

XML – based technologies



Αναγνωστόπουλος Ιωάννης - Λυκουρέντζου Ιωάννα

Περιεχόμενα



- XML
- Έλεγχος εγκυρότητας XML
 - DTD
 - Schema
- XML DOM
- Περαιτέρω δυνατότητες και τεχνολογίες της XML
 - XPath
 - XSLT
 - XPointer
 - XLink
 - XQuery
- Web services με τη βοήθεια των τεχνολογιών XML
 - Web services
 - WSDL
 - SOAP
 - RDF και OWL
 - RSS
 - WAP και WML

XML



- Extensible Markup Language – Απόγονος της (Standard Generalised Markup Language)
- Σχεδιασμένη για να εισάγει την sgml στο διαδίκτυο
- Μια μετα-γλώσσα (meta-language)
 - Μια γλώσσα που παράγει γλώσσες
 - XHTML – η πιο πρόσφατη έκδοση της HTML
 - VoiceXML – Για διαδραστική φωνητική επικοινωνία ανθρώπου – υπολογιστή μέσω τηλεφώνου, PDA ή υπολογιστή
 - WSDL – για την περιγραφή των web services
 - WAP και WML – πρωτόκολλο και γλώσσα σήμανσης (markup) για φορητές συσκευές
 - RSS γλώσσες για τροφοδότες νεων (news feeds)
 - RDF/OWL – για περιγραφή πόρων και οντολογιών
 - Synchronised Multimedia Integration Language (SMIL) – για περιγραφή πολυμέσων στον Ιστό (smil+voiceXML)

Voice XML - παράδειγμα



```
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml">
  <form>
    <block>
      <prompt>
        Hello world!
      </prompt>
    </block>
  </form>
</vxml>
```

Αναμενόμενα κέρδη



- Αποθήκευση μια φορά και μορφοποίηση πολλές φορές
- Ανεξαρτησία από υλικό – λογισμικό
 - Αποθήκευση σε μορφή απλού κειμένου (plain text format)
 - Διευκόλυνση της μεταφοράς/ανταλλαγής δεδομένων ανάμεσα σε ασύμβατες κατά τα άλλα εφαρμογές
 - Διαθεσιμότητα των δεδομένων σε μεγαλύτερο εύρος εφαρμογών αλλά και μέσων (PDAs, news feeds, voice machines)
- Συγκέντρωση δεδομένων μια φορά και ανταλλαγή πολλές φορές
- Ταχύτερη εστιασμένη αναζήτηση
- Μικρότερη συμφόρηση του δικτύου
- Οι μηχανές αναζήτησης αναζητούν συγκεκριμένες ετικέτες (tags) στον κώδικα XML
 - Ταχύτερα
 - Με μεγαλύτερη ακρίβεια

XML



- HTML
 - Χρησιμοποιεί tags ανάμεσα στο κείμενο...
 - ...για να περιγράψει το layout της σελίδας
 - `<p> Alan, 42 years, <i>agb@abc.com</i>`
 - Σχεδιάστηκε ειδικά για να περιγράψει την παρουσίαση και όχι το περιεχόμενο
- XML
 - Σχεδιάστηκε ειδικά για να περιγράψει το περιεχόμενο (content) και όχι την παρουσίαση μιας σελίδας
 - Βασικές διαφορές από την HTML
 - Μπορεί κανείς να ορίσει νέα tags κατά βούληση
 - Nested tags σε οποιοδήποτε βάθος
 - Ένα έγγραφο XML μπορεί προαιρετικά να περιέχει μια περιγραφή της γραμματικής του
 - Τα tags δομούν το περιεχόμενο
 - `<person><name>Kostas</name>...</person>`
 - Το πως θα εμφανιστούν ορίζεται ξεχωριστά από κάποιο stylesheet (XSL)

XML



- Ο ρόλος της XML
 - Προτάθηκε σαν μια markup γλώσσα περιγραφής εγγράφων
 - Καταγωγή απο την SGML (ψηφιακές βιβλιοθήκες)
 - Εξελίσσεται όμως σε ένα παγκόσμιο πρότυπο για ανταλλαγή πληροφορίας
- Βασικό συστατικό της XML είναι το element
 - Τιμή του element → Κείμενο που περικλείεται από ένα ζεύγος tags
 - Εκτός από την λογική δομή (elements), τα tags περιγράφουν και την φυσική δομή (entities)

```
<person>
  <name>Alan</name>
  <age>42</age>
  <email>alan@abc.com</email>
  <email>abrown@mail.com</email>
</person>
```

XML – Elements/Attributes



```
<root>
  <child>
    <subchild>....</subchild>
  </child>
</root>
```

