το οποίο διαθέτει γνωρίσματα είναι πάντα σύνθετου τύπου. Απαιτείται επομένως η χρήση της δήλωσης **complexType** προκειμένου να περιγραφεί το περιεχόμενο του στοιχείου αυτού. Οι δηλώσεις των γνωρισμάτων ενός στοιχείου τοποθετούνται στο τέλος της δήλωσης του σύνθετου τύπου που αντιστοιχεί στο στοιχείο, ακριβώς πριν από τη δήλωση τέλους του σύνθετου τύπου (πριν από το **</xs:complexType>**).

Παράδειγμα 6.4. Στο τμήμα του παραδείγματος που ακολουθεί, παρουσιάζεται μια παραλλαγή του ορισμού του στοιχείου **φοιτητής** όπως εμφανίζεται στο Παράδειγμα 6.1. έτσι ώστε να διαθέτει και 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>
```

Με βάση την παραπάνω δήλωση το στοιχείο **φοιτητής** περιλαμβάνει τα υποστοιχεία **όνομα** και **επώνυμο**, ενώ έχει και ένα γνώρισμα το οποίο έχει το όνομα **ηλικία**, εμφανίζεται υποχρεωτικά σε κάθε στοιχείο **φοιτητής**, και παίρνει σαν τιμή έναν θετικό ακέραιο.

❑

6.3.3. Περιορισμοί συχνότητας εμφάνισης στοιχείων και γνωρισμάτων

Σε αντίθεση με άλλες γλώσσες ορισμού σχήματος (όπως για παράδειγμα η DTD), η XML Schema μας επιτρέπει να δηλώσουμε το πλήθος των

εμφανίσεων ενός στοιχείου με εξαιρετική ακρίβεια. Αυτό γίνεται δίνοντας τιμές σε δύο γνωρίσματα του στοιχείου **element**, τα οποία είναι το **minOccurs** (μέσω του οποίου δηλώνουμε τον ελάχιστο αριθμό εμφανίσεων του στοιχείου), και το **maxOccurs** (μέσω του οποίου δηλώνουμε τον μέγιστο αριθμό εμφανίσεων του στοιχείου). Οι τιμές που μπορούν να πάρουν τα γνωρίσματα αυτά είναι μη αρνητικοί ακέραιοι αριθμοί. Όταν κάποιο από τα γνωρίσματα **minOccurs** ή **maxOccurs** παραλείπεται (και τα δύο ή ένα από αυτά) τότε θεωρείται ότι έχει σαν τιμή το **1**. Εννοείται ότι, όταν εμφανίζονται και τα δύο αυτά γνωρίσματα στη δήλωσε ενός στοιχείου, η τιμή του **maxOccurs** πρέπει να είναι μεγαλύτερη ή ίση από την τιμή του **minOccurs**.

Παράδειγμα 6.5. Στη δήλωση:

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

η τιμή **unbounded** για το γνώρισμα **maxOccurs** καθορίζει ότι μπορούμε να έχουμε απεριόριστο αριθμό εμφανίσεων του στοιχείου **φοιτητής**, ενώ είναι αποδεκτό και να μην έχουμε καμιά εμφάνιση του στοιχείου αυτού (λόγω της τιμής **0** του γνωρίσματος **minOccurs**).

❑

Αναφορικά με την εμφάνιση των γνωρισμάτων θα πρέπει να σημειώσουμε τα ακόλουθα. Είναι γνωστό ότι ένα γνώρισμα είναι δυνατό να εμφανίζεται σε ένα στιγμιότυπο στοιχείου μία φορά ή να μην εμφανίζεται καθόλου. Δεν μπορεί όμως το ίδιο γνώρισμα να εμφανίζεται σε ένα στοιχείο περισσότερες από μια φορές (με τη ίδια ή διαφορετική τιμή). Η XML Schema μας παρέχει τη δυνατότητα να προσδιορίσουμε την υποχρεωτικότητα ή μη της εμφάνισης ενός γνωρίσματος σε ένα στοιχείο μέσα από την τιμή του γνωρίσματος **use** το οποίο μπορούμε (όπως είδαμε στο Παράδειγμα 6.3) να συμπεριλάβουμε στη δήλωση του γνωρίσματος. Έτσι, δίνοντας την τιμή **required** στο γνώρισμα **use** δηλώνουμε ότι το γνώρισμα στο οποίο αναφέρεται η δήλωση οφείλει να εμφανίζεται υποχρεωτικά στο στοιχείο στο οποίο αναφέρεται, ενώ δίνοντας την τιμή **optional** υποδηλώνουμε προαιρετική εμφάνιση του συγκεκριμένου γνωρίσματος.

