0301402 INTRODUCTION TO XML

UNIT	MODULES	WEIGHTAGE
1	Introduction to XML	20 %
2	Document Type Definition (DTD)	20 %
3	XML Namespace	20 %
4	XML Schema	20 %
5	Extensible StyleSheet Language (XSL)	20 %

TEXT BOOK

- XML & Related Technologies
  - Publisher : - Pearson
  - Author : - Atul Kahate

- XML Related Technologies and Programming with JAVA
  - Publisher : - PHI
  - Author : - IBM

UNIT -1 Introduction to XML

- Need of XML
- XML Terminology
- XML Standards
- Basic Structure
- The Idea of Markup
- Organizing Information in XML
- Creating Well – formed XML Document
- XML Declaration
- XML Namig Rules

UNIT -1 Introduction to XML

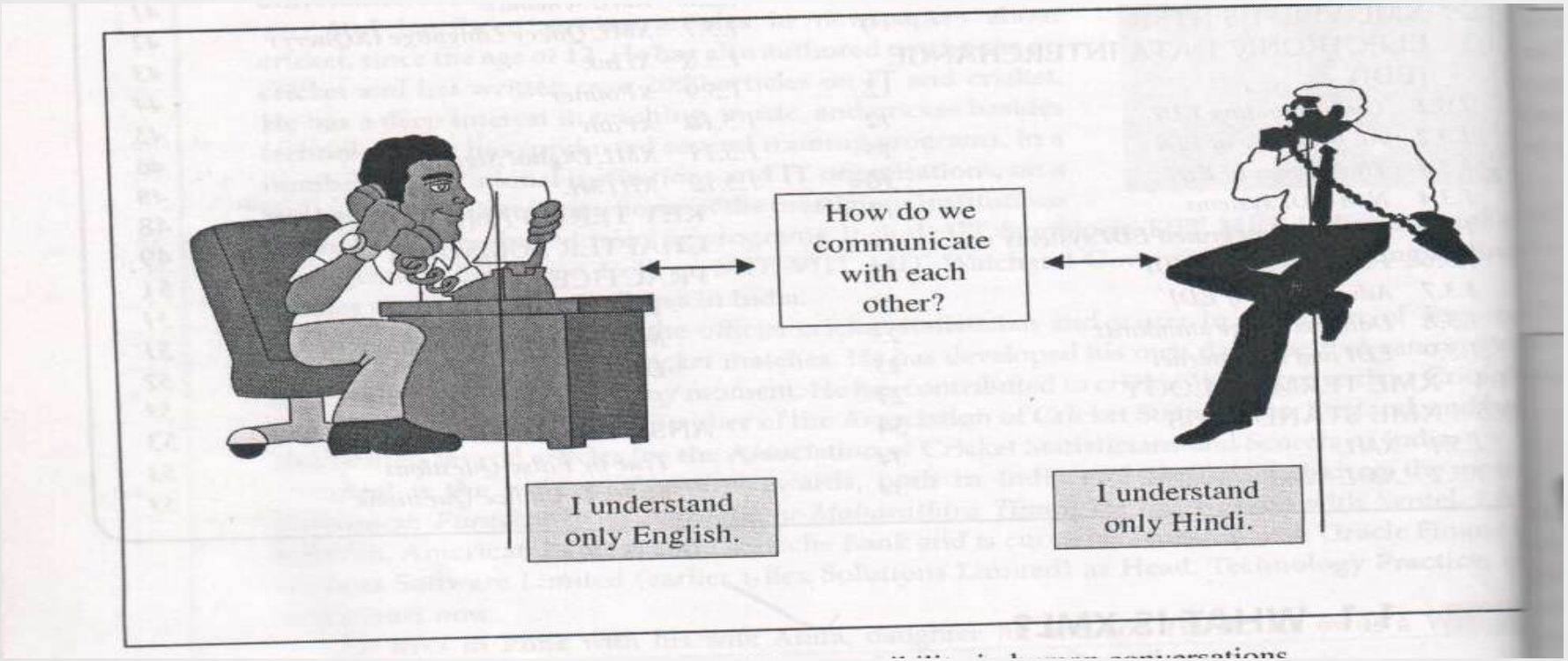
- Element Tag – Rules
- Element Attibutes – Rules
- Element Content
- Comments
- Well – Formed versus Valid
- HTML versus XML

Need of XML (What is XML?)

- **Extensible Markup Language**
- Unlike Programming platform, it is **not easy to imagine the end use and applications of XML**.
- XML syntax and semantic are well known, but w**here to use it is usually not clear.**

Need of XML (What is XML?)

The Problem of incompatibility in humman conversations



Need of XML (What is XML?)

Approach 1: Use of a traslator to solvve the problem of incompatibility

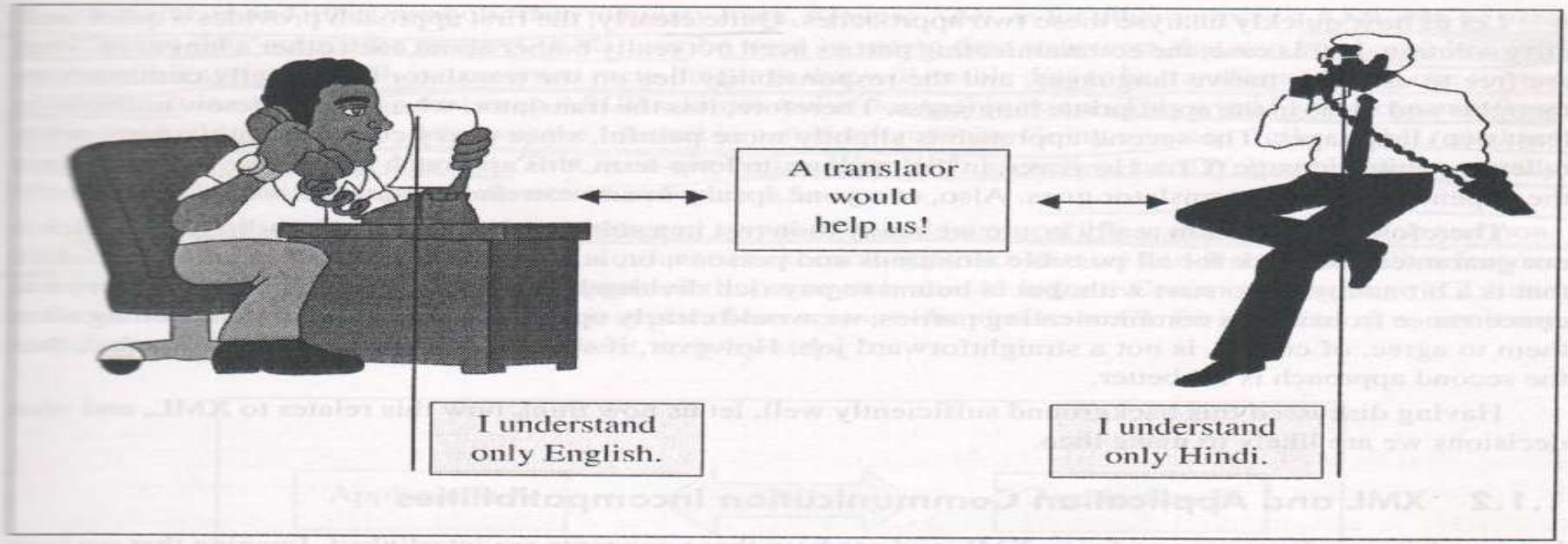


Figure 1.2 Approach 1: Use of a translator to solve the problem of incompatibility

2. Think about a Common Language (let us call this as CL for the sake of brevity) that both the persons

Need of XML (What is XML?)

Approach 2 : Making the communicating parties use a Common Language (CL)

