Egyptian E-Learning University (EELU)
Faculty of Computers and Information Technology


## The Web Engineering 3

# Introduction to Semantic Web

## Lecture 6

## Structured web documents in XML

**Presented by**
**Prof. Khaled Wassif**

# Course Topics

- Introduction to the semantic web.
- Semantic web technologies and layered approach.
- Structured web documents in XML.
- Describing web resources in basic elements of Resource Description Framework (RDF).
- Web Ontology Language: OWL.
- Ontologies Applications.

# Course References

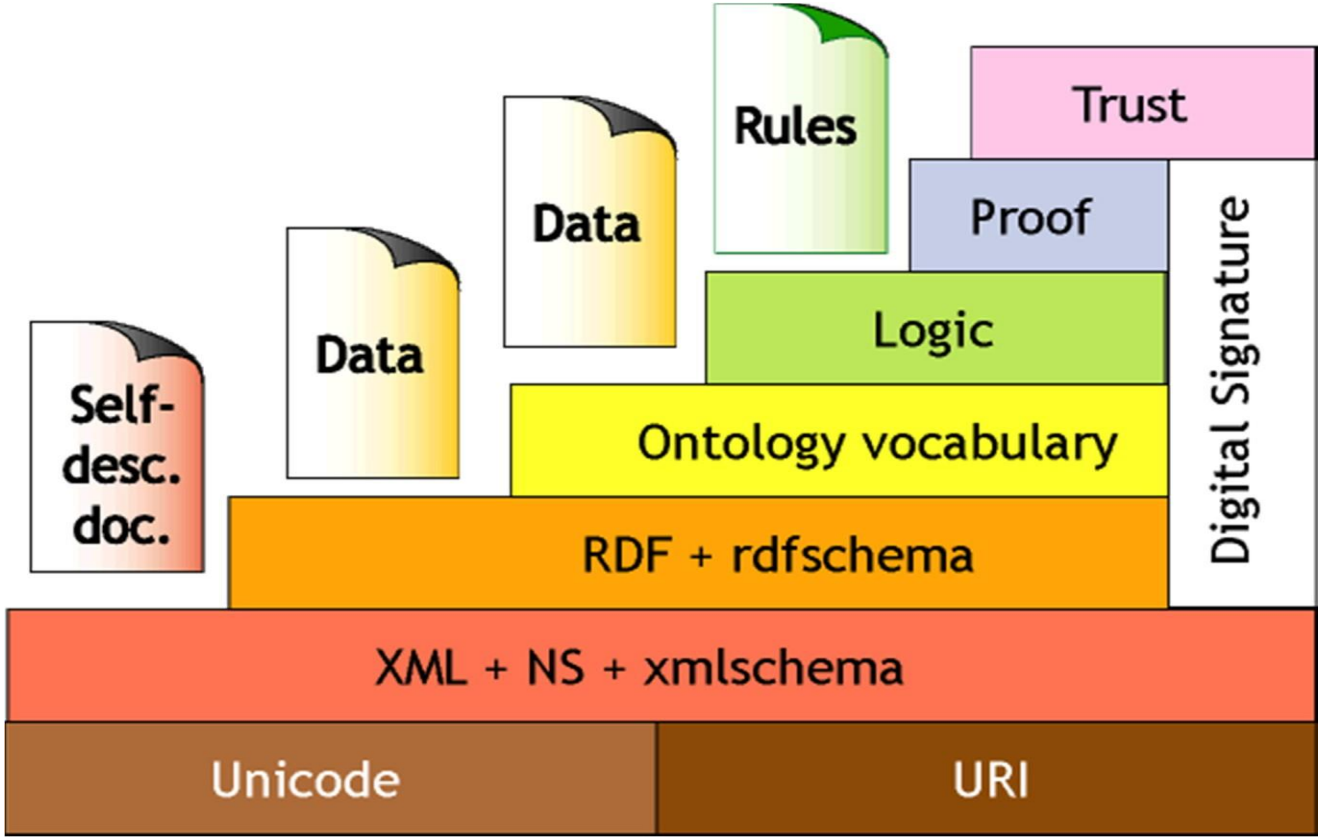

1. Grigoris Antoniou, Paul Groth, Frank van Harmelen, Rinke Hoekstra, "A Semantic Web Primer", 2012.
2. John Domingue, Dieter Fensel, James A. Hendler, "Introduction to the Semantic Web Technologies", 2011.