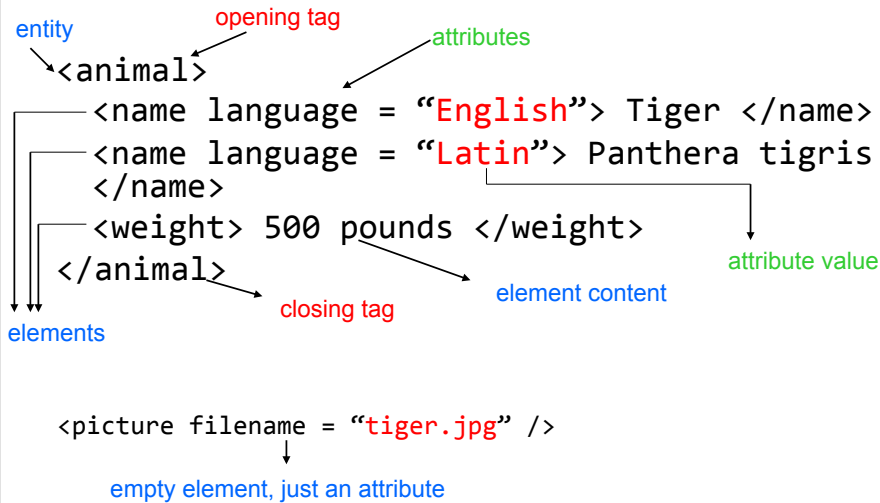
```
<person sex="female">
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>
```

```
<person>
  <sex>female</sex>
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>
```

Elements
Person, firstname, lastname

Attributes
sex (value="female")

Βασικά συνθετικά του συντακτικού της XML



XML attributes



- Ένα element μπορεί να έχει κανένα, ένα ή πολλά attributes
 - Περιγράφουν ιδιότητες (properties) του element

```

<product>
  <name language="French">trompette no 6</name>
  <price currency="Euro"> 420.12 </price>
  <address format="XL1245" language="French">
    <street>31 rue Croix-Bosset</street>
    <zip>92874</zip>
  </address>
</product>

```

- Όπως και τα elements, τα attributes ορίζονται από τον χρήστη

XML - Παράδειγμα

```

<a>
  <b id="x1">
    <c>Κώστας</c>
    <c>Γιάννης</c>
  </b>
  <d/>
  <b id="x2">
    <c>Ηλίας</c>
  </b>
</a>
        
```

File Format
(Instance)

Tree Structure
(instance)

Schema
(Document Type
Definition, DTD)

Παράδειγμα – XML Δένδρου

Root element
<bookstore>

Parent Child

Ποιο το file format instance ?

Επιπλέον στοιχεία της XML



- Στοιχεία που δεν χρειάζονται στην ανταλλαγή δεδομένων
- Σχόλια

```
<!-- this is a comment -->
```
- Αρχική γραμμή

```
<?xml version="1.0"?>
```
- Προαιρετικό: Document Type Definition (DTD)
 - Ορίζει την γραμματική του κειμένου

```
<?xml version="1.0"?>
<!DOCTYPE name [markupdeclarations]>
<name>...</name>
```

XML – Name conflicts



- Εφόσον τα ονόματα των element στην XML δεν είναι προκαθορισμένα (δεσμευμένα), ενδέχεται δύο διαφορετικά XML documents να χρησιμοποιούν το ίδιο element name

```
<table>
  <tr>
    <td>Apples</td>
    <td>Bananas</td>
  </tr>
</table>
```

HTML κώδικας
Το <table> δηλώνει
πίνακα της HTML

```
<table>
  <name>African Coffee Table</name>
  <width>80</width>
  <length>120</length>
</table>
```

XML κώδικας
Το <table> περιέχει πληροφορία
για ένα τραπέζι

XML Namespaces



- Λύση... τα XML Namespaces
- Τα ονόματα των elements ενός αρχείου χαρακτηρίζονται από μια ετικέτα (super label)
- Έτσι, δύο conflicting elements πλέον έχουν διαφορετική ετικέτα και επομένως δεν υπάρχει περίπτωση σύγχυσης
- Το label αυτό ονομάζεται *namespace* και αποτελεί το όνομα της συλλογής των ονομάτων του αρχείου
- Το XML namespace attribute τοποθετείται στο αρχικό tag ενός element και έχει την παρακάτω σύνταξη:

```
xmlns:namespace-prefix="namespaceURI"
```

Ή

```
xmlns=namespaceURI" (obsolete)
```

- Όταν ένα namespace καθορίζεται στο αρχικό tag ενός element, όλα τα child elements με το ίδιο namespace-prefix συνδέονται με το ίδιο namespace
- Το URI στο όνομα ενός namespace δεν έχει *link πουθενά!* Χρησιμοποιείται αυτή η αναπαράσταση γιατί είναι *μοναδική*
<http://www.domain.com/ns/rivers/1.1>

