Università della Svizzera italiana | Facoltà di scienze informatiche

XML
Extensible Markup Language

Prof. Cesare Pautasso
http://www.pautasso.info
cesare.pautasso@unisi.ch

Università della Svizzera italiana | Facoltà di scienze informatiche

XML is the future (**1999**)

XML will be **foundation** for
future Web standards

XML will become the **language** for
international desktop and Web publishing

XML will become the
universal data exchange format
between heterogeneous environments

XML will replace all
existing Word Processing **storage** formats

Predictions by John Bosak (Sun), 1999

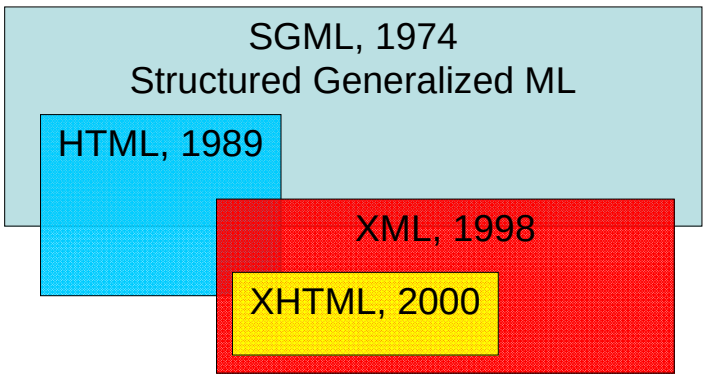
Università della Svizzera italiana | Facoltà di scienze informatiche

Contents

- Motivation: **Universal Syntax**
- History
- HTML vs. XML
- What is the Extensible Markup Language?
  - XML Syntax
  - XML Structure
- A critical look at XML
- XML Technology Landscape

Università della Svizzera italiana | Facoltà di scienze informatiche

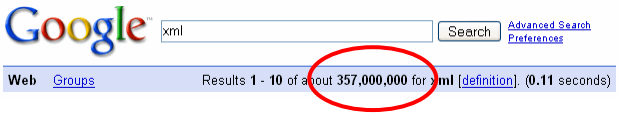
XML History



(X)HTML Example

- Write a program that downloads some amazon.com web page and extracts the title, author and price of a book (given its ISBN number)

- Write a program that asks google.com how many Web pages are there about a certain topic



HTML Scraping

Università della Svizzera italiana | Facoltà di scienze informatiche

XML Example

- Would not be better if Google would return the search results in a format that would make it easier for a program to figure it out?

```
<search keyword="xml">
  <results total="357,000,000">
      <site url="www.xml.com">XML.com</site>
  </results>
</search>
```

- Represent and structure the data using the "most appropriate" markup tags so that we can give it some conventional semantic meaning and write programs that can "understand" it.

Università della Svizzera italiana | Facoltà di scienze informatiche

HTML not enough

- Forgiving syntax check
  - No formal validation "at compile-time" of content correctness
  - Difficult to parse from your own programs (need to handle lots of garbage)
- Fixed set of tags – limited extensibility:
  - Users cannot customize the syntax
  - Extended by browser makers with lots of interoperability problems
- Predefined semantics of tags
  - Documents written in the HyperText Markup Language are very good for representing HyperText pages, but very bad for everything else!

Università della Svizzera italiana | Facoltà di scienze informatiche

XHTML vs. XML

- XHTML – hypertext markup language that follows the XML syntax rules
- XHTML Tags describe **Web pages** (hypertext nodes)
- XHTML Documents displayed by browsers for people to read

- XML – generic *meta*-language that defines syntax rules for a whole family of markup languages
- Can be used to describe the structure and markup any type of content
- XML documents typically processed automatically by software, not read by people

