XML

Evolution of XML

- XML is a descendant of SGML, the Standard Generalized Markup Language. The language that would eventually become SGML was invented by Charles F. Goldfarb, Ed Mosher, and Ray Lorie at IBM in the 1970s and developed by several hundred people around the world until its eventual adoption as ISO standard 8879 in 1986.
- SGML was intended to solve many of the same problems XML solves in much the same way XML solves them. It is a semantic and structural markup language for text documents. SGML is extremely powerful and achieved some success in the U.S. military and government, in the aerospace sector, and in other domains that needed ways of efficiently managing technical documents that were tens of thousands of pages long.

Evolution of XML

- In 1996, Jon Bosak, Tim Bray, C. M. Sperberg-McQueen, James Clark, and several others began work on a “lite” version of SGML that retained most of SGML’s power while trimming a lot of the features that had proven redundant, too complicated to implement, confusing to end users, or simply not useful over the previous 20 years of experience with SGML. The result, in February of 1998, was XML 1.0, and it was an immediate success. Many developers who knew they needed a structural markup language but hadn’t been able to bring themselves to accept SGML’s complexity adopted XML whole-heartedly.

Evolution of XML

- Extensible Markup Language (XML) is a markup language and file format for storing, transmitting, and reconstructing arbitrary data. It defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The World Wide Web Consortium's XML 1.0 Specification of 1998 and several other related specifications ,all of them free open standards—define XML.

Use of XML

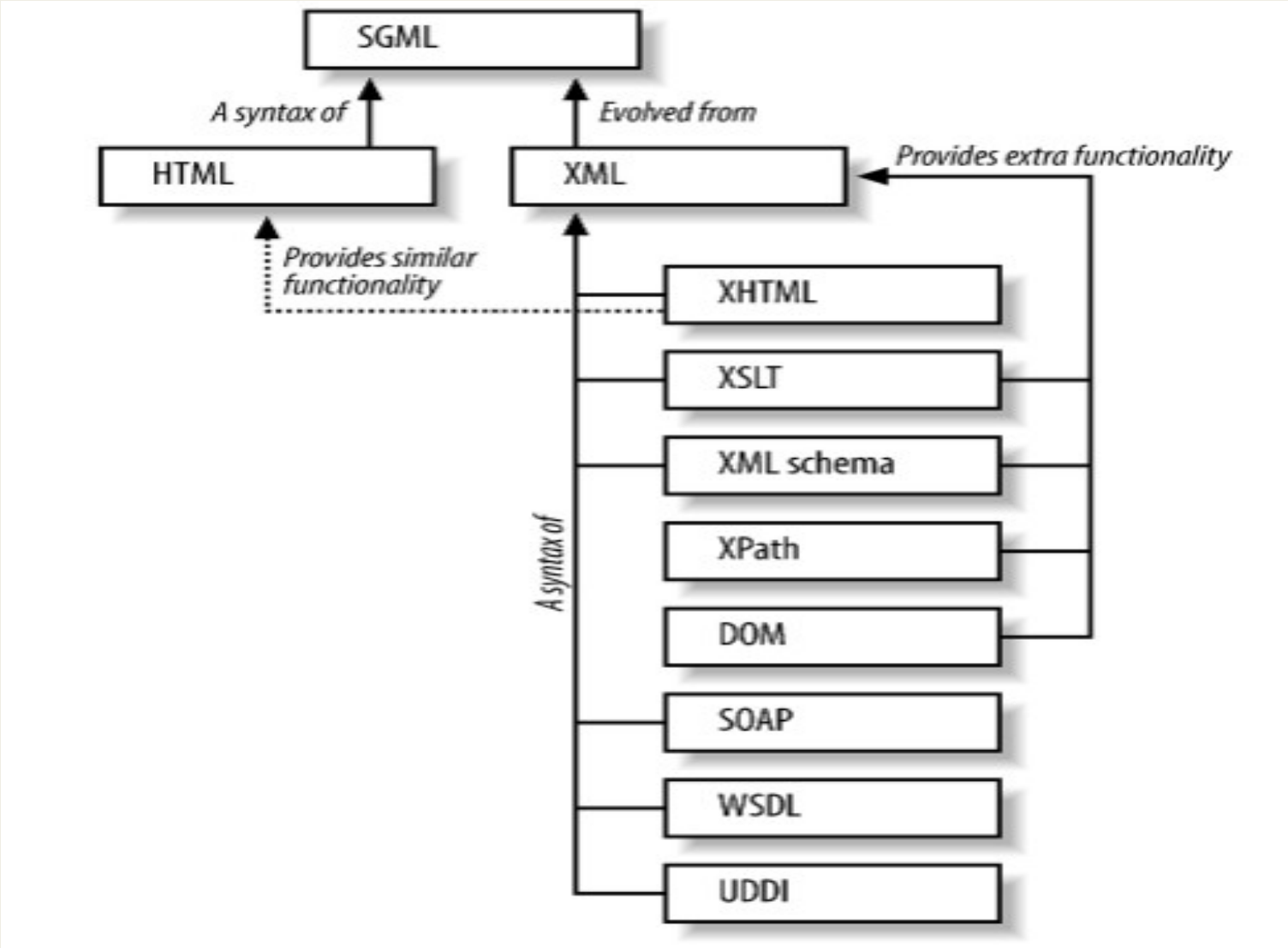
- You can use XML to create your own markup language that includes a set of rules and tags that describe information suited to your needs, for example, name, title, address, and zip code. You define this markup language in a document type definition (DTD) or XML Schema file that functions as the standard way to describe your information. Using XML to share standardized information means you are no longer required to write programs to focus on proprietary software or convert and translate different data formats.