XML Namespaces - Παράδειγμα



```
<root>
```

```
<h:table xmlns:h="http://www.w3.org/TR/html4/">
  <h:tr>
    <h:td>Apples</h:td>
    <h:td>Bananas</h:td>
  </h:tr>
</h:table>
```

} Namespace για το HTML Tag

```
<f:table xmlns:f="http://www.w3schools.com/furniture">
  <f:name>African Coffee Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>
```

} Namespace για το XML Tag

```
</root>
```


Παράδειγμα



```
<table>
  <tr>
    <td>Apples</td>
    <td>Bananas</td>
  </tr>
</table>
```

```
<table>
  <name>African Coffee Table</name>
  <width>80</width>
  <length>120</length>
</table>
```

```
<h:table xmlns:h="http://www.w3.org/TR/html4/">
  <h:tr>
    <h:td>Apples</h:td>
    <h:td>Bananas</h:td>
  </h:tr>
</h:table>
```

```
<f:table xmlns:f="http://www.w3schools.com/furniture">
  <f:name>African Coffee Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>
```

```
<table xmlns="http://www.w3.org/TR/html4/">
  <tr>
    <td>Apples</td>
    <td>Bananas</td>
  </tr>
</table>
```

```
<table xmlns="http://www.w3schools.com/furniture">
  <name>African Coffee Table</name>
  <width>80</width>
  <length>120</length>
</table>
```

Έλεγχος εγκυρότητας XML

XML DTD
XML Schema



DTD – Document Type Definition (από SGML)



- Προτάθηκε σαν μια γραμματική για τα XML έγγραφα – Χρησιμοποιείται για τον έλεγχο της εγκυρότητας τους
- Αν κάποιος έγγραφο XML αναφέρεται σε ένα DTD αλλά δεν το ακολουθεί, τότε το έγγραφο XML δεν γίνεται δεκτό ως έγκυρο
- Σε κάποιο βαθμό μπορεί να θεωρηθεί ως σχήμα για δεδομένα μορφοποιημένα σε XML
- Επιτρέπει
 - Σε κάθε έγγραφο XML να έχει περιγραφή της δομής του
 - Σε ανεξάρτητες ομάδες χρηστών να συμφωνήσουν στη χρήση μιας κοινής δομής για την ανταλλαγή δεδομένων
 - Σε μία εφαρμογή διαδικτύου να επικυρώσει τα δεδομένα τα οποία δέχεται
- Πολλοί οργανισμοί δημιουργούν DTDs για μία ευρεία γκάμα εφαρμογών (οικονομικές/ιατρικές συναλλαγές κ.τ.λ)

Δηλώσεις DTD



- Εσωτερικές
 - Internal DOCTYPE declaration
 - `<!DOCTYPE root-element [element declarations]>`
- Εξωτερικές (σε τοπικό filesystem)
 - External DOCTYPE declaration
 - `<!DOCTYPE root-element SYSTEM "filename">`
- Εξωτερικές (σε URI)
 - External DOCTYPE declaration
 - `<!DOCTYPE root-element PUBLIC "URI">`

XML - Internal Doctype Declaration



Internal DOCTYPE declaration
`<!DOCTYPE root-element [element-declarations]>`

```
<?xml version="1.0"?>
<!DOCTYPE note [
  <!ELEMENT note (to,from,heading,body)>
  <!ELEMENT to (#PCDATA)>
  <!ELEMENT from (#PCDATA)>
  <!ELEMENT heading (#PCDATA)>
  <!ELEMENT body (#PCDATA)>]
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend</body>
</note>
```

root-element

element declarations

XML - External Doctype Declaration



External DOCTYPE declaration
`<!DOCTYPE root-element SYSTEM "filename">`

```
<?xml version="1.0"?>
<!DOCTYPE note SYSTEM "{pathlink}\note.dtd">
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

note.dtd

