Learning Objectives

- define the purpose of SGML, HTML, and XML

There are four central problems in data management: capture, storage, retrieval, and exchange of data. The purpose of this book is to address **XML, a technology for managing data exchange**. The foundational XML chapters in this book are structured by a 'data model' approach. The first chapter introduces the reader to the XML document, XML schema, and XML stylesheet with a single entity example. Subsequent chapters expand upon the XML basics with multiple-entity examples and a one-to-one relationship, a one-to-many relationship, or a many-to-many relationship.

XML is a tool used for data exchange. Data exchange has long been an issue in information technology, but the Internet has elevated its importance. Electronic data interchange (EDI), the
traditional data exchange standard for large organizations, is giving way to XML, which is likely to become the data exchange standard for all organizations, irrespective of size.

EDI supports the electronic exchange of standard business documents and is currently the major data format for electronic commerce. A structured format is used to exchange common business documents (e.g., invoices and shipping orders) between trading partners. In contrast to the free form of e-mail messages, EDI supports the exchange of repetitive, routine business transactions. Standards mean that routine electronic transactions can be concise and precise. The main standard used in the United States and Canada is known as X.12, and the major international standard is UN/EDIFACT. Firms adhering to the same standard can share data electronically.

The Internet is a global network potentially accessible by nearly every firm, with communication costs typically less than those of traditional EDI. Consequently, the Internet has become the electronic transport path of choice between trading partners. The simplest approach is to use the Internet as a means of transporting EDI documents. But because EDI was developed in the 1960s, another approach is to reexamine the technology of data exchange. A result of this rethinking is XML, but before considering XML we need to learn about SGML, the parent of XML.

For a typical U.S. firm, it is estimated that document management consumes up to 15 percent of its revenue, nearly 25 percent of its labour costs, and anywhere between 10 and 60 percent of an office
worker’s time. The Standard Generalized Markup Language (SGML) is designed to reduce the cost and increase the efficiency of document management.

A markup language embeds information about a document within the document's text. In the following example, the markup tags indicate that the text contains details of a city. Note also that the city's name, state, and population are identified by specific tags. Thus, the reader—a person or a computer—is left in no doubt as to meaning of Athens, Georgia, or 100,000. Note also the latitude and location of the city are explicitly identified with appropriate tags. SGML’s usefulness is based upon both recording text and the meaning of that text.

Exhibit 1: Markup language

```html
<city>
  <cityname>Athens</cityname>
  <state>GA</state>
  <description>Home of the University of Georgia</description>
  <population>100,000</population>
  <location>Located about 60 miles Northeast of Atlanta</location>
  <latitude>33° 57’ 39” N</latitude>
  <longitude>83° 22’ 42” W</longitude>
</city>
```

SGML is a vendor-independent International Standard (ISO 8879) that defines the structure of documents. Developed in 1986 as a meta language, SGML is the parent of both HTML and XML. Because SGML documents are standard text files, SGML provides cross-system portability. When technology is rapidly changing, SGML provides a stable platform for managing data exchange. Furthermore, SGML files can be transformed for publication in a variety of media. The use of SGML preserves textual information independent of how and when it is presented. Organizations reap long-term benefits when they can store documents in a single, independent standard that can then be converted for display in any desired media.

SGML has three major advantages for data management:

- **Reuse**: Information can be created once and reused many times.
- **Flexibility**: SGML documents can be published in any format. The same content can be printed, presented on the Web, or delivered with a text synthesis. Because SGML is content-oriented, presentation decisions can be delayed until the output format is decided.
- **Revision**: SGML supports revision and version control. With content version control, a firm can readily track the changes in documents.

A short section of SGML demonstrates clearly the features and strength of SGML (see Exhibit 2). The tags surrounding a chunk of text describe its meaning and thus support presentation and retrieval. For example, the pair of tags `<airline>` and `</airline>` surrounding “Delta” identify the airline making the flight.
Exhibit 2: SGML example

<flight>
  <airline>Delta</airline>
  <flightno>22</flightno>
  <origin>Atlanta</origin>
  <destination>Paris</destination>
  <departure>5:40pm</departure>
  <arrival>8:10am</arrival>
</flight>

The preceding SGML code can be presented in several ways by applying a style sheet to the file. For example, it might appear as

Delta flight 22 flies from Atlanta to Paris leaving 5:40pm and arriving 8:10am

or as

<table>
<thead>
<tr>
<th>Airline</th>
<th>Flight</th>
<th>Origin</th>
<th>Destination</th>
<th>Departure</th>
<th>Arrival</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delta</td>
<td>22</td>
<td>Atlanta</td>
<td>Paris</td>
<td>5:40pm</td>
<td>8:10am</td>
</tr>
</tbody>
</table>

If the data are stored in HTML format and rendered on a Web site (as in Exhibit 3), then the meaning of the data has to be inferred by the reader. This is generally quite easy for humans, but impossible for machines. Furthermore, the presentation format is fixed and can only be altered by rewriting the HTML. If you are not familiar with HTML, you should read the WikiBooks chapter on XHTML, an extension of HTML, before reading the next chapter.

Exhibit 3: HTML rendering example

Delta flight 22 flies from Atlanta to Paris leaving 5:40pm and arriving 8:10am

Meaning and presentation should be independent, and this is an important reason why SGML is more powerful than HTML.

SGML is a markup language that defines the structure of documents and is preferred to HTML as it can be transformed into a variety of media.

Many computer systems contain data in incompatible formats. A time-consuming challenge is to exchange data between such systems. XML is a generic data storage format that comes bundled with a number of tools and technologies that should make it easier to exchange specific XML 'applications' between incompatible systems. Since XML is open and generic, it is expected that as time progresses,
more and more organizations and people will jump onto the XML bandwagon, both developers and data users. This should make XML the ultimate viable technology for certain types of data exchange.

XML is used not only for exchanging information, but also for publishing Web pages. XML’s very strict syntax allows for smaller and faster Web browsers and as such is well suited for use with Personal Digital Assistants (PDAs) and cellphones. Web browsers that interpret HTML documents, on the other hand, are bloated with programming code to compensate for HTML’s not so strict coding.

The types of data generally well suited for encoding as XML are those where field lengths are unknown and unpredictable and where field contents are predominantly textual.

An XML schema allows for the exchange of information in a standardized structure. A schema defines custom markup tags that can contain attributes to describe the content that is enclosed by these tags. Information from the tagged data in the XML document can be extracted using an application called a “parser”, and with the use of an XML stylesheet the data can be formatted for a Web page.

XML’s power lies in the combination of custom markup tags and content in a defined XML document. The purpose of eXtensible Markup Language (XML) is to make information self-describing. Based on SGML, XML is designed to support electronic commerce. The definition of XML, completed in early 1998 by the World Wide Web Consortium (W3C), describes it as a meta language — a language to generate languages. XML should steadily replace HTML on many Web sites because of some key advantages. The major differences between XML and HTML are captured in the following table.

**Exhibit 4: XML vs HTML**

<table>
<thead>
<tr>
<th>XML</th>
<th>HTML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information content</td>
<td>Information presentation</td>
</tr>
<tr>
<td>Extensible set of tags</td>
<td>Fixed set of tags</td>
</tr>
<tr>
<td>Data exchange language</td>
<td>Data presentation language</td>
</tr>
<tr>
<td>Greater hypertext linking</td>
<td>Limited hypertext linking</td>
</tr>
</tbody>
</table>

The eXtensible in XML means that a new data exchange language can be created by defining its structure and tags. For example, the OpenGIS Consortium designed a Geography Markup Language (GML) to facilitate the electronic exchange of geographic information. Similarly, the Open Tourism Consortium is working on the definition of TourML to support exchange of tourism information.

The insurance industry uses data corresponding to the XML based standard ACORD for electronic data exchange. Another good example of XML in action is NewsML™.
In this text we will cover all the features of XML, but at this point let us introduce a few of the key features.

**Applications of XML:**

Before we start learning more about how an XML document is structured, let us point out what XML can be used for. The four major implementations of XML are:

**Publication:** Database content can be converted into XML and afterwards into HTML by using an XSLT stylesheet. Making use of this technique, complex websites as well as print media like PDF files can be generated. Information no longer has to be stored in different formats (i.e. RTF, DOC, PDF, HTML). Content can be stored in the neutral XML format and then, using appropriate layout style sheets and transformations, brochures, websites, or datalists can be generated (See more in Chapter 17.)

An example of the capability of XML and XSLT can be found at [http://www.emimusic.de](http://www.emimusic.de): This website contains approximately 20,000 pages with profiles of the artists, their products and the titles of the songs. These pages are generated using a XSLT script. Based on the script used it will also be possible to create a catalog in PDF format. Please see below for more details.

**Interaction:** XML can be used for accessing and changing data interactively. This man<>machine communication usually happens via a web browser (see Chapter 12).