# Lecture Outline

- **Introduction**
- Detailed Description of XML
- Structuring
  a) DTDs
  b) XML Schema
- Namespaces
- Accessing, querying XML documents: Xpath
- Transformations: XSLT

# The Semantic Web Structure

- The development of the Semantic Web proceeds in steps.
- Each step building a layer on top of another.



# An HTML Example

Ex: for a book with title “Real-time Reasoning: Context - Dependent Reasoning”. The authors of this book are V. Marek and M. Truszczynski. This book is published in 1993 through springer under ISBN 0387976892.

## The HTML Code

```
<h2>Real-time Reasoning: Context - Dependent Reasoning</h2>
<i>by <b>V. Marek</b> and <b>M. Truszczynski</b></i><br>  Springer 1993<br> ISBN 0387976892
```

---

<h> header </h>

  This is a bold and italicized text  

**An HTML tag**: is a piece of HTML language used to indicate the beginning and end of an HTML element in an HTML document and only used to maintain interface displaying.

# An HTML Example

<h2>Real-time Reasoning: Context - Dependent Reasoning</h2>

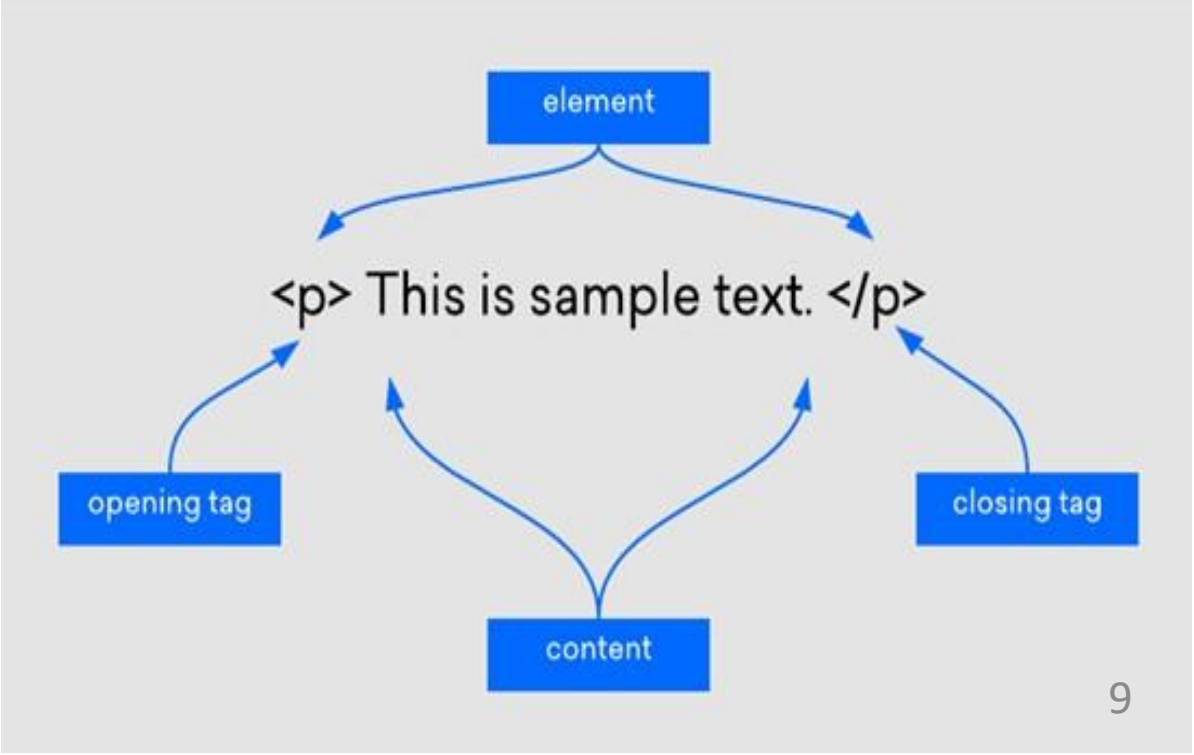
by V. Marek and M. Truszczynski<br> Springer 1993<br> ISBN 0387976892

