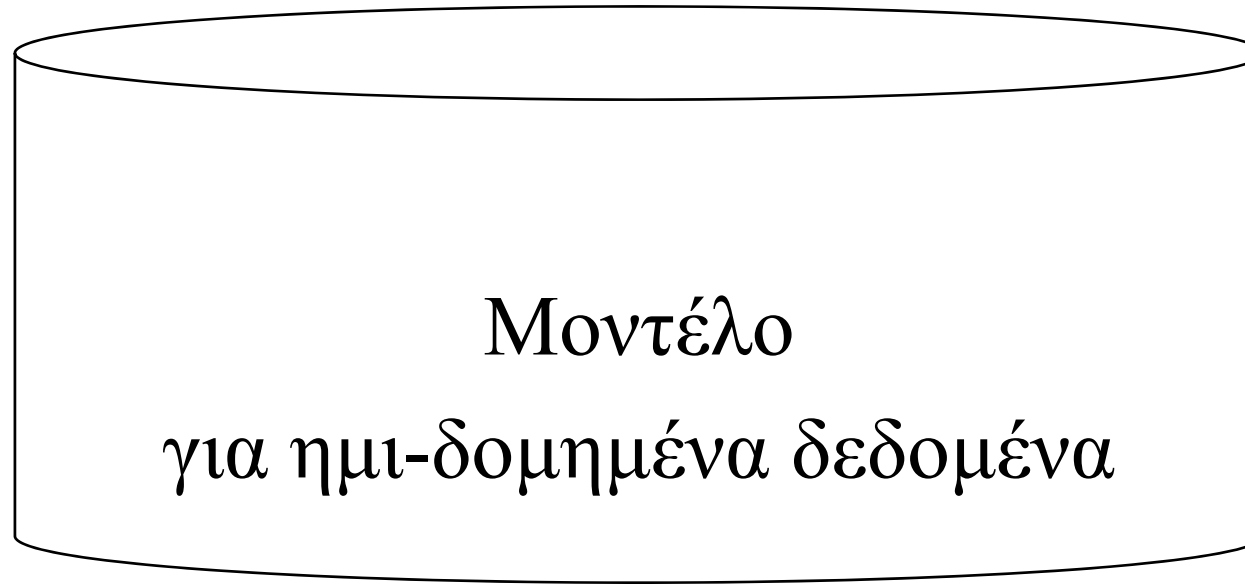




- Ημι-δομημένα δεδομένα
- XML
- DTD – XML Schema

Εισαγωγικά

- 3 βασικές κατηγορίες δεδομένων:
 - Δομημένα (structured): τα δεδομένα σε ΒΔ, όπου η αναπαράσταση γίνεται σύμφωνα με αυστηρά καθορισμένο σχήμα. Το ΣΔΒΔ είναι υπεύθυνο για να ελέγχει και να διασφαλίζει όλα τα δεδομένα να είναι εναρμονισμένα με το σχήμα και να μην παραβιάζουν τους περιορισμούς.
 - Ημι-δομημένα (semi-structured): επόμενες διαφάνειες
 - Μη δομημένα (unstructured): υπάρχει πολύ λίγη πληροφορία σχετικά με τον τύπο δεδομένων, π.χ. δεδομένα σε HTML σελίδες.