Università della Svizzera italiana | Facoltà di scienze informatiche

XML Syntax

XML Syntax

- Elements (enclosed in tags)
- Attributes (of elements)
- Text (inside elements)
- Namespace Declarations (special attributes)
- Prolog `<?xml version="1.0" encoding="UTF-8"?>`
- Entities (&) Character Data (CDATA)
- Comments `<!-- -->` (à la SGML)
- Processing Instructions `<? ?>` (SGML legacy)

- Note: **Order** of elements in document is preserved, order of attributes within element is not

XML Elements Syntax

- Markup Elements enclosed in Tags

`<element>…</element>`

`<element/>` (empty element tag)

- Elements tags can be nested and mixed with text

```
<element>…
   <subElement>…</subElement>…
</element>
```

- Warning: make sure nesting is well formed!

```
</element><sub><element></sub>
```

XML Attribute Syntax

- Element attributes

```
<element attribute="…">…</element>
<element attribute="…"/>
```

- Do we really need attributes?

```
<element>
   <attribute>…</attribute>
</element>
```

Attributes represent "hidden" data about the element, while element text is "visible"

- Warning: Attribute names must be unique!

```
<element a="1" a="ABC"></element>
```


XML Entities

- Special characters can be encoded (or escaped) using character entities
- In general, any Unicode character can be entered with the notation:

Character	Entity
<	<
>	>
&	&
"	"
'	'

&#N; (N is decimal)

&#xM; (M is hex)

Università della Svizzera italiana | Facoltà di scienze informatiche

XML Prolog Syntax

- Header found at the beginning of all XML documents:

```
<?xml version="1.0" encoding="UTF-8"?>
```

  - Specify the encoding
    usually Unicode (UTF-8, UTF-16) or ISO- 8859-1
  - Also XML version "1.1" exists, not yet widely used.

Università della Svizzera italiana | Facoltà di scienze informatiche

XML CDATA

- **Character Data** Sections are used to stop parsing the XML markup and treat the enclosed data as is:

```
<element>
  <![CDATA[<not An Element>]]>
</element>
```

XML Whitespace

- Should whitespace (spc, tabs, cr, lf) be ignored?
- It depends:
  - Data-oriented XML – ignore whitespace (more compact documents, faster processing, less readability)
  - Document-oriented XML – whitespace is part of the content and must be preserved:

```
<chapter xml:space="preserve">
  <section name="Introduction">
    Once upon a time…
  </section>
</chapter>
```

`xml:space="default"`
Application decides
how to handle it

XML Namespaces: Problem

- Problem:
  - What happens if two XML documents are pasted together?
  - How to distinguish "tag vocabularies" of different data sources?
- Solution:
  - Use **namespaces** to qualify and distinguish the elements of an XML subtree so that tag name collisions can be avoided.

```
<color>
  <blue/>
  <orange/>
  <red/>
  <black/>
  <white/>
</color>
```

```
<fruit>
  <apple/>
  <orange/>
  <ananas/>
  <banana/>
</fruit>
```

Example: mix XHTML tags
with your own XML data
to store "rich text" descriptions

Working with Namespaces

```
<ns:element
  xmlns:ns="URI"…>…
</ns:element>
```

- Declare a namespace prefix `ns` uniquely identified by `URI` using the special attribute **`xmlns`**
- The namespace declaration scope contains all attributes and all children elements

```
<mix xmlns:color="URI1"
  xmlns:fruit="URI2">
  <color:orange/>
  <fruit:orange/>
</mix>
```

- Prefixes are used to avoid repeating long URI for each element/attribute
- Elements without prefix belong to the **default** namespace (declared with `xmlns="DefaultURI"`)

XML Document Correctness

Well formed documents

- Verify the basic XML **syntax** constraints
- Only well formed documents can be parsed

Valid documents

- Well formed documents that satisfy additional **structural** constraints defined in a "schema"
- Non valid documents can still be processed even if no guarantees can be made on their "meaning"

