

UNIVERSITY OF  
CALGARY

**A tutorial report for SENG 609.22**

**Agent Based Software Engineering**

**Course Instructor: Dr. Behrouz H. Far**

# **XML Tutorial**

Yanan Zhang

Department of Electrical and Computer Engineering

University of Calgary

zhangy@enel.ucalgary.ca

# 1. What is XML

- XML stands for **Extensible Markup Language**, which is a markup language for documents containing structured information
- Both XML itself and its related technologies are **developed and approved by the World Wide Web Consortium (W3C)**, which was created in October 1994 to lead the World Wide Web to its full potential by developing common protocols that promote its evolution and ensure its interoperability. W3C has more than 500 Member organizations from around the world and has earned international recognition for its contributions to the growth of the Web.
- XML **was designed to describe data**. It is a method for putting structured data in a text file.
- XML looks a bit **like HTML** but **isn't HTML**

Like HTML, XML makes use of *tags* (words bracketed by '<' and '>') and *attributes* (of the form name="value"), but while HTML specifies what each tag & attribute means (and often how the text between them will look in a browser), XML uses the tags only to delimit pieces of data, and leaves the interpretation of the data completely to the application that reads it.

- XML **is text**, but **isn't meant to be read**.

XML files are text files but even less than HTML are they meant to be read by humans. They are text files, because that allows experts (such as programmers) to more easily *debug* applications, and in emergencies, they can use a simple text editor to fix a broken XML file. But the rules for XML files are much stricter than for HTML. A forgotten tag, or an attribute without quotes makes the file unusable, while in HTML such practice is often explicitly allowed, or at least tolerated. It is written in the official XML specification: applications are not *allowed* to try to second-guess the creator of a broken XML file; if the file is broken, an application has to stop right there and issue an error.

- XML is verbose, but that is not a problem

Since XML is a text format, and it uses tags to delimit the data, XML files are nearly always larger than comparable binary formats. That was a conscious decision by the XML developers. The advantages of a text format are evident, and the disadvantages can usually be compensated at a different level. Disk space isn't as expensive anymore as it used to be, and programs like zip and gzip can compress files very well and very fast. Those programs are available for nearly all platforms (and are usually free). In addition, communication protocols such as modem protocols and HTTP/1.1 (the core protocol of the Web) can compress data on the fly, thus saving bandwidth as effectively as a binary format.

- XML is new, but not that new.

Development of XML started in 1996 and it is a W3C standard since February 1998, which may make you suspect that this is rather immature technology. But in fact the technology isn't very new. Before XML there was SGML, developed in the early '80s, an ISO standard since 1986, and widely used for large documentation projects. And of course HTML, whose development started in 1990. The designers of XML simply took the best parts of SGML, guided by the experience with HTML, and produced something that is no less powerful than SGML, but vastly more regular and simpler to use. Some evolutions, however, are hard to distinguish from revolutions... And it must be said that while SGML is mostly used for technical documentation and much less for other kinds of data, with XML it is exactly the opposite.

- XML is license-free, platform-independent and well-supported

By choosing XML as the basis for some project, you buy into a large and growing community of tools (one of which may already do what you need!) and engineers experienced in the technology. Opting for XML is a bit like choosing SQL for databases: you still have to build your own database and your own programs/procedures that manipulate it, but there are many tools available and many people that can help you. And since XML, as a W3C technology, is license-free, you can build your own software around it without paying anybody anything. The large and growing support means that you are also not tied to a single vendor. **XML isn't always the best solution, but it is always worth considering.**

## 2. XML is a family of technologies

- XML1.0**                      W3C Recommendation Feb '98

The specification that defines what "tags" and "attributes" are, but around XML 1.0, there is a growing set of optional modules that provide sets of tags & attributes, or guidelines for specific tasks.

