CprE 450/550X
Distributed Systems and Middleware

Distributed Object-based Systems

Yong Guan
3216 Coover
Tel: (515) 294-8378
Email: guan@ee.iastate.edu

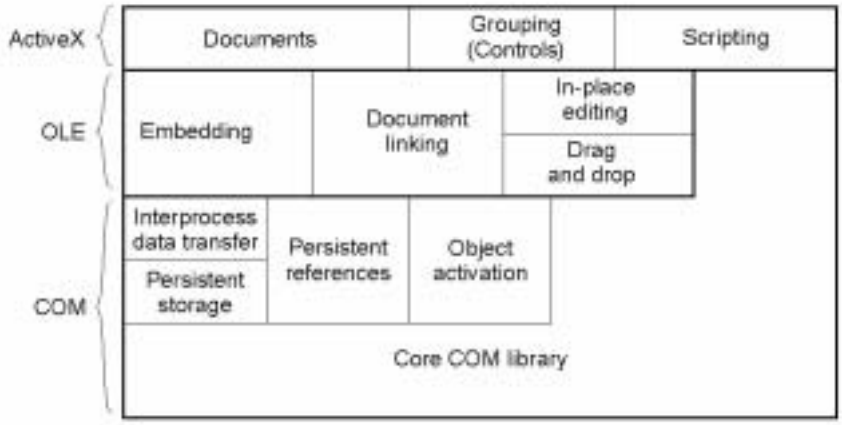
March 11, 2003

Readings for Today's Lecture

- References
  - Chapter 9 of "Distributed Systems: Principles and Paradigms"
  - http://www.corba.org/
  - http://www.omg.org/
  - "Understanding CORBA"
  - "Introduction to Distributed Object Programming with CORBA ", http://www.cs.wustl.edu/~schmidt/PDF/corba4.pdf
  - DCOM, http://www.microsoft.com/com/tech/DCOM.asp
  - J2EE Tutorial, http://java.sun.com/j2ee/tutorial/1_3-fcs/
  - Microsoft .NET, http://www.microsoft.com/net/basics/

3

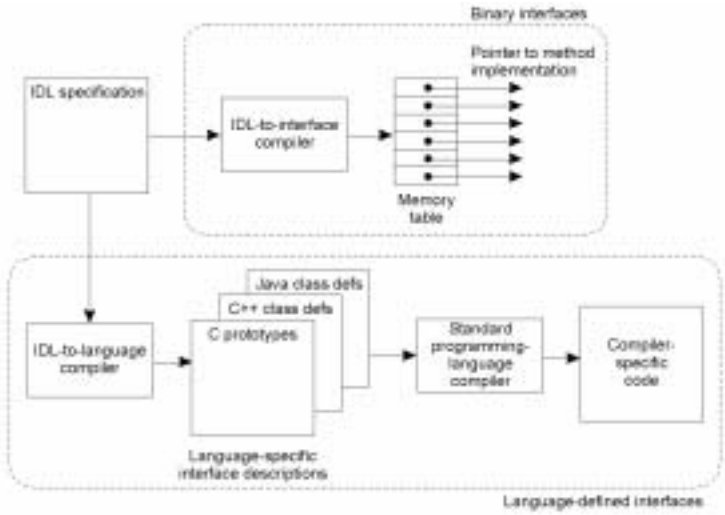
Overview of DCOM



The general organization of ActiveX, OLE, and COM.

4

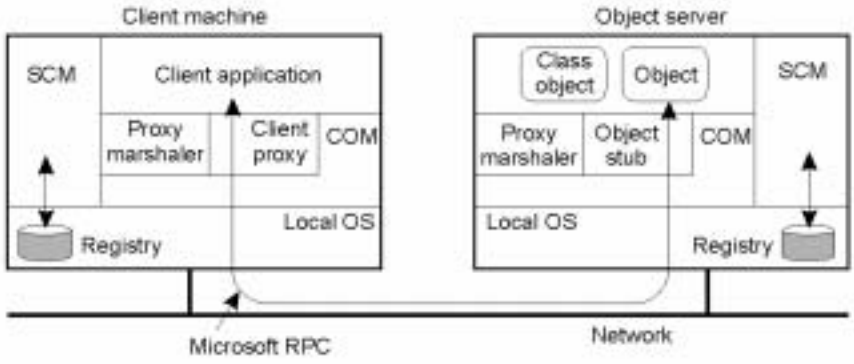
Object Model



The difference between language-defined and binary interfaces.

5

Tape Library and Registry



The overall architecture of DCOM.

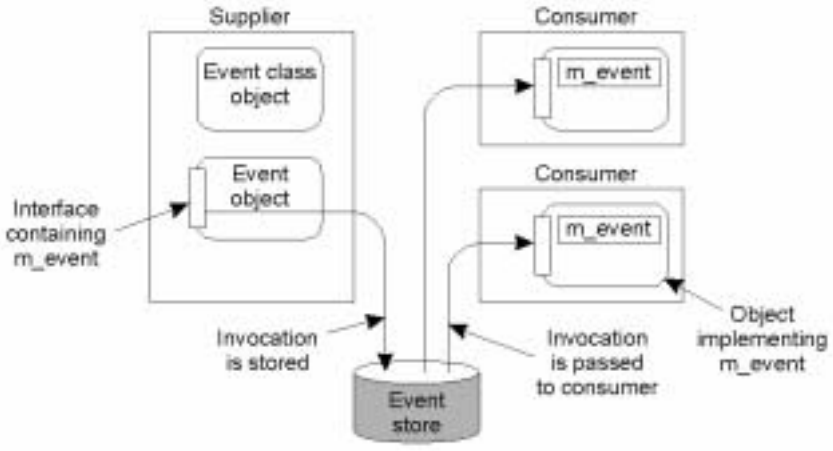
6

DCOM Services

CORBA Service	DCOM/COM+ Service	Windows 2000 Service
Collection	ActiveX Data Objects	-
Query	None	-
Concurrency	Thread concurrency	-
Transaction	COM+ Automatic Transactions	Distributed Transaction Coordinator
Event	COM+ Events	-
Notification	COM+ Events	-
Externalization	Marshaling utilities	-
Life cycle	Class factories, JIT activation	-
Licensing	Special class factories	-
Naming	Monikers	Active Directory
Property	None	Active Directory
Trading	None	Active Directory
Persistence	Structured storage	Database access
Relationship	None	Database access
Security	Authorization	SSL, Kerberos
Time	None	None

7

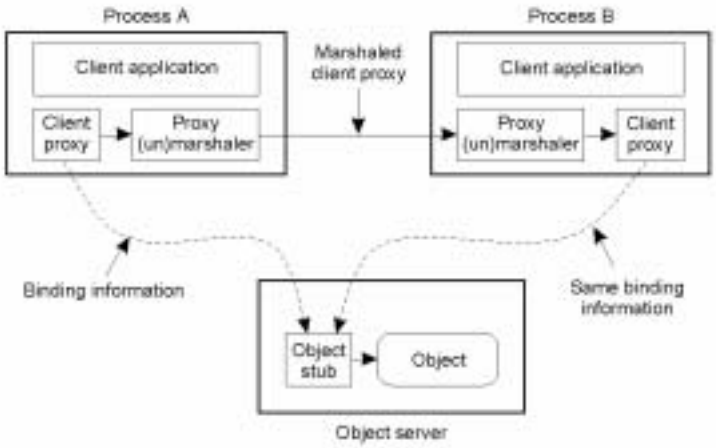
Events



Event processing in DCOM.

8

Clients



Passing an object reference in DCOM with custom marshaling.

9

Monikers (1)

Step	Performer	Description
1	Client	Calls BindMoniker at moniker
2	Moniker	Looks up associated CLSID and instructs SCM to create object
3	SCM	Loads class object
4	Class object	Creates object and returns interface pointer to moniker
5	Moniker	Instructs object to load previously stored state
6	Object	Loads its state from file
7	Moniker	Returns interface pointer of object to client

Binding to a DCOM object by means of file moniker.

10

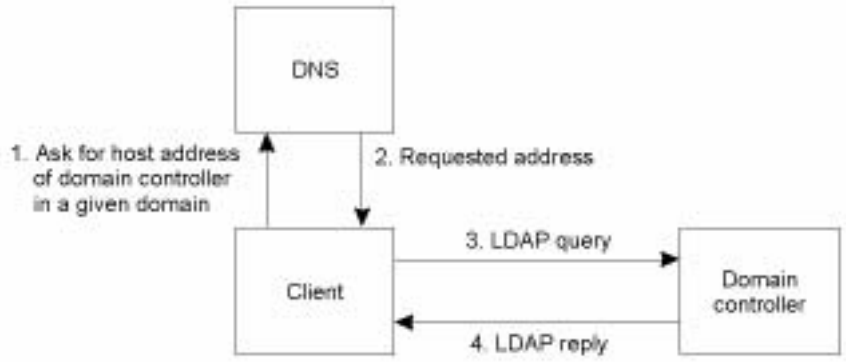
Monikers (2)

Moniker type	Description
File moniker	Reference to an object constructed from a file
URL moniker	Reference to an object constructed from a URL
Class moniker	Reference to a class object
Composite moniker	Reference to a composition of monikers
Item moniker	Reference to a moniker in a composition
Pointer moniker	Reference to an object in a remote process

DCOM-defined moniker types.

11

Active Directory

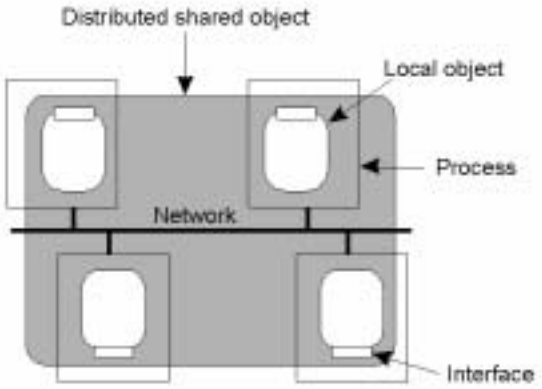


The general organization of Active Directory.

12

13

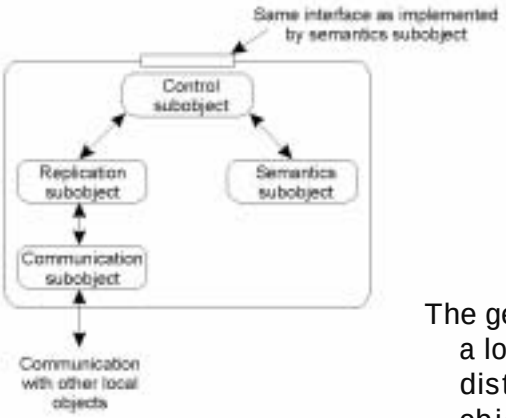
Globe Object Model (1)



The organization of a Globe distributed shared object.

14

Globe Object Model (2)



The general organization of a local object for distributed shared objects in Globe.

15

Globe Object Model (3)

Document Interface	
Method	**Description**
AddElement	Add an element to the current set of elements
DeleteElement	Remove an element from the Web document
AllElements	Return a list of the elements currently in the document
SetRoot	Set the root element
GetRoot	Return a reference to the root element
Content Interface	
Method	**Description**
GetCotent	Return the content of an element as an array of bytes
PutContent	Replace the content of an element with a given array of bytes
PutAllContent	Replace the content of an entire document

Interfaces implemented by the semantics subobject of a GlobeDoc object.

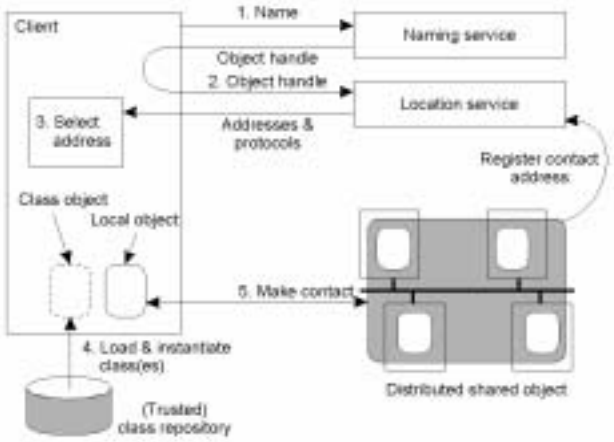
16

Globe Object Model (4)

Property Interface	
Method	**Description**
GetProperties	Return the list of (attribute, value)-pairs of an element
SetProperties	Provide a list of (attribute, value)-pairs for an element
Lock Interface	
Method	**Description**
CheckOutElements	Check out a series of elements that require modification
CheckInElements	Check in a series of modified elements
GetCheckedElements	Get a list of elements that are currently checked out

Interfaces implemented by the semantics subobject of a GlobeDoc Object.

Process-to-Object Binding



Binding a process to an object in Globe.

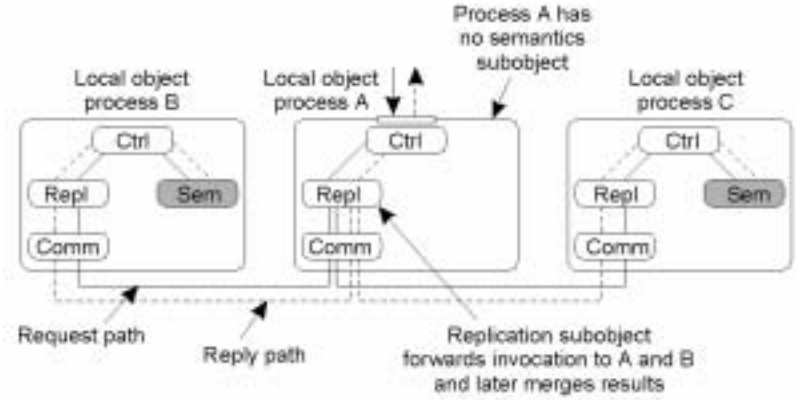
Globe Services

Service	Possible Implementation in Globe	Available
Collection	Separate object that holds references to other objects	No
Concurrency	Each object implements its own concurrency control strategy	No
Transaction	Separate object representing a transaction manager	No
Event/Notification	Separate object per group of events (as in DCOM)	No
Externalization	Each object implements its own marshaling routines	Yes
Life cycle	Separate class objects combined with per-object implementations	Yes
Licensing	Implemented by each object separately	No
Naming	Separate service, implemented by a collection of naming objects	Yes
Property/Trading	Separate service, implemented by a collection of directory objects	No
Persistence	Implemented on a per-object basis	Yes
Security	Implemented per object, combined with (local) security services	Yes
Replication	Implemented on a per-object basis	Yes
Fault tolerance	Implemented per object combined with fault-tolerant services	Yes

Overview of possible Globe implementations of typical distributes-systems services.

19

Communication



Invoking an object in Globe that uses active replication.

20

Globe Server

Method	Description
Bind	Lets the server bind to a given object, unless it is already bound
AddBinding	Lets the server bind to an object, even if it is already bound
CreateLR	Lets the server create a local object for a new distributed object
RemoveLR	Lets the server remove a local object of a given object
UnbindDSO	Lets the server remove all local objects of a given object
ListAll	Returns a list of all local objects
ListDSO	Returns a list of all local objects for a given objects
StatLR	Get the status of a specific local object

Operations on a Globe object server.

21

Object References and Contact Addresses (1)

Field	Description
Protocol identifier	A constant representing a (known) protocol
Protocol address	A protocol-specific address
Implementation handle	Reference to a file in a class repository

The representation of a protocol layer in a stacked contact address.

22

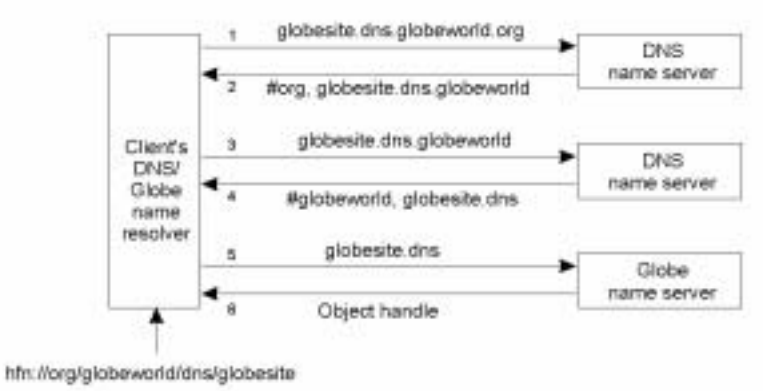
Object References and Contact Addresses (2)

Field	Description
Implementation handle	Reference to a file in a class repository
Initialization string	String that is used to initialize an implementation

The representation of an instance contact address.

23

Globe Naming Service



Iterative DNS-based name resolution in Globe.

24

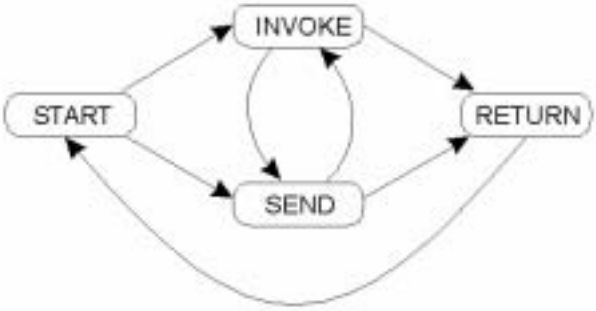
Replication (1)

Method	Description
Start	Indicate that a new method invocation has been locally requested
Send	Pass the marshaled invocation request to the replication subobject
Invoked	Indicate that the invocation on the semantics object has completed

The interface of the replication subobject as made available to the control subobject.

25

Replication (2)



The behavior of the control subobject as a finite state machine.

26

Examples of Replication in Globe (1)

Read method			
State	**Action to take**	**Method call**	**Next state**
START	None	Start	INVOKE
INVOKE	Invoke local method	Invoked	RETURN
RETURN	Return results to caller	None	START
Modify method			
State	**Action to take**	**Method call**	**Next state**
START	None	Start	SEND
SEND	Pass marshaled invocations	Send	INVOKE
INVOKE	invoke local method	Invoked	RETURN
RETURN	Return results to caller	None	START

State transitions and actions for active replication.

Examples of Replication in Globe (2)

Read method			
State	**Action to take**	**Method call**	**Next state**
START	None	Start	INVOKE
INVOKE	Invoke local method	Invoked	RETURN
RETURN	Return results to caller	None	START

Modify method at backup replica			
State	**Action to take**	**Method call**	**Next state**
START	None	Start	SEND
SEND	Pass marshaled invocation	Send	RETURN
RETURN	Return results to caller	None	START

Modify method at primary replica			
State	**Action to take**	**Method call**	**Next state**
START	none	Start	INVOKE
INVOKE	invoke local method	Invoked	RETURN
RETURN	Return results to caller	None	START

- State transitions and actions with primary-backup replication.

Summary (1)

Issue	CORBA	DCOM	Globe
Design goals	Interoperability	Functionality	Scalability
Object model	Remote objects	Remote objects	Distributed objects
Services	Many of its own	From environment	Few
Interfaces	IDL based	Binary	Binary
Sync. communication	Yes	Yes	Yes
Async. communication	Yes	Yes	No
Callbacks	Yes	Yes	No
Events	Yes	Yes	No
Messaging	Yes	Yes	No
Object server	Flexible (POA)	Hard-coded	Object dependent
Directory service	Yes	Yes	No
Trading service	yes	No	No

Continued …

Comparison of CORBA, DCOM, and Globe.

29

Summary (2)

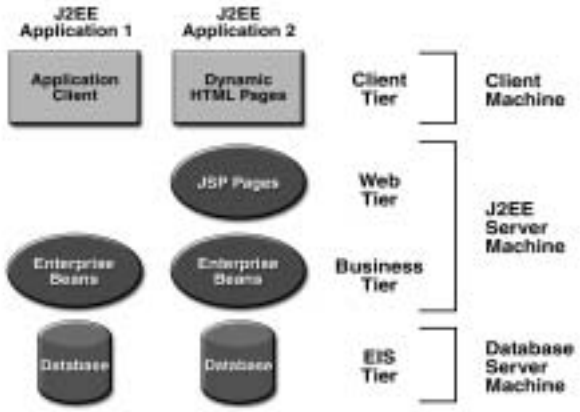
Issue	CORBA	DCOM	Globe
Naming service	Yes	Yes	Yes
Location service	No	No	Yes
Object reference	Object's location	Interface pointer	True identifier
Synchronization	Transactions	Transactions	Only intra-object
Replication support	Separate server	None	Separate subobject
Transactions	Yes	Yes	No
Fault tolerance	By replication	By transactions	By replication
Recovery support	Yes	By transactions	No
Security	Various mechanisms	Various mechanisms	More work needed

Comparison of CORBA, DCOM, and Globe.

30

31

J2EE: Java™ 2 Platform, Enterprise Edition



Multitiered Distributed Application model

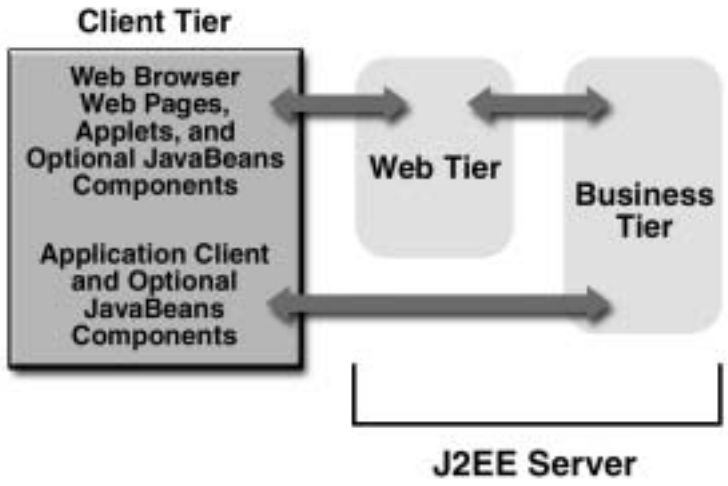
32

J2EE Components

- Application clients and applets are components that run on the client.
- Java Servlet and JavaServer Pages (JSP) technology components are Web components that run on the server.
- Enterprise JavaBeans (EJB) components (enterprise beans) are business components that run on the server.

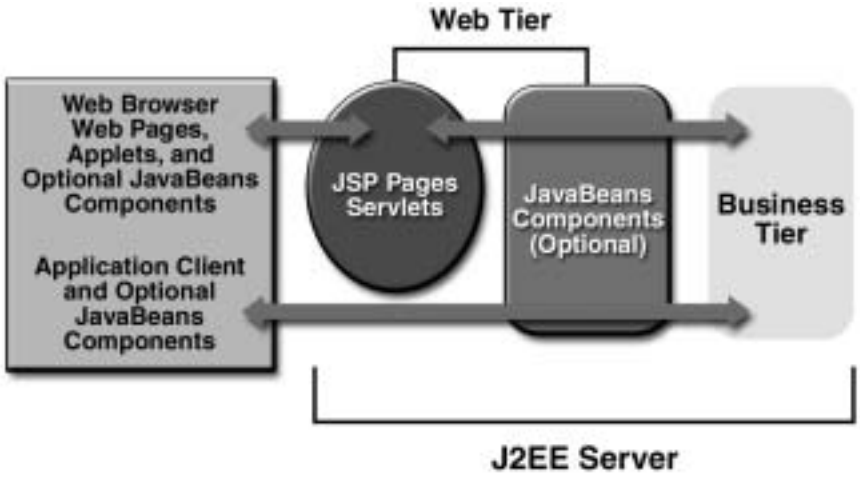
33

J2EE Server Communications



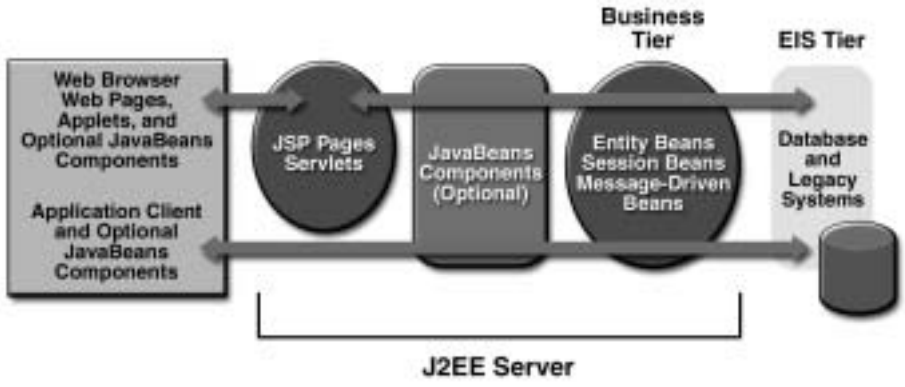
34

J2EE Web components



35

J2EE Business Components



36

Microsoft .Net

- A set of Microsoft software technologies for connecting information, people, systems, and devices. It enables a high level of software integration through the use of XML Web services—small, discrete, building-block applications that connect to each other as well as to other, larger applications over the Internet.

Next class, we are going to study synchronization (Chapter 5).