Università della Svizzera italiana | Facoltà di scienze informatiche

Summary: **Well Formed** XML Syntax Rules

1. XML is case sensitive
2. Single Root Element for a document
3. Elements enclosed in angle brackets </>
   `<element></element>` or `<element/>`
4. Correct element nesting (XML tree)
5. Attributes must have unique names
6. Attribute values must be `"quoted"`
7. Document begins with XML prolog `<?xml …?>`
8. Namespaces prefixes must be declared before use

Università della Svizzera italiana | Facoltà di scienze informatiche

XML Structure

Document Type Definition (DTD)

XML Schema

Università della Svizzera italiana | Facoltà di scienze informatiche

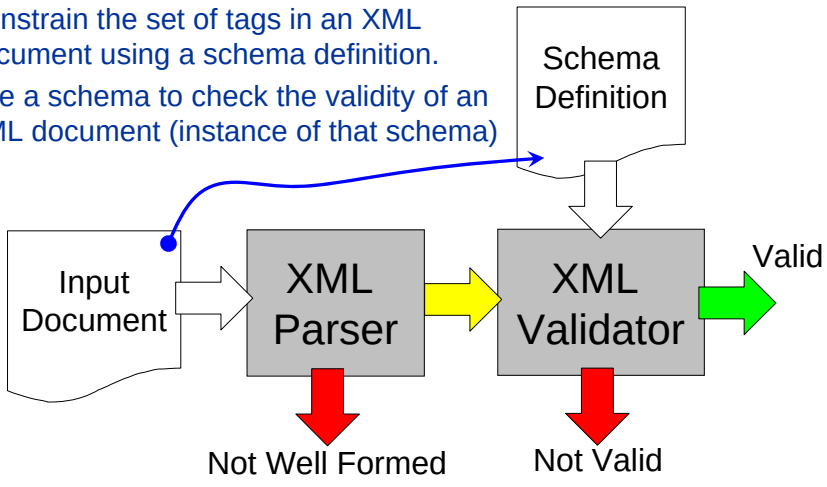
XML Structure

- Easy to start writing XML documents, simply make it well formed and make up the tags as you go.
- Challenges:
  - How to ensure multiple documents use the same set of tags and have the same structure?
  - How to write an application if we cannot be sure of which tags are used in the documents?
  - How to agree in advance on a specific XML format for exchanging a document between programs?
- Solution: use a "schema" to restrict and constrain the tags and the structure of an XML document

Università della Svizzera italiana | Facoltà di scienze informatiche

Schemas

- Constrain the set of tags in an XML document using a schema definition.
- Use a schema to check the validity of an XML document (instance of that schema)



Defining XML Structure

- Different schema languages are available to specify new XML-based languages.
- **Document Type Definition (DTD)** – part of the XML 1.0 specification. A relic from SGML, it does not use XML syntax, nor it supports namespaces (which were invented later)

  To address some of the limitations of DTD, other languages for constraining the structure of XML documents have been introduced:
- **XML Schema** – the new XML type system from W3C. Uses XML syntax. Very complex data modeling language.
- **RelaxNG** – a simpler schema language from OASIS with both XML syntax and a simplified notation.

Document Type Definitions DTD

- The Document Type Definition (DTD) specifies the tree structure of XML documents:
  - Nesting of Elements
  - Attributes of Elements
  - Values of Attributes
- A DTD contains a set of definitions of:
  1. Element Type
  2. Attribute List

Università della Svizzera italiana | Facoltà di scienze informatiche

Element Type Definition

```
<!ELEMENT name content>
```

The element with tag name contains certain elements, text, or a combination thereof.

```
<!ELEMENT name EMPTY>
```

The element name is empty: `<name/>`

```
<!ELEMENT name (#PCDATA)>
```

The element name only contains character data:

```
<name>Text without other elements</name>
```

```
<!ELEMENT name (mix)>
```

The element name can contain a mixture of elements and text:

```
<name>Text with <other/> elements</name>
```