# The Same Example in XML

```
<book>
    <title>Real-time Reasoning: Context - Dependent Reasoning</title>
    <author>V. Marek</author>
    <author>M. Truszczynski</author>
    <publisher>Springer</publisher>
    <year>1993</year>
    <ISBN>0387976892</ISBN>
</book>
```

# The Same Example in XML

```
<book>
    <title>Real-time Reasoning: Context - Dependent Reasoning</title>
    <author>V. Marek</author>
    <author>M. Truszczynski</author>
    <publisher>Springer</publisher>
    <year>1993</year>
    <ISBN>0387976892</ISBN>
</book>
```



# HTML and XML: Similarities

- Both use tags (e.g. <h2> and </year>)
- Tags may be nested (tags within tags)
- Human users can read and interpret both HTML and XML representations quite easily
- … But how about machines?

# Problems with Automated Interpretation of HTML Documents

An intelligent agent (in search engines) trying to retrieve the names of the authors of the book:

<h2>Real-time Reasoning: Context - Dependent Reasoning</h2>

by V. Marek and M. Truszczynski<br> Springer 1993<br>
ISBN 0387976892

- Authors' names could appear immediately after the title
- or immediately after the word "by"
- Are there two authors?
- Or just one, called "V. Marek and M. Truszczynski"?

# 1) HTML vs XML: Structural Information

- HTML documents do not contain structural information: just pieces of the document and their displaying relationships.
- XML more easily accessible to machines because:
  - Every piece of information is described.
  - Relations are also defined due to the nesting structure.
  - E.g., the **<author>** tags appear within the **<book>** tags, so they describe one property called author of a particular book.

# 1) HTML vs XML: Structural Information

- A machine processing the XML document would be able to conclude that:
  - the **author** element refers to the attached **book** element instead of using proximity considerations as in HTML.

- XML allows the definition of constraints on values:
  - E.g. a year must be a number of four digits

# 3) HTML vs XML: Formatting (font, design, and ….

- XML formatting is based on HTML formatting
- The HTML representation provides more details than the XML representation (displaying process):
  - The formatting of the document is also described (such as fonts – text coloring – text restrictions)
- The main use of an HTML document is to display information:
  - It must define formatting.
- XML: separation of content from display:
  - Same information can be displayed in different ways.

# HTML vs XML: Another Example

- In HTML

```
<h2>Relationship force-mass</h2>
<i> F = M × a </i>
```

- In XML

```
<equation>
    <meaning>Relationship force-mass</meaning>
    <leftside> F </leftside>
    <rightside> M × a </rightside>
</equation>
```

# 4) HTML vs XML: Different Use of Tags

- In both HTML and XML using same tags.

But

- In XML completely different.

- HTML tags define display: color, lists, font, …

But

- XML tags not fixed: user (developer) definable tags.

XML meta markup language: language for defining other markup languages such as HTML

# XML Vocabularies

- Web applications must agree on common vocabularies to communicate and collaborate.
- Communities and business sectors are defining their specialized vocabularies for example:
  - Mathematics (MathML): **MathML** is a low-level format for describing mathematics as a basis for machine-to-machine communication. MathML is intended to facilitate the use and re-use of mathematical and scientific content on the Web. MathML developed by W3C.
  - Bioinformatics (BSML)
  - Human resources (HRML)
  - …

# Lecture Outline

- Introduction
- **Detailed Description of XML**
- Structuring
    a) DTDs
    b) XML Schema
- Namespaces
- Accessing, querying XML documents: Xpath
- Transformations: XSLT

# The XML Language

An XML document consists of:

- a prolog
- a number of elements
- an optional epilog (not discussed)

# Prolog of an XML Document

The prolog consists of:

- an XML declaration.
- an optional reference to external structuring documents

Example:

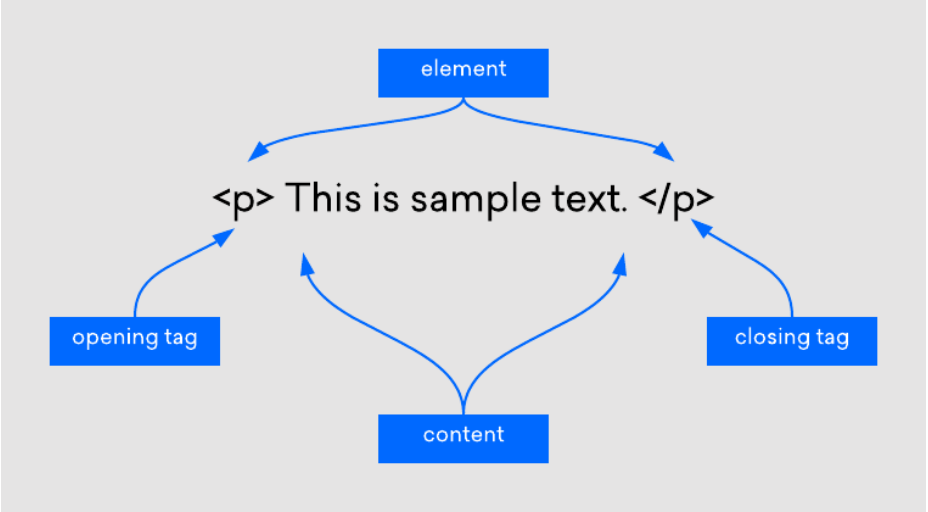
```
<?xml version="1.0" encoding="UTF-16"?>

<!DOCTYPE book SYSTEM "book.dtd">
```

# XML Elements

- any “things” the XML document talks about
  - E.g. books, authors, publishers

- An element consists of:
  - an opening tag
  - the content
  - a closing tag




Example:

<lecturer>Yasser Ibrahim</lecturer>

# XML Elements Writing Constraints

- Tag names can be chosen almost freely.
- The first character must be a letter, an underscore, or a colon
- No name may begin with the string “xml” in any combination of cases (capital or small letters)
  - E.g. “Xml”, “xML”

# Content of XML Elements

- Content may be text, or other elements, or nothing

```
<lecturer>
    <name>Yasser Ibrahim</name>
    <phone> +2−010−3875 507 </phone>
</lecturer>
```

- If there is no content, then the element is called empty; it is declared as follows:

<lecturer/> for <lecturer></lecturer>

# XML Attributes

- Attributes are part of XML elements.
- An element can have multiple unique attributes.
- Attribute gives more information about XML elements. To be more accurate, attributes define the properties of elements.
- An XML attribute is always a name-value pair.
- An attribute is a name-value pair inside the opening tag of an element:

```
<order>
 <orderNo="23456" customer="John Smith"  date="October 15, 2002">
          <item itemNo="a528" quantity="1"/>
          <item itemNo="c817" quantity="3"/>
</order>
```

# XML Attributes: An Example

```
<order>
 <orderNo="23456" customer="John Smith"  date="October 15, 2002">
          <item itemNo="a528" quantity="1"/>
          <item itemNo="c817" quantity="3"/>
</order>
```

# The Same Example without Attributes

```
<order>
    <orderNo>23456</orderNo>
    <customer>John Smith</customer>
    <date>October 15, 2002</date>
    <item>
          <itemNo>a528</itemNo>
          <quantity>1</quantity>
    </item>
    <item>
          <itemNo>c817</itemNo>
          <quantity>3</quantity>
          </item>
</order>
```

# XML Elements vs Attributes

- Attributes can be replaced by elements.
- Use of elements are the same as use of attributes.
- But note that attributes **cannot** be nested.

# Another Components of XML Docs

- Comments:
  - A piece of text that is to be ignored by parser
  - **<!-- This is a comment -->**

