ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

Εισαγωγή στην Επιστήμη και Τεχνολογία των Υπηρεσιών

Ενότητα 1: Εισαγωγή

Χρήστος Νικολάου
Τμήμα Επιστήμης Υπολογιστών

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται στην άδεια χρήσης **Creative Commons** και ειδικότερα

Αναφορά – Μη εμπορική Χρήση – Όχι Παράγωγο Έργο v. 3.0

(Attribution – Non Commercial – Non-derivatives)

- Εξαιρείται από την ως άνω άδεια υλικό που περιλαμβάνεται στις διαφάνειες του μαθήματος, και υπόκειται σε άλλου τύπου άδεια χρήσης. Η άδεια χρήσης στην οποία υπόκειται το υλικό αυτό αναφέρεται ρητώς.

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Κρήτης**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

XML
Introduction
605.444 / 635.444

David Silberberg

Lesson 1

What is XML?

- XML stands for *eXtensible Markup Language*
- XML is a language to define other languages
  - This is called a *meta-language*
- XML's predecessor is SGML (Standard Generalized Markup Language)
  - XML is simpler
- XML uses a tag set to define some data construct
  - Tag names have meaning within some context
  - Tags are arranged in hierarchies
  - Thus, the tag placement within the hierarchy has meaning as well

Brief History of Markup Languages

- Originally, people created text or binary files to represent data
  - People needed to develop standards to ensure good interchange
  - However, standards were created for each set of applications
  - People recognized that there is a need for a standard markup language
- SGML was born
  - Complicated
  - Handles huge amounts of complex data
  - HTML was born from it
  - XML was born from it, too

HTML Tag Examples

- In HTML, tags are defined with specific meanings
  - <TABLE>, <H1>, <P>, etc. all have predefined meanings
  - Web browser software interprets tags and performs some corresponding action(s)
    - When <TABLE> is encountered, Web browsers set up tables
    - When <H1> is encountered, Web browsers set up first-level headers
    - When <P> is encountered, Web browsers break for a new paragraph
  - If a tag is undefined, the corresponding software may generate errors (or they might ignore it)
    - <BIRD>, <DOG>, <HARRY>, etc. have no meaning to Web browsers
    - Web browsers that encounter these terms might generate errors or they might ignore the tags
- HTML grammar defines the proper usage of the tags

HTML Tag Attributes

- Attributes augment some aspect of the tag
  - Grammar defines the attributes that correspond to tags
  - For example, <FONT> has attributes SIZE and COLOR
  - Usage: <FONT SIZE="3" COLOR="ORANGE">
  - BACKGROUND is not a valid attribute for <FONT>
- Tags may have corresponding end tags
  - <TABLE> starts a table while </TABLE>ends a table
  - <P> does not require its corresponding end tag
- Tags may be embedded within other tags

```
<HEAD>
   <TITLE>My Web Page</TITLE>
</HEAD>
```

XML Grammar

- All of the aspects defined above contribute to the HTML grammar
  - Tag names
  - Attributes
  - End tags
  - Tag nesting hierarchy
- XML does not define specific tags or a specific grammar
  - Completely extensible
  - You can choose:
    - Tag names
    - Attributes
    - Nesting hierarchy

XML Example

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

<department>
    <name>Computer Science</name>
    <professor lastname="Smith" firstname="Fred">
        <rank>Full</rank>
        <specialty>XML</specialty>
        <office>23-204</office>
    </professor>
    <professor lastname="Chu" firstname="Harry">
        <rank>Associate</rank>
        <specialty>Artificial Intelligence</specialty>
        <office>23-225</office>
    </professor>
</department>
```

Another XML Example