Università della Svizzera italiana | Facoltà di scienze informatiche

Choice
Element Type Definition

```
<!ELEMENT name (first | last)>
<!ELEMENT first (#PCDATA)>
<!ELEMENT last (#PCDATA)>
```

The element name can contain a **choice** between two child elements: `first` xor `last`:

```
<name><first>Cesare</first></name>
<name><last>Pautasso</last></name>
```

Invalid (name contains both elements)

```
<name><last>Pautasso</last><first>Cesare</first></name>
```

Sequence
Element Type Definition

```
<!ELEMENT book (title, author, price))>
      <!ELEMENT title (#PCDATA)>
   <!ELEMENT author (first | last)>
      <!ELEMENT price (#PCDATA)>
```

The element book can contain a **sequence** of title, author, price elements (found in this order)

```
<book> <title>XML for dummies</title>
       <author>
          <first>Ed</first>
          <last> Tittle</last>
       </author>
       <price>$0.68</price>
</book>
```


Repeating elements (cardinality)

```
      <!ELEMENT book (title, author+, edition*))>
            <!ELEMENT title (#PCDATA)>
         <!ELEMENT author (first | last)>
            <!ELEMENT price (#PCDATA)>
   <!ELEMENT edition (#PCDATA | year? | price?)>
```

The element book can contain a sequence of exactly one title, at least one author, an optional price, and zero or more edition elements (which may include optional year and price elements).

```
<book>
      <title>XML for dummies</title>
      <author>Ed Tittle</author>
      <author>Ramesh Chandak</author>
      <edition>Paperback <year>1998</year></edition>
      <edition>Hardcover <price>$0.68</price></edition>
</book>
```

Element Type Definition Summary

<!ELEMENT name **content**>

content	
ANY	any contents
EMPTY	<name/>
name	an element
,	Sequence (Concatenation) of elements
|	Choice (Any Order) between elements
?	Optional
*	Zero or More
+	One or More

If no cardinality is specified, then the child element **must** be present only **once**

Attribute List Definition

<!ELEMENT element content>

<!ATTLIST element attribute type options>

Define the attribute for an element. Attributes have types. Options are used to control their presence and default values.

```
<!ATTLIST book number ID #REQUIRED>
<!ATTLIST price currency CDATA CHF>
<!ATTLIST author prefix (Prof|Dr) #IMPLIED>
<!ATTLIST edition number CDATA #REQUIRED>
```

Università della Svizzera italiana | Facoltà di scienze informatiche

Attribute Types

<!ATTLIST element attribute type options>

Attribute type	
CDATA	Character Data (String)
ID	Unique value within document to identify element
IDREF	Reference to other element ID
IDREFS	Multiple space-separated Ref.
(value | …)	Enumeration

Università della Svizzera italiana | Facoltà di scienze informatiche

Attribute Default Options

<!ATTLIST element attribute type options>

Attribute options	
#REQUIRED	Attribute must be present
#IMPLIED	Optional attribute w/o default
value	Optional attribute w/ default
#FIXED value	Set value no matter if attribute is present or not

Link XML Document to DTD

Local embedded definitions:

```
<!DOCTYPE root [definitions]>
```

External definitions: Use this one!

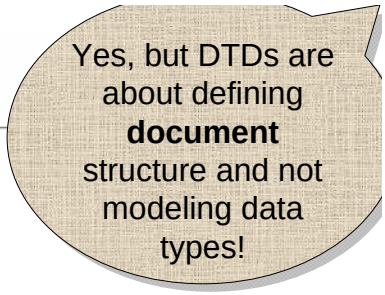
```
<!DOCTYPE root SYSTEM "url/filename">
<!DOCTYPE root PUBLIC "FPI">
<!DOCTYPE root PUBLIC "FPI" "url">
```

(FPI = Formal Public Identifier)