Ημι-δομημένα δεδομένα

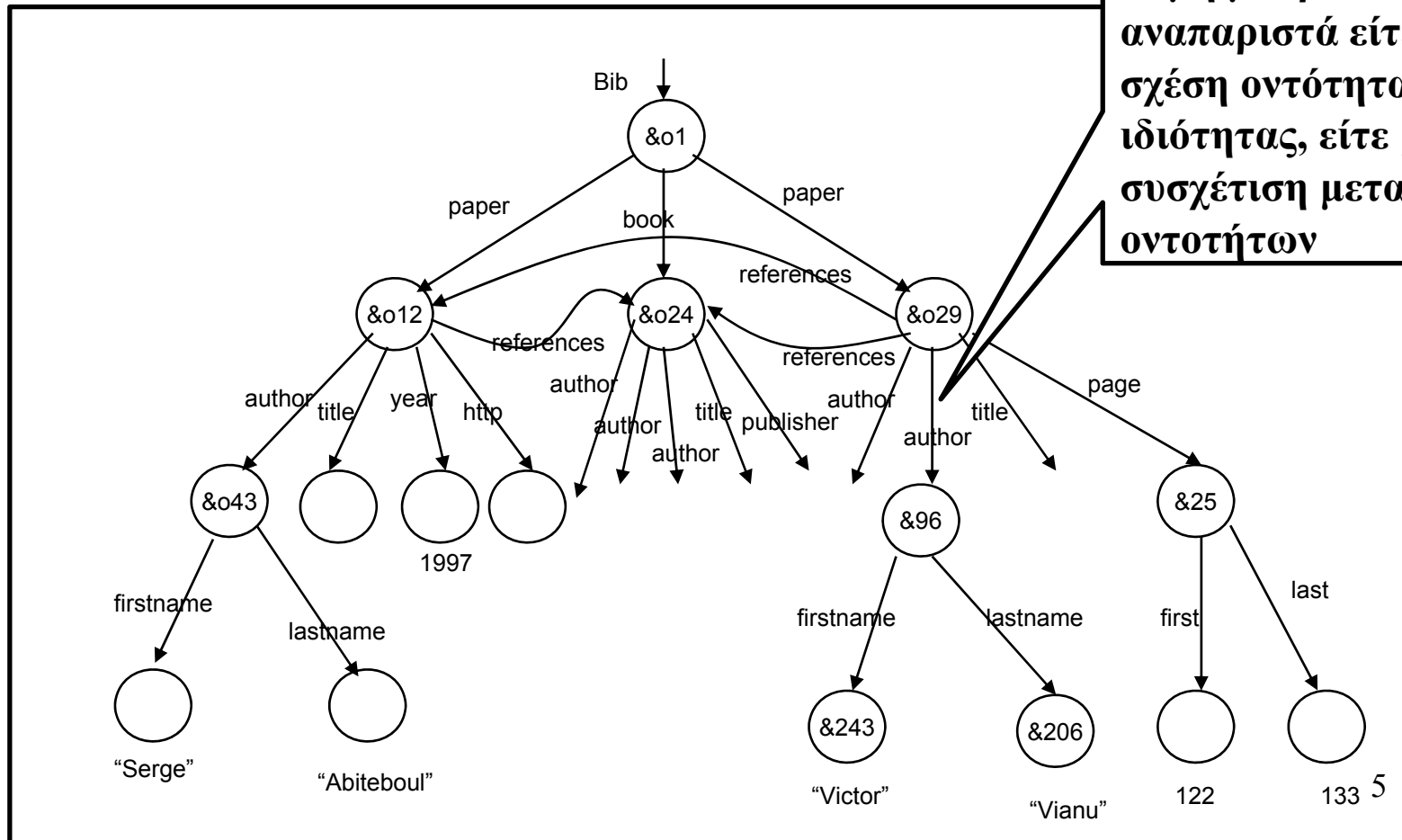
- Τα δεδομένα μπορεί να έχουν κάποια δομή αλλά αυτή η δομή δεν είναι όμοια για όλα τα δεδομένα.
- Η πληροφορία για το σχήμα βρίσκεται μαζί με τα δεδομένα καθώς δεν είναι γνωστό εκ των προτέρων ποιες ιδιότητες υπάρχουν σε κάθε αντικείμενο/οντότητα.
- Αναφέρονται και ως αυτοπεριγραφόμενα δεδομένα (self-describing) .

Κίνητρο για ημι-δομημένα δεδομένα/μοντέλο

- Παροχή ευελιξίας όσον αφορά τα δεδομένα, καθώς δεν απαιτείται η αυστηρή εναρμόνιση με κάποιο σχήμα.
 - Ουσιαστικά το σχήμα μπορεί να μεταβάλλεται αυθαίρετα με το χρόνο και ανάμεσα στα δεδομένα της ΒΔ την ίδια χρονική στιγμή.
 - Υπάρχει τίμημα όσον αφορά την αποδοτικότητα στην επεξεργασία ερωτημάτων.
- Καλύτερη υποστήριξη ολοκλήρωσης ΒΔ (database integration) και διαμοιρασμού πληροφορίας στον παγκόσμιο ιστό.

