

Object-Oriented Programming Design



1. Introduction to Object-Oriented Concepts
2. How to think in Terms of Objects
3. Advanced Object-Oriented Concepts
4. The anatomy of a Class
5. Class Design guidelines
6. Designing with objects
7. Mastering Inheritance and Composition
8. Frameworks and Reuse: Designing with interfaces and Abstract classes
9. Building objects
10. Creating Object Models with UML

11. Objects and Portable Data: XML

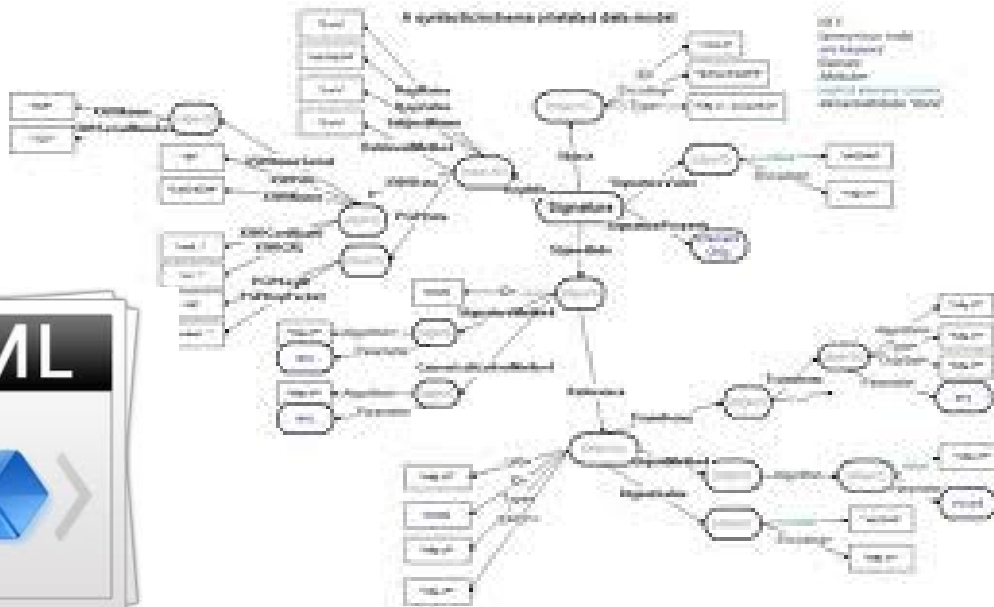
12. Persistent Objects: Serialization and Relational Databases
13. Objects and the Internet
14. Objects and Client/Server Applications
15. Design Patterns





11. Objects and Portable Data: XML

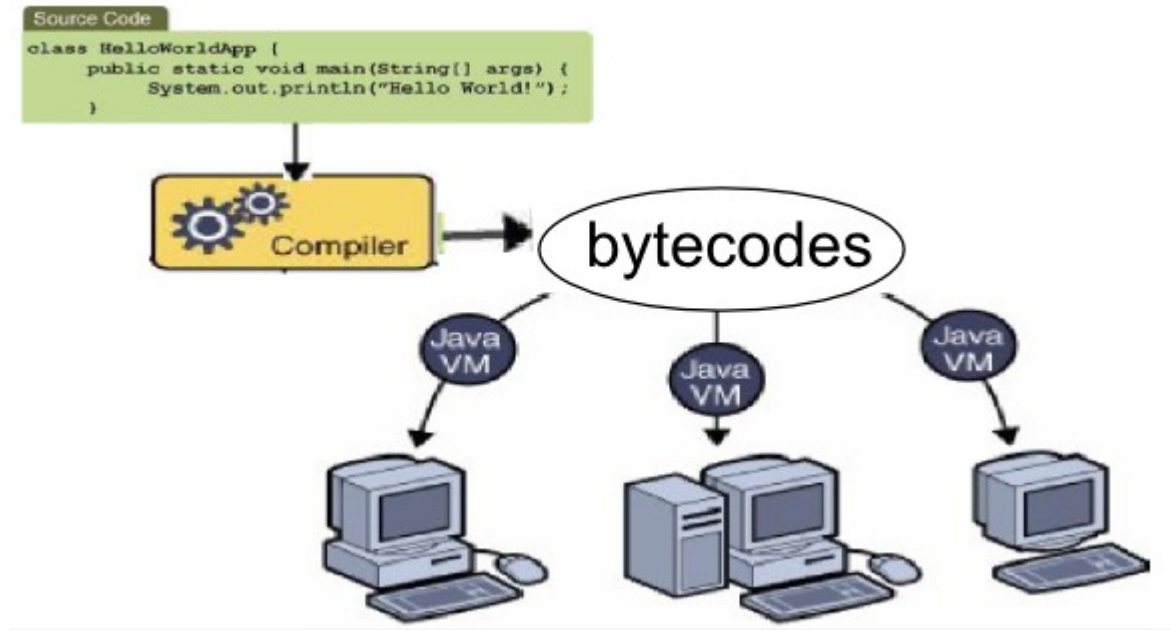
- ▶ Object-oriented technologies have made major inroads in recent years. Objects have become a major technology in the application development industry. Objects have also made major headway in the definition and movement of data as well.