- XML has a variety of uses for Web, e-business, and portable applications.
- The following are some of the many applications for which XML is useful:

- **Web publishing**: XML allows you to create interactive pages, allows the customer to customize those pages, and makes creating e-commerce applications more intuitive. With XML, you store the data once and then render that content for different viewers or devices based on style sheet processing using an Extensible Style Language (XSL)/XSL Transformation (XSLT) processor.
- **Web searching and automating Web tasks**: XML defines the type of information contained in a document, making it easier to return useful results when searching the Web:For example, using HTML to search for books authored by Tom Brown is likely to return instances of the term 'brown' outside of the context of author. Using XML restricts the search to the correct context (for example, the information contained in the <author> tag) and returns only the information that you want. By using XML, Web agents and robots (programs that automate Web searches or other tasks) are more efficient and produce more useful results.

- **General applications**: XML provides a standard method to access information, making it easier for applications and devices of all kinds to use, store, transmit, and display data.
- **e-business applications**: XML implementations make electronic data interchange (EDI) more accessible for information interchange, business-to-business transactions, and business-to-consumer transactions.
- **Metadata applications**: XML makes it easier to express metadata in a portable, reusable format.
- **Pervasive computing**: XML provides portable and structured information types for display on pervasive (wireless) computing devices such as personal digital assistants (PDAs), cellular phones, and others. For example, WML (Wireless Markup Language) and VoiceXML are currently evolving standards for describing visual and speech-driven wireless device interfaces.

XML Family


SGML
A syntax of
Evolved from
HTML
XML
Provides extra functionality
Provides similar functionality
XHTML
XSLT
XML schema
XPath
DOM
SOAP
WSDL
UDDI
A syntax of

XML Family

- DOM
- You can manipulate an XML document’s tree structure through the Document Object Model (DOM). The DOM specification was developed to introduce a platform-independent model for XML documents.

- XPath
- You will sometimes want to locate a particular element or attribute in the content of an XML document. The XPath specification provides the mechanism used to navigate an XML document.