Αναπαράσταση

- Ως ένας κατευθυνόμενος γράφος
 - όχι απαραίτητα δένδρο



Η ετικέτα σε μία ακμή μπορεί να αναπαριστά είτε μία σχέση οντότητας-ιδιότητας, είτε μία συσχέτιση μεταξύ οντοτήτων

XML (eXtensible Markup Language)

- Ο βασικός και πιο διαδεμένος τρόπος αναπαράστασης ημιδομημένων δεδομένων.
- Είναι πρότυπο του World Wide Web Consortium και αναπαριστά τα ημιδομημένα δεδομένα γραμμικά.
- Η XML μπορεί να θεωρηθεί:
 - Ως μία γλώσσα σήμανσης για έγγραφα.
 - Ένα μοντέλο για ιεραρχικά δεδομένα.
 - Ένας συμβολισμός (notation) για επικοινωνία δεδομένων μαζί με τη δομή τους.
 - Υπόβαθρο για Υπηρεσίες Ιστού (Web Services) και σημασιολογικό ιστό (Semantic Web).
- Αλλά ουσιαστικά είναι μία γλώσσα σήμανσης όπου η συλλογή ετικετών είναι επεκτάσιμη.

Σημασιολογικές Ετικέτες

- Περιγράφουν τα δεδομένα.
 - Όχι απλώς την παρουσίαση τους όπως στην HTML.
- Οι ετικέτες τυπικά είναι ζευγάρια, π.χ. `<FOO> ... </FOO>`.
 - Μπορούν όμως να είναι και μονές, π.χ. `<FOO/>`.
- Ανάμεσα σε ένα ζευγάρι ετικετών μπορεί να βρίσκεται κείμενο και άλλα ζευγάρια ετικετών.
- Δεν υπάρχει περιορισμός στον βαθμό και τον τρόπο που μπορούν οι ετικέτες να είναι εμφωλιασμένες.
- Στο όνομά τους παίζει ρόλο αν χρησιμοποιούνται μικρά ή κεφαλαία γράμματα.
- Ένα στοιχείο (element) αποτελείται από ένα ζευγάρι ετικέτες και ό,τι βρίσκεται ανάμεσα σε αυτές.

