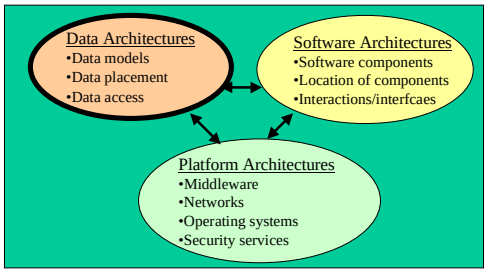
Enterprise Data Architectures

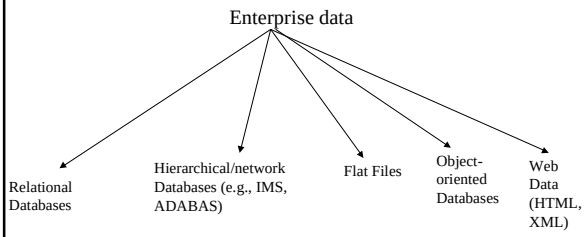
- Enterprise data overview
- Data architectures in Web-XML environments
- Data architecture steps
  - –Requirements
  - –Data modeling
  - –Logical database design
  - –Clustering partitioning
  - –Allocation
  - –Data connectivity

Amjad Umar

Data Architectures as Part of Solutions Architectures

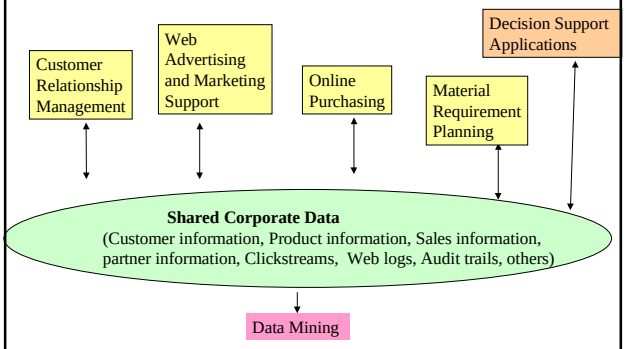


Enterprise Data Overview



Enterprise Data: Enterprise Data (also known as Corporate Data) is the information that is used or created by a corporation in conducting business and is shared across the business processes of the corporation.

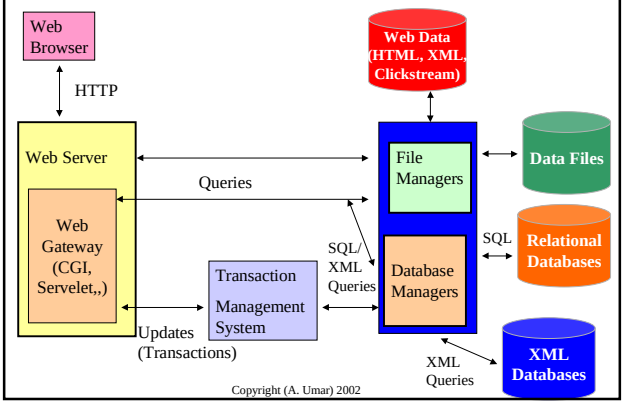
Shared Corporate (Enterprise) Data



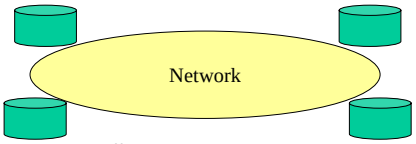
Enterprise Data in Web Environments

- Management of enterprise data in a distributed environment is a challenging task
  - Data at several sites in different formats
  - Several users need to access and update
  - How to design distributed databases
  - How to maintain data quality
  - How to use available technologies
- What is distributed data?
  - Different data at different computers (d1 at c1, d2 at c2)
  - Duplicate data- partial or complete (d1 at c1, c2, c3)
  - Partitioned (d1' at c1, d1" at c2)

Web-based Data and Transaction Management



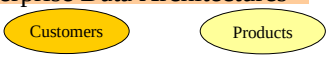
Distributed Data and Transaction Management



- Databases at different sites appear as one.
- User issues an SQL call: Select name, address from customers.
- The system finds the needed data.
- Distributed data and transaction management (DDTM)
  1. Retrieval transparency - simple as well as joins
  2. Update transparency - data updated at many sites automatically
  3. Transaction transparency - transaction may be decomposed and executed at many sites
  4. Failure transparency - if one site fails, the data can be accessed from other sites