- XSLT
- Different organizations often develop different markup languages for the same problem domain. In those cases, it can be useful to transform an existing XML document in one format into another document in another format. XML Stylesheet Language Transformations (XSLT) was developed to enable you to convert XML documents into other XML and non-XML formats

XML Family

- XML Schema
- The original XML specification included the Document Type Description (DTD), which allows you to specify the structure of an XML document. The XML Schema standard allows you to constrain an XML document in a more formal manner than DTD. Using an XML Schema, you can ensure that a document structure and content fits the expected model.
- Serialization
- In addition to the XML technologies listed above, there are specific XML syntaxes used for specific purposes. One such purpose is serializing objects into XML. Objects can be serialized to an arbitrary XML syntax, or they can be serialized to the Simple Object Access Protocol (SOAP)
- Web Services
- Web Services allows for the sharing of resources on a network as if they were local through XML syntaxes such as SOAP, Web Services Definition Language (WSDL), and Universal Description, Discovery, and Integration (UDDI). Web Services provides the foundation for .NET remoting, although Web Services is, by its nature, an open framework that is operating system- and hardware-independent.

XML Family

- Data
- Most modern software applications are concerned in some way with storing and accessing data. While XML can itself be used as a rudimentary data store, relational database management systems, such as SQL Server, DB2, and Oracle, are much better at providing quick, reliable access to large amounts of data.

XML Namespaces

- As defined by the W3C Namespaces in XML Recommendation , an XML namespace is a collection of XML elements and attributes identified by an Internationalized Resource Identifier (IRI); this collection is often referred to as an XML "vocabulary."

- One of the primary motivations for defining an XML namespace is to avoid naming conflicts when using and re-using multiple vocabularies. XML Schema is used to create a vocabulary for an XML instance, and uses namespaces heavily. Thus, having a sound grasp of the namespace concept is essential for understanding XML Schema and instance validation overal

XML Namespaces

- Namespaces are similar to packages in Java in several ways:
- A package in Java can have many reusable classes and interfaces. Similarly, a namespace in XML can have many reusable elements and attributes.
- To use a class or interface in a package, you must fully qualify that class or interface with the package name. Similarly, to use an element or attribute in a namespace, you must fully qualify that element or attribute with the namespace.
- A Java package may have an inner class that is not directly inside the package, but rather "belongs" to it by the virtue of its enclosing class. The same is true for namespaces: there could be elements or attributes that are not directly in a namespace, but belongs to the namespace by virtue of its parent or enclosing element. This is a transitive relationship. If a book is on the table, and the table is on the floor, then transitively, the book is on the floor; albeit the book is not directly on the floor.