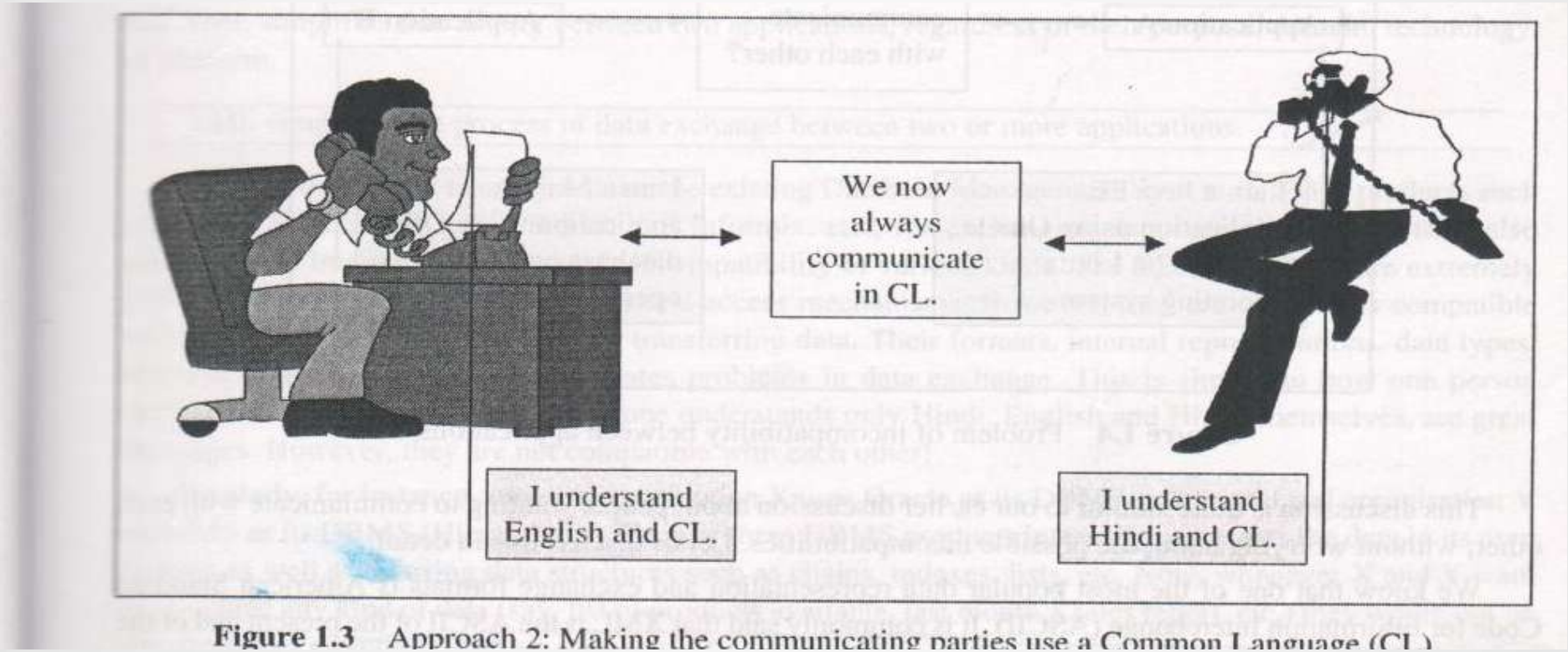
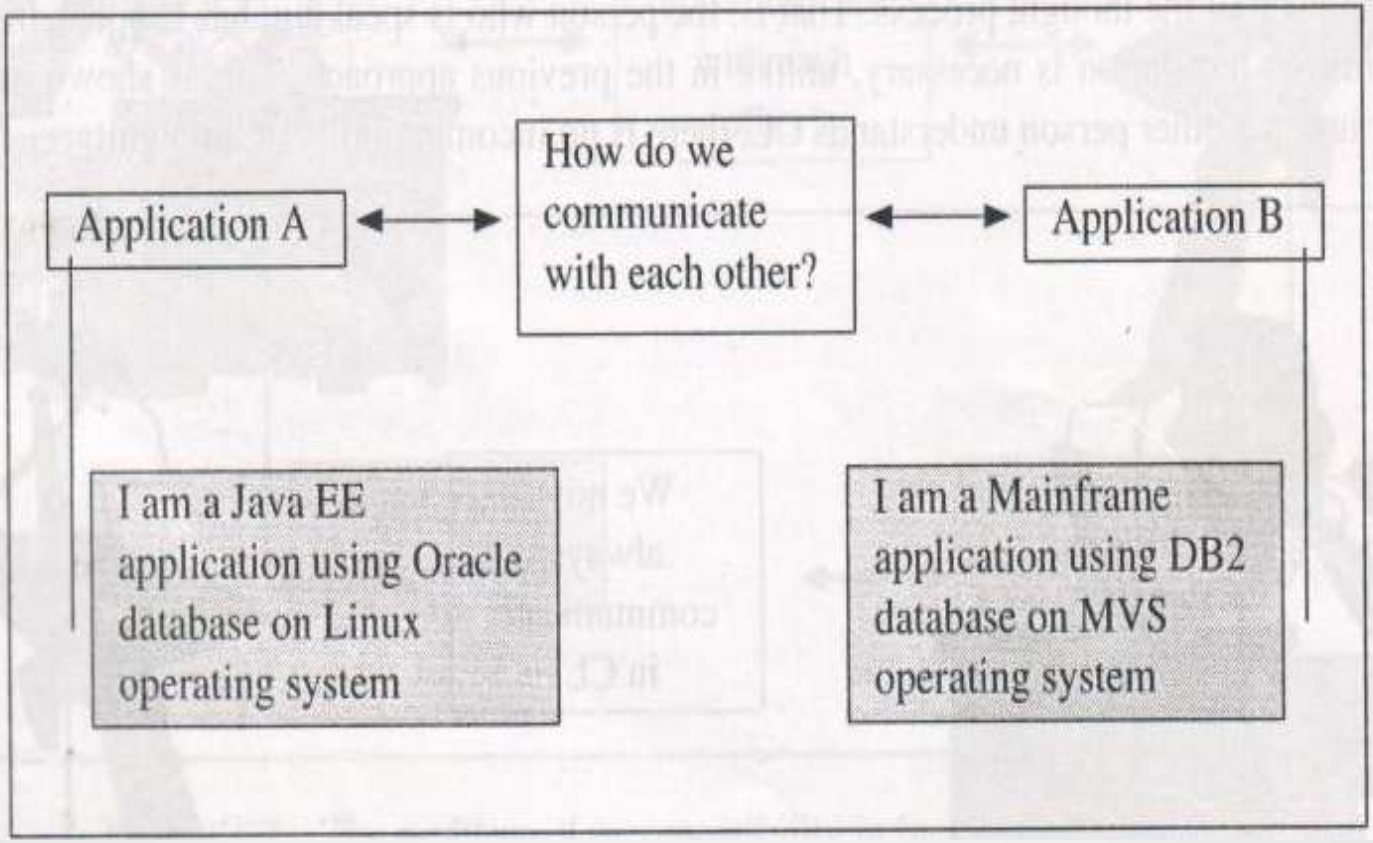


Figure 1.3 Approach 2: Making the communicating parties use a Common Language (CL)

Need of XML (What is XML?)

Problem of Incompatibility between Applications



Need of XML (What is XML?)

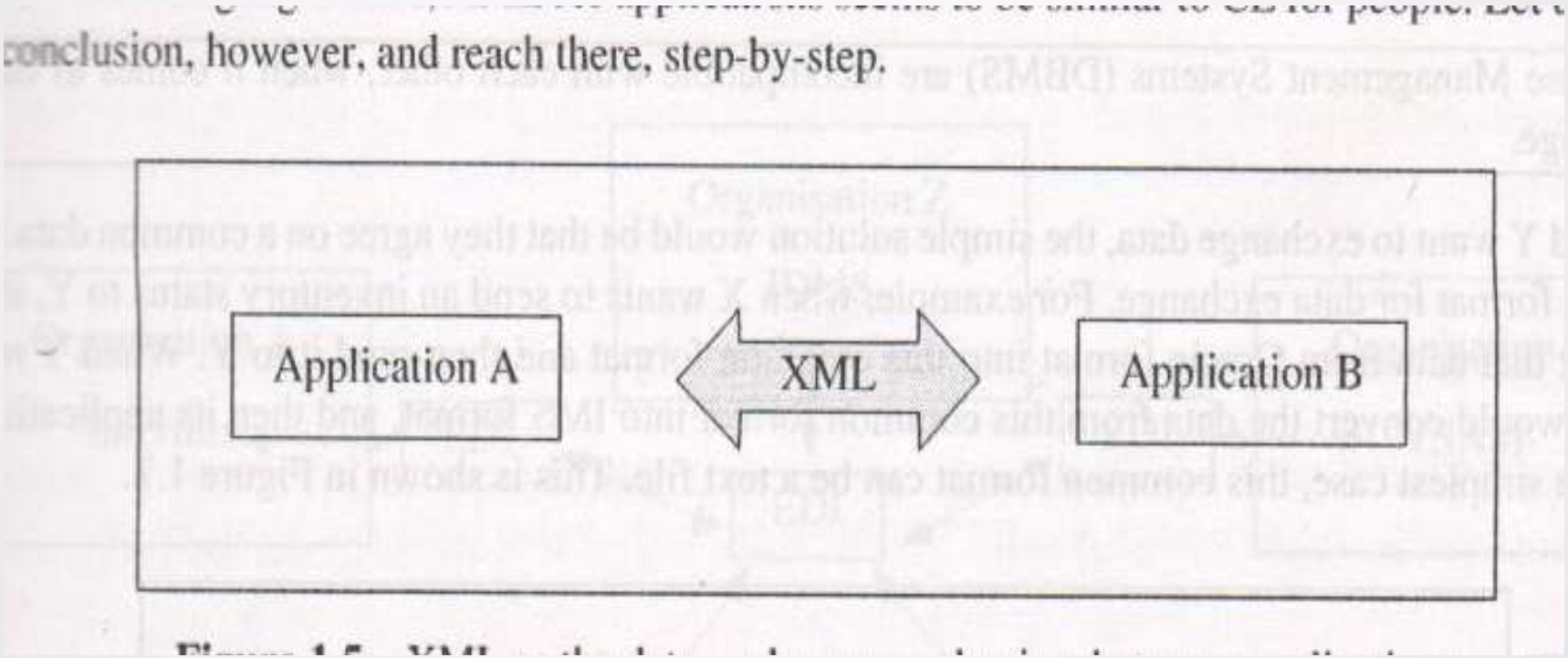
- XML can be **used to exchange data across the Internet**.
- XML can be used to **create data structures that can be shared between incompatible system**.
- XML is a **common meta-language that will enable data to be transformed from one format to another**.
- This would allow organisations and individuals to **exchange data over the Internet in a uniform manner.**

Need of XML (What is XML?)