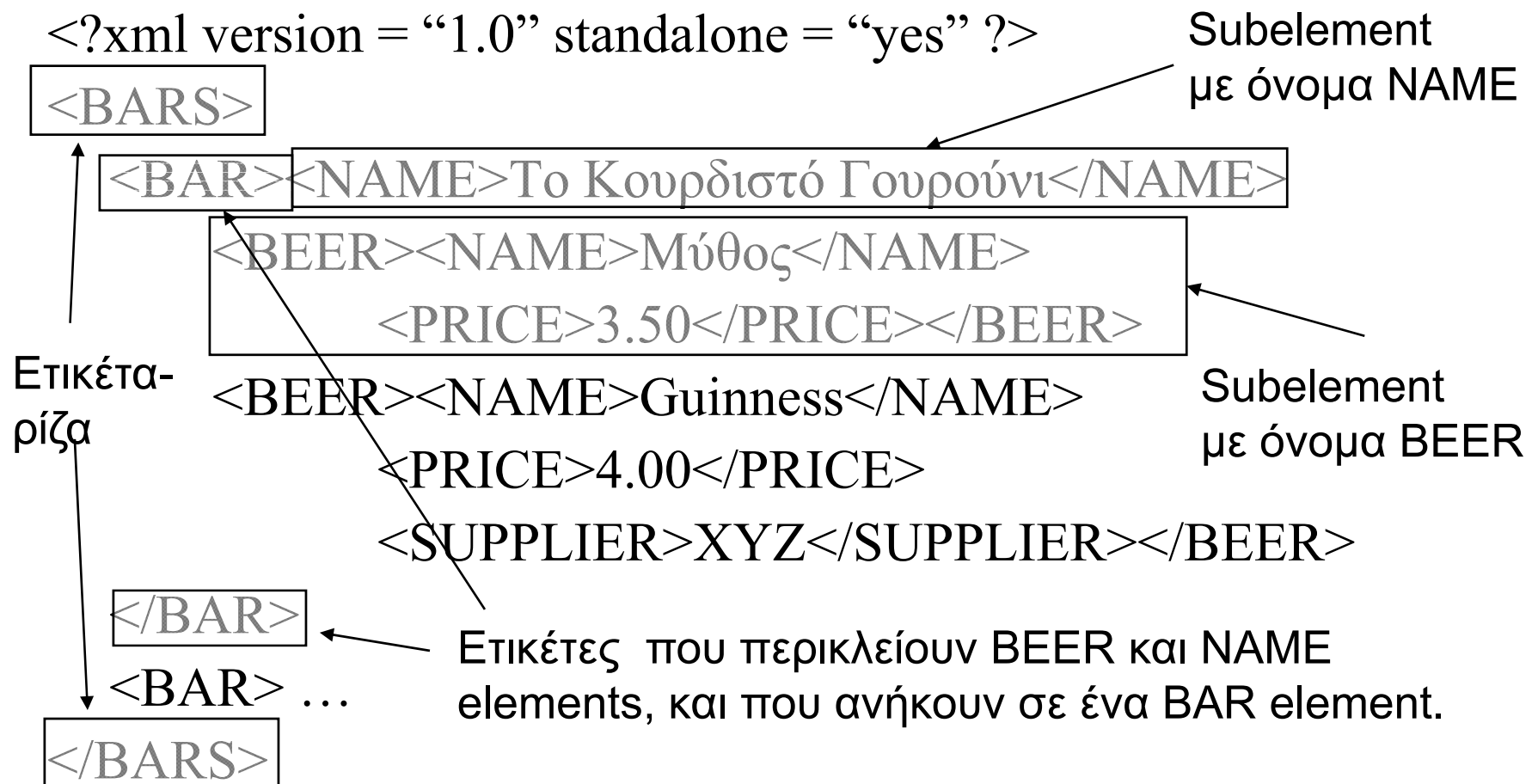
Τρόποι Χρήσης

- Η XML μπορεί να χρησιμοποιηθεί με ή χωρίς την ύπαρξη κάποιου σχήματος:
 1. Η καλώς ορισμένη (well-formed) XML δίνει απόλυτη ελευθερία στον χρήστη όσον αφορά τις ετικέτες που μπορεί να χρησιμοποιεί.
 - Είναι πολύ κοντά στο μοντέλο ΗΔΔ.
 2. Η έγκυρη (valid) XML εναρμονίζεται με κάποιο Document Type Definition (DTD) ή XML Schema, που ορίζει τις επιτρεπόμενες ετικέτες και τη γραμματική του XML αρχείου.
 - Είναι κάπου ανάμεσα σε αυστηρά καθορισμένα μοντέλα και μοντέλα ΗΔΔ.

Καλώς ορισμένη (Well-Formed) XML

- Το αρχείο αρχίζει με μία δήλωση που περικλείεται από `<?xml ... ?>` .
- Μία τυπική δήλωση είναι:
`<?xml version = "1.0" standalone = "yes" ?>`
 - “standalone” σημαίνει ουσιαστικά ότι δεν υπάρχει DTD.
- Υπάρχει μία ετικέτα-ρίζα (root tag) η οποία περικλείει όλες τις υπόλοιπες (εμφωλιασμένες) ετικέτες.

Καλώς ορισμένη (Well-Formed) XML: Παράδειγμα



Ιδιότητες (Attributes)

- Τα στοιχεία μπορεί να περιέχουν ιδιότητες.
- Οι ιδιότητες έχουν τη μορφή *όνομα-τιμή*.
- Βρίσκονται μέσα στην αρχική ετικέτα του στοιχείου.
- Οι ιδιότητες μπορούν να αναπαραστήσουν και αυτές ακμές στο γράφο του μοντέλου ΗΔΔ.
 - Ιδίως ακμές που καθιστούν τον γράφο να μην είναι δένδρο.
- Δεν υπάρχει γενικός κανόνας πότε μία οντότητα ή αντικείμενο πρέπει να αναπαριστάται ως (υπο-)στοιχείο ή ιδιότητα, αλλά μια γενική πρακτική είναι οι ιδιότητες να χρησιμοποιούνται κυρίως για μεταδεδομένα.