6.4. Απλοί τύποι

Η XML Schema διαθέτη μια πλούσια συλλογή από ενσωματωμένους απλούς τύπους. Ανάμεσα σ' αυτούς περιλαμβάνονται οι τύποι **string**, **byte**, **integer**, **positiveInteger**, **negativeInteger**, **decimal**, **int**, **long**, **float**, **double**, **boolean**, **date**, **dateTime**, **ID**, **IDREF**, **IDREFS**, **ENTITY**, **ENTITIES**, **NMTOKEN**, **NMTOKENS**.

Εκτός από τους ενσωματωμένους απλούς τύπους, η XML Schema δίνει τη δυνατότητα στο χρήστη να ορίσει νέους απλούς τύπους μέσω δηλώσεων που περιγράφουν τον τρόπο που μπορούν οι τύποι αυτοί να παραχθούν από απλούς τύπους (ενσωματωμένους ή παραγόμενους) που υπάρχουν ήδη. Για τον ορισμό και την ονομασία νέων απλών τύπων χρησιμοποιείται το στοιχείο **simpleType** με τη βοήθεια του οποίου μπορούμε να παράγουμε ένα νέο απλό τύπο επιβάλλοντας επιπλέον περιορισμούς σε ένα τύπο που ήδη υπάρχει.

Παράδειγμα 6.6. Με την παρακάτω 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**, ο οποίος προκύπτει από τον απλό τύπο integer της XML Schema, επιβάλλοντας επιπλέον περιορισμούς. Ο τύπος πάνω στον οποίο βασιζόμαστε για να ορίσουμε το νέο τύπο, δίνεται σαν τιμή στο γνώρισμα **base** του στοιχείου **restriction**. Οι επιπλέον περιορισμοί στο παράδειγμα αυτό ορίζουν σαν ελάχιστη επιτρεπτή τιμή το **0**, και σαν μέγιστη επιτρεπτή τιμή το **100**. Οι περιορισμοί αυτοί ορίζονται μέσω των στοιχείων **minInclusive** και **maxInclusive** αντίστοιχα. Ο νέος τύπος επομένως που δημιουργείται με την παραπάνω δήλωση, φέρει το όνομα **myInteger**, και περιλαμβάνει όλους τους ακέραιους αριθμούς από το **0** έως το **100**.

❑

Η XML Schema παρέχει μια πλούσια ποικιλία «*όψεων*» (facets) που μπορεί κανείς να χρησιμοποιήσει για να επιβάλει περιορισμούς ώστε να ορίσει νέους τύπους. Δύο από αυτές τις όψεις είναι τα στοιχεία **minInclusive** και **maxInclusive** που συναντήσαμε στο προηγούμενο παράδειγμα. Μια ακόμη χρήσιμη όψη είναι το στοιχείο **enumeration** το οποίο περιορίζει έναν απλό τύπο σε ένα σύνολο διακριτών τιμών. Στο παράδειγμα που ακολουθεί μπορούμε να δούμε τη χρήση της όψης **enumeration** για τον ορισμό νέων τύπων:

Παράδειγμα 6.7. Στην περιγραφή σε XML Schema που ακολουθεί:

```
<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 ένας νέος τύπος που φέρει το όνομα **νόμισμα** ενώ οι τιμές τις οποίες περιλαμβάνει είναι τα (διεθνή) σύμβολα των νομισμάτων των διαφόρων χωρών (GRD, EURO, USD, κ.λ.π.).