```
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
```

Δομικά στοιχεία DTD



- Elements
 - Μπορούν να περιέχουν κείμενο, άλλα elements ή να είναι άδεια
 - HTML element: `<body>some text</body>`
 - XML element: `<message>some text</message>`
 - Empty element: `<hr>`
- Attributes
 - Παρέχουν επιπλέον πληροφορία σχετικά με τα elements
 - Τοποθετούνται πάντοτε εσωτερικά ενός element
 - Συντάσσονται σε ζεύγη (name/value)
 - Παράδειγμα: ``
- PCDATA
 - Parsed character data
 - Εξετάζεται από τον parser
 - Δεν πρέπει να περιέχει χαρακτήρες: `&`, `<`, `>`. Πρέπει να αντικαθίστανται με `&`, `<` και `>`
- CDATA
 - Character data
 - Δεν εξετάζεται από τον parser

Δήλωση elements



- Δήλωση
 - `<!ELEMENT element-name category>`
ή
`<!ELEMENT element-name (element-content)>`
- Empty element
 - `<!ELEMENT element-name EMPTY>`
- Element με PCDATA
 - `<!ELEMENT element-name (#PCDATA)>`
 - Παράδειγμα: `<!ELEMENT from (#PCDATA)>`
- Element με οποιοδήποτε περιεχόμενο
 - `<!ELEMENT element-name ANY>`
 - `<!ELEMENT note ANY>`
- Elements με elements-παιδιά
 - `<!ELEMENT element-name (child1)>`
ή
`<!ELEMENT element-name (child1,child2,...)>`
 - Παράδειγμα: `<!ELEMENT note (to,from,heading,body)>`
 - Τα elements-παιδιά πρέπει να δηλώνονται με την ακριβή σειρά εμφάνισής τους

Δήλωση elements



- Ακριβώς μία εμφάνιση element
 - <!ELEMENT element-name (child-name)>
 - Παράδειγμα: <!ELEMENT note (message)>
 - Το element message πρέπει να υπάρχει ακριβώς μία φορά εντός του element note
- Τουλάχιστον μία εμφάνιση
 - <!ELEMENT element-name (child-name+)>
 - Παράδειγμα: <!ELEMENT note (message+)>
- Προαιρετική εμφάνιση-όσες φορές χρειαστεί
 - <!ELEMENT element-name (child-name*)>
 - Παράδειγμα: <!ELEMENT student (course_left*)>
- Προαιρετική εμφάνιση-μονάχα μία φορά
 - <!ELEMENT element-name (child-name?)>
 - <!ELEMENT student (diploma_thesis?)>
- Είτε/είτε περιεχόμενο
 - Παράδειγμα: <!ELEMENT note (to,from,header,(message|body))>

Δήλωση attributes



- <!ATTLIST element-name attribute-name attribute-type default-value>
- Παράδειγμα:
 - XML κώδικας: <payment type="check" />
 - DTD: <!ATTLIST payment type CDATA "check">
- Attribute type (παραδείγματα)
 - CDATA → Character data
 - (en1|en2|..) → Τιμή από μία αριθμημένη λίστα
 - ID → Μοναδικό id
 - IDREF → To id ενός άλλου element
 - IDREFS → Μία λίστα άλλων ids
- Attribute value
 - Value → Η προκαθορισμένη τιμή του attribute
 - #REQUIRED To attribute περιλαμβάνεται υποχρεωτικά
 - #IMPLIED → To attribute δεν περιλαμβάνεται υποχρεωτικά
 - #FIXED value → Η τιμή του attribute είναι προκαθορισμένη και σταθερή

DTD:
 <!ATTLIST person number
 CDATA #REQUIRED>
 Έγκυρο XML:
 <person number="5677" />
 Άκυρο XML:
 <person />

DTD:
 <!ATTLIST contact fax CDATA #IMPLIED>
 Έγκυρο XML:
 1. <contact fax="555-667788" />
 2. <contact />

DTD:
 <!ATTLIST sender company CDATA #FIXED "Microsoft">
 Έγκυρο XML:
 <sender company="Microsoft" />
 Άκυρο XML:
 <sender company="W3Schools" />

- Άλλος ορισμός default values σε attribute
 - <!ATTLIST element-name attribute-name (en1|en2|..) default-value>
 - <!ATTLIST payment type (check|cash) "cash">

Χρήση attributes/elements

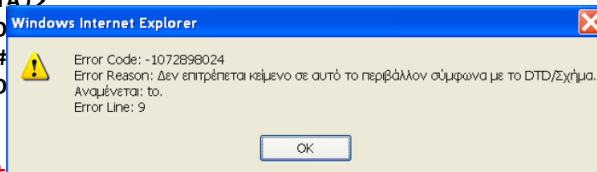


- Η XML δεν ορίζει ξεκάθαρους κανόνες σχετικά με το πότε πρέπει να χρησιμοποιούνται attributes και πότε elements
- Ωστόσο τα attributes
 - Δεν λαμβάνουν πολλαπλές τιμές
 - Δεν επεκτείνονται εύκολα
 - Δεν μπορούν να περιγράψουν δομές
- Σαν γενικό κανόνα:
 - Τα metadata αποθηκεύονται σαν attributes (π.χ element id)
 - Τα δεδομένα αποθηκεύονται ως elements

Validation errors - DTD