<BEER source="http://oles-oi-times.gr" validUntil="1/1/2010">

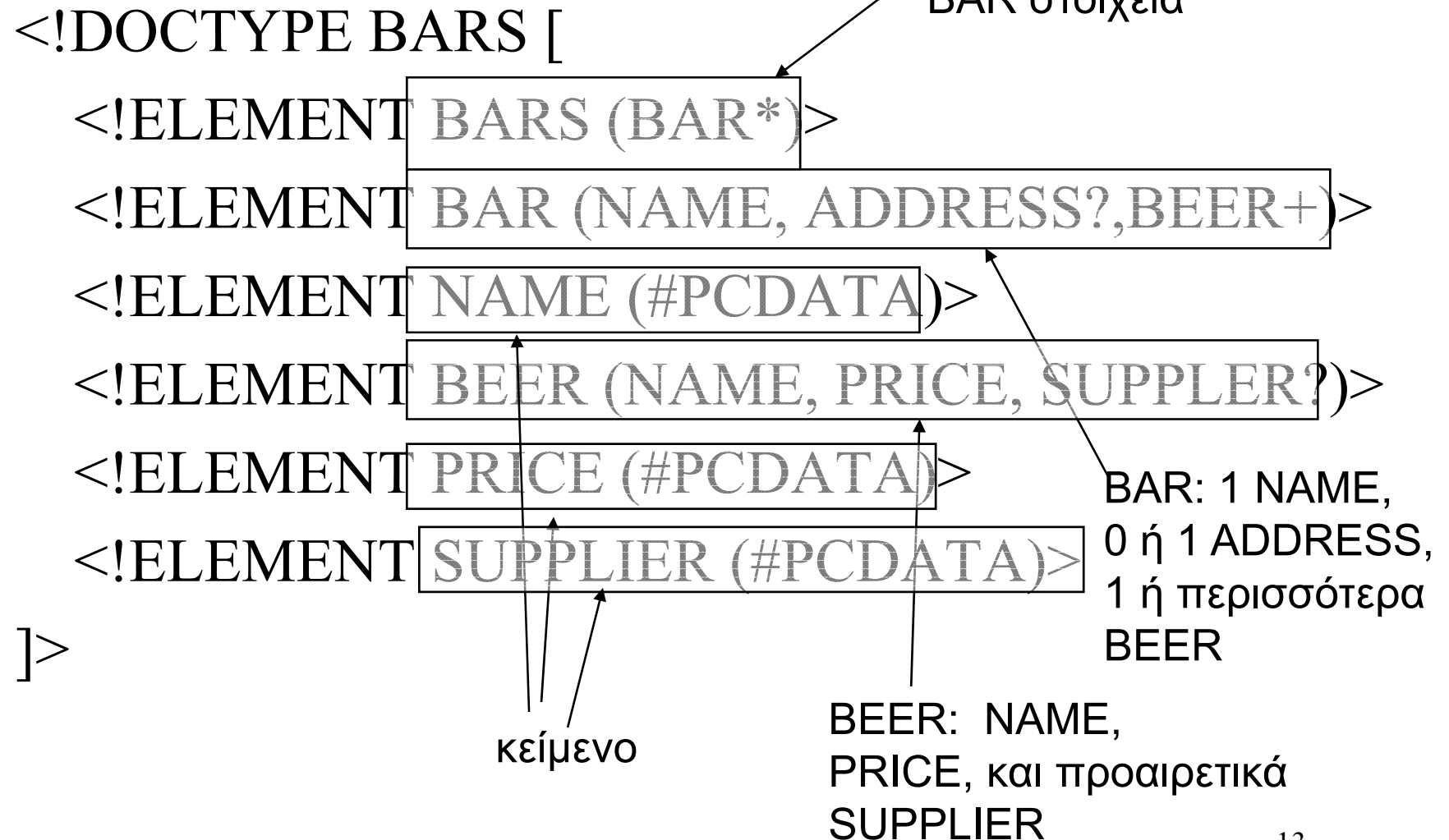
<NAME>Guinness</NAME><PRICE>4.00</PRICE>

</BEER>

Document Type Definitions - DTD

- Βασική δομή:
 <!DOCTYPE <root tag> [
 <!ELEMENT <name>(<components>)>
 ... more elements ...
]>
• Πεδίο «!ELEMENT»: Η περιγραφή ενός στοιχείου που αποτελείται από το όνομα και μέσα σε παρένθεση
 - είτε τη σειρά+πλήθος των υποστοιχείων,
 - είτε, αν είναι φύλλο, την κωδική λέξη #PCDATA (Parsed Character DATA), που χονδρικά αντιστοιχεί σε string.

Παράδειγμα DTD



Περιγραφές στοιχείων σε DTD

- Τα υποστοιχεία εμφανίζονται με την διάταξη που ορίζεται.
- Το πλήθος ορίζεται με τα ακόλουθα σύμβολα:
 - * = 0..m
 - + = 1..m
 - ? = 0 ή 1.
- Το σύμβολο «|» συνδέει εναλλακτικές διατάξεις.

```
<!ELEMENT CONTACT (  
  (ADDRESS, ZIP, CITY?, COUNTRY) | EMAIL)>
```

Περιγραφές ιδιοτήτων σε DTD

```
<BEER source=http://oles-oi-times.gr validUntil="1/1/2010">  
  <NAME>Guinness</NAME><PRICE>4.00</PRICE>  
</BEER>
```

Παράδειγμα DTD ορισμού:

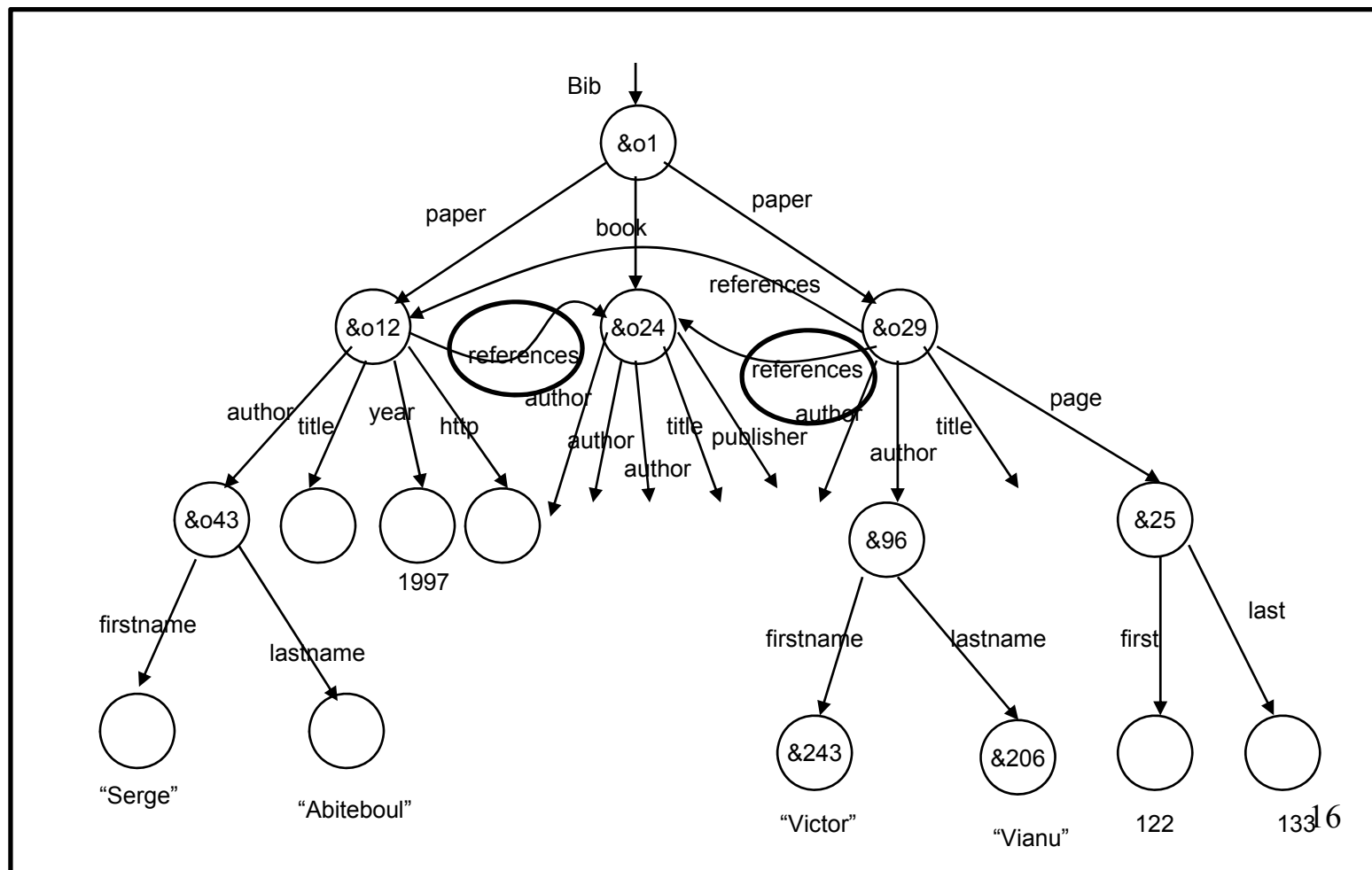
```
<!ATTLIST BEER source(http://oles-oi-times.gr |  
                    http://ola-ta-bar.gr) #IMPLIED  
          validUntil CDATA #REQUIRED)
```

Γενική μορφή:

```
<!ATTLIST element-name attribute-name type  
          #IMPLIED | #REQUIRED>
```