11. Objects and Portable Data: XML

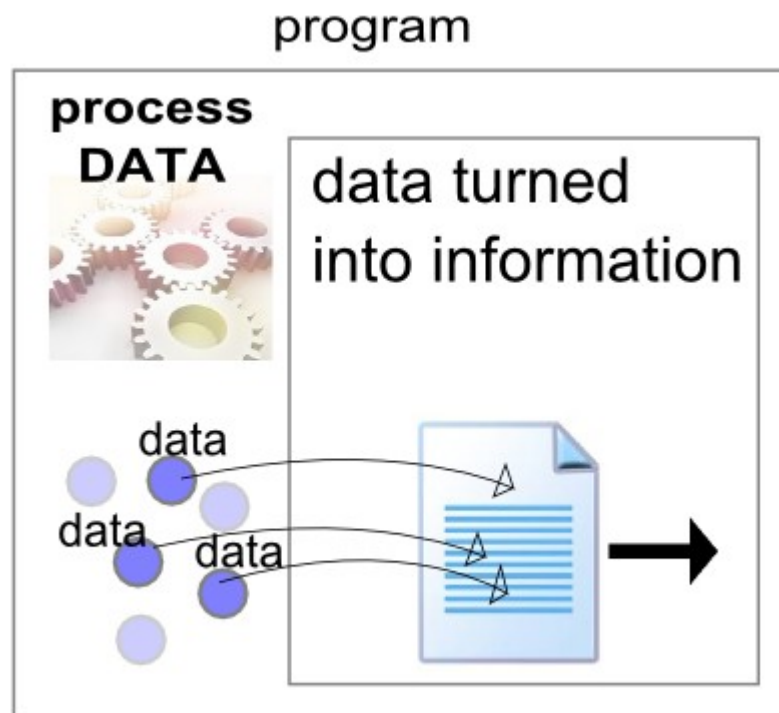
- ▶ Much of Java's success was due to the fact that it was highly portable across multiple platforms. The bytecodes produced by Java could be executed on various platforms, as long as the system has a Java virtual machine loaded.





11. Objects and Portable Data: XML

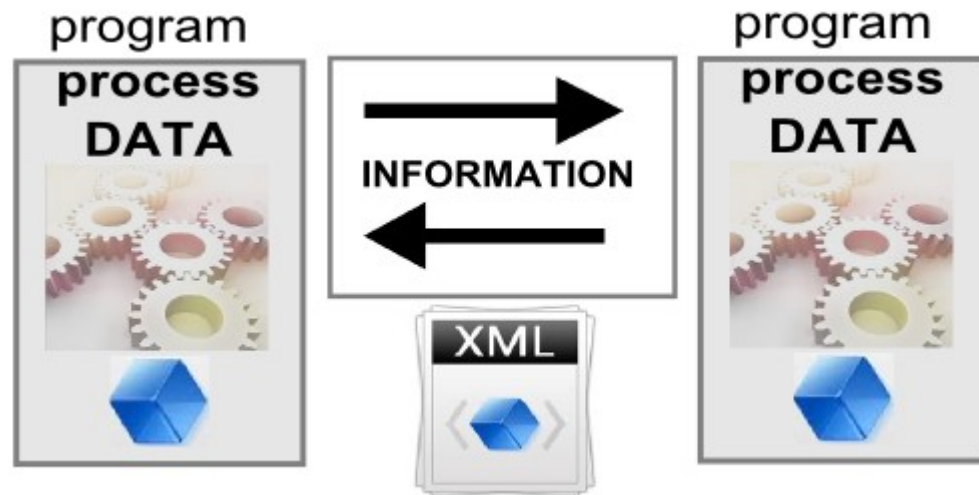
- ▶ The programs process data, and this data must be turned into information. It is this information that drives businesses.