- XML can be **used for web as well as non – web applications.**
- XML can be used to **exchange data between comapatible and incompatible applications in Web and non-Web applications.**
- **XML simlifies the process of data exchange between two or more applications.**

Need of XML (What is XML?)

XML as the data exchangemechanism between applications



Need of XML (What is XML?)

- Database Management System (DBMS) are incompatible with each other, when it comes to data exchange.

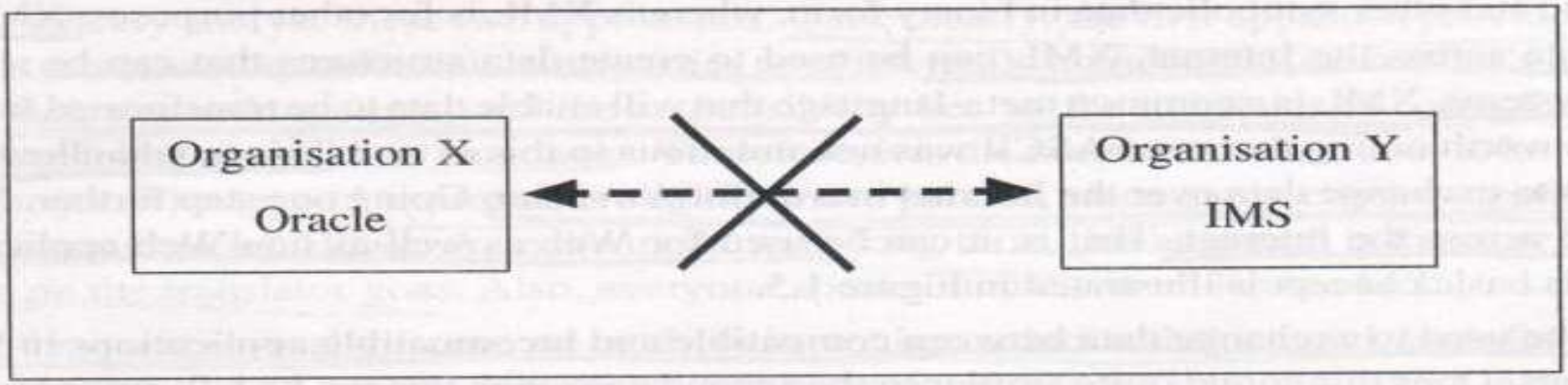
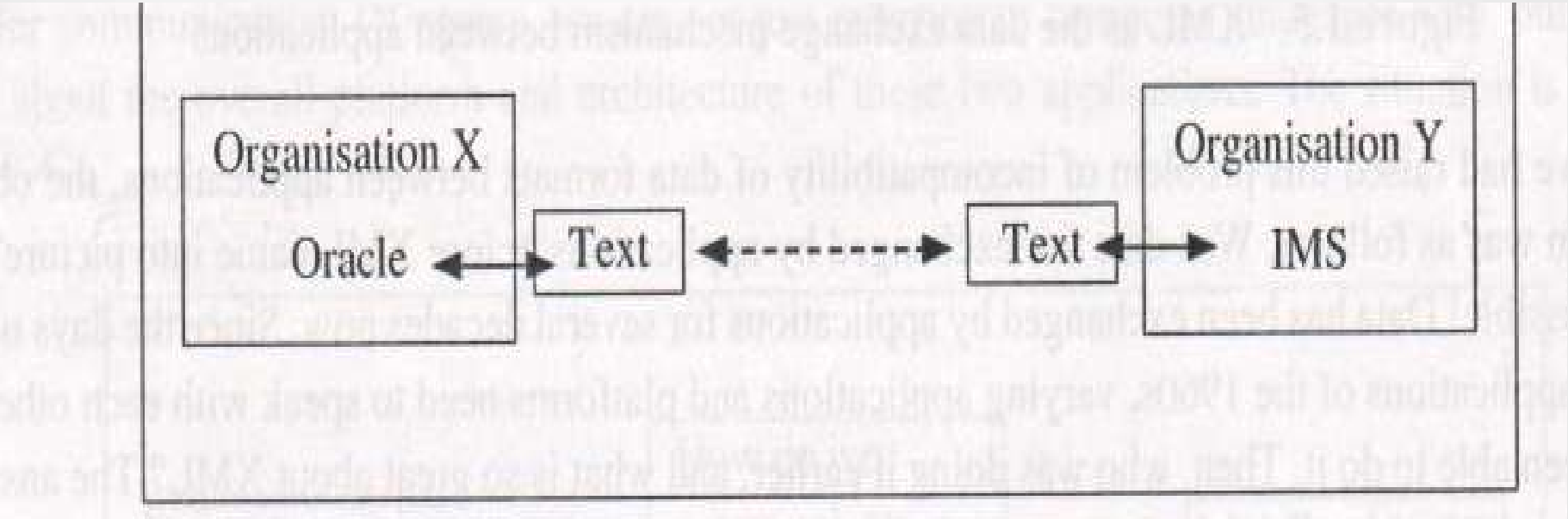


Figure 1.6 Incompatible data formats

Need of XML (What is XML?)

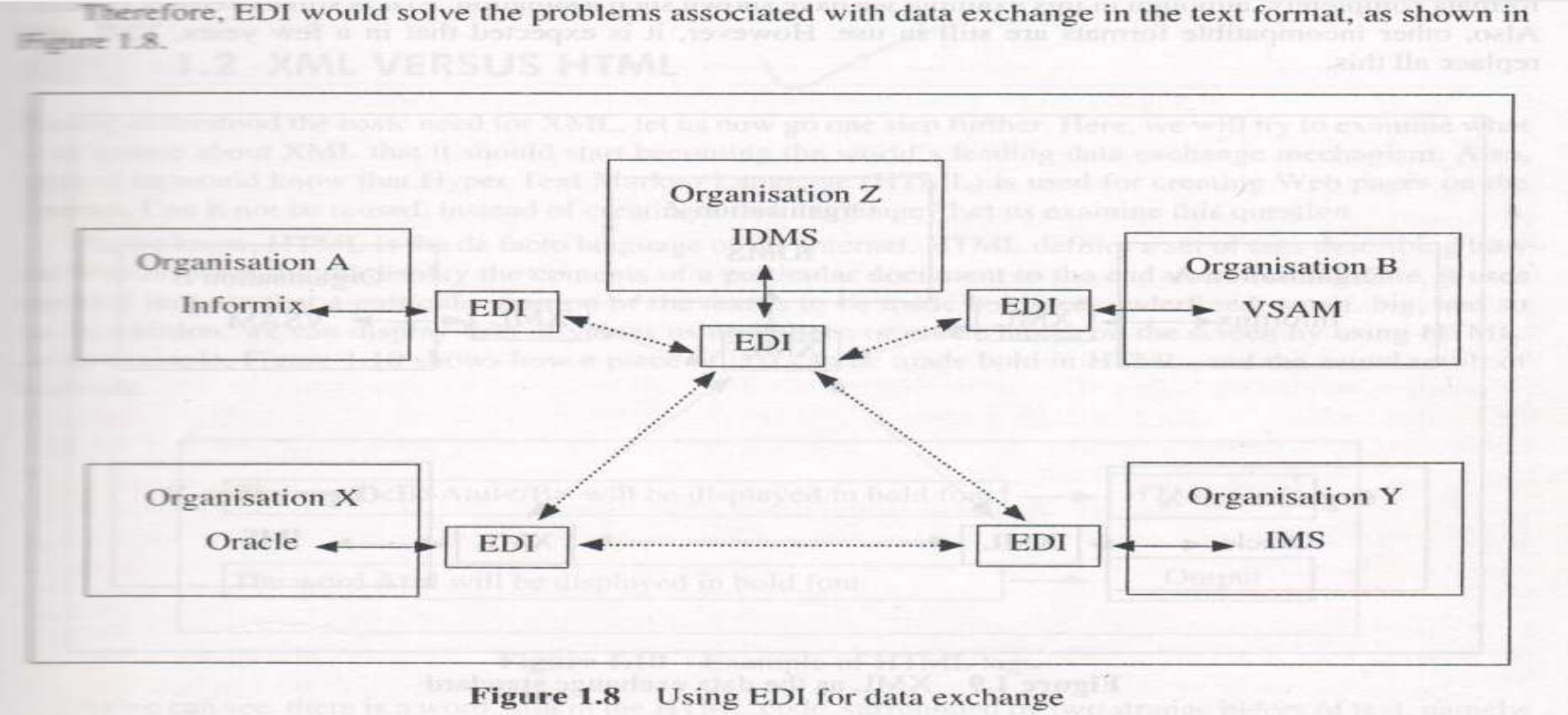
- Solution for data Exchnage between Database Management System (DBMS)



Data Exchange in a text Format

Need of XML (What is XML?)

- Solution for data Exchnage between Database Management System (DBMS)



Data Exchange using EDI (Electronic Data Interchange)

Need of XML (What is XML?)

- Solution for data Exchnage between Database Management System (DBMS)

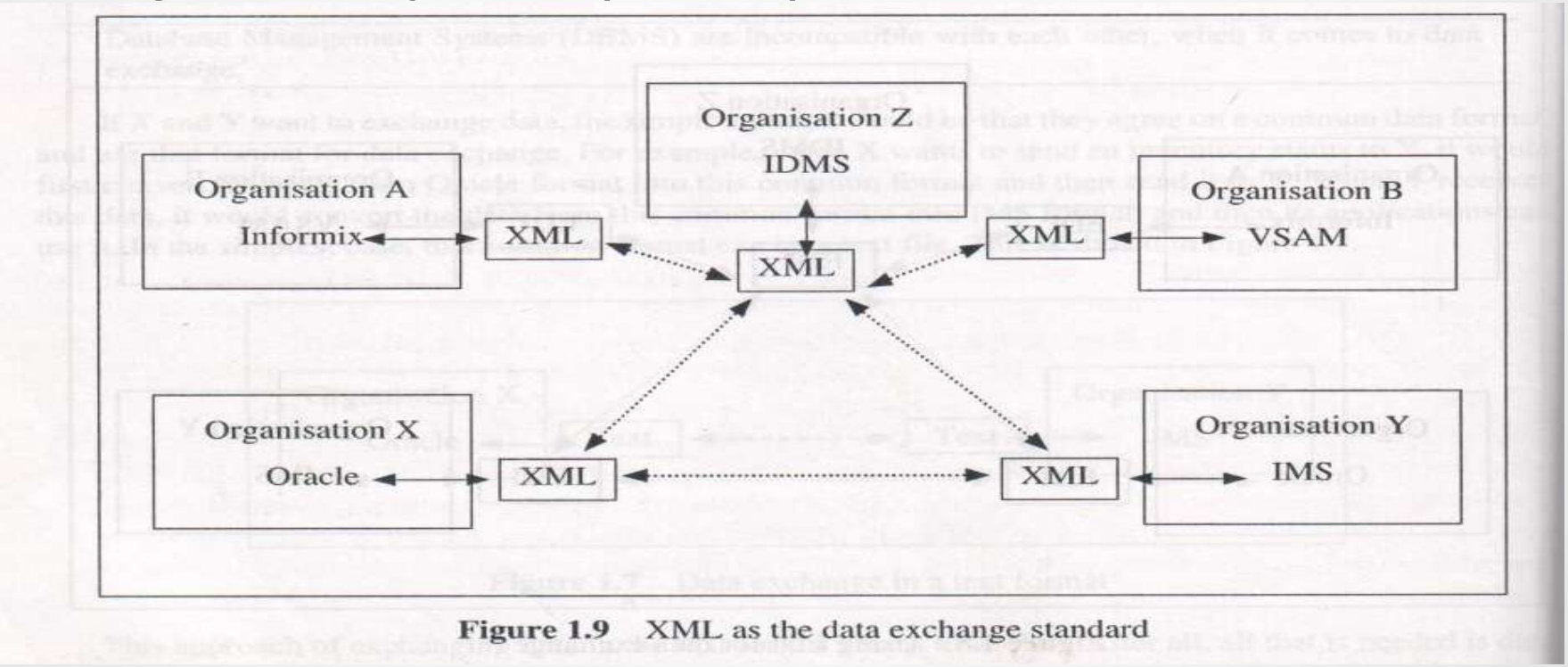


Figure 1.9 XML as the data exchange standard

Data Exchange using XML

XML Versus HTML

HTML

- HTML defines a set of tags describing how the web browser should display the content of a document.
- HTML is an information presentation language.
- HTML focuses on display of DATA

XML

- XML used tag to organise documents and the contents there in.
- XML is an information description language
- XML focuses on representation of DATA

XML Versus HTML

HTML	XML
• HTML can not process, sort, encrypting data .	• XML can process, sort, encrypting data.
• HTML formate would convey nothing about meaning of the data.	• XML describes the meaning of the data.

XML Versus HTML

The Role of HTML and XML

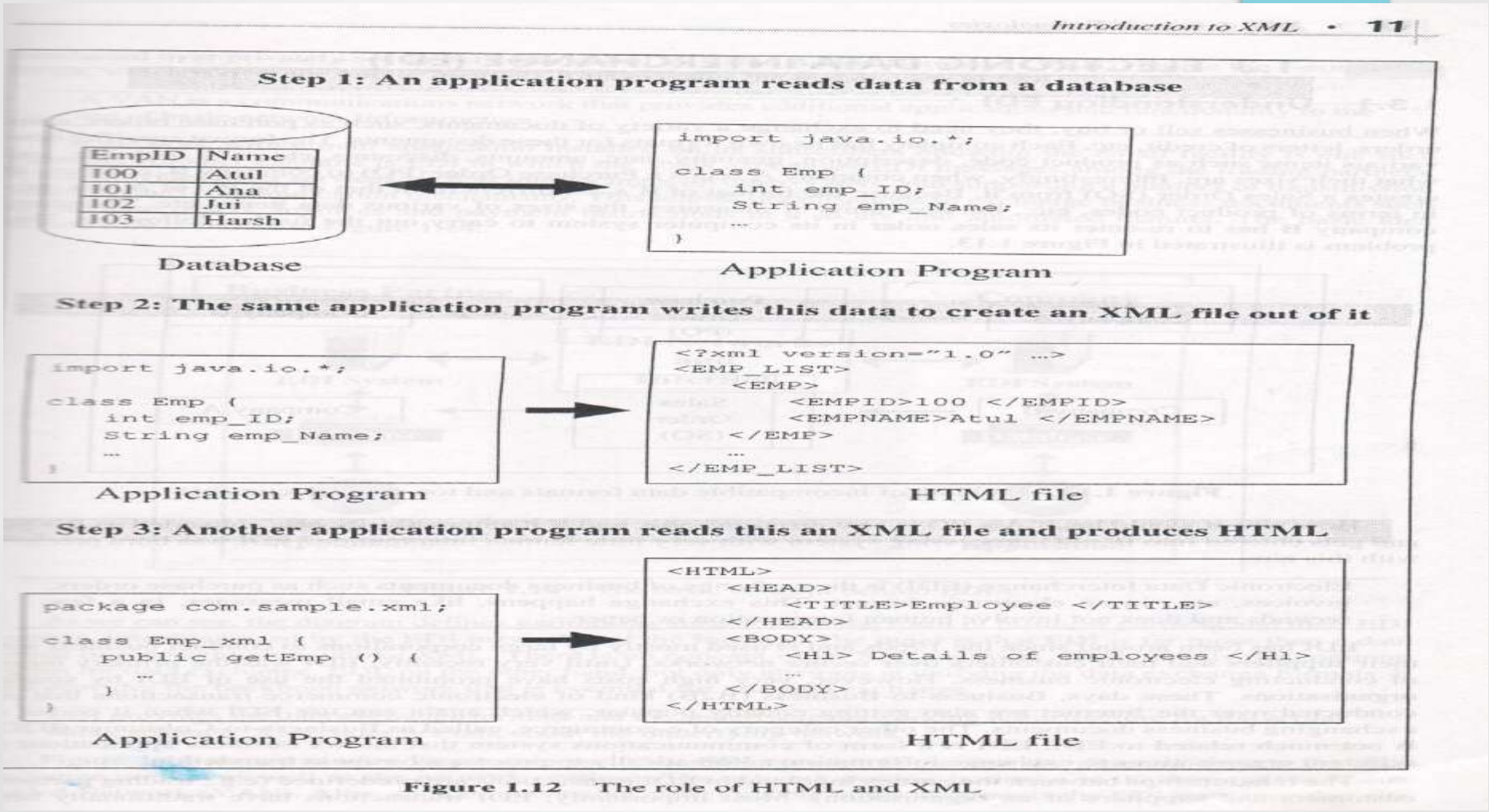


Figure 1.12 The role of HTML and XML

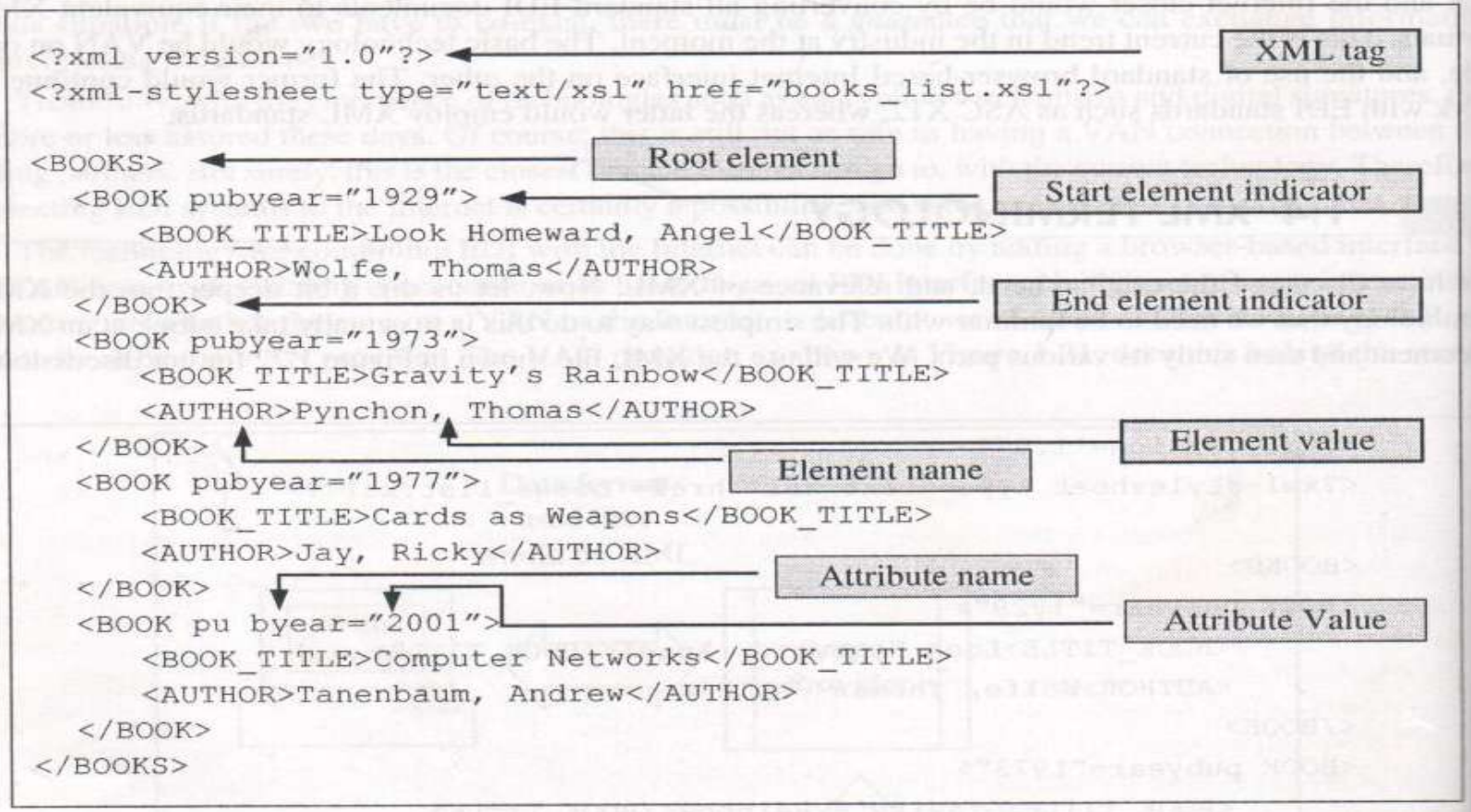
XML Terminology

- Every XML file has an extension of .xml
- Demo 1: Book.xml

XML Terminology

Terminology in XML

Figure 1.23 shows a short pictorial explanation of this XML document. A detailed explanation is provided in Table 1.1.



XML Terminology

Contentsof the XML File	Description
<?xml version="1.0" encoding="UTF-8"?>	This line idetifies that files is an XML document. Every XML document must begin with this line.
<BOOKS>	This line implicitly indicates the start of the actual contents in the XML file.
<BOOK Pubyear='1973'>	Pubyear='1973' is called as an attribute in XML. An Attribute serves the purpose of providing more information about an element.
<BOOK_TITLE>	This is another element declaration. Information in a hierachical manner comes into picture.
</BOOK>	This declaration indicates the end of the first <BOOK> element.

XML Terminology

Terminology in XML

Exercise 3: Suppose we want to store information regarding employees in the following format in XML. Show such a file with one example:

Employee ID	Numeric	5 positions
Employee Name	Alphanumeric	30 positions
Employee Department	Alphanumeric	2 positions
Role	Alphanumeric	20 positions
Manager	Alphanumeric	30 positions

Solution 3:

```
<?xml version="1.0"?>
```

XML Terminology

- **EXAMPLE**
- Suppose we want to sotre information regarding employees in the following format in XML. Show such a file with example

Employee Id	Numeric
Employee Name	Alphanumeric
Employee Department	Alphanumeric
Role	Alphanumeric
Manager	Alphanumeric

- Demo – 3 Employee.XML

XML STANDARDS

- The World Wide Consortium (W3C) has developed a number of standards for XML.
- Standards are :
  - XML
  - XML Namespaces
  - Document Type Definitions (DTD)
  - Cascading Style Sheets (CSS)
  - Extensible Stylesheet Language (XSL)
  - XML Schemas

XML STANDARDS

- Standards are :
  - XML Query Language (Xquery)
  - XLink
  - Xpointer
  - XPath
  - XML Digital Signatures
  - XHTML

XML STANDARDS - XML

- The first version of XML Standard was 1.0 finalised in feb, 1998.
- Which are as follow:
  - XML shall be **easily implemented** on the Internet
  - XML shall be **usable in a wide variety of applications**
  - XML shall **comply with SGML** (Standard Generalized Marked up Language)
  - XML documents **shall be readable to humans**
  - XML document **design shall be formal and compact**
  - XML document **design shall be done quickly**
  - XML documents shall be easy to create
  - XML markup is not required to be abrupt

XML STANDARDS –

- **XML Namespaces**
  - To avoid confusing situation in terms of tags duplications, Namespace help us resolve conflict.

- **Document Type Definitions (DTD)**
  - A DTD file allows us to specify the rules associated with an XML file.

XML STANDARDS - XMLNampespaces

- The XML Namespace allowing the users of XML to associate certain tags with their own identifiers.
- User A creating <book> tag in Book1.xml
- User B creating <book> tag in Book2.xml

- If Book1.xml and Book2.xml going to merge this would be an ambiguity.

XML STANDARDS –

- **Document Type Definitions (DTD)**

A Document Type Definition (DTD) file allows us to specify the rules associated with an XML file. Figure 1.25 shows the idea.

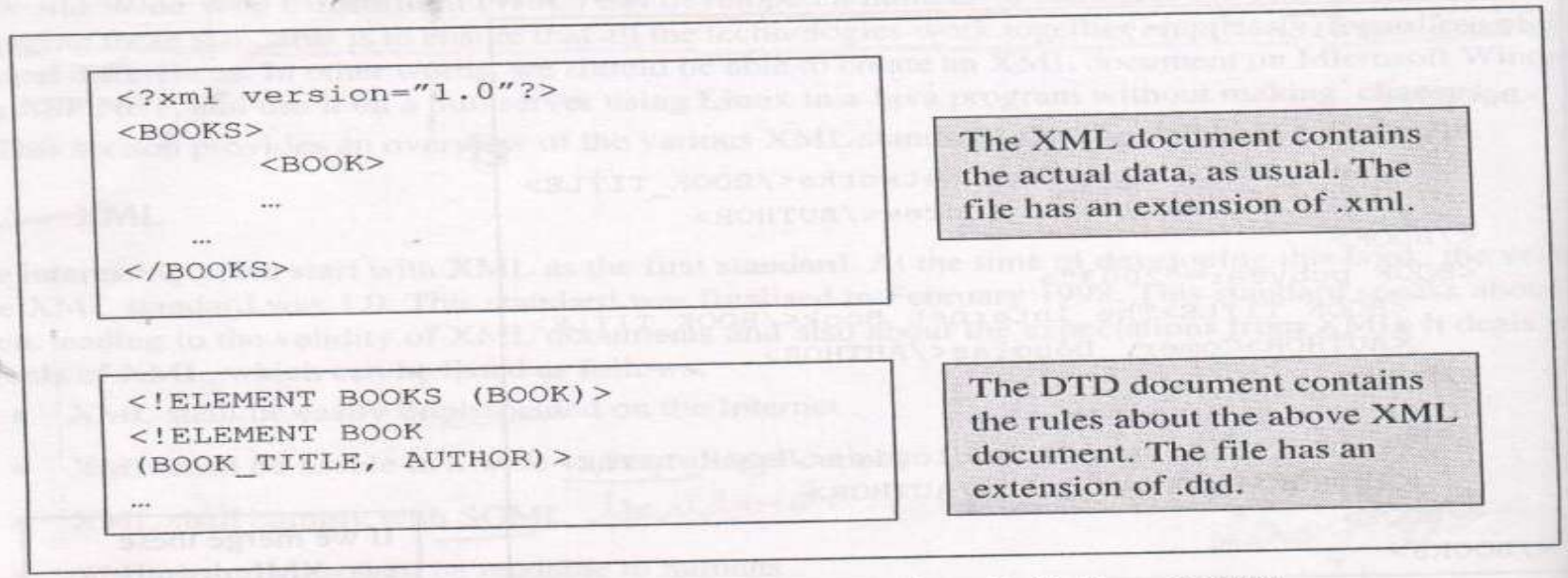


Figure 1.25 Concept of Document Type Definitions (DTD)

XML STANDARDS –

- **Document Type Definitions (DTD)**

various elements of an XML document. For example, it can help us in ensuring that ... is a sub-element of the BOOK element, but that the reverse is not possible. This is shown in Figure 1.26.

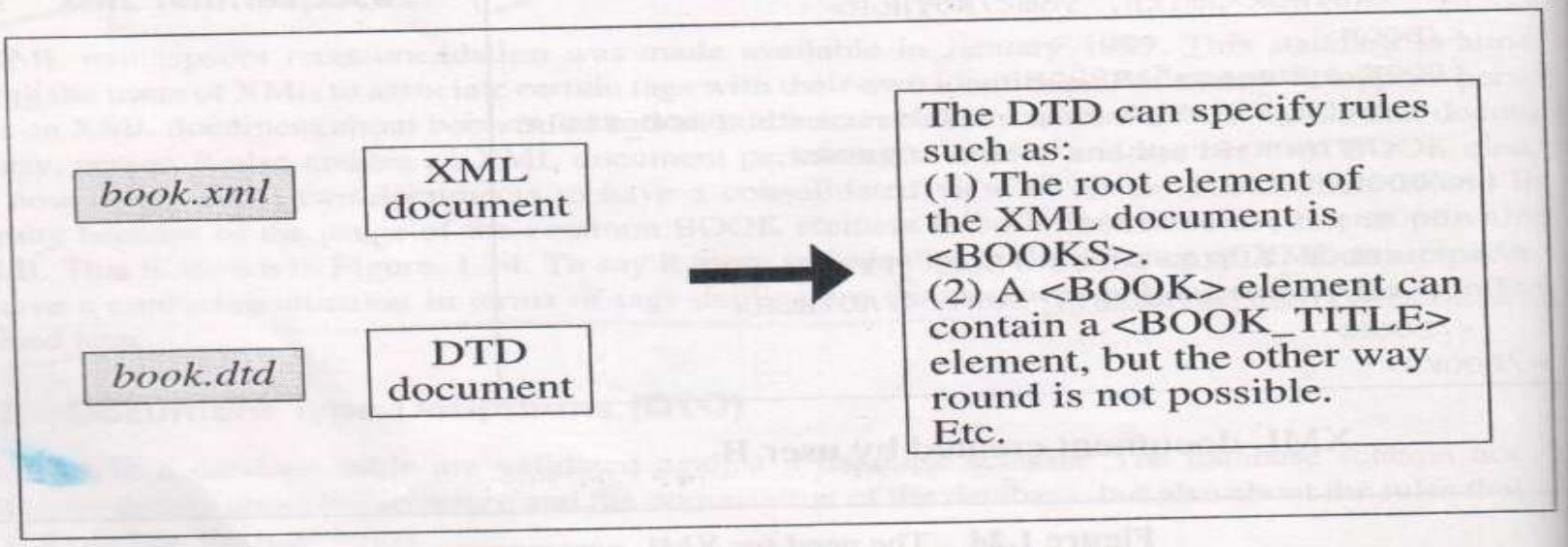


Figure 1.26 DTD concept illustrated further

XML STANDARDS –

- **Cascading Style Sheets (CSS)**
  - The **Cascading Styling Sheets technology allows the formatting of HTML documents** in a standard uniform manner.
- **Extensible Stylesheet Language (XSL)**
  - The Extensible Stylesheet Language standard provides means for formatting and **transforming XML documents in the desired manner.**
  - XSL technology consists of two parts.
    - XSL Transformation (XSLT)
    - XSL Formatting Objects (XSL - FO)

XML STANDARDS –

- **Extensible Stylesheet Language (XSL)**

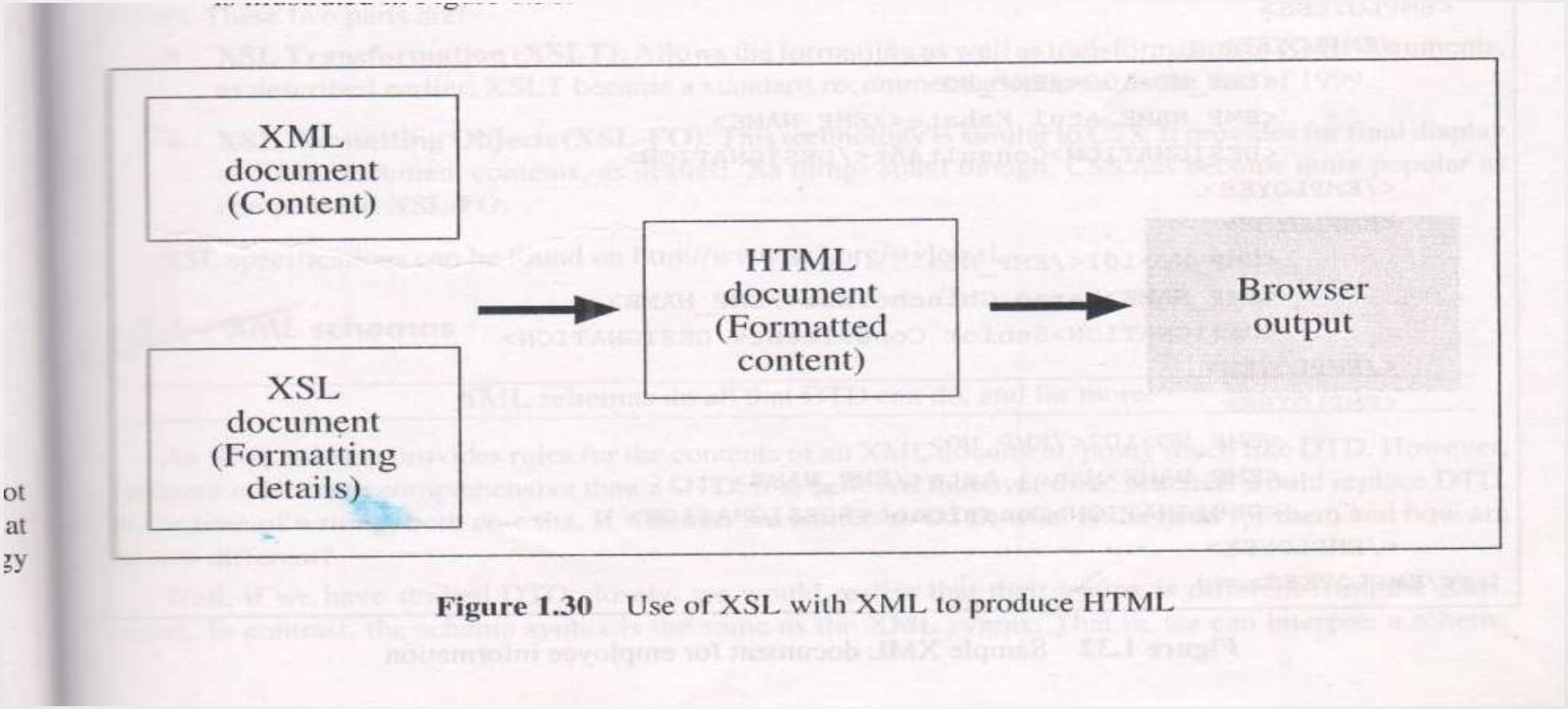


Figure 1.30 Use of XSL with XML to produce HTML

XML STANDARDS –

- **Extensible Stylesheet Language (XSL)**

concept is shown in Figure 1.31.