Ταυτότητες και αναφορές

- Επιπλέον τύποι ιδιοτήτων: ID, IDREF, IDREFS



Παράδειγμα

Έστω οι ορισμοί:

```
<!ELEMENT OWNER (NAME) >
```

```
  <!ATTLIST OWNER ownerID ID #REQUIRED
                                owns IDREFS #IMPLIED>
```

```
<!ELEMENT BAR (NAME)>
```

```
  <!ATTLIST BAR barID ID #REQUIRED
                                ownedBy IDREF #IMPLIED>
```

Απόσπασμα από XML αρχείο:

```
<OWNER ownerID="o001" owns="b1 b2">
```

```
  <NAME>Όνομα Επίθετο</NAME>
```

```
</OWNER>
```

```
<BAR barID="b1" ownedBy="o001"> <NAME>Όνομα 1</NAME></BAR>
```

```
<BAR barID="b2" > <NAME>Όνομα 2</NAME></BAR>
```

Τρόπος Χρήσης

<?xml version = "1.0" standalone = "no"?>

<!DOCTYPE BARS [

 <!ELEMENT BARS (BAR*)>

 <!ELEMENT BAR (NAME, ADDRESS?, BEER+)>

 <!ELEMENT NAME (#PCDATA)>

 <!ELEMENT BEER (NAME, PRICE, SUPPLIER?)>

 <!ELEMENT PRICE (#PCDATA)>

 <!ELEMENT SUPPLIER (#PCDATA)>

<BARS>

 <BAR><NAME>Το Κουρδιστό Γουρούνι</NAME>

 <BEER><NAME>Μύθος</NAME>

 <PRICE>3.50</PRICE></BEER>

 </BAR>

</BARS>

Εναλλακτικά: <!DOCTYPE BARS SYSTEM "filename.dtd">

Βασικά Μειονεκτήματα

- Περιορισμοί στους τύπους δεδομένων.
- Το DTD έχει τη δική του σύνταξη και χρειάζεται ειδικούς επεξεργαστές
 - Είναι καλύτερο το σχήμα ενός XML αρχείου να ορίζεται με μία γραμματική βασισμένη πάλι σε XML ώστε οι ίδιοι επεξεργαστές για τα αρχεία να χρησιμοποιούνται και για τα σχήματα.
- Βάσει του DTD τα στοιχεία πρέπει να ακολουθούν ορισμένη διάταξη.

XML Schema

- Είναι πιο γενική μέθοδος με μεγαλύτερες δυνατότητες.
- Οι ορισμοί είναι και αυτοί XML αρχεία.
 - Η περιγραφή των στοιχείων γίνεται με στοιχεία.
- Είναι πρότυπο του W3C για μοντελοποίηση με XML
- Τα αρχεία ενός XML Schema έχουν τυπικά κατάληξη .xsd

XML Schema: Σύνοψη δυνατοτήτων

- Τι μπορεί να οριστεί:
 - Ποια στοιχεία/ιδιότητες είναι υποχρεωτικά και ποια προαιρετικά.
 - Οι τύποι των στοιχείων και ιδιοτήτων και το εύρος τιμών.
 - Αριθμός Εμφανίσεων.
 - Σχετική Διάταξη.
 - Ορισμός κλειδιών (πρωτευόντων και ξένων).

Δομή ενός αρχείου XML schema

```
<? xml version = ... ?>
```

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

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

Ορίζει ως "xs" το
namespace που
περιγράφεται στο
αντίστοιχο URL.

Συνεπώς, η προσθήκη του
"xs" στα στοιχεία του σχήματος
σημαίνει αναφορά σε ετικέτες
που ορίζονται στο συγκεκριμένο
Namespace, που είναι το
XML-Schema namespace.

Το xs:element στοιχείο

- Έχει ως ιδιότητες:
 - name = Το όνομα της ετικέτας για το στοιχείο που ορίζεται.
 - type = Ο τύπος του στοιχείου.
 - Μπορεί να είναι βασικός τύπος ορισμένος ήδη (πχ. string, integer, positiveInteger, negativeInteger, short, long, date, dateTime, time, id, idref, anyURI).
 - Ή τύπος που έχει οριστεί στο συγκεκριμένο αρχείο.
- Παράδειγμα:
`<xs:element name = "NAME" type = "xs:string" />`
Περιγράφει στοιχεία όπως `<NAME>Κουρδιστό Γουρούνι</NAME>`

Σύνθετοι Τύποι - xs:complexType

- Χρησιμοποιείται για περιγραφή στοιχείων που περιέχουν υποστοιχεία.
- Τα πιο συχνά υποστοιχεία του xs:complexType είναι:
 - xs:sequence: μία σειρά στοιχείων που πρέπει να εμφανιστούν σύμφωνα με την ορισμένη διάταξη.
 - xs:all: μία σειρά στοιχείων, τα οποία πρέπει να εμφανιστούν όλα μία το πολύ φορά με αυθαίρετη διάταξη.
 - xs:choice: ένα μόνο στοιχείο μπορεί υπάρξει κάθε φορά.

Παράδειγμα με xs:complexType

```
<xs:complexType name = "beerType">
  <xs:sequence>
    <xs:element name = "NAME" type = "xs:string" minOccurs = "1" maxOccurs = "1" />
    <xs:element name = "PRICE" type = "xs:float" minOccurs = "0" maxOccurs = "1" />
  </xs:sequence>
</xs:complexType>

<xs:element name = "BAR">
  <xs:complexType >
    <xs:sequence>
      <xs:element name = "NAME" type = "xs:string" minOccurs = "1" maxOccurs = "1" />
      <xs:element name = "BEER" type = "beerType" minOccurs = "0" maxOccurs = "unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Παράδειγμα με xs:complexType

```
<xs:complexType name = "beerType2">  
  <xs:choice>  
    <xs:element name = "ALE"  
      type = "xs:string"  
      minOccurs = "1" maxOccurs = "1" />  
    <xs:element name = "BLACK"  
      type = "xs:string"  
      minOccurs = "1" maxOccurs = "1" />  
  </xs:choice>  
</xs:complexType>
```

xs:attribute

- Ορίζονται ως υποστοιχεία του xs:complexType.
- Ιδιότητες:
 - name = Το όνομα της ιδιότητας για το στοιχείο που ορίζεται.
 - type = Ο τύπος της ιδιότητας.
 - use: “required” ή “optional”
- Παράδειγμα:

```
<xs:complexType name = "beerType">  
  <xs:attribute name = "name" type = "xs:string" use = "required" />  
  <xs:attribute name = "price" type = "xs:float" use = "optional" />  
</xs:complexType>
```

Περιγράφει στοιχεία όπως

```
<XXX name="Κουρδιστό Γουρούνι" price="4.00"/>
```

Περιορισμοί σε απλούς τύπους - xs:simpleType

- Εκφράζονται είτε ως απαριθμήσεις είτε ως καθορισμός εύρους τιμών.

```
<xs:simpleType name = "Country">  
  <xs:restriction base = "xs:string">  
    <xs:enumeration value = "Greece" />  
    <xs:enumeration value = "EU" />  
    <xs:enumeration value = "Other" />  
  </xs:restriction>  
</xs:simpleType>  
  
<xs:simpleType name = "priceAllowed">  
  <xs:restriction  
    base = "xs:float"  
    minInclusive = "0.01"  
    maxExclusive = "1000.00" />  
</xs:simpleType>
```

Ορισμός κλειδού – xs:key

- Ορίζεται ως υποστοιχείο του xs:element.
- Το νόημα είναι πως μέσα στα περιεχόμενα του δεδομένου στοιχείου, όλα τα υποστοιχεία τα οποία μπορούν να προσπελασθούν με συγκεκριμένο μονοπάτι (selector) έχουν μοναδική τιμή για το συνδυασμό των ορισμένων πεδίων (fields).
- Παράδειγμα: Μέσα στο στοιχείο BAR, η ιδιότητα name μαζί με το υποστοιχείο SUPPLIER του υποστοιχείου BEER, είναι μοναδικά.

```
<xs:element name = "BAR" ... >
```

```
...
```

```
  <xs:key name = "barKey">
```

```
    <xs:selector xpath = "BEER" />
```

```
    <xs:field xpath = "@name" />
```

```
    <xs:field xpath = "SUPPLIER" />
```

```
  </xs:key>
```

```
</xs:element>
```

Ορισμός ξένου κλειδού – xs:key ref

- Σύνταξη:

```
<xs:keyref name = όνομα-ξένου-κλειδιού refer = όνομα κλειδιού  
    <xs:selector xpath = μονοπάτι />  
    <xs:field xpath = μονοπάτι />  
</xs:keyref>
```

Βιβλιογραφία

- The W3C Consortium Tutorial is short but informative:
 - <http://www.w3schools.com/schema/>
- D. Fallside, XML Schema Part 0: Primer, 2001:
 - <http://www.w3.org/TR/xmlschema-0/>
- N. Bradley, The XML Companion (3rd Edition), Addison-Wesley, 2002.