```
<?xml version="1.0" ?>
<!DOCTYPE note [
  <!ELEMENT note (to,from,heading,body)>
  <!ELEMENT to (#PCDATA)>
  <!ELEMENT from (#PCDATA)>
  <!ELEMENT heading (#PCDATA)>
  <!ELEMENT body (#PCDATA)>
]>
<note>
  <date>1/12/2008</date>
  <to>Tove/to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget our meeting this weekend!</body>
</note>
```



Παράδειγμα



trompette.dtd

```
<!ELEMENT product ((name, price, address))>
<!ELEMENT name (#PCDATA)>
<!ATTLIST name
    language CDATA #FIXED "French">
<!ELEMENT price (#PCDATA)>
<!ATTLIST price
    currency CDATA #FIXED "Euro">
<!ELEMENT address ((street, zip))>
<!ATTLIST address
    format CDATA #FIXED "XL1245"
    language CDATA #FIXED "French">
<!ELEMENT street (#PCDATA)>
<!ELEMENT zip (#PCDATA)>
```

Ο κώδικας



trompette

```
<product>
  <name language="French">trompette no 6</name>
  <price currency="Euro"> 420.12 </price>
  <address format="XL1245" language="French">
    <street>31 rue Croix-Bosset</street>
    <zip>92874</zip>
  </address>
</product>
```

XML Schema



- Περιγράφει τη δομή ενός εγγράφου XML
- Περισσότερες δυνατότητες από ότι το DTD
 - Γραμμένα σε γλώσσα XML-επεκτάσιμα (επαναχρησιμοποίηση ενός schema σε άλλα)
 - Πιο επεκτάσιμα
 - Υποστηρίζουν namespaces
 - Υποστηρίζουν τύπους δεδομένων
- Ιδιαίτερα η υποστήριξη τύπων δεδομένων
 - Καθιστά ευκολότερη την επικύρωση της ορθότητας των δεδομένων
 - Καθιστά ευκολότερη τη διατύπωση περιορισμών πάνω στα δεδομένα
 - Καθιστά ευκολότερη τη μετατροπή ανάμεσα σε τύπους δεδομένων
- Η γλώσσα του XML Schema ονομάζεται XML Schema Definition (XSD)

Παράδειγμα XML Schema



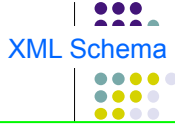
```
<?xml version="1.0"?>
  <note>
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
  </note>
```

Κώδικας XML

Παράδειγμα XML Schema

Στοιχείο - ρίζα

Τα στοιχεία που χρησιμοποιεί το schema προέρχονται από αυτό το namespace



Τα στοιχεία που δημιουργεί το schema είναι σε αυτό το namespace

```

<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.mySite.com">
  <xs:element name="note">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="to" type="xs:string"/>
        <xs:element name="from" type="xs:string"/>
        <xs:element name="heading" type="xs:string"/>
        <xs:element name="body" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>


```

complexType:
 Περιέχει και άλλα elements

simpleType:
 Δεν περιέχει άλλα elements

- Το xml Schema δεν δεσμεύει "λέξεις"
 - Περιέχει το namespace στο σώμα του

XSD Simple Types - Elements



- Περιέχουν μόνο κείμενο ή απλούς τύπους (boolean, string, date, κ.α)
- Δεν μπορούν να περιέχουν άλλα elements ή attributes
- Μπορεί να περιοριστεί το περιεχόμενο σε συγκεκριμένο τύπο ή να οριστεί έτσι ώστε να ακολουθεί συγκεκριμένο μοτίβο (pattern)

Υπάρχουν πολλοί built-in τύποι
 Παραδείγματα:

- xs:string
- xs:decimal
- xs:integer
- xs:boolean
- xs:date
- xs:time

```

<lastname>Smith</lastname>
<age>40</age>
<id>12532</id>

```

↓

```

<xs:element name="lastname" type="xs:string"/>
<xs:element name="age" type="xs:integer"/>
<xs:element name="id" type="xs:integer" fixed="12532">

```

Επίσης default και fixed τιμές

XSD - Attributes



- Κάθε attribute δηλώνεται σαν simple type
- Τα simple elements δεν μπορούν να έχουν attributes. Τα elements που έχουν attributes θεωρούνται complex types

```
<lastname lang="EN">Smith</lastname>
```

```
<xs:attribute name="lang" type="xs:string"/>
<xs:attribute name="lang" type="xs:string" default="EN"/>
<xs:attribute name="lang" type="xs:string" fixed="EN"/>
<xs:attribute name="lang" type="xs:string" use="required"/>
```

default

fixed

Optional/required attributes
(+use)

Παραδείγματα built-in τύπων:

- xs:string
- xs:decimal
- xs:integer
- xs:boolean
- xs:date
- xs:time

Μην ξεχνιόμαστε → XML Principles