```
<?xml version="1.0"?>
<store>
    <name>Giant</name>
    <location>Columbia</location>
    <section name="Produce">
        <vegetable>
            <name>Carrot</name>
            <quantity>275</quantity>
        </vegetable>
        <fruit>
            <name>Kiwi</name>
            <quantity>500</quantity>
        </fruit>
    </section>
    …
</store>
```

The same tag may occur in several contexts

Properties of XML Tags

- You can use pretty much any tag name you want
- You can associate any attribute you want with the tags
- You can define any nesting you want
- XML is flexible
- However, its flexibility also makes it ambiguous
  - Does quantity mean actual number of carrots or packages of carrots?
  - Does rank mean actual rank or rank aspiring to?
  - Should name be an attribute or an embedded tag?
  - Could *teacher* have been used instead of *professor*?
  - Is the grocery list an inventory list or a purchasing list?

XML Definition

- So what is XML?
  - Usually, XML refers not only to the language, but the entire tool suite associated with it.
  - XML defines the core language
  - Metadata framework
- Why use XML? Why make up a new language?
  - Couldn't we just impose a syntax on our files?
    - For example, addresses always are specified in the following order: name, street, city, state, zip
    - How could we differentiate between a 1- and 2-line street address?
    - How do we know which part is the salutation (Mr., Ms., Dr., etc.)?
  - XML provides a way to tag and describe all the data
  - A specific XML syntax permits:
    - General tools (e.g., parsers) to be created and used
    - A general format to be applied to many domains

How to Use XML

- XML is a data structure representation language
- XML alone would not fare too well
- However, there are many tools associated with it that can help
  - Java has a robust set of tools and APIs
  - Most of the tools exist for C, C++, and Perl
- As with any data structure, you must first parse it, then manipulate the data
- You might want to write out a new data structure as a result

Technologies to be Covered

- Transformation technologies
  - XSLT – *Extensible Stylesheet Language*
  - CSS – *Cascading Stylesheets*
  - SAX – *Simple API for XML*
  - DOM – *Document Object Model*
  - JDOM – *Java DOM*
  - JAXP – *Java API for XML Parsing*
  - TRAX – *Transformation API for XML*
- Search technologies
  - XPath – *XML Path*
  - XQuery – *XML Query*

Technologies to be Covered (2)

- Knowledge Representation
  - DTD – *Document Type Definition*
  - XML Schema
  - Namespaces
  - RDF – *Resource Description Framework*
  - RDFS – *RDF Schema*
  - OWL – *Web Ontology Language*
- Services
  - Web Services
  - JAXB – *Java Architecture for XML Binding*
  - RESTful Services
- Other
  - XLink – *XML Link*
  - XPointer – *XPointer*

Processing Instructions (PI) in XML

- A *processing instruction* is a command to the application and is embedded in the XML file
  - Mostly, XML represents data structures
  - Occasionally, processing instructions must sent to applications or parsers that read XML files
- Syntax: <?target instructions?>
  - Example <?xml version="1.0"?>
  - Instruction is: *version="1.0"*
  - Target is: *xml*
  - All XML parsers have a standard set of instructions that they should recognize

PIs (continued)

- Parsers may route PIs with external targets to other applications
- Example
  - <?purchasing buy_date="Monday"?>
  - Our grocery list may be a purchasing list
  - The PI indicates that the list must be routed to the Purchasing program

Document Type Definition (DTD)

- Establishes constraints for a set of XML documents
- Defines the grammar of XML documents
- Can be defined in
  - The XML document itself
  - Another (only one) external document
  - Both
- Defines
  - Valid tags: <professor>, <vegetable>, etc.
  - Valid attributes:<professor lastname="" firstname="">, etc.
  - Valid values: section name can be one of *produce, dairy, meat, etc.*
  - Valid nesting of tags: <vegetable> is embedded within <section>

XML Schema

- XML Schema replaces and improves the function of DTDs
  - Addresses limitations of DTD
    - More powerful definitions of XML files
    - More expressive
    - Represents what people want to express in XML
  - Uses XML-style approach to defining XML documents
  - Includes knowledge of hierarchy
  - Handles namespace conflicts
  - Can specify relationships among XML documents

- http://www.w3.org/XML/Schema

Namespaces

- A *namespace* is a mapping between an element prefix and a URI (Uniform Resource Indicator)
- Handles namespace collisions.
- Allows parsers to handle collisions
- Suppose we wanted to make <name> unambiguous
  - We could use tags like <store_name>, <fruit_name>, or <vegetable_name>
  - However, this is unnatural and can become tedious
  - Namespaces allow a more natural solution: <store:name>, <fruit:name>, and <vegetable:name>
- http://www.w3.org/TR/REC-xml-names

Namespaces

- A *namespace* is a mapping between an element prefix and a URI (Uniform Resource Indicator)
- Handles namespace collisions.
- Allows parsers to handle collisions
- Suppose we wanted to make <name> unambiguous
  - We could use tags like <store_name>, <fruit_name>, or <vegetable_name>
  - However, this is unnatural and can become tedious
  - Namespaces allow a more natural solution: <store:name>, <fruit:name>, and <vegetable:name>
- http://www.w3.org/TR/REC-xml-names

XSL and XSLT Example

- XML document