- Xlink**                      XML Linking Language

XML Linking Language, which allows elements to be inserted into XML documents in order to create and describe links between resources. It uses XML syntax to create structures that can describe links similar to the simple unidirectional hyperlinks of today's HTML, as well as more sophisticated links.

- **Xpointer**                      Supports addressing into the internal structures of XML documents.

Xpointer is **based on the XML Path Language (XPath)**, supports addressing into the internal structures of XML documents.

- XSL**                      The Extensible Stylesheet Language

It consists of three parts: **XSL Transformations (XSLT)**: a language for transforming XML documents; **the XML Path Language (XPath)**, an expression language used by XSLT to access or refer to parts of an XML document. (XPath is also used by the XML Linking specification). The third part is **XSL Formatting Objects**: an XML vocabulary for specifying formatting semantics.

- DOM**                      Document Object Model

The Document Object Model is a platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents.

- XML Namespaces**    Prevent name conflicts between your tags and somebody else's.

XML is designed to be modular and extensible, providing a simple method for qualifying element and attribute names used in Extensible Markup Language documents by associating them with namespaces identified by URI references. So you should be able to reuse modules from other DTDs in your own work.

- XML Schemas**        Inherited from SGML the DTD mechanism for **defining grammar rules**.

XML Schema **several advantages over DTDs**:

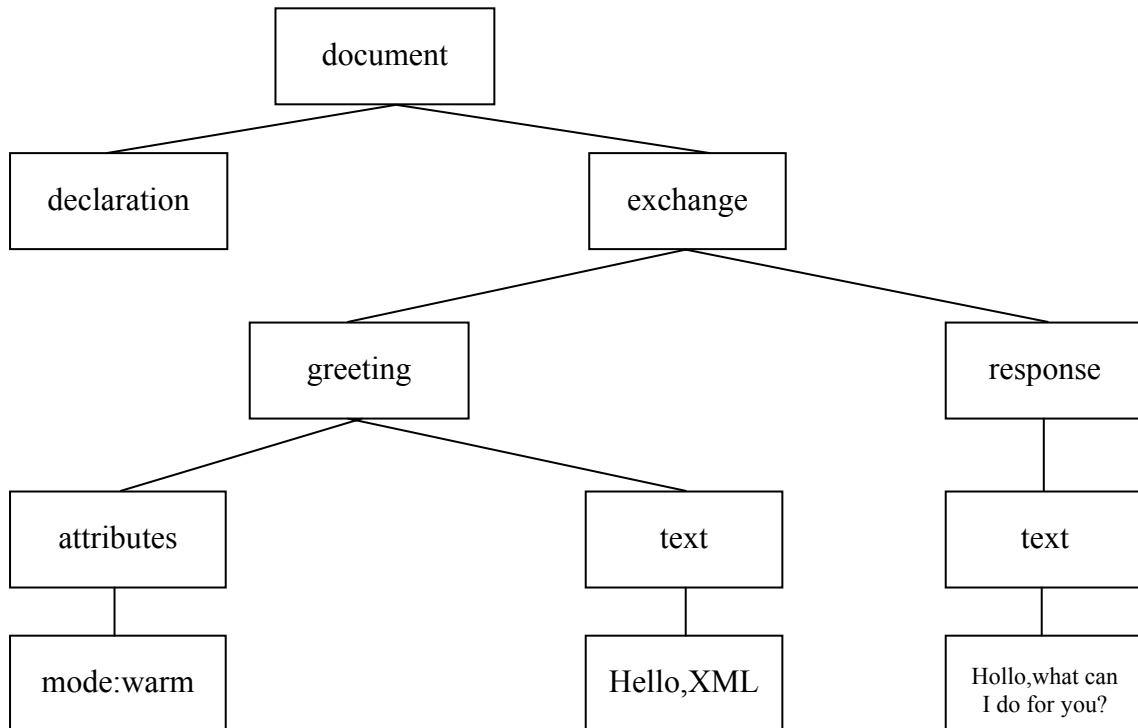
- 1.It uses XML syntax, and so can be produced and edited using standard XML tools.
- 2.It is a more detailed system of data types.
- 3.Users can define their own data types, including structured data types.