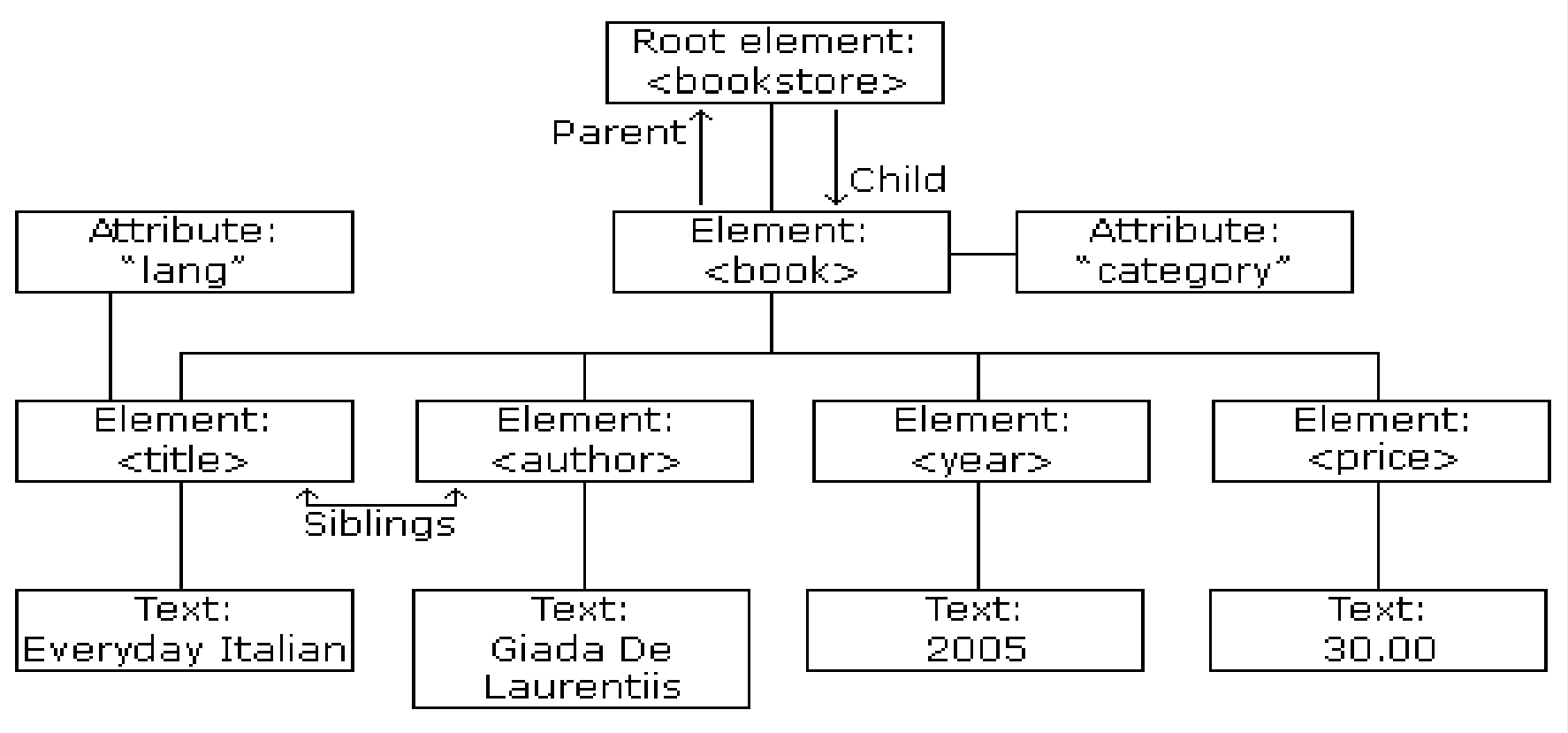
XML Namespaces

- Namespaces are declared as an attribute of an element. It is not mandatory to declare namespaces only at the root element; rather it could be declared at any element in the XML document. The scope of a declared namespace begins at the element where it is declared and applies to the entire content of that element, unless overridden by another namespace declaration with the same prefix name–where, the content of an element is the content between the <opening-tag> and </closing-tag> of that element. A namespace is declared as follows:

- <someElement xmlns:pfx="http://www.foo.com" />

- In the attribute xmlns:pfx, xmlns is like a reserved word, which is used only to declare a namespace. In other words, xmlns is used for binding namespaces, and is not itself bound to any namespace. Therefore, the above example is read as binding the prefix "pfx" with the namespace "http://www.foo.com."

Structure of XML file



Syntax Rules for XML Declaration

- The XML declaration is case sensitive and must begin with "**<?xml>**" where "**xml**" is written in lower-case.
- If document contains XML declaration, then it strictly needs to be the first statement of the XML document.
- The XML declaration strictly needs be the first statement in the XML document.
- An HTTP protocol can override the value of *encoding* that you put in the XML declaration.

XML Attributes

- An attribute specifies a single property for the element, using a name/value pair. An XML-element can have one or more attributes. For example −

- Tutorialspoint!
- An element can have multiple unique attributes. Attribute gives more information about XML elements. To be more precise, they define properties of elements. An XML attribute is always a name-value pair.

- An XML attribute has the following syntax −

- <element-name attribute1 attribute2 >
- ....content..
- < /element-name>