```
<xs:attribute name="Lang" type="xs:string"/>
```

XSD Simple Types – Restrictions



- Οι περιορισμοί χρησιμοποιούνται για να οριστούν οι αποδεκτές τιμές των attributes/elements

Περιορισμός τιμής

```
<xs:element name="weekday">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="1"/>
      <xs:maxInclusive value="7"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Περιορισμός ομάδας τιμών

```
<xs:element name="course">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Internet Programming"/>
      <xs:enumeration value="Databases"/>
      <xs:enumeration value="Information Visualization"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Περιορισμός μοτίβου

```
<xs:element name="letter">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[a-z]"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Περιορισμός κενών χαρακτήρων

```
<xs:element name="address">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:whiteSpace value="replace"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Επίσης
και
preserve

XSD Simple Types – Restrictions



Περιορισμός	Περιγραφή
enumeration	Λίστα αποδεκτών τιμών
fractionDigits	Μέγιστος αριθμός επιτρεπτών δεκαδικών. Πρέπει να είναι μεγαλύτερο ή ίσο με 0.
length	Ακριβής αριθμός χαρακτήρων ή στοιχείων λίστας που επιτρέπονται. Πρέπει να είναι μεγαλύτερο ή ίσο με 0.
maxExclusive	Άνω όριο αριθμητικής τιμής
maxLength	Μέγιστος αριθμός χαρακτήρων ή στοιχείων λίστας που επιτρέπονται. Πρέπει να είναι μεγαλύτερο ή ίσο με 0.
minExclusive	Κάτω όριο αριθμητικής τιμής
minLength	Ελάχιστος αριθμός χαρακτήρων ή στοιχείων λίστας που επιτρέπονται. Πρέπει να είναι μεγαλύτερο ή ίσο με 0.
pattern	Καθορίζει την ακριβή σειρά χαρακτήρων που επιτρέπονται.
totalDigits	Καθορίζει τον ακριβή αριθμό ψηφίων που επιτρέπονται. Πρέπει να είναι μεγαλύτερο ή ίσο με το 0
whiteSpace	Καθορίζει το χειρισμό του κενού χώρου (tabs, κενά, αλλαγές γραμμών)

XSD-Complex Types



- Ένα complex element μπορεί να περιέχει άλλα elements ή attributes
- είδη complex elements
 - Elements που περιέχουν άλλα elements
 - Ο,τιδήποτε Element(s) με/χωρίς κείμενο που περιέχουν attribute(s) ή/και άλλα Elements

Παραδείγματα

```
<person id="Z8746"/>
```

```
<food type="dessert">Ice cream</food>
```

```
<person>
  <firstname>John</firstname>
  <lastname>Smith</lastname>
</person>
```

```
<description>
  It happened on
  <date lang="english">04.10.09</date>
</description>
```

XSD Complex Types – Τρόποι ορισμού



```
<person>
  <firstname>John</firstname>
  <lastname>Smith</lastname>
</person>
```

Απ' ευθείας

```
<xs:element name="person">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Παράδειγμα μόνο με elements

Τα elements πρέπει να εμφανιστούν Με τη συγκεκριμένη σειρά που ορίστηκαν

```
<xs:element name="person" type="personinfo"/>
<xs:complexType name="personinfo">
  <xs:sequence>
    <xs:element name="firstname" type="xs:string"/>
    <xs:element name="lastname" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

Με αναφορά (επιτρέπει τη χρήση του ίδιου complex type από πολλά elements - reuse)

Παράδειγμα

```
<xs:element name="student" type="personinfo"/>
```

XSD Complex Types – Τρόποι ορισμού



```
<person>
  <firstname>John</firstname>
  <lastname>Smith</lastname>
</person>
```

```
<xs:element name="person" type="fullpersoninfo"/>
<xs:complexType name="personinfo">
  <xs:sequence>
    <xs:element name="firstname" type="xs:string"/>
    <xs:element name="lastname" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="fullpersoninfo">
  <xs:complexContent>
    <xs:extension base="personinfo">
      <xs:sequence>
        <xs:element name="address" type="xs:string"/>
        <xs:element name="city" type="xs:string"/>
        <xs:element name="country" type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Επέκταση complex type και προσθήκη επιπλέον elements

```
<person>
  <firstname>John</firstname>
  <lastname>Smith</lastname>
  <address>Brandon Rd, 13A</address>
  <city>London</city>
  <country>UK</country>
</person>
```