Example:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```


DTD Limitations



- Limited data types for element content and attributes
  - only CDATA, ID/IDREF or Enumeration
- Does not support Namespaces
- Definitions not specified using XML syntax
- References too simple
  - Cannot constrain ID uniqueness within a document subtree
  - Cannot have primary keys across multiple attributes
  - Does not support referential integrity
- Context insensitive definitions
  - Cannot constrain structure based on parents of an element
  - Cannot have elements present if an attribute has a certain value or if also another element is present (co-occurrence)

Università della Svizzera italiana | Facoltà di scienze informatiche

DTD vs. XML Schema

- ~~Limited data types for element content and attributes~~
  - ~~only CDATA, ID/IDREF or Enumeration~~
- ~~Does not support Namespaces~~
- ~~Definitions not specified using XML syntax~~
- ~~References too simple~~
  - ~~Cannot constrain ID uniqueness within a document subtree~~
  - ~~Cannot have primary keys across multiple attributes~~
  - ~~Does not support referential integrity~~
- Context insensitive definitions
  - Cannot constrain structure based on parents of an element
  - Cannot have elements present if an attribute has a certain value or if also another element is present (co-occurrence)

Rich built-in XML Data Types (44)

Namespace Support

XML Syntax

Key+References

Università della Svizzera italiana | Facoltà di scienze informatiche

XML Schema (XSD)

- XML language to define the structure of XML documents. W3C standard from 2001.
- Schemas are composed of:
  - Data Type Definitions
    - Simple Types - Restrictions
    - Complex Types
      - Constructors: Sequence, All (=Choice)
      - Cardinality (minOccurs, maxOccurs)
  - Element/Attribute Declarations (of type)
    - Key Identifiers

Università della Svizzera italiana | Facoltà di scienze informatiche

XML Schema Example

```
<xs:element name="person">
   <xs:complexType>
        <xs:sequence>
               <xs:element ref="name"/>
                <xs:element ref="email" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref="url" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref="link" minOccurs="0" maxOccurs="1"/>
        </xs:sequence>
        <xs:attribute name="id" type="xs:ID" use="required"/>
         <xs:attribute name="note" type="xs:string“/>
         <xs:attribute name="contr" default="false">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="true"/>
                        <xs:enumeration value="false"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="salary" type="xs:integer“/>
   </xs:complexType>
</xs:element>
```

Hands-Out Example

Università della Svizzera italiana | Facoltà di scienze informatiche

XML Schema Example
Complex Types/Element

```
<xs:element name="person">
 <xs:complexType>
  <xs:sequence>
   <xs:element ref="name"/>
   <xs:element ref="email"  minOccurs="0" maxOccurs="unbounded"/>
   <xs:element ref="url"    minOccurs="0" maxOccurs="unbounded"/>
   <xs:element ref="link"   minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
 </xs:complexType>
</xs:element>
```

Anonymous Type Definition

Cardinality
default (`minOccurs="1",maxOccurs="1"`)

`sequence` (all, in order)
`all` (all, in any order)
`choice` (only one)
`any`

Università della Svizzera italiana | Facoltà di scienze informatiche

XML Schema Example
Simple Types/Attribute

```
<xs:element name="person">
 <xs:complexType>
  <xs:attribute name="id" type="xs:ID" use="required"/>
  <xs:attribute name="note" type="xs:string"/>
  <xs:attribute name="contr" default="false">
   <xs:simpleType>
    <xs:restriction base="xs:string">
     <xs:enumeration value="true"/>
     <xs:enumeration value="false"/>
    </xs:restriction>
   </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="salary" type="xs:integer"/>
 </xs:complexType>
</xs:element>
```

Custom Defined Type

Built-in Types (ID, integer, string, …)

Università della Svizzera italiana | Facoltà di scienze informatiche

Link XML Document to XSD

- Add a pointer in the root element of the XML document

```
<root xmlns="namespace"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="namespace url"
  />
```

- Configure namespaces in the schema

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

Università della Svizzera italiana | Facoltà di scienze informatiche

XML Critique

Università della Svizzera italiana | Facoltà di scienze informatiche

The X factor

If I would invent another programming language, its name will contain the letter X

Niklaus Wirth,
2001

XML as the Universal Syntax

The Extensible Markup Language is used in all aspects of a Web application:

- Data Representation
  - regular structure (like in database management)
  - semi-structured
  - unstructured textual documents (e.g., XHTML pages)
- Meta-Data
  - Define rules on how the data should be structured
- Code
  - Specify how data should be processed/transformed

Text vs. Structured Data

Text

The Conductor says the Cisalpino train is arriving in Lugano in 5 minutes

```
<message from="Conductor">
The Cisalpino train will arrive in
   Lugano in 5 minutes
</message>
```

XML

```
<message from="Conductor">
   <train>Cisalpino</train>
   <event>arrival</event>
   <station>Lugano</station>
   <time unit="minutes">5</time>
</message>
```

Data-oriented XML

Document-oriented XML

```
<message from="Conductor">
The <train>Cisalpino</train> will arrive
in <station>Lugano</station>
in <time>5 <unit>minutes</unit></time>
</message>
```

Università della Svizzera italiana | Facoltà di scienze informatiche

Where is XML used?

- **More than the Web (XHTML)**
- Blogs (Rich Site Summary – RSS Feeds)
- Document Markup Formats
  - Office Documents (OOXML/ODF, WordML)
  - Wireless Mobile Applications (WML)
  - Chemistry (CML)
  - Theology (ThML)
  - Music (MusicML)
  - Speech (VoiceML)
  - Graphics (Scalable Vector Graphics – SVG)
  - Many more!
- Meta-Data (XML Schema, WSDL, Resource Description Framework – RDF, Adobe Extensible Metadata Platform – XMP)
- Logs (Common Base Events – CBE Format)
- Configuration Files (J2EE Deployment Descriptors, ANT build scripts)
- Communication Protocols (Web Services)
- Databases

Università della Svizzera italiana | Facoltà di scienze informatiche

XML Benefits

1. Text-based (Unicode) format
   - Human readable
   - Machine readable
2. W3C Standard
   - *Independent* of Platform, OS, Vendor, Programming Language, Communication Protocol
3. Easy to start writing well formed XML documents
4. Rich Technology Tool-chain
   - More than general data representation format: choose XML and you can reuse a rich family of data management and processing technologies

Università della Svizzera italiana | Facoltà di scienze informatiche

XML Limitations

1. Text-based (Unicode) format
   - **Verbose** compared to a binary or simpler textual format (like JSON)
2. Not a graph, only a tree
   - The notion of reference not well integrated
3. "Self-Describing" Data
   - XML is about syntax (well-formed) and structure (valid, according to a schema)
   - Tags do not have any implicit semantics: it depends on application interpretation

Università della Svizzera italiana | Facoltà di scienze informatiche

XML Technology Landscape

- Data Representation
  - XML Syntax
  - XML Information Set (InfoSet)
  - XML Namespaces
  - XML Schema, DTD
  - XLink, XPointer
- Data Processing
  - XPath
  - XSLT – Extensible Stylesheet Transformation Language
  - XQuery
  - XUpdate
- Data Processing API
  - DOM
  - SAX
  - JAXP
- Communication Protocols
  - XML Forms
  - XML Web Services (SOAP, WSDL, UDDI)
  - XML Encryption
  - XML Digital Signature

Università della Svizzera italiana | Facoltà di scienze informatiche

References

- Anders Moller and Michael Schwartzbach, **An Introduction to XML and Web Technologies**, Addison-Wesley, 2006
- Elliotte Rusty Harold and W. Scott Means, **XML in a Nutshell**, O'Reilly, 3rd Ed. 2004
- John Bosak, **XML Ubiquity and the Scholarly Community,** Computers and the Humanities, Volume 33, Numbers 1-2, April 1999 , pp. 199-206(8)