11. Objects and Portable Data: XML

- ▶ XML is a standard mechanism for defining and transporting data between potentially disparate systems. XML provides a mechanism for independent application to share data.





11. Objects and Portable Data: XML

PORTABLE DATA

- ▶ Historically, a major business problem has been the diversity of data storage formats.



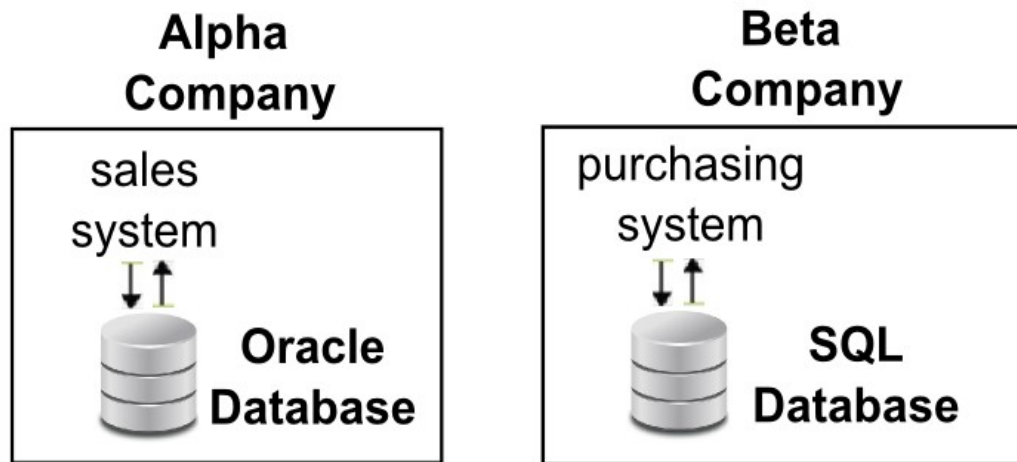
diversity of data storage formats



11. Objects and Portable Data: XML

PORTABLE DATA

- ▶ **For example**, assume that Alpha Company uses an Oracle database system to operate its sales system. Assume further that Beta Company uses a SQL Server database system to operate its purchasing system.