Αν θέλαμε να προσθέσουμε tel, zip code ; _

XSD Complex Types-Empty Elements



- Ένα κενό element δεν περιέχει κείμενο, μόνο attributes

```
<product prodid="123" />
```

```
<xs:element name="product">
  <xs:complexType>
    <xs:attribute name="prodid" type="xs:positiveInteger"/>
  </xs:complexType>
</xs:element>
```

```
<xs:element name="product" type="prodtype"/>
<xs:complexType name="prodtype">
  <xs:attribute name="prodid" type="xs:positiveInteger"/>
</xs:complexType>
```

Η

Επιτρέπει τη χρήση του ίδιου complex Type (ποιο είναι εδώ;) από άλλα elements

XSD Complex Types – Mixed περιεχόμενο



```
<letter>
  Dear Mr.<name>John Smith</name>.
  Your order <orderid>1032</orderid>
  will be shipped on <shipdate>2001-07-13</shipdate>.
</letter>
```

Κείμενο, attributes και elements

```
<xs:element name="letter">
  <xs:complexType mixed="true">
    <xs:sequence>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="orderid" type="xs:positiveInteger"/>
      <xs:element name="shipdate" type="xs:date"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

XSD Complex Types – Indicators



- Επιτρέπουν τον έλεγχο του πως εμφανίζονται τα elements σε έναν complex type
 - Ενδείκτες σειράς:
 - All → Εμφάνιση των elements σε οποιαδήποτε σειρά. Κάθε element εμφανίζεται μόνο μία φορά.
 - Choice → Μπορεί να εμφανιστεί μόνο ένα από τα elements
 - Sequence → Εμφάνιση των elements με τη σειρά ορισμού
 - Ενδείκτες μέγιστου/ελάχιστου αριθμού εμφάνισης:
 - maxOccurs → Μέγιστες εμφανίσεις του element
 - minOccurs → Ελάχιστες εμφανίσεις του element
 - Ενδείκτες ομάδας:
 - Group name → Ορισμός ομάδας από element
 - attributeGroup name → Ορισμός ομάδας από attributes

XSD Complex Types – Indicators - Παραδείγματα



```
<xs:element name="person">
  <xs:complexType>
    <xs:all>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
    </xs:all>
  </xs:complexType>
</xs:element>
```

```
<xs:element name="person">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="full_name" type="xs:string"/>
      <xs:element name="child_name" type="xs:string" maxOccurs="10"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
<xs:group name="persongroup">
  <xs:sequence>
    <xs:element name="firstname" type="xs:string"/>
    <xs:element name="lastname" type="xs:string"/>
    <xs:element name="birthday" type="xs:date"/>
  </xs:sequence>
</xs:group>
```

XSD Complex Types – Επέκταση με μη καθορισμένα elements



- Τα στοιχεία <any> και anyAttribute επιτρέπουν την επέκταση του XML εγγράφου με elements και attributes που δεν ορίζονται από το σχήμα του

```
<xs:element name="person">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
      <xs:any minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

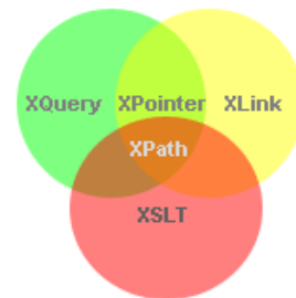
```
<xs:element name="person">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
    </xs:sequence>
    <xs:anyAttribute/>
  </xs:complexType>
</xs:element>
```

Περαιτέρω δυνατότητες και τεχνολογίες της XML



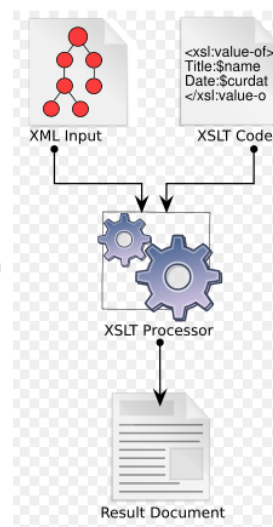
Περαιτέρω τεχνολογίες

- Με βάση τη γλώσσα XML δημιουργήθηκαν διάφορες τεχνολογίες οι οποίες μας επιτρέπουν τη μορφοποίηση και επεξεργασία των εγγράφων XML
 - XSLT
 - XPointer
 - Xlink
 - XPath
 - XQuery



XSLT

- EXtensible Stylesheet Language
- Γλώσσα μορφοποίησης εγγράφων XML
- Δυνατότητες
 - Προσθήκης/αφαίρεσης, αναδιάταξης/ταξινόμησης, απεικόνισης/απόκρυψης elements και attributes στο αρχείο εξόδου (output file)



