Class announcements
- Midterm Tuesday!
  - Study guide to be posted soon.

Student Presentation

Three-tier client/server

System integration

Emergence

Subsystems are more specialized and simpler functionality
Higher-level system functionality arises from the interaction of subsystems
Emergence includes capabilities that arise purely from that interaction (desired or not)
  - e.g. airplane flies, but subsystems can't
Why system decomposition?

- Divide and conquer approach to containing complexity
- Reuse
- Consonant with industry structure (unless system is to be supplied by one company)
- Others?

Networked computing infrastructure

by
David G. Messerschmitt

Layering

Elaboration or specialization

<table>
<thead>
<tr>
<th>Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Existing layers</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Example of Layering: networking

```
+-----------------+     +-----------------+     +-----------------+
| Application     |     | Message         |     | Physical        |
|                 |     | Frames          |     |                |
| Transport       |     |                 |     |                |
|                 |     |                 |     |                |
| Network         |     |                 |     |                |
|                 |     |                 |     |                |
| Link            |     |                 |     |                |
|                 |     |                 |     |                |
| Physical        |     | Bits            |     |                |
|                 |     |                 |     |                |
```

Software Layering

```
+-----------------+     +-----------------+     +-----------------+
| Application     |     | Message         |     | Physical        |
|                 |     | Frames          |     |                |
|                 |     |                 |     |                |
| Transport       |     |                 |     |                |
|                 |     |                 |     |                |
| Network         |     |                 |     |                |
|                 |     |                 |     |                |
| Link            |     |                 |     |                |
|                 |     |                 |     |                |
| Physical        |     | Bits            |     |                |
|                 |     |                 |     |                |
```

Operating system functions

- Graphical user interface (client only)
- Hide details of equipment from the application
- Multitasking
- Resource management
  - Processing, memory, storage, etc
  - etc
Middleware Functions

- Capabilities that can be shared by many applications, but that is not part of OS
  - Example: Database Management System (DBMS)
- Hide details of OS from application
  - Java Virtual Machine
- More purposes we’ll talk about later.

What’s a database?

Database
- File with specified structure
- Example: relational table

A Database

<table>
<thead>
<tr>
<th>Year</th>
<th>City</th>
<th>Accommodation</th>
<th>Tourists</th>
</tr>
</thead>
<tbody>
<tr>
<td>2002</td>
<td>Oakley</td>
<td>Bed&amp;Breakfast</td>
<td>14</td>
</tr>
<tr>
<td>2002</td>
<td>Oakland</td>
<td>Resort</td>
<td>290</td>
</tr>
<tr>
<td>2002</td>
<td>Oakland</td>
<td>Bed&amp;Breakfast</td>
<td>390</td>
</tr>
<tr>
<td>2002</td>
<td>Berkeley</td>
<td>Camping</td>
<td>12000</td>
</tr>
<tr>
<td>2002</td>
<td>Berkeley</td>
<td>Resort</td>
<td>28000</td>
</tr>
<tr>
<td>2002</td>
<td>Albany</td>
<td>Bed&amp;Breakfast</td>
<td>3240</td>
</tr>
<tr>
<td>2003</td>
<td>Oakley</td>
<td>Resort</td>
<td>320</td>
</tr>
<tr>
<td>2003</td>
<td>Oakland</td>
<td>Bed&amp;Breakfast</td>
<td>280</td>
</tr>
<tr>
<td>2003</td>
<td>Oakland</td>
<td>Resort</td>
<td>210</td>
</tr>
<tr>
<td>2003</td>
<td>Berkeley</td>
<td>Camping</td>
<td>110000</td>
</tr>
<tr>
<td>2003</td>
<td>Berkeley</td>
<td>Resort</td>
<td>419000</td>
</tr>
<tr>
<td>2003</td>
<td>Albany</td>
<td>Bed&amp;Breakfast</td>
<td>6750</td>
</tr>
</tbody>
</table>

Storage Middleware example: DBMS

- Database Management System (DBMS)
  - Manage Multiple databases
  - Allow multiple applications to access common databases
  - Implement standard data "lookup" (query) functions.

The Internet

by
David G. Messerschmitt

Intranet

Private internet

Often connected to Internet

- Firewall creates a protected enclave
**Extranet**

An Extranet is composed of:
- Intranets connected through an unprotected domain (typically the Internet)
- Encryption and other security technologies used to:
  - protect proprietary information
  - prevent imposters, vandals, etc.

---

**What is the Internet?**

- An internet is a “network of networks”
- Interconnect standard for LAN’s, MAN’s, and WAN’s
- Internet = the major global internet
- A private internet is called an intranet
- An extranet is an interconnection of intranets through the Internet

---

**Client - Server Computing**

---

**Client Server Example**

---

**Client Server Example - Layers Revealed**

---
3-Tier Client Server Architecture example

Client

Application Server

Web Server

Database

Common Gateway Interchange

In some implementations, Application Logic and Web Server can be put on different machines.

Shared data

Client

What is Bob's balance?

$0.50

$0.50

Bob $0.50 Bronze

Charles $1000000 Gold

Relational Database

<table>
<thead>
<tr>
<th>Customer</th>
<th>Balance</th>
<th>Customer Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>$527</td>
<td>Silver</td>
</tr>
<tr>
<td>Bob</td>
<td>$0.50</td>
<td>Bronze</td>
</tr>
<tr>
<td>Charles</td>
<td>$1000000</td>
<td>Gold</td>
</tr>
</tbody>
</table>

DBMS Responsibilities

- Hide Changes in the Database hardware from the Application
- Standard operations on the data, including searches, such a search is called a query.
- Separate Database Management from Applications, so that many applications can access the same data.
- Security, Integrity, Backup, fault tolerance, etc.
3-Tier Client Server Architecture in General

- Takes inputs from client
- Accepts instructions from user
- Makes requests of server

- Takes inputs from client
- Decides what to be done next
- Decides what shared data to access
- Processes shared data

- Support multiple applications with common data
- Protect critical data
- Decouple data administration and application administration

Shared data

Peer to peer

- Client
- Server
- Fulfillment logic
- Book distributors
- Databases

Sun Case

- Workstations
- Servers
- Software

Sun N-tier case

- What does Sun make?

Financial institution

Book distribution centers

Consumer

Enterprise

Inter-enterprise

Slide adapted from slides for Understanding Networked Applications
By David G Messerschmitt. Copyright 2000. See copyright notice
How Successful had Sun been up to 1998?
- Founded in 1982
- Open Standards Workstation
  - Unix Operating System (Solaris)
  - TCP/IP networking
- 1988 - Revenues $1 billion
- 1993 - Market value $3.0 billion
- 1997 - Jumped from 3rd to 1st in Unix Server Market.

How Successful had Sun been up to 1998?
- 1993 - "The network is the computer."
- 1994 - Internet explodes in popularity

Microsoft mid to late 90s
- Dominated Desktop software
  - Users familiar with Windows, Office, etc.
- NT servers
  - Fine for small intranets, "not industrial strength"

Sun N-Tier Case
- What is Java?
  - Programming Language
  - Portable between computers with different operating systems
  - Easy to write programs in
  - Easier re-use
  - But, programs are slow

What problems did the micro era produce?
- Desktops are expensive to maintain
  - TCO for windows PC $9900!
- Every PC had a lot of software that had to be maintained
  - Office, Windows, etc...
- Small differences, like the order in which software is installed, could make different PCs behave differently!

In the Networking Era
- These "bloated" PCs are networked and termed fat clients.
- But networking of PCs offered the possibility of
  - Putting most of the functionality into servers
  - Getting rid of much of the software on the client
  - These clients would be called thin clients.
  - Sun, Oracle, and others saw it as the future.
**Hardware for thin clients**

- A **Network Computer** (NC) – a computer with minimal hardware that depends on a network connection to a server to function
  - Be careful not to confuse it with the phrase “networked computer!”
  - Example: Sun’s JavaStation (1996–2000)
  - It is the hardware one would use to implement a thin-client computing model.

**Another term from that era..**

- A **NetPC** was a PC introduced by Microsoft and Intel in 1996
  - Same software as a normal PC
  - Did not allow users to install their own software
  - NetPC died out
  - Features of it, and Microsoft’s Zero Administration Kit, live on in today’s version of Windows.

**Microsoft Vision**

- Keep “fat-client” model
- Add some features to Windows to reduce administration costs

**Sun’s Vision**

- Thin Client model.
- Application Servers with Applications written in Java.
- NCs could retrieve applications from application server as needed.
- Applications compatible with any NC hardware and OS.
- Applications could be fixed, added, updated at the server level, rather than maintaining each PC.

**SUN 3 - Tier**

**Sun N-tier**
Today

- 3-tier model common.
- Sun's version of 4-tier model not-common.
- N-tier model where Webserver and Application Server on separate equipment also common.
- Sun's hardware business not strong.
  - Linux on cheap PCs most common servers
  - Microsoft desktops replacing Sun workstations

Sun's Performance

<table>
<thead>
<tr>
<th>Year</th>
<th>Net Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>1995</td>
<td>355.8</td>
</tr>
<tr>
<td>1996</td>
<td>476.4</td>
</tr>
<tr>
<td>1997</td>
<td>762.4</td>
</tr>
<tr>
<td>1998</td>
<td>762.9</td>
</tr>
<tr>
<td>1999</td>
<td>1,031.3</td>
</tr>
<tr>
<td>2000</td>
<td>1,854.0</td>
</tr>
<tr>
<td>2001</td>
<td>927.0</td>
</tr>
<tr>
<td>2002</td>
<td>(587.0)</td>
</tr>
<tr>
<td>2003</td>
<td>(3,429.0)</td>
</tr>
<tr>
<td>2004</td>
<td>(388.0)</td>
</tr>
<tr>
<td>2005</td>
<td>(106.0)</td>
</tr>
</tbody>
</table>

Java

- Common in Server implementations
  - Example: Java Servlet implementing application logic in a banking application.
- Often used to push simple applets onto client
- Not common
  - For “big” desktop applications
  - Office Suite in Java not popular
- Microsoft is still in business...