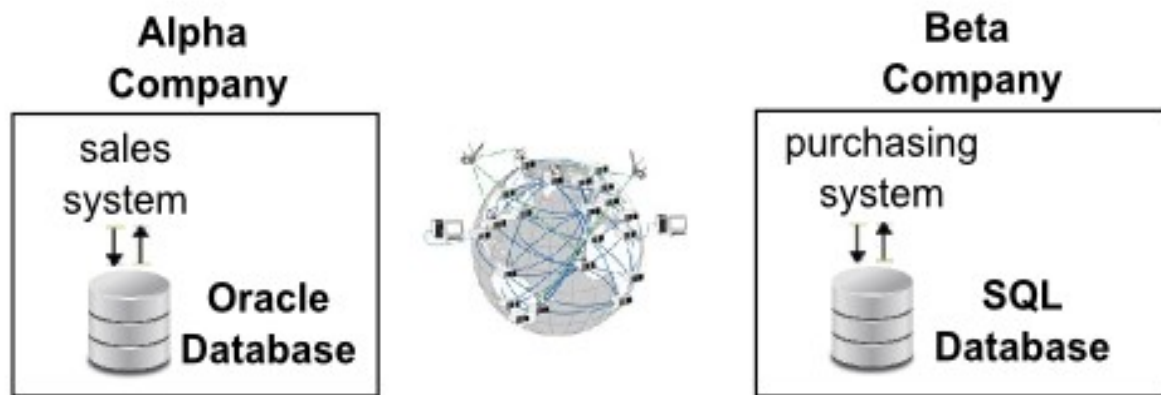




11. Objects and Portable Data: XML

PORTABLE DATA

- ▶ Now consider the problem that occurs when Alpha Company and Beta Company want to do business over the Internet.

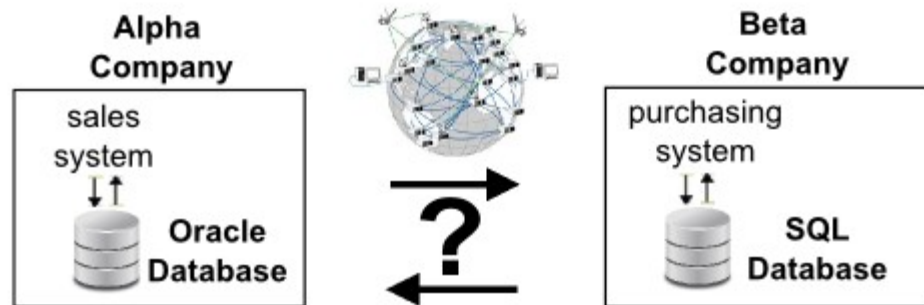




11. Objects and Portable Data: XML

PORTABLE DATA

- ▶ The two databases are not directly compatible.

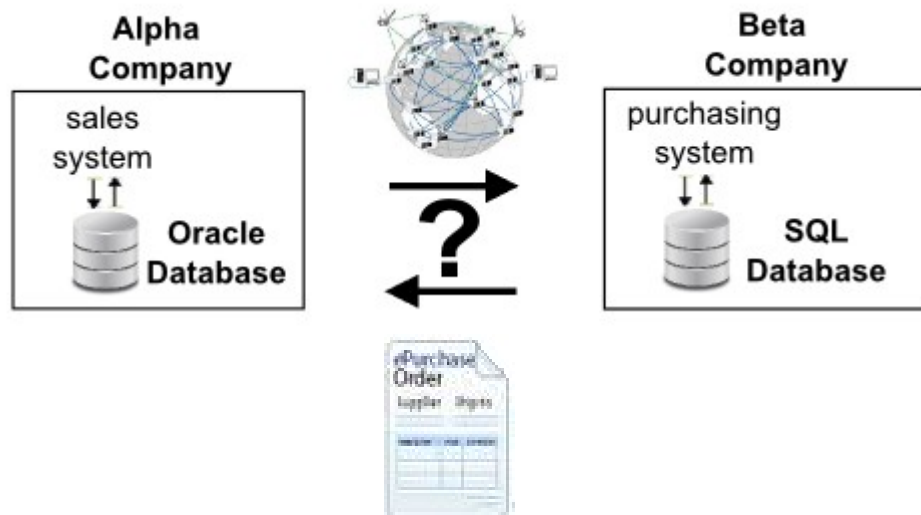




11. Objects and Portable Data: XML

PORTABLE DATA

- ▶ Our goal is to create an electronic purchase order for Beta Company using SQL Server, which will interact directly with Alpha Company's sales system, which uses Oracle.





11. Objects and Portable Data: XML

PORTABLE DATA

- ▶ XML provides standards to move data in a variety of ways. Often we can think of as moving vertically and horizontally. The term *Vertical* means that data is meant to move through multiple industry groups.



11. Objects and Portable Data: XML

PORTABLE DATA

- ▶ Industry groups such as those in accounting and finance (FpML, Financial products Markup Language) has developed their own markup languages that provide standard data definitions. These vertical applications provide the specific business models and terminology to move information across multiple industries. These standards are often called a *vocabulary*. Thus, industry groups are using XML to form a vocabulary.



11. Objects and Portable Data: XML

PORTABLE DATA

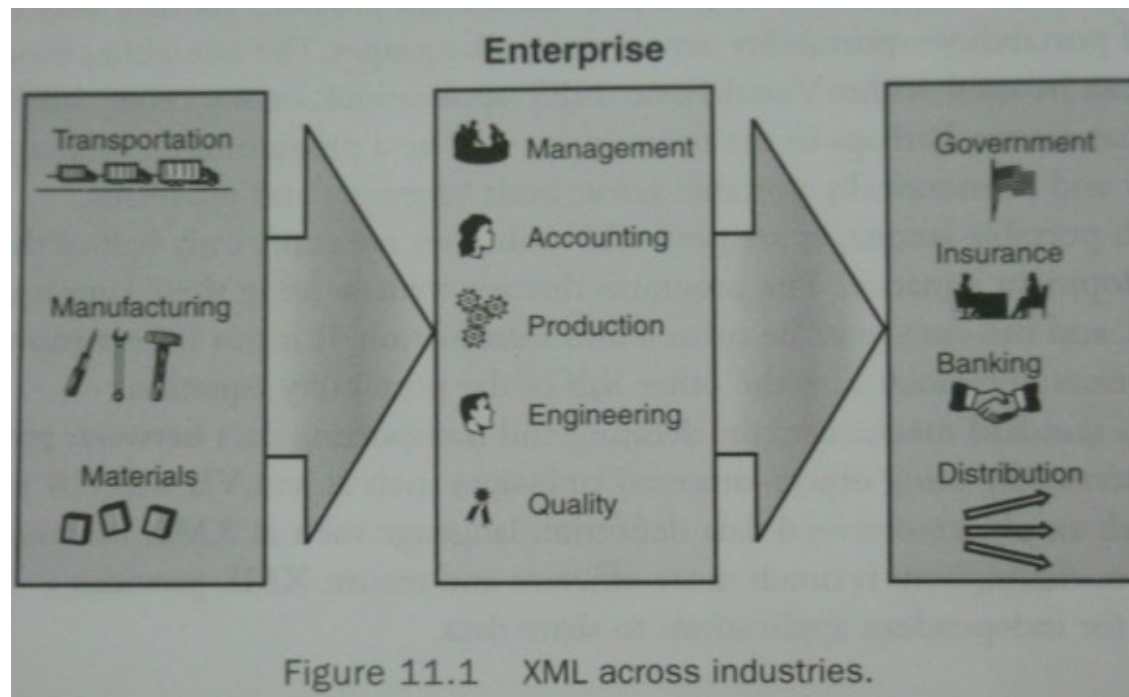
- ▶ The other approach to XML standards is that of horizontal applications. Horizontal applications are specific to a particular industry, such as retail or transportation. In all electronic commerce applications, the sharing data is paramount.



11. Objects and Portable Data: XML

PORTABLE DATA

- ▶ It represents how data can move vertically and horizontally through various industries. Some of the industries with XML-based standards include legal, hospitality, accounting, retail, travel, finance and education.





11. Objects and Portable Data: XML

PORTABLE DATA

- ▶ Here is where we consider the concept of portable data. Although the low-level data (at the machine level) is certainly not portable, we want to create a higher-level portability at the information level. XML provides this information portability that we are looking for.



11. Objects and Portable Data: XML

The Extensible Markup Language (XML)

- ▶ You probably are already familiar with another markup language called HTML (Hypertext Markup Language). Both XML and HTML are descendants of SGML (Standard Generalized Markup Language).



11. Objects and Portable Data: XML

The Extensible Markup Language (XML)

- ▶ The primary function of HTML is to present data in a browser. HTML is meant to format and present data, not to define and verify it.



11. Objects and Portable Data: XML

The Extensible Markup Language (XML)

- ▶ The primary function of HTML is to present data in a browser. HTML is meant to format and present data, not to define and verify it.