How to Build Enterprise Data Architectures

Step1: Information Requirements



Step2: Data Modeling

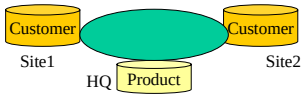
Step3: Logical&Physical Database Design

•Customer (cname, cno, product)
•Product (pid, pname, cost)

Step4: Data Partitioning and Clustering

•Customer (cname, cno, product)- site1
•Customer (cname, cno, product)-site2
•Product (pid, pname, cost)

Step5: Data Allocation



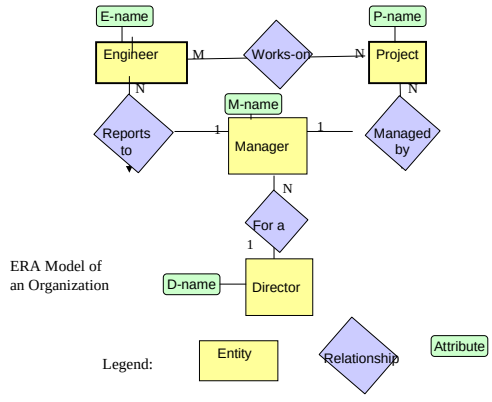
Step6: Data Access and Interoperability

- Choose database access paradigm
- Determine how data will flow between systems

Step1: Gather Information Requirements

- Understand the business processes to be served
- Document the transactions that will be executed by the users
- Identify the information needed for the transactions.
- Document operational requirements (used in later steps)
  - Size information (no. of users, frequency of use)
  - Response time (average, worst case) requirements
  - Scaling and growth requirements
  - Data security requirements
  - Data availability restrictions
  - Data synchronization requirements (i.e., how frequently duplicate data should be synchronized: immediately, hourly, daily, etc.)
  - Connectivity needed from users
  - Interoperability and interfaces
  - Backup/recovery needs and disaster handling
  - Policy restrictions (e.g., standards, application control and auditing restrictions)

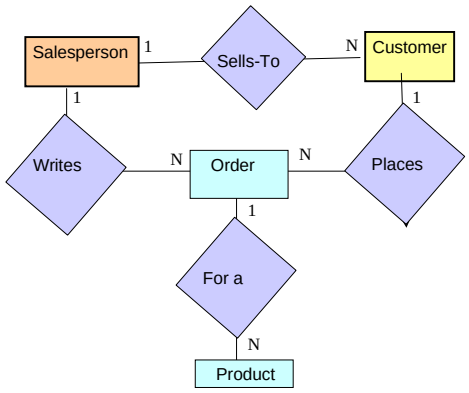
Step 2: Build a Data Model



ERA Model of an Organization

Legend:

Data Model for Order Processing



Step 3: Build Logical Database Design

Project

P-name		

Engineer

E-name		

Manager

M-name	P-name	E-name

Assignment

E-name	P-name

Director

D-name	M-name	

Database Design of an Org

Logical Database Design of order processing

Customer

Cust-no	Cust-name	

Product

Prod-no	Prod-name	

Order

Order-no	Sales-name	Cust-no

Order-line

Order-no	Prod-no

Salesperson

Sales-name	Addr	Dept	Job-level

Note: This database design can be represented in XML

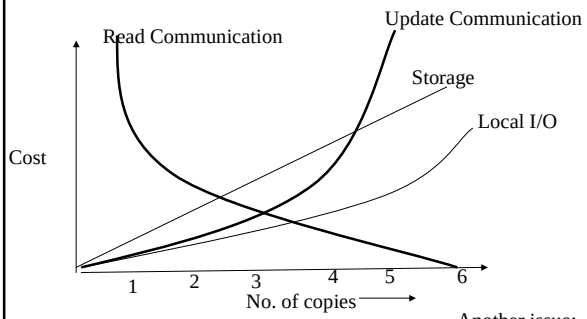
Step 4: Partitioning and Clustering

- Logical database design leads to a set of relational tables.
- These tables may be allocated to computers (database servers)
- Allocation of data in computing networks is a complex decision
  - where should the data be placed,
  - should there be a duplicate copy
  - how frequently the duplicated copy should be synchronized, etc.
  - These questions, if not addressed correctly, can cause major discomforts and misery later on in life cycle.
- Data partitioning and clustering step is a prepatory step for data allocation and, in some cases, can be included in the allocation step
  - partition the tables to improve the availability and performance of data access to different user communities.
  - group (cluster) some of these fragments into datasets, where a dataset is a unit of data that can be allocated as a unit.