### 3. Main Uses of XML

- Creating documents for publishing on the Web and elsewhere
- Encoding data, including data extracted from relational database
- Configuring computer programs
- Developing domain-specific vocabularies and DTDs

## 4. A Simple Example

```
<?xml version="1.0" ?>
<!DOCTYPE exchange [
<!ELEMENT exchange (greeting, response)>
<!ELEMENT greeting (#PCDATA)>
<!ELEMENT response (#PCDATA)>
<!ATTLIST greeting
    mode(warm|luke-warm|cool|cold) "warm" <!-- "warm" is the default-- >
>
]>
<exchange>
    <greeting mode="warm" wording = "semi-formal"> Hello, XML</greeting>
    <response> Hello, what can I do for you?</response>
</exchange>
```



## 5. How it works

- An XML document is an information unit that can be viewed in two ways: as a linear sequence of characters that contains character data and markup, or as an abstract data structure that is a tree of nodes.
- To switch it from the linear view to the structure view, you need an XML processor, also known as an XML parser.
- Document Type Definition or DTD is a formally defined ‘grammar’ that expresses constraints on the document’s content.



## 6. XML Applications

- **SVG** (Scalable Vector Graphics)  
SVG is a language for describing two-dimensional graphics in XML. SVG allows for three types of graphic objects: vector graphic shapes (e.g., paths consisting of straight lines and curves), images and text. Graphical objects can be grouped, styled, transformed and composited into previously rendered objects.
- **JSP** (JavaServer Pages)  
JSP is to simplify the creation and management of dynamic web pages, by separating content and presentation. JSPs are basically files that combine standard HTML or XML and new scripting tags. They get translated into Java servlets the first time they are invoked by a client.
- **RDF** (Resource Description Framework)  
The Resource Description Framework using XML as an interchange syntax. The RDF specifications provide a lightweight ontology system to support the exchange of knowledge on the Web. The Resource Description Framework (RDF) is an infrastructure that enables the encoding, exchange and reuse of structured metadata.  
Metadata is machine understandable information for the web.  
It is an application of XML that imposes needed structural constraints to provide unambiguous methods of expressing semantics. RDF additionally provides a means for publishing both human-readable and machine-processable vocabularies designed to encourage the reuse and extension of metadata semantics among disparate information communities. The structural constraints RDF imposes to support the consistent encoding and exchange of standardized metadata provides for the interchangeability of separate packages of metadata defined by different resource description communities.
- Multiple languages for industry, finance, sciences, news and entertainment

## 7. Why XML is Great

- It's easy, both technically and practically, fast, reliable, ubiquitous and free.
- No special arrangements are needed between the sender and the recipient: XML data is always network-ready.
- Explicit public standards define not only the inputs to XML processors but also their outputs. Applications don't have to worry about which processor they use, and can concentrate on using and transforming the documents as data objects.

## 8. XML with MultiAgent System

Once the web has been sufficiently "populated" with rich metadata, what can we expect? First, searching on the web will become easier as search engines have more information available, and thus searching can be more focused. Doors will also be opened for automated software agents to roam the web, looking for information for us or transacting business on our behalf. The web of today, the vast unstructured mass of information, may in the future be transformed into something more manageable - and thus something far more useful. The effective use of metadata among applications requires common conventions about semantics, syntax, and structure. The design of enabling infrastructures such as RDF to support these constructs provides the necessary foundations to support the management of information on the web and provides the ability for transforming the web into a more useful and powerful information resource.

## 9. Reference

- [www.xml.com](http://www.xml.com)
- [www.w3.org](http://www.w3.org)
- [agents.umbc.edu](http://agents.umbc.edu)
- [www.multiagent.com](http://www.multiagent.com)
- Professional Java Server Programming J2EE Edition