11. Objects and Portable Data: XML

The Extensible Markup Language (XML)

- ▶ XML was designed to represent data
- ▶ XML is much more strict with its format than HTML.
- ▶ XML is not proprietary and the World Wide Web Consortium (W3C) is the organization that proposes recommendations and that promotes the distribution of its standards.



```
<Books>
  <Book ISBN="0553212419">
    <title>Sherlock Holmes: Complete Novels...
    <author>Sir Arthur Conan Doyle</author>
  </Book>
  <Book ISBN="0743273567">
    <title>The Great Gatsby</title>
    <author>F. Scott Fitzgerald</author>
  </Book>
  <Book ISBN="0684826976">
    <title>Undaunted Courage</title>
    <author>Stephen E. Ambrose</author>
  </Book>
  <Book ISBN="0743203178">
    <title>Nothing Like It In the World</title>
    <author>Stephen E. Ambrose</author>
  </Book>
</Books>
```



11. Objects and Portable Data: XML

The Extensible Markup Language (XML)

- ▶ We will see how XML is used within various object-oriented technologies such as distributed computing, object persistence, and so on.



```
<Books>
  <Book ISBN="0553212419">
    <title>Sherlock Holmes: Complete Novels...
    <author>Sir Arthur Conan Doyle</author>
  </Book>
  <Book ISBN="0743273567">
    <title>The Great Gatsby</title>
    <author>F. Scott Fitzgerald</author>
  </Book>
  <Book ISBN="0684826976">
    <title>Undaunted Courage</title>
    <author>Stephen E. Ambrose</author>
  </Book>
  <Book ISBN="0743203178">
    <title>Nothing Like It In the World</title>
    <author>Stephen E. Ambrose</author>
  </Book>
</Books>
```



11. Objects and Portable Data: XML

XML versus HTML

- ▶ HTML and XML are designed for different purposes. HTML presents data, and XML describes the data.
- ▶ Both HTML and XML are important tools in the development of Web-based systems.





11. Objects and Portable Data: XML

XML versus HTML

- ▶ XML actually looks a lot like HTML. This is not surprising, because they come from the same source. However, XML provides two primary advantages that HTML does not: **validity** and **well-formed** documents.



```
<Books>
  <Book ISBN="0553212419">
    <title>Sherlock Holmes: Complete Novels...</title>
    <author>Sir Arthur Conan Doyle</author>
  </Book>
  <Book ISBN="0743273567">
    <title>The Great Gatsby</title>
    <author>F. Scott Fitzgerald</author>
  </Book>
  <Book ISBN="0684826976">
    <title>Undaunted Courage</title>
    <author>Stephen E. Ambrose</author>
  </Book>
  <Book ISBN="0743203178">
    <title>Nothing Like It In the World</title>
    <author>Stephen E. Ambrose</author>
  </Book>
</Books>
```





11. Objects and Portable Data: XML

XML versus HTML

- ▶ HTML tags are all predefined. Tags such as `<HTML>`, `<HEAD>`, `<BODY>`... are all defined in the HTML specification. You cannot add your own tags. Because HTML is intended for formatting purposes, this is not really a problem.





11. Objects and Portable Data: XML

XML versus HTML

- ▶ XML, however is meant to define data. To define data you need to create your own tag names. This is where a document called Document Type Definition (DTD) comes in play. The DTD is where you define the tags that describe your data. You can use a DTD to check the validity of the XML document.

DTD

```
<?xml version="1.0"?><!DOCTYPE books [  
  <!ELEMENT title (#PCDATA)>  
  <!ELEMENT author (#PCDATA)>  
  <!ELEMENT authors (author)+>  
  <!ELEMENT subject (#PCDATA)>  
  <!ATTLIST subject class CDATA "">  
  <!ELEMENT book (title,authors,subject)>  
  <!ATTLIST book  
    bookid CDATA #REQUIRED  
    pubdate CDATA #REQUIRED>  
>  
<books name="My books">  
  <book bookid="1" pubdate="03/01/2002">  
    <title>Java Web Services</title>  
    <authors>
```



11. Objects and Portable Data: XML

XML versus HTML

- ▶ For example, if you are creating a purchase order system, you might want to create a tag called `<purchaseOrder>` in the DTD. If you misspell the tag, this problem will be detected, and the document will be flagged as invalid. A validated document makes XML documents more robust, a necessity when dealing with data.



DTD

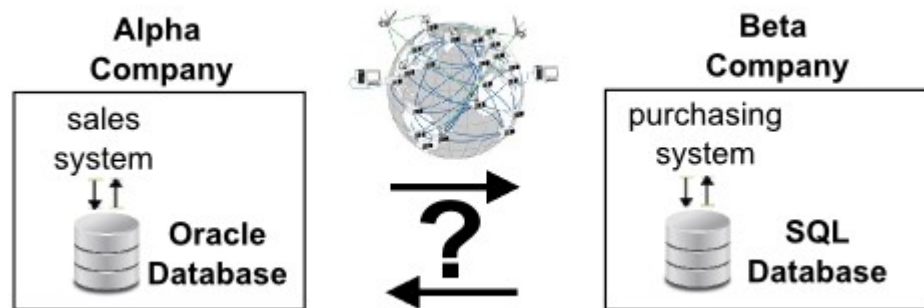
```
<?xml version="1.0"?><!DOCTYPE books [  
  <!ELEMENT title (#PCDATA)>  
  <!ELEMENT author (#PCDATA)>  
  <!ELEMENT authors (author)+>  
  <!ELEMENT subject (#PCDATA)>  
  <!ATTLIST subject class CDATA "">  
  <!ELEMENT book (title,authors,subject)>  
  <!ATTLIST book  
    bookid CDATA #REQUIRED  
    pubdate CDATA #REQUIRED>  
>  
<books name="My books">  
  <book bookid="1" pubdate="03/01/2002">  
    <title>Java Web Services</title>  
    <authors>
```



11. Objects and Portable Data: XML

XML and Object-Oriented Languages

- ▶ XML works hand-in-hand with object-oriented languages.
- ▶ For example. Let's revisit the example earlier: Beta Company wants to purchase some products from Alpha company. All transactions will be handled electronically over the internet. The problem is that the data is stored in two totally different databases would most likely be designed differently. The goal is to share data between Alpha Co. And Beta Co.

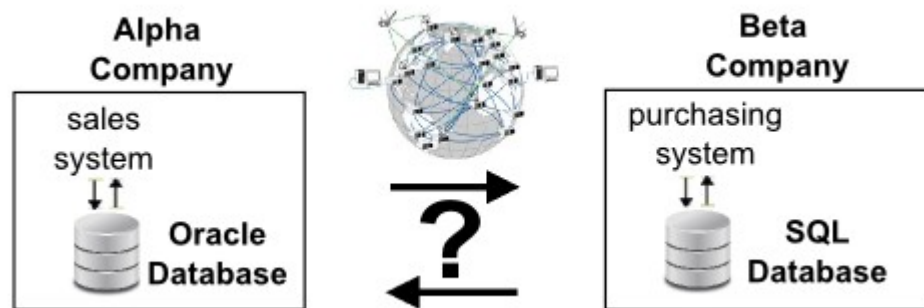




11. Objects and Portable Data: XML

XML and Object-Oriented Languages

- ▶ We cannot have a direct physical connection between the databases. The issue is how to transact business. One company sending a purchase order and the receiving company processing it.

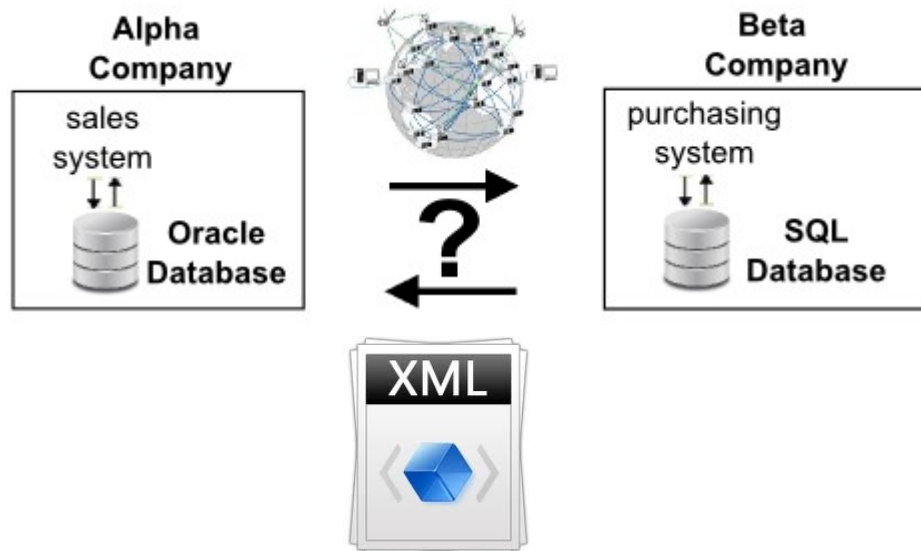




11. Objects and Portable Data: XML

PORTABLE DATA

- ▶ To accomplish the goal of connecting the systems of the two companies, Alpha Co. Comes up with an XML specification describing what information is required to complete a transaction and store the information in its database.





11. Objects and Portable Data: XML

PORTABLE DATA

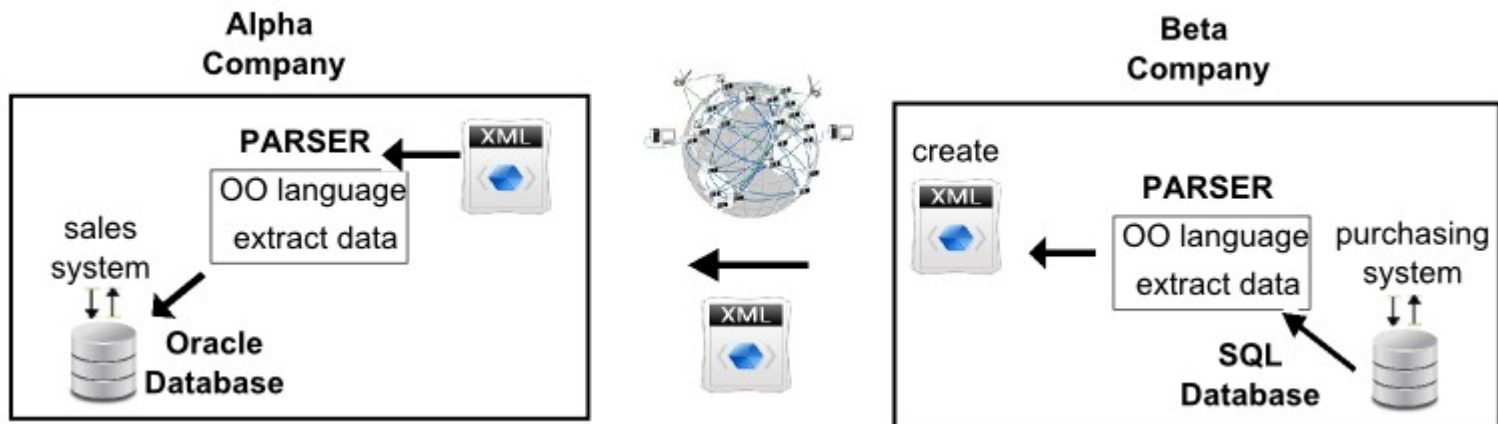
- ▶ Here is where the Object-oriented languages come in. A language such as Java, or C# can be used to extract the data from Beta Company's Sql database and create an XML document based on the agreed-upon standards.



11. Objects and Portable Data: XML

PORTABLE DATA

- ▶ This XML document can be sent over the internet to Alpha Company, the information will be extracted from the XML document and enters it into its Oracle database.



1. Introduction to Object-Oriented Concepts
2. How to think in Terms of Objects
3. Advanced Object-Oriented Concepts
4. The anatomy of a Class
5. Class Design guidelines
6. Designing with objects
7. Mastering Inheritance and Composition
8. Frameworks and Reuse: Designing with interfaces and Abstract classes
9. Building objects
10. Creating Object Models with UML
11. Objects and Portable Data: XML

12. Persistent Objects: Serialization and Relational Databases

13. Objects and the Internet
14. Objects and Client/Server Applications
15. Design Patterns