```
<?xml version="1.0"?>
<?xml-stylesheet href="hello.xsl" type="text/xsl"?>
<store>
    <name>Giant</name>
    <location>Columbia</location>
</store>
```

XML and XSLT Example (continued)

- XSLT stylesheet

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/store/">
<html>
 <head>
  <title>
   <xsl:value-of select="name"/>
  </title>
 </head>
 <body>
  <xsl:apply-templates/>
 </body>
</html>
</xsl:template>
```

XML and XSLT Example (continued)

```
<xsl:template match="location">
  <p>
      <xsl:apply-templates/>
  </p>
</xsl:template>
```

Output

```
<html>
    <head>
        <title>
                Giant
        </title>
    </head>
    <body>
        Columbia
    </body>
</html>
```

XPath

- XML Path (XPath) expressions specify how to locate items in XML documents
- Considers documents to be trees with nodes (elements, attributes, and text)
- XPath expressions identify nodes or subtrees that match the expressions
- Used heavily by XSLT, DOM and XQuery

- http://www.w3.org/TR/xpath

XQuery

- Many XML query languages have surfaced
  - XQL
  - UnQL - UPenn
  - Lorel -Stanford
  - A host of others
- XQuery is the new standard XML query language
  - Based on XML Query Algebra
- Allows document elements to be queried with respect to their inter-relationships

- http://www.w3.org/XML/Query

Sample XML Document

```
<?xml version="1.0"?>
<inventory>
    <book>
        <title>JAVA and XML</title>
        <author>Brett McLaughlin</author>
        <publisher>O'Reilly</publisher>
        <year>2000</year>
    </book>
    <book>
        <title>Beginning XML</title>
        <author>David Hunter</author>
        <publisher>Wrox</publisher>
        <year>2000</year>
    </book>
</inventory>
```

Sample XQuery Statement

List the titles of books published by O'Reilly in 2000.

```
FOR     $b IN document("bib.xml")//book
WHERE   $b/publisher = "O'Reilly" AND
        $b/year = "2000"
RETURN  $b/title
```

Result:

```
<title>JAVA and XML</title>
```

Parsers

- SAX (Simple API for XML)
  - SAX is not a parser, per se, but a framework for parsers
  - Provide access to data in an XML document
  - It is event-driven
    - APIs are defined for events like startDocument() and endDocument()
    - Parser authors supply their own code for what should happen at these events
- DOM (Document Object Model)
  - Provides a means to manipulate (not only access) XML document data
  - DOM reads the entire document first (taxes resources)
  - XML document is represented as a tree structure

Parsers (continued)

- Comparison of SAX and DOM
  - DOM is more powerful because it enables manipulation
  - However, it is slow because the whole document must be read at once
  - SAX is faster because you operate on data as you go
  - But, you can't manipulate the entire XML document at once
- JAXP (Sun's Java API for XML Parsing)
  - Provides cohesive and standard layer for parser functions
  - SAX and DOM sit underneath JAXP