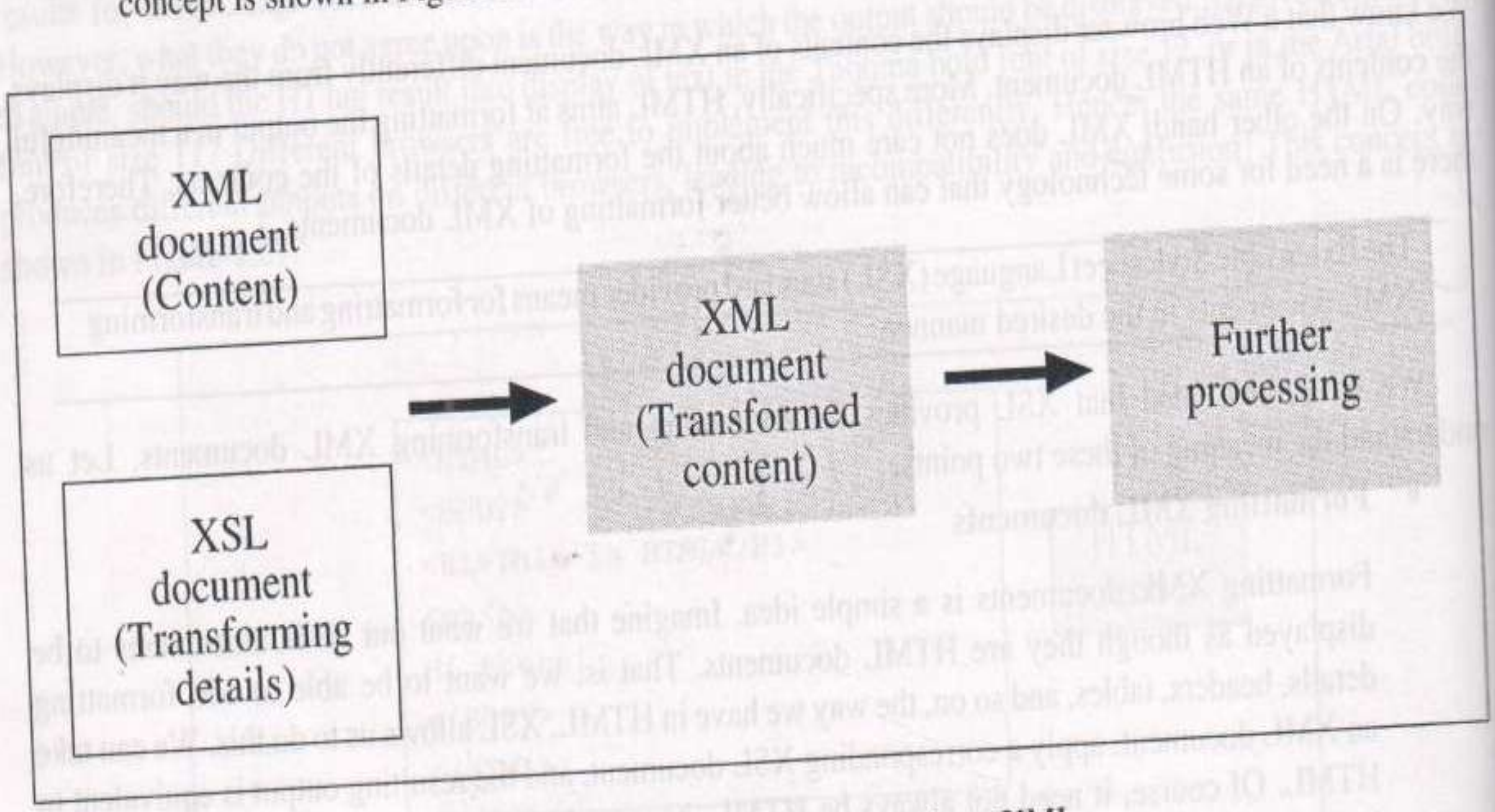


Figure 1.31 Use of XSL to transform XML

XML STANDARDS –

- **XML Schemas**
  - An XML schema provides rules for the contents of an XML document, pretty much like DTD.
  - It is far more than DTD.
- **XML Query Language (Xquery)**
  - The XML query language, also called as Xquery, is still being developed.
  - Xquery specifications can be found on
  - http:// www.w3.org/tr/xmlquery-req/.

XML STANDARDS –

- **Xlink**
  - Xlink defines a standard mechanism for **creating hyperlinks** in XML documents.
    - Xlink is a language for creating hyperlinks in XML doc.
    - Xlink is similar to HTML links.
    - Any elements in an XML document can behave as an Xlink
    - Xlink supports two types of links
      - Simple link
      - Extended links

XML STANDARDS –

- **Xpointer**
  - Xpointer allows the **hyperlinks to point to more specific parts in the XML** documents.
- **Xpath**
  - Xpath is used **to refer to specific portions of an XML document using XSLT and Xpointer.**
- **XML Digital Signatures**
  - XML digital signature **provide a means of message integirty** and non- repudiation for XML documents.
  -

XML STANDARDS –

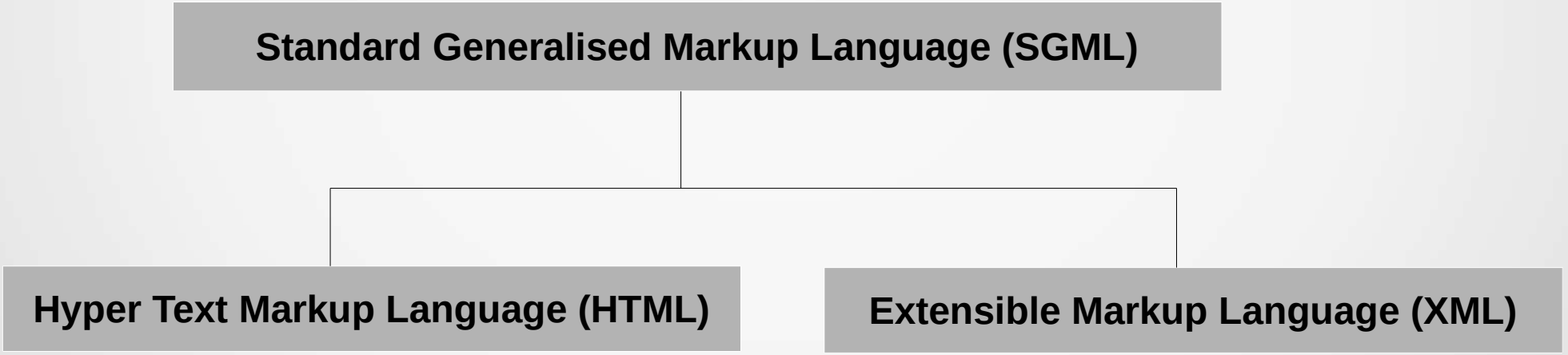
- **XHTML**
  - It is a revised version of HTML with rules from XML.

The Idea of Markup

- In XML, tag containg values. This **tags specify certain ruels.**
- Therefore, we can say that **XML is nothing but a set of rules.**
- XML has been written in s**uch a manner that it can be extended easily, depending on the business domain, particular sets of requirements or technology.**

The Idea of Markup

- XML is based on yet another language, called as Standard Generalised Markup Language (SGML).
- SGML is the parent of almost all important modern markup languages.



The Idea of Markup

- **The following features of Values:**
  - Easy to read for humans
  - Easy to use
  - Easy to Use for Computer
  - Easy to debug
  - Easy to modify suitably for any industry or domain
  - Works with all leading programming languages, database and formats such as spredsheets and drawing.

Organising Information in XML

- Desiging an XML document is similar to designing a database table.
- This process break down into three steps:
  - **Classifiying information as per its importance**
  - **Adding the details**
  - **Transforming information into XML format**
    - **Identifying elements**
    - **Identifying attributes**

Organising Information in XML

- **Classifiying information as per its importance for BOOK**
  - Titler
  - Author
  - Publication
  - Price
  - Publishing Year
  - Reprint number
  - Edition number
  - Book Website

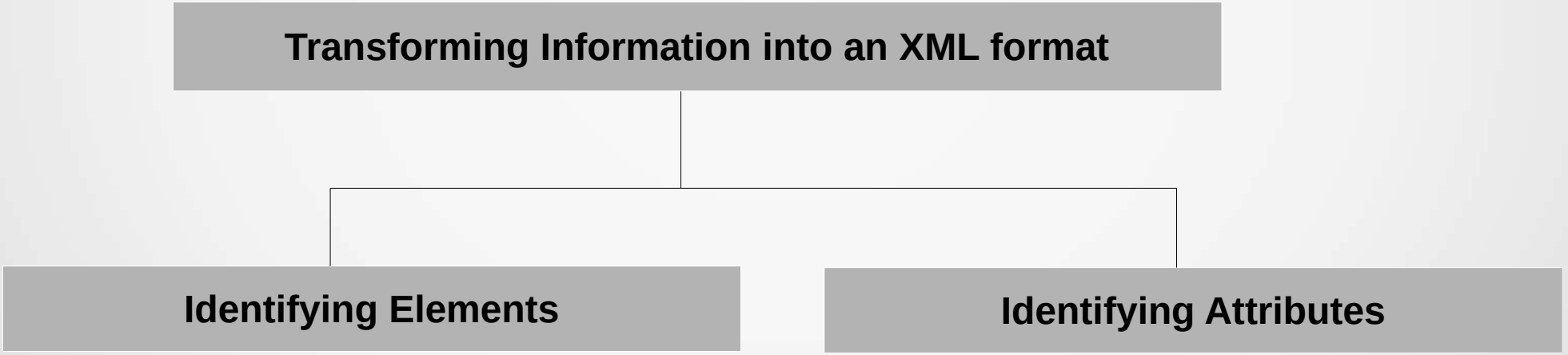
Organising Information in XML

- **Adding the details**

Primary Info	Details we want to capture	Details we can ignore
Title	Main Title Sub - Title	-
Autor	First Name Last Name	Full Name Affiliations
Publication	Name of Publisher	Full Address
Price	In local Currency	In more Currency
Edition	Number	-
Book Website	URL	-

Organising Information in XML

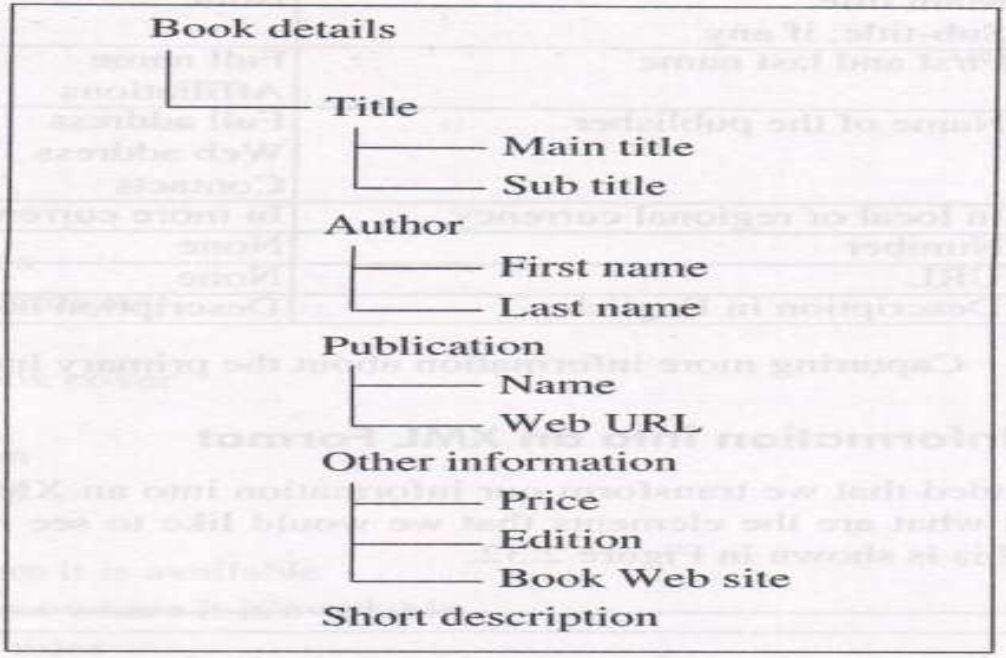
- **Transforming information into XML format**
  - **Identifying elements**
  - **Identifying attributes**



Organising Information in XML

- **Identifying Elements**

Figure 2.33 depicts the hierarchy of elements for our books example.