- Processing Instructions (PIs):
  - Define procedural attachments

<?stylesheet type="text/css" href="mystyle.css"?>

**CSS** is the language we use to style an HTML document. **CSS** describes how HTML elements should be displayed.

# Well-Formed XML Documents

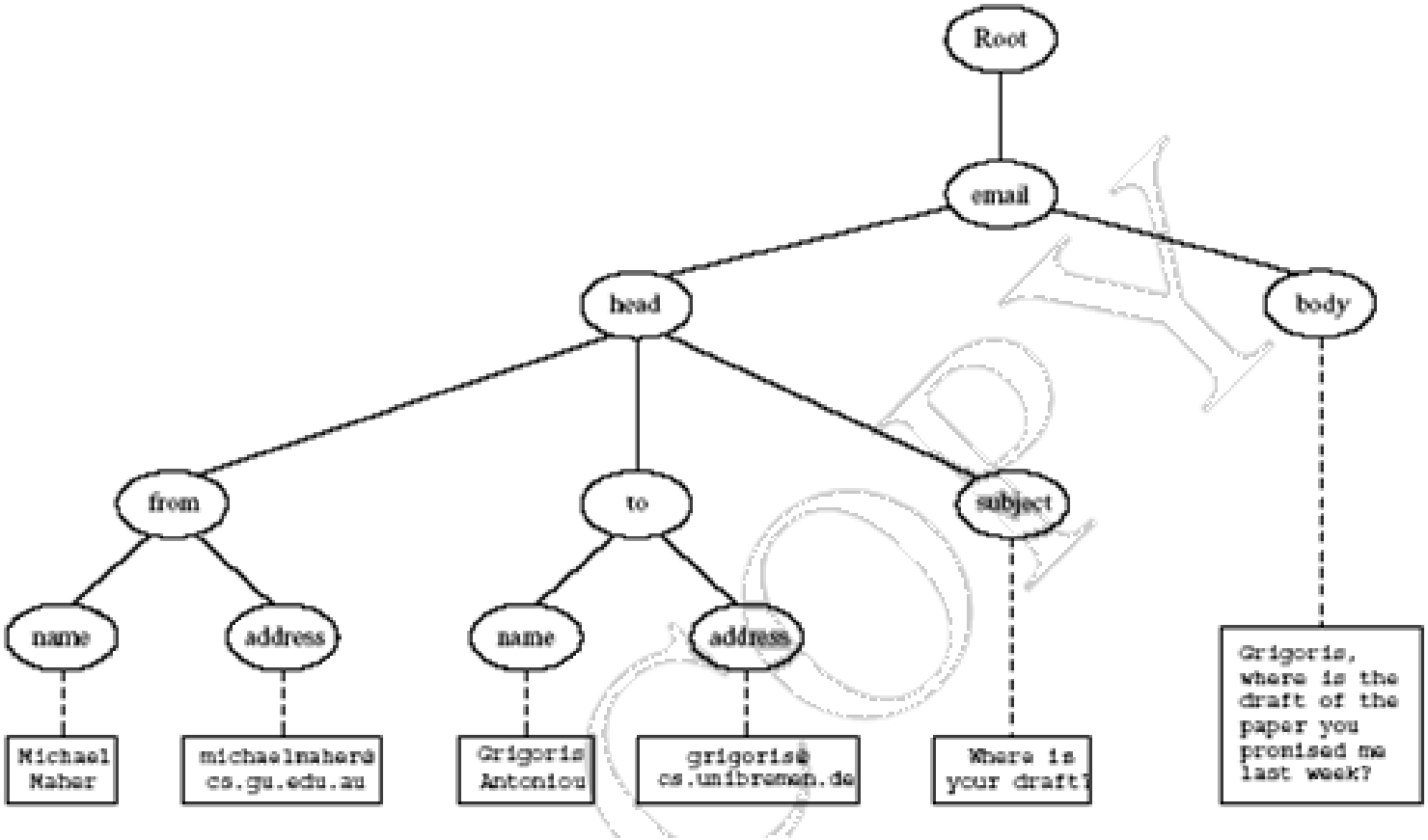
- Focuses on a correct syntax:
  - Some syntactic rules:
    - Only one outermost element (called root element)
    - Each element contains an opening and a corresponding closing tag
    - Tags must not overlap
      - **<author><name>Lee Hong</author></name>** **Error notation**

      **<author>**

      **<name>Lee Hong</name>**

      **</author<**
    - Attributes within an element have unique names
    - Element and tag names must be allowed

# The Tree Model of XML Docs



# The Tree Model of XML Docs

**An example:**

```
<email>
    <head>
        <from name="Michael Maher"
              address="michaelmaher@cs.gu.edu.au"/>
        <to name="Grigoris Antoniou"
              address="grigoris@cs.unibremen.de"/>
        <subject>Where is your draft?</subject>
    </head>
    <body>
        Grigoris, where is the draft of the paper you promised me
        last week?
    </body>
</email>
```

# The Tree Model of XML Docs

- The tree representation of an XML document is an ordered labeled tree: (developed for facilitate the organizing of XML Docs)
  - There is exactly one root
  - Each non-root node has exactly one parent
  - Each node has a label.
  - There are no cycles or overlapping
  - The order of elements is important
  - … but the order of attributes is not important

# Thank you

**Prof. Khaled Wassif**