Other XML Technologies

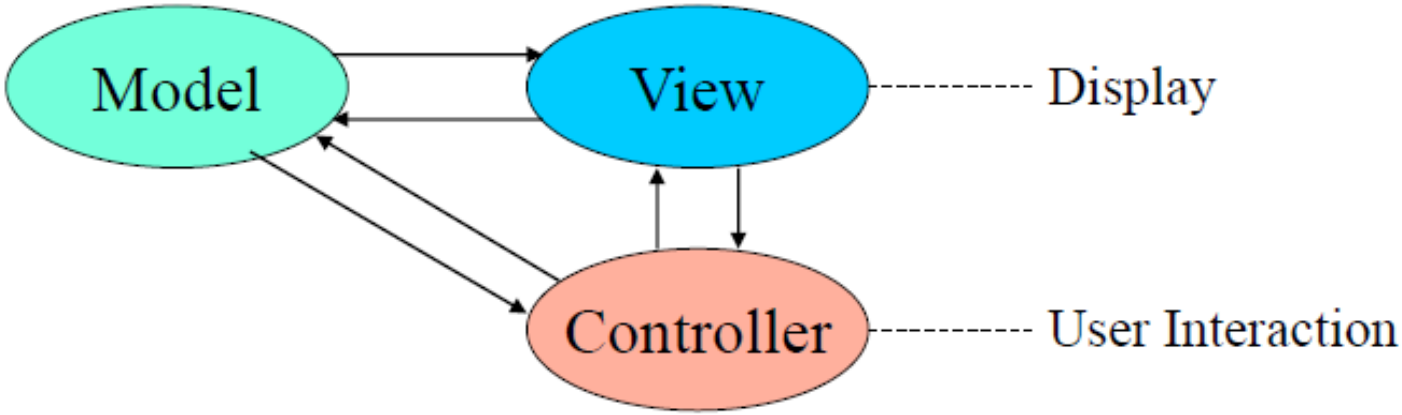
- Resource Description Framework (RDF)
  - Rudimentary definition of resources and objects
  - Basis of XML ontologies specifications (e.g., Web Ontology Language (OWL), DARPA Agent Markup Language (DAML))
- XLink & XPointer
  - HTML on steroids!
  - Super links
  - Can create your own web site environment based on other existing web sites.
- Web Services
  - Distributed systems development using XML
  - Ubiquitous data representation (XML) and ubiquitous transport protocol (HTTP)
  - RESTful web services
- Tons more
  - Check out http://www.w3.org/ !

How is XML Used Today?

- XML is not only a hot topic, but is used in many applications today
- It has caught on very quickly!
- Java itself has caught on very rapidly
- Together, they form a powerful technology
  - Java is portable code
  - XML is portable data
  - Let's explore some of XML's uses

XML for Presentation

- There are two aspects of data
  - Content - the model of the data
  - Presentation - the view of the data
- Has its roots in Smalltalk's Model-View-Controller framework