**Integration:** Using XML, homogenous and heterogenous applications can be integrated. In this case, XML is used to describe data, interfaces, and protocols. This machine-machine communication helps integrate relational databases (i.e. by importing and exporting different formats).

**Transaction:** XML helps to process transactions in applications like online marketplaces, supply chain management, and e-procurement systems.

---

**Key features of XML**

- Elements have both an opening and a closing tag
- Elements follow a strict hierarchy, with documents containing only one root element
- Elements cannot overlap other elements
- Element names must obey XML naming conventions
- XML is case sensitive

XML will improve the efficiency of data exchange in several important ways, which include:
write once and format many times: Once an XML file is created it can be presented in multiple ways by applying different XML stylesheets. For instance, the information might be displayed on a web page or printed in a book.

hardware and software independence: XML files are standard text files, which means they can be read by any application.

write once and exchange many times: Once an industry agrees on a XML standard for data exchange, data can be readily exchanged between all members using that standard.

Faster and more precise web searching: When the meaning of information can be determined by a computer (by reading the tags), web searching will be enhanced. For example, if you are looking for a specific book title, it is far more efficient for a computer to search for text between the pair of tags <booktitle> and </booktitle> than search an entire file looking for the title. Furthermore, spurious results should be eliminated.

10 reasons to use XML

1. XML is a widely accepted open standard.
2. XML allows to clearly separate content from form (appearance).
3. XML is text-oriented.
4. XML is extensible.
5. XML is self-describing.
6. XML is universal; meaning internationalization is no problem.
7. XML is independent from platforms and programming languages.
8. XML provides a robust and durable format for information storage.
9. XML is easily transformable.
10. XML is a future-oriented technology.

The major XML elements

The major XML elements are:

- **XML document**: An XML file containing XML code.
- **XML schema**: An XML file that describes the structure of a document and its tags.
XML stylesheet: An XML file containing formatting instructions for an XML file.

In the next few chapters you will learn how to create and use each of these elements of XML.

Creating a markup file

Any text editor can be used to create a markup file (e.g., an HTML file). In this book, we use the text editor within NetBeans, an open source Integrated Development Environment (IDE) for Java, because NetBeans supports editing and validation of XML files. Before proceeding, you should download and install NetBeans from http://www.NetBeans.org/.

The examples in this book use NetBeans to illustrate proper XML code. For an alternative to NetBeans, see Exchanger XML Lite.

Case Studies in XML Implementation

XML at United Parcel Service (UPS)

“UPS is a service company and it is all about scale and speed,” says Geoff Chalmers, Project Leader at UPS eSolutions Department. In 2003, UPS had $33.5 billion annual revenue and 357,000 employees worldwide. Six percent of the United States’ Gross Domestic Product (GDP) on any given day is in the UPS system.

UPS uses technology extensively. The Information Systems department employs 4,000 people. The company’s web site has 166 different country home pages and is supported by 44 applications.

UPS delivers around 13 million packages every day, and customers can track these shipments via the UPS Web site, which receives around 200 million hits daily. Nineteen of the applications within ups.com are XML OnLine Tool (Web services) applications.

UPS’s online tools are developed specifically to be integrated with customers’ applications. This makes the customer’s task simpler, easier, and faster. UPS verified the importance of simplicity and speed, via “CampusShip,” a product that has been one of the UPS’s most successful in the last 10 years. UPS CampusShip® is a Web-based, UPS-hosted shipping system. Using an Internet connection, employees can ship their own packages and letters from any desktop, while management maintains overall control of shipping activities. UPS CampusShip® allows simultaneous shipper autonomy and managerial cost-control within the organization. This product has been successful because no installation or software maintenance is required and it is quick to implement. XML Online Tools enabled cheap and fast evolution of CampusShip®.

UPS favors XML especially because it is agnostic; platform and language independent. These features make XML very flexible and powerful. It is also decoupled and scalable. XML has enabled UPS to
target a broader market and reduce customer interaction, and thus the cost of customer service. Another positive feature of XML is that it is backward compatible. The adoption of XML has reduced maintenance, implementation, and usage costs significantly within UPS.

However these advantages don’t come without a price. “XML is inefficient in so many ways” says Chalmers. XML unfortunately takes more CPU and bandwidth than the other technologies. Yet bandwidth and CPU are cheap and getting cheaper everyday, so this is a gradually disappearing problem.

Nevertheless, Chalmers also thinks that XML doesn’t work well in databases. He says that it is too wordy and it is an exchange medium rather than a database medium. There were some early attempts to tightly integrate XML and databases. Because databases do supply structure and identification to data as does XML, the value-add of XML-database integration is limited to applying hierarchical structure. On the other hand, if data is to be stored as a blob, then XML makes sense. Another problem that he points out about XML is that business rules cannot be expressed in XML schemas. Finally, raw XML programming and debugging can be challenging. Therefore, UPS’s enterprise customers are starting to explore the code generators and embedded facilities to be found in .NET and BEA. However, hand coding by experienced in-house engineers is a must for the high availability, scalability, and performance that UPS requires for the UPS OnLine Tools.

**[edit]XML at EMI Music**

How is it used?

EMI Music Germany GmbH & Co. KG, a famous German record label, displays information about the artists it is affiliated with on its website. Visitors are able to explore all their audio or video productions. The whole website consists of nearly 20,000 pages that contain information about artists and their products (CD, DVD, LP). Everything is properly linked and systematically grouped.

After all, there is data to be provided for every artist, albums, samples, pictures, descriptions or article codes. The site is updated on a daily basis and is subject to change by a web editor whenever it’s necessary. Now this is a fairly complex and large amount of data to be handled.

This is where XML comes into play. The data, which is stored in a database, has been transformed into XML code. Now an XSLT stylesheet converts this data into HTML code, which can be easily read by any web browser (e.g. Internet Explorer or Firefox).

What’s the benefit?

The advantage of XML is that the programming effort is considerably lower as compared to other formats. This is because XML lies at the point of intersection of XSLT and HTML.
It's also no problem for the web editor to update the website. Using XML makes it easy for the person in charge to deal with this large amount of data.

**Going beyond...** On the basis of the XML scripts thus far produced by EMI Music, the company could easily produce a PDF-formatted catalog or design i-Mode pages for the current mobile phone generation. Thanks to XML, this can be done with little extra effort.

**[edit]**A brief history of XML

In the late 60s Charles Goldfarb, Raymond Lorie and Edward Mosher all working for IBM started to develop GML (Generalized Markup Language), a text formatting language. The language was successfully applied for internal documentation procedures. As it used to be common, the document editing was performed in the batch-mode. GenCode, another procedure to define generic formatting codes for the typesetting systems of various software producers, was developed by the GCA (Graphic Communications Association) at about the same time. Both of these technologies, GML syntactically and GenCode semantically, served as basis for the development of SGML (Standard Generalized Markup Language). The process of standardization started at the U.S. Standardization institute ANSI in the early 80s and in 1996 SGML finally passed as ISO standard ISO2879:1986.

SGML is reckoned to be a complex and comprehensive language (the specification extends 500 pages). However, the success of HTML (Hyper Text Markup Language) proved that the concepts of SGML were appropriate. SGML-based HTML was developed by Tim Berners-Lee in Geneva, in the early 90s in order to illustrate and link documents in the Internet. Meanwhile, HTML developed as the most successful format for all electronical documents. The Internet was originally designed as a space for human-human and human-machine communication but lately machine-machine communication has gained tremendous importance, putting a completely new challenge on the computer languages used.

HTML is a descriptive language for the presentation of documents. The main focus is on the presentation, meaning that an HTML-document mixes the presented data and its formatting instruction. A human being may recognize the displayed semantic by means of the presentation and the context meaning; a machine or (better-said) software is unable to.

In 1996 a team under the guidance of Jos Bosak attending the W3C-consortium was established to make SGML web-suitable. The result was a 30-page specification, which received in February 1998 the status of a "W3C-recommendation" and was named "Extensible Markup Language (XML)".

The most important goals developing XML were:

- XML should be compatible with SGML
XML should be easy to use in the Internet
- The number of optional characteristics should be minimized
- XML-documents should be easy to generate and human-readable
- XML should be supported by a variety of application
- It should be easy to write programs for XML
- XML should be put into practice on time

In the terminology of markup languages, a description formulated in XML is called a XML-document, albeit the content has nothing to do with text processing.

Why is this book not an XML document?

If you have accepted the ideas presented in this chapter, the question is very pertinent. The simple answer is that we have been unable to find the technology to support the creation of an open text book in XML. We need several pieces of technology

- An XML language for describing a book. DocBook is such a language, but the structure of a book is quite complex, and DocBook (reflecting this complexity) cannot be quickly mastered
- A Wiki that works with a language such as DocBook
- A XML stylesheet that converts XML into HTML for displaying the book's content

There is a project to create WikiML (Wiki MarkupLanguage), and this might be used at some point.

References

Initiating author Richard T. Watson, University of Georgia