Organising Information in XML

- **Identifying Elements**

is to transform the visual form of the hierarchy into an XML-like syntax. The resulting structure Figure 2.34.

```
<BOOK_DETAILS>
    <TITLE>
        <MAIN_TITLE> </MAIN_TITLE>
        <SUB_TITLE> </SUB_TITLE>
    </TITLE>
    <AUTHOR>
        <FIRST_NAME> </FIRST_NAME>
        <LAST_NAME> </LAST_NAME>
    </AUTHOR>
    <PUBLICATION>
        <NAME> </NAME>
        <WEB_URL> </WEB_URL>
    </PUBLICATION>
    <OTHER_INFO>
        <PRICE> </PRICE>
        <EDITION> </EDITION>
        <BOOK_WEB_SITE> </BOOK_WEB_SITE>
    </OTHER_INFO>
    <SHORT_DESC> </SHORT_DESC>
</BOOK_DETAILS>
```

Organising Information in XML

- **Identifying Attributes**

ements, for instance, that of the TITLE. This approach is illustrated in Figure 2.38.

```
<BOOK_DETAILS>
    <TITLE category = "…">
        <MAIN_TITLE> </MAIN_TITLE>
        <SUB_TITLE> </SUB_TITLE>
    </TITLE>
    <AUTHOR>
        <FIRST_NAME> </FIRST_NAME>
        <LAST_NAME> </LAST_NAME>
    </AUTHOR>
    <PUBLICATION>
        <NAME> </NAME>
        <WEB_URL> </WEB_URL>
    </PUBLICATION>
    <OTHER_INFO>
        <PRICE> </PRICE>
        <EDITION> </EDITION>
        <BOOK_WEB_SITE> </BOOK_WEB_SITE>
    </OTHER_INFO>
    <SHORT_DESC> </SHORT_DESC>
</BOOK_DETAILS>
```

Figure 2.38 Adding the category attribute to a book

Creating Well – Formed XML Documents

- **The <?xml> tag**
  - This tag identifies **our document as an XML Document.**
  - It **must be first line** of the Document.
  - It **specifies the version of the XML** specifications it is following.
  - It also **specifies the encodeing.**

    <?xml version="1.0" encoding="UTF-8"?>
  - The Character encoding allows us to specify the language based on the ISO standards or Unicode standards, which use to creat markup and contents of the documents.

Creating Well – Formed XML Documents

- **The root Element**
  - XML document must have exactly one root element.
  - Root element must be the first element immediately after the <?XML> tag

```
<?xml version="1.0" encoding="UTF-8"?>
<BOOKS>
    rest of the xml document
</BOOKS>
```

Creating Well – Formed XML Documents

- **Opening and Closing Tags (Element Tag Rules>**
  - All elements have an opening tag. Optionally, element also have a closing tag.

<BOOK Pubyear='1973'>

<BOOK_TITLE> LOOK Homeward </BOOK_TITLE>

<AUTHOR> Wolfe, Thomas </AUTHOR>

</BOOK>

Creating Well – Formed XML Documents

- **Empty Elements**
  - Empty Elements in XML can be represented in two ways.

(1) We can either use the tag pair <> and </> containing the element name to depict this, without content in between.

<Book_Name></Book_Name>

(2) We can just use the single tag </>

<Book_Name/>

Creating Well – Formed XML Documents

- **Empty Elements**
  - Show the customer name including the first and the last name, but the middle name should be empty.

<Name>

<First> Atul </First>

<Middle></Middle>

<Last> Patel </Last>

</Name>

Creating Well – Formed XML Documents

- **Opening and Closing Tags**
  - All elements have an opening tag. Optionally, element also have a closing tag.

<BOOK Pubyear='1973'>

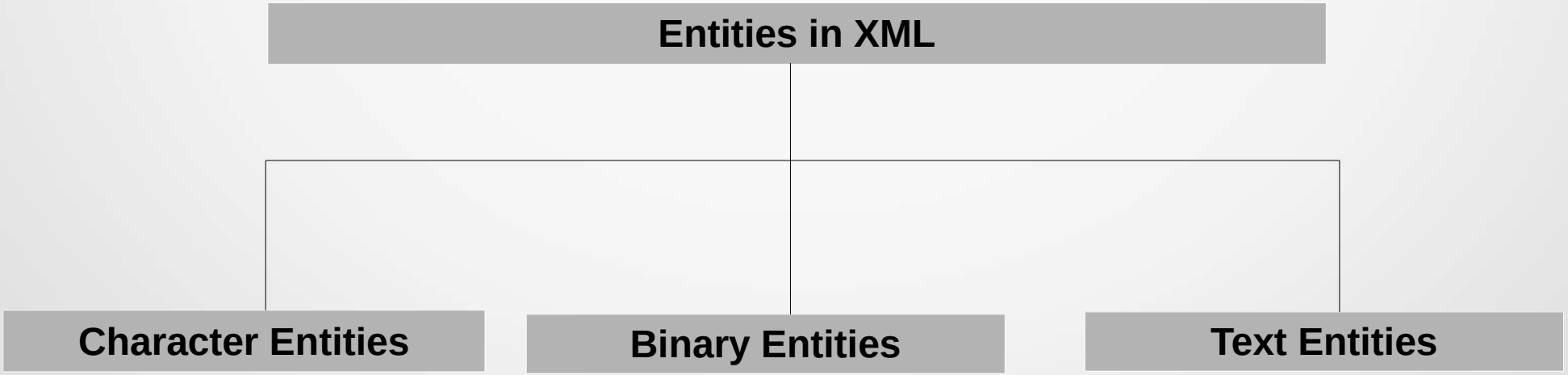
<BOOK_TITLE> LOOK Homeward </BOOK_TITLE>

<AUTHOR> Wolfe, Thomas </AUTHOR>

</BOOK>

Creating Well – Formed XML Documents

- **Entities**
  - An Entity inXML represents a text that you want to use repeatedly without having to write it every time.
  - We define it at one place, and refer to it from other place.
  - There are three types of entities in XML:



Creating Well – Formed XML Documents

- **Character Entities**
  - Character entity references are special character code that assign a different meaning to a special symbol.

Character Entity	Meaning
&amp;	& Character
&apos;	' Character
&gt;	> Character
&lt;	< Character
&quot;	“ Character

Creating Well – Formed XML Documents

- **Character Entities**
  - Character entity references are special character code that assign a different meaning to a special symbol.
  - **Contents in XML:**

    Please make sure that your offer is >$500
  - **Interpretation :**

    Please make sure that your offer is > $500

Creating Well – Formed XML Documents

- **Text Entities**
  - Text entities are used to associate large or repeated blocks of text with a name and replace the text with the entity name.
  - Declaring Syntax

    <!ENTITY name “content”>

    <!ENTITY country “INDIA”>

  - Demo4.xml

Creating Well – Formed XML Documents

- **Binary Entities**
  - Binary entities are used to associate a name with binary data (such as an image or a video) and use the entity name insted of the actual binary data.

<!ENTITY city SYSTEM “delhi.html” NDATA html>

Creating Well – Formed XML Documents

- **Element Nameing**
  - Should contain at least one latter : a-z or A-Z
  - Can start with an alphabet or an underscore
  - Can contain latters, digits, hypens, underscores, full stops
  - XML names are case sensitive.
  - Names cannot contain spces
  - Name cannot beused any prefix
  - i.e

    <Name05>

    <Name.05>

    <_05Name>

Creating Well – Formed XML Documents

- **Nesting Conventions**
  - In XML, child elements must be nested completely inside the parent element.
- **Adding Attributes**
  - Attributes allow us to specify more information about XML elements.
  - Attributes merely provide an alternative to sub-elements
  - Attributes conist of a name=”value” pair
  - Attributes are placed in the start tag of the element.
  - An element may have several attributes, each uniquely named.
  - Attributes must have a value
  - Values must be quoted with either double or single quotes

Creating Well – Formed XML Documents

- Comments

  <!-- THIS IS COMMENT IN XML -->

Element Content

- Element content is handled in one of two ways:

(1) Parsed Character Data (PCDATA): it is examined by the XML parser to discover XML content embedded within it.

Character Entity	Meaning
&amp;	& Character
&apos;	' Character
&gt;	> Character
&lt;	< Character
&quot	“ Character

Element Content

- Element content is handled in one of two ways:

  (2) Character Data (CDATA) : CDATA is not parsed and is treated as it is. It is useful for embedding other languages within the XML as :

  - HTML documents
  - XML documents
  - JavaScript documents
  - Etc.

- Assignment Submission
  - Theory : / 12 / 2022
  - Practical : / 12 / 2022
- CEC Submission
  - Theory : / 12 / 2022
  - Practical : / 12 / 202

UNIT 1 COMPLETED