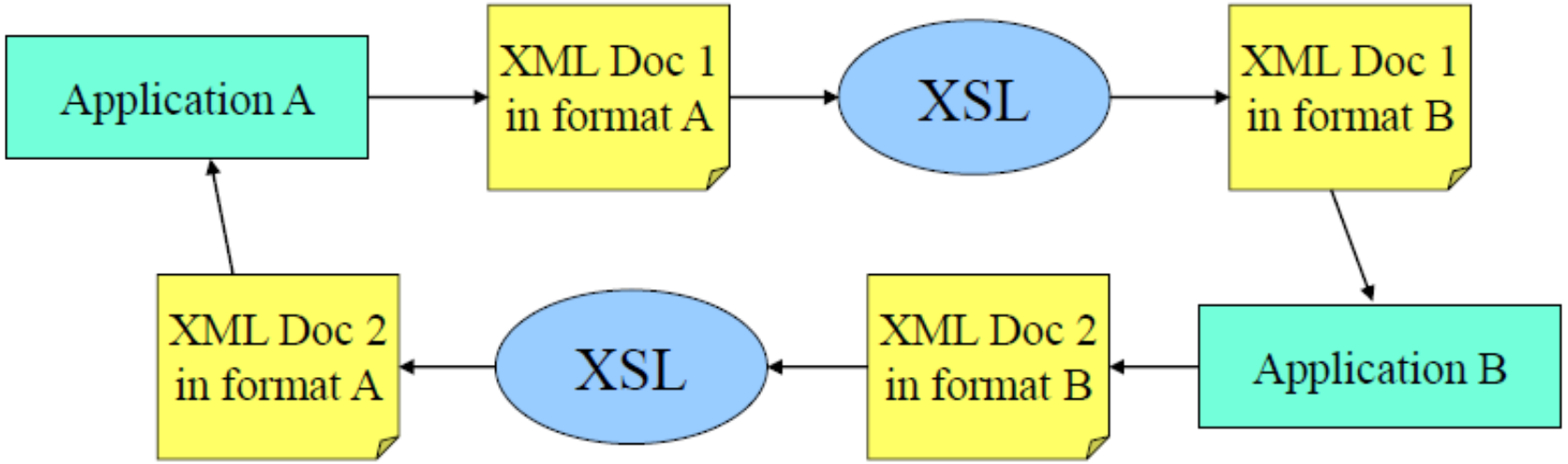


XML for Presentation (cont.)

- Data is represented independent of display format
- Allows multiple applications to use the same data, but display them in different formats
- Different presentation formats are possible
  - Swing
  - HTML
  - WML (wireless)
  - In fact, the same data might need to be displayed using many different display styles
- Usually, XSL or XSLT converts format to display

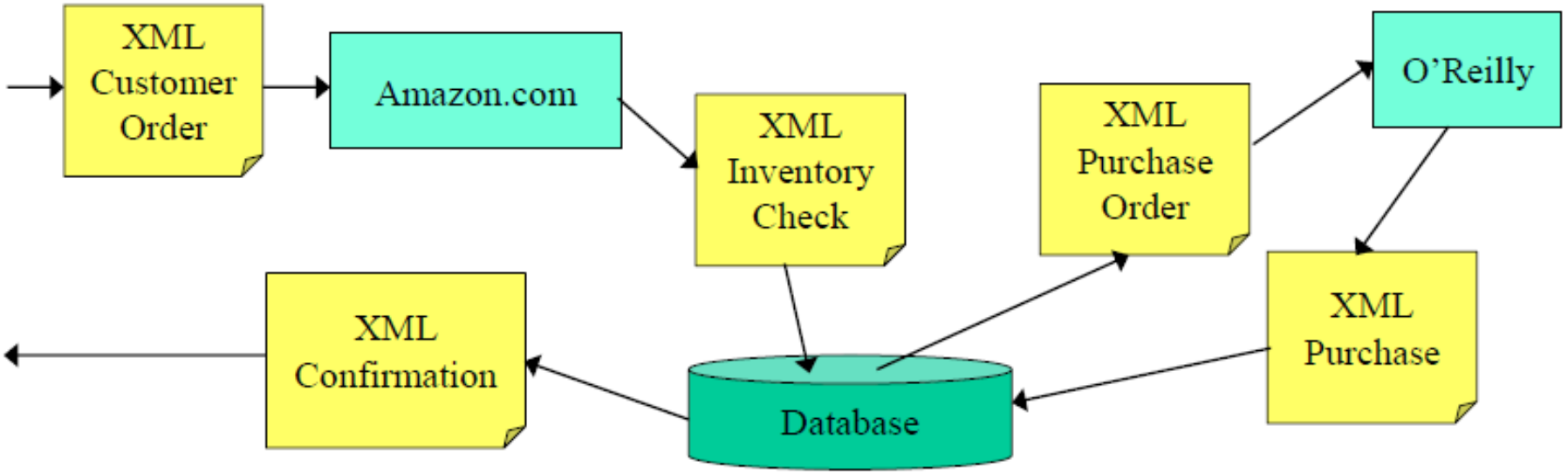
XML for Communication

- XML is not tied to any application or client
- Same data can be passed around among applications
- Data representation is easily passed around the network
- Programs transform data and pass them to other programs



Business-to-Business

- XML provides standard data exchange among business
- Passes data among multiple applications of multiple companies/departments
- Possibly, each application modifies and augments the XML document



Commercial Product Support

- Parsers
  - SAX, DOM, JDOM, JAXP, XSLT, TRAX, etc.
- Publishing frameworks
  - Apache Cocoon, SilkPage, etc.
- Editors and IDE
  - **oXygen**, XML Spy, Amaya, Komodo, etc.

Τέλος Ενότητας

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης