







ion Systems Analysis and Design

© 2002 Jaelson Castro and John Mylopoulos

Distributed Information Systems

- Most information systems are distributed.
- This means that the objects that participate in a particular use case need not be on the same machine with other objects and users they are supposed to interact.
- One can use Remote Procedure Calls-RPC (C/C++) and Remote Method Invocation-RMI (Java).
- The object-oriented industrial standard for distributed objects is CORBA (Common Object Request Broker Architecture)

© 2002 Jaelson Castro and John Mylop

OO Database De

CSC34

OO Database Design 2

Information Systems Analysis and Design

© 2002 Jaelson Castro and John Mylopoulos

CORBA

- CORBA separates the interface of a class from its implementation. The implementation runs on one machine, the interface can be compiled on several other machines.
- When accessed by a client program, an object is treated as though it is in memory on the client machine; however, the object may actually be located on another machine.
- When the client program sends an object a message to invoke one of its operations, the message and parameters are converted into a format that can be sent over the network (*marshalling*)
- At the other end, the server unmarshals the data back into a message with arguments, and passes these on to the implementation of the target object.
- CORBA achieves this by means of programs known as ORBs (Object Request Brokers) that run on each machine.
- The ORBs communicate with each other by means of an Inter-ORB Protocol (IOP).
- Over the Internet, the protocol used is IIOP (Internet IOP).

formation Systems Analysis and Design

CSC34

00 Database Design 7

Designing Data Management Classes

- Idea is to not use a DBMS (Relational or Object-Oriented.)
- Instead, design data management classes which handle persistence, caching, etc.
- These classes decouple applications from their persistent storage.
- Use data management classes whenever you need to:
 Store an application object persistently;
 - Search for or retrieve stored objects;
 - Interface with an external database.
- This solution won't work for large data sets!

© 2002 Jaelson Castro and John Mylopoulos





	Database Broker Approach
:	Each persistent class could be responsible for its own storage but < highly coupled (to storage mechanism); < lessens class cohesion; < class must now have expert knowledge of storage tasks; < these are unrelated to application tasks. Solution; defined on the storage tasks;
-	Separates the business objects from their data storage implementation. The classes that provide the data storage services will be held in a separate package. For each business class that needs to be persistent, there will be an aeromiterid attabase berging days.

CSC34

00 Database Design &







Information Systems Analysis and Design	CSC340
Additional Reading	
 Rumbaugh et al. Object-Oriented Modeling and Design. Prentice Hall, 1991; Chapter 17 - Relational Databases Larman, Applying UML and Patterns. Prentice-Hall, 199 Chapter 38 - Frameworks, Patterns and Persistence Coad, Object Models - Strategies, Patterns and Application Prentice-Hall, 1997; Appendix C - Data Management 	:0-)8. 75.
© 2002 Jaalson Castro and John Millonoulos	Design 18

CSC34