❑

6.5. Ορισμός γνωρισμάτων σε στοιχεία απλού τύπου

Με βάση όσα έχουμε πει μέχρι τώρα για να δηλώσουμε ένα στοιχείο με όνομα **ονοματεπώνυμο**, ο τύπος του οποίου είναι ο απλός τύπος **string**, αρκεί μια δήλωση της μορφής:

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

Τι γίνεται όμως όταν θέλουμε αυτό το στοιχείο να διαθέτει και ένα γνώρισμα, για παράδειγμα το γνώρισμα **ΑΔΤ** (αριθμός δελτίου ταυτότητας); Όπως έχουμε ήδη αναφέρει, οι απλοί τύποι (στοιχεία απλού τύπου) δεν επιτρέπεται να έχουν γνωρίσματα. Επομένως, προκειμένου να δηλώσουμε ότι ένα στοιχείο απλού τύπου περιλαμβάνει ένα γνώρισμα, θα πρέπει να ορίσουμε ένα σύνθετο τύπο. Το περιεχόμενο του στοιχείου αυτού εξακολουθούμε να επιμένουμε όμως να είναι απλού τύπου (του τύπου **string** για την περίπτωση μας). Αυτό μπορεί να γίνει όπως φαίνεται στο Παράδειγμα:

Παράδειγμα 6.8. Δήλωση στοιχείου «απλού τύπου» το οποίο όμως έχει γνώρισμα:

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

❑

Στο Παράδειγμα 6.8 χρησιμοποιούμε το στοιχείο **complexType** για να ξεκινήσουμε τον ορισμό του νέου (ανώνυμου) τύπου. Για να υποδηλώσουμε ότι το περιεχόμενο του νέου τύπου περιλαμβάνει μόνο ακολουθία απλών

χαρακτήρων και όχι (υπό)στοιχεία χρησιμοποιούμε το στοιχείο **simpleContent**. Τέλος, για τον ορισμό του νέου τύπου, επεκτείνουμε τον απλό τύπο **string**. Η επέκταση συνίσταται στην προσθήκη του γνωρίσματος **ΑΔΤ** μέσω μιας συνήθους δήλωσης γνωρίσματος. Το στοιχείο **ονοματεπώνυμο** θα διαθέτει τώρα και το γνώρισμα **ΑΔΤ**.

6.6. Ομαδοποίηση στοιχείων

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

6.6.1. Ομαδοποίηση στοιχείων με το "sequence"

Το στοιχείο **sequence** της XML Schema χρησιμοποιείται για να δηλώσει μια (διατεταγμένη) ακολουθία (υπο)στοιχείων.

Παράδειγμα 6.9. Στο παρακάτω τμήμα περιγραφής σε XML Schema:

```
<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 του τύπου **studentType** πρέπει να περιλαμβάνει ένα υποστοιχείο με όνομα **όνομα** ακολουθούμενο από ένα υποστοιχείο με όνομα **επώνυμο** και στη συνέχεια από ένα υποστοιχείο με όνομα **πατρώνυμο**. Ο τύπος και των τριών αυτών (υπό)στοιχείων είναι ο απλός τύπος **string**.

❑

6.6.2. Ομαδοποίηση στοιχείων με το στοιχείο "choice"

Το στοιχείο ομαδοποίησης **choice** επιτρέπει σε ένα μόνο από τα (υπο)στοιχεία που περιλαμβάνει να εμφανίζεται στο XML τεκμήριο σε κάθε στιγμιότυπο του στοιχείου το οποίο περιγράφει.

Παράδειγμα 6.10. Στο παρακάτω τμήμα περιγραφής σε XML Schema:

```
<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>
```

στο οποίο δηλώνεται ότι κάθε στοιχείο του τύπου **studentType** περιλαμβάνει είτε το στοιχείο **ονοματεπώνυμο** σαν μοναδικό (υπο)στοιχείο, είτε ένα στοιχείο **όνομα** ακολουθούμενο από ένα στοιχείο **επώνυμο**.

❑

6.6.3. Ομαδοποίηση στοιχείων με το "all"

Σε αντίθεση με το στοιχείο **sequence** το οποίο επιβάλει συγκεκριμένη σειρά με την οποία πρέπει να εμφανίζονται τα στοιχεία τα οποία περικλείει, το στοιχείο **all** χρησιμοποιείται για να δηλώσει ότι θα πρέπει να εμφανίζεται υποχρεωτικά (ή προαιρετικά αν συνοδεύεται με το γνώρισμα **minOccurs=0**) κάθε στοιχείο της ομάδας. Οποιαδήποτε όμως σειρά εμφάνισης θεωρείται αποδεκτή. Οι επιτρεπτές τιμές των γνωρισμάτων **minOccurs** και **maxOccurs** στις δηλώσεις των στοιχείων που περιλαμβάνονται σε μια τέτοια ομάδα είναι **0** και **1**.

Παράδειγμα 6.11. Στο παρακάτω τμήμα περιγραφής σε XML Schema:

```
<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** πρέπει να περιλαμβάνει από μια εμφάνιση καθενός από τα στοιχεία **όνομα**, **επώνυμο** και μια ή καμία εμφάνιση του στοιχείου **πατρώνυμο** αλλά με οποιαδήποτε σειρά[6].

❑

[6] Σκεφτείτε σαν άσκηση πως θα δηλώσετε το ίδιο πράγμα σε DTD.

Θα πρέπει ακόμη να αναφερθεί ότι το XML Schema επιβάλλει ότι μια ομάδα που δηλώνεται με το **all** πρέπει να εμφανίζεται σαν το μοναδικό υποστοιχείο του ορισμού του περιεχομένου ενός σύνθετου τύπου.

Παράδειγμα 6.12. Δεν είναι επιτρεπτή μια δήλωση 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>
```

❑

Να σημειωθεί επίσης ότι μέσα σε μια ομάδα που δηλώνεται με το **all** επιτρέπεται να περιλαμβάνονται μόνο ξεχωριστά στοιχεία και όχι ομάδες στοιχείων.

6.7. Ομαδοποίηση γνωρισμάτων

Εκτός από την ομαδοποίηση των στοιχείων που περιγράψαμε στην προηγούμενη ενότητα, η XML Schema επιτρέπει και την ομαδοποίηση γνωρισμάτων. Για τον σκοπό αυτό χρησιμοποιείται το στοιχείο **attributeGroup**.

Παράδειγμα 6.13. Στο XML Schema που ακολουθεί ορίζεται μια ομάδα γνωρισμάτων με όνομα **personAttributes**, η οποία περιλαμβάνει τρία γνωρίσματα, τα γνωρίσματα **ΑΔΤ** (αριθμός δελτίου ταυτότητας), **ΑΦΜ** (αριθμός φορολογικού μητρώου) και **φύλο**. Το πρώτο είναι τύπου **string**, το δεύτερο είναι τύπου **positiveInteger**, ενώ για το τρίτο έχει οριστεί ένας ανώνυμος τύπος ο οποίος περιλαμβάνει τις δύο τιμές **άνδρας** και **γυναίκα**, που είναι και οι μόνες αποδεκτές τιμές για το γνώρισμα **φύλο**:

```
<xs:attributeGroup name="personAttributes">
   <xs:attribute name="ΑΔΤ"  type="xs:string" use="required"/>
   <xs:attribute name="ΑΦΜ" 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>
```

❑

Έχοντας στη διάθεση μας τον ορισμό μιας ομάδας γνωρισμάτων, όπως αυτόν της ομάδας με όνομα **personAttributes** του Παραδείγματος 6.13, μπορούμε να τον επικαλεστούμε σε άλλο σημείο της περιγραφής για να δηλώσουμε ότι ένα στοιχείο διαθέτει τα γνωρίσματα που περιγράφονται στην συγκεκριμένη ομάδα.

Παράδειγμα 6.14. Στο τμήμα περιγραφής σε XML Schema που ακολουθεί γίνεται επίκληση της δήλωσης του συνόλου γνωρισμάτων **personAttributes** του Παραδείγματος 6.13, μέσω του γνωρίσματος **ref** που μας παρέχει η σύνταξη της XML Schema, προκειμένου να δηλώσουμε ότι το στοιχείο **φοιτητής** έχει ως γνωρίσματα, 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>
```