Step 5: Data Allocation

- Datasets D1, D2,.., Dn can be allocated by using one of the following strategies:
  1. All datasets are allocated to a centralized machine.
  2. Datasets are uniquely allocated to database server machines where they are most frequently accessed. .
  3. Datasets are allocated as described in strategy 2, however one or more duplicate copies are kept at other computers for read-only purposes.
  4. Datasets are allocated as described in strategy 3, however the duplicate copies are updated simultaneously (i.e., if a transaction updates D1 at one computer, then all copies of D1 at other computers must also be updated before the transaction completes).
- Tradeoffs in data duplication:
  - Duplication increases availability
  - Increases synchronization costs
  - May increase security risks (more data to break into)
- Many data allocation algorithms to optimize data locations

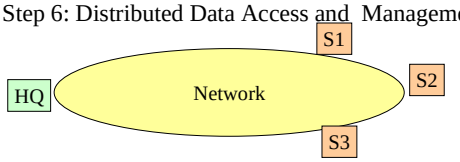
Tradeoffs in Data Duplication



Another issue: Fault tolerance versus security

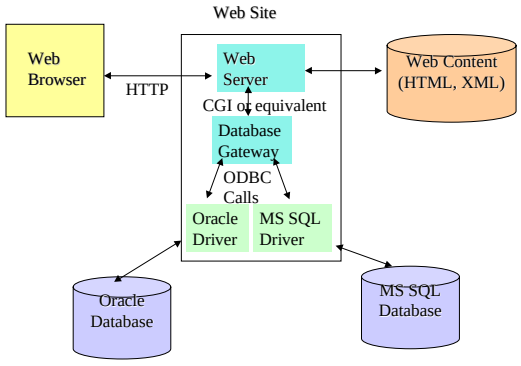
- In practice, large number of copies are not practical
- Allocation of F files to N computers requires F. 2(N)
- Large number of decisions
- Reduce by reducing N and F

Step 6: Distributed Data Access and Management

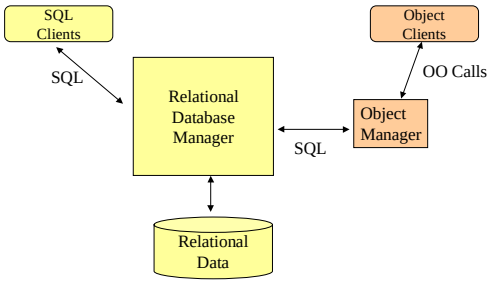


- Databases at different sites appear as one.
- User issues an SQL call: Select name, address from customers.
- The system finds the needed data.
- Distributed data management (DDM) supports transparency
  Retrieval transparency -
  Simple retrieval of ;information from a remote database is similar to local
  Remote joins: joins between multiple sites are supported
- . Failure transparency - if one site fails, the data can be accessed from other sites

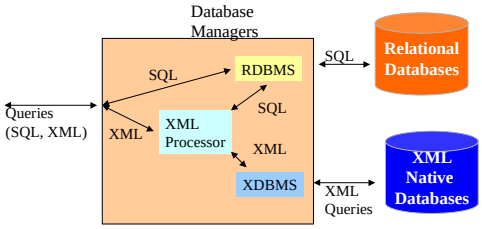
How to Access Databases From Web



Object Relational Databases allow OO access to relational data

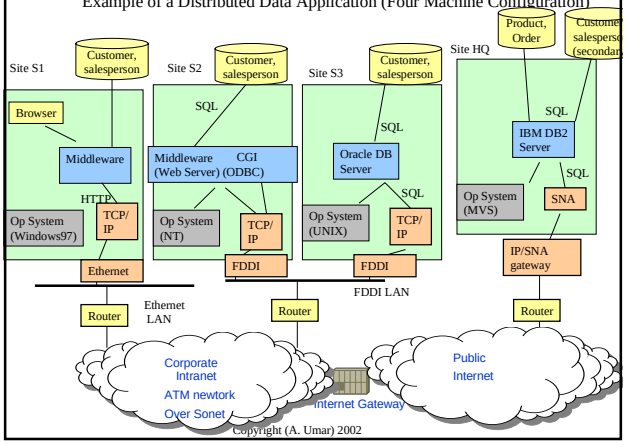


Mixing XML and SQL Databases



RDBMS = Relational Database Management System
XDBMS = XML Database Management System ("Native")

Example of a Distributed Data Application (Four Machine Configuration)



Key Points

- Enterprise data is an important assset
- Data architectures in Web-XML environments require mixing of old with new
- Data architecture needs several steps
  - Requirements
  - Data modeling
  - Logical database design
  - Clustering partitioning
  - Allocation
  - Data connectivity