<http://en.wikipedia.org/wiki/XSLT>

XLink



- XML Linking Language
- Χρησιμοποιείται για τη δημιουργία συνδέσμων στα έγγραφα XML
- Κάθε element ενός XML εγγράφου μπορεί να γίνει σύνδεσμος
- Υποστήριξη
 - Απλών συνδέσμων (παρόμοιοι με συνδέσμους HTML)
 - Εκτεταμένων συνδέσμων (σύνδεση πολλών πόρων ταυτόχρονα)
- Δυνατότητα ορισμού των συνδέσμων έξω από τα αρχεία που θα συνδεθούν

Xlink - Παράδειγμα



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ucg xmlns:xlink="http://www.w3.org/1999/xlink">
<department name="DIB">
  <description xlink:type="simple" xlink:href="http://dib.ucg.gr/" xlink:show="new">
    Το Τμήμα αυτό εδρεύει στην πόλη της Λαμίας...
  </description>
</department>
<department name="POA">
  <description
    xlink:type="simple" xlink:href="http://poa.ucg.gr/" xlink:show="new">
    Το Τμήμα αυτό εδρεύει στην πόλη της Λιβαδειάς...
  </description>
</department>
</ucg>
```

XPath



- Πλοήγηση στα elements και τα attributes της XML
- Χρησιμοποιεί εκφράσεις (path expressions) για αυτή την πλοήγηση
 - Ομοιότητα αυτών των εκφράσεων με τις εκφράσεις πλοήγησης στο filesystem ενός υπολογιστή
- Περιέχει βιβλιοθήκες από standard functions
- Συνέργια με XQuery

<http://en.wikipedia.org/wiki/XPath>

XPath - example



XPath:

```
/wikimedia/projects/project/editions/*[2]
```

XML document:

```
<?xml version="1.0" encoding="utf-8"?>
<wikimedia>
  <projects>
    <project name="Wikipedia" launch="2001-01-05">
      <editions>
        <edition language="English">en.wikipedia.org</edition>
        <edition language="German">de.wikipedia.org</edition>
        <edition language="French">fr.wikipedia.org</edition>
        <edition language="Polish">pl.wikipedia.org</edition>
      </editions>
    </project>
    <project name="Wiktionary" launch="2002-12-12">
      <editions>
        <edition language="English">en.wiktionary.org</edition>
        <edition language="French">fr.wiktionary.org</edition>
        <edition language="Vietnamese">vi.wiktionary.org</edition>
        <edition language="Trukish">tr.wiktionary.org</edition>
      </editions>
    </project>
  </projects>
</wikimedia>
```

http://en.wikipedia.org/wiki/File:XPath_example.svg

XQuery



- Η XQuery συνιστά για την XML ότι η SQL για τις βάσεις δεδομένων
- Γλώσσα αναζήτησης σε XML δεδομένα
- Μπορεί να χρησιμοποιηθεί:
 - Για την εξαγωγή πληροφορίας για χρήση σε ένα web service
 - Για την αναζήτηση σε έγγραφα Ιστού
 - Για τη δημιουργία συνοπτικών αναφορών (summary reports) σχετικά με δεδομένα τα οποία βρίσκονται αποθηκευμένα σε μία βάση XML

XQuery (παράδειγμα)



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bookstore>
  <book category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="CHILDREN">
    <title lang="en">Harry Potter</title>
    <author>J. K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title lang="en">XQuery Kick Start</title>
    <author>James McGovern</author>
    <author>Per Bothner</author>
    <author>Kurt Cagle</author>
    <author>James Linn</author>
    <author>Vaidyanathan Nagarajan</author>
    <year>2003</year>
    <price>49.99</price>
  </book>
</bookstore>
```

```
doc("books.xml")/bookstore/book/title
```



```
<title lang="en">Everyday Italian</title>
<title lang="en">Harry Potter</title>
<title lang="en">XQuery Kick Start</title>
```

```
doc("books.xml")/bookstore/book[price<30]
```



```
<book category="CHILDREN">
  <title lang="en">Harry Potter</title>
  <author>J. K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>
```

XPointer



- XML Pointer Language
- Με την XLink μπορούμε να δημιουργήσουμε συνδέσμους στο έγγραφο XML
- Το XPointer επιτρέπει στους συνδέσμους να «δείχνουν» σε συγκεκριμένα τμήματα ενός εγγράφου XML
- Χρησιμοποιεί εκφράσεις XPath για την πλοήγηση μέσα σε ένα έγγραφο

Xpointer - Παράδειγμα



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

<universities xmlns:xlink="http://www.w3.org/1999/xlink">

<university xlink:type="simple" xlink:href="http://ucg.gr/departments.xml#DIB">
  <students>400</students>
</university>
...
...
</universities>
```