Η παραπάνω δήλωση εξασφαλίζει ότι το στοιχείο φοιτητής θα διαθέτει τα τρία γνωρίσματα που περιγράψαμε προηγουμένως. Αξίζει επίσης να σημειωθεί ότι αν το τεκμήριο μας επιθυμούμε να περιλαμβάνει και άλλα στοιχεία τα οποία να διαθέτουν την ίδια ομάδα γνωρισμάτων (π.χ. τα στοιχεία **μέλοςΔεπ**, **εργαζόμενος** κ.λ.π.), αυτό μπορεί να γίνει με δηλώσεις ορισμού των στοιχείων παρόμοιες με αυτήν του στοιχείου **φοιτητής**, στις οποίες θα υπάρχει μια απλή αναφορά στην ίδια ομάδα γνωρισμάτων, χωρίς να χρειάζεται να οριστεί ξανά η ομάδα αυτή.

❑

Χρησιμοποιώντας ομάδες γνωρισμάτων κατά τον τρόπο που εξετάσαμε στα Παραδείγματα 6.13 και 6.14, μπορούμε να διαμορφώσουμε πιο ευανάγνωστες περιγραφές σχήματος, και να 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 από πολλούς ορισμούς. Μια ομάδα γνωρισμάτων είναι δυνατό να περιλαμβάνει με τη σειρά της άλλες ομάδες γνωρισμάτων. Αξίζει επίσης να τονιστεί ότι τόσο οι δηλώσεις γνωρισμάτων όσο και οι αναφορές σε ομάδες γνωρισμάτων πρέπει να τοποθετούνται στο τέλος των ορισμών συνθέτων τύπων.

6.8. Επίλογος

Τελειώνοντας αυτήν τη σύντομη περιγραφή των δυνατοτήτων της XML Schema, θα θέλαμε να τονίσουμε ότι η περιγραφή αυτή απέχει πολύ από το να είναι πλήρης. Στόχος μας δεν ήταν να αναπτύξουμε με πληρότητα το σύνολο των δυνατοτήτων της XML Schema, η οποία είναι μια εξαιρετικά πλούσια γλώσσα περιγραφής σχήματος, αλλά να εισάγουμε τον αναγνώστη στις βασικές ιδέες, στη βασική φιλοσοφία της γλώσσας. Για μια πλήρη περιγραφή της XML Schema, ο αναγνώστης παροτρύνεται να ανατρέξει στη βιβλιογραφία [4,5,6] που προτείνεται στο τέλος του κεφαλαίου.

Βιβλιογραφία

1. S. Abiteboul, P. Buneman, D. Suciu "*Data on the Web: From Relations to Semistructured Data and XML*" Morgan Kaufmann Publishers 2000.
2. "*Extensible Markup Language (XML) 1.0 (Second Edition)*", W3C Recommendation, 6 October 2000. Διαθέσιμο από την διεύθυνση: http://www.w3.org/TR/REC-xml
3. "Namespaces in XML", W3C Recommendation, World Wide Web Consortium 14 January1999. Διαθέσιμο από την διεύθυνση: http://www.w3.org/TR/1999/REC-xml-names-19990114
4. "XML Schema Part 0: Primer" W3C Recommendation, 2 May 2001. Διαθέσιμο από την διεύθυνση: http://www.w3.org/TR/xmlschema-0/
5. "XML Schema Part 1: Structures" W3C Recommendation, 2 May 2001. Διαθέσιμο από την διεύθυνση: http://www.w3.org/TR/xmlschema-1/
6. "XML Schema Part 2: Datatypes" W3C Recommendation, 02 May 2001. Διαθέσιμο από την διεύθυνση: http://www.w3.org/TR/xmlschema-2/

Ευρετήριο Όρων