Class announcements

- Midterm Thursday Feb 11, 2010
- Syllabus: Everything covered till Feb 09, 2010 Tue.
Class announcements

- Business Paper Assessment:
  - In addition to the details mentioned on the website, we will consider the following
  - At the end of the business paper you have to mention which team member has taken care of which section.
    - You can also mention that the whole team has worked together to prepare each of the sections if that is the case.
  - If there is a huge difference between the quality of overall paper and the individual sections then the grading will be done accordingly.
  - Remember this project is to inculcate team-work!
Student Presentation
Architecture

by

David G. Messerschmitt
©Copyright David G. Messerschmitt, 2000. This material may be used, copied, and distributed freely for educational purposes as long as this copyright notice remains attached. It cannot be used for any commercial purpose without the written permission of the author.
What is Architecture?

How do you architect a solution?
A system is decomposed into interacting subsystems.

Each subsystem may have a similar internal decomposition.
Three elements of architecture

Decomposition
Organization
Functionality
Responsibility
Interaction
Cooperation
System examples

Let’s quickly look at some system decomposition examples

- Quick tour of information technology systems
Time sharing

ASCII terminal
(no graphics)

Point-to-point wire
(no network)

Mainframe
(database and application server)
Two-tier client/server

Local-area network

Server/Mainframe
Three-tier client/server

Client

Application server

Enterprise data server
Inter-organizational computing

Global internet
Consumer access
System integration

Architecture

-> subsystem implementation

-> system integration

Bring together subsystems and make them cooperate properly to achieve desired system functionality

- Always requires testing
- May require modifications to architecture and/or subsystem implementation
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

Services

Existing layers
Example of Layering: networking

- **Physical**: Bits, Signals
- **Link**: Frames
- **Network**: Packets
- **Transport**: Messages
- **Application**
Software Layering

- Application
- Middleware
- Operating System
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>Oakley</td>
<td>Resort</td>
<td>190</td>
</tr>
<tr>
<td>2002</td>
<td>Oakland</td>
<td>Bed &amp; Breakfast</td>
<td>340</td>
</tr>
<tr>
<td>2002</td>
<td>Oakland</td>
<td>Resort</td>
<td>230</td>
</tr>
<tr>
<td>2002</td>
<td>Berkeley</td>
<td>Camping</td>
<td>120000</td>
</tr>
<tr>
<td>2002</td>
<td>Berkeley</td>
<td>Bed &amp; Breakfast</td>
<td>3450</td>
</tr>
<tr>
<td>2002</td>
<td>Berkeley</td>
<td>Resort</td>
<td>390800</td>
</tr>
<tr>
<td>2002</td>
<td>Albany</td>
<td>Camping</td>
<td>8790</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>Bed &amp; Breakfast</td>
<td>55</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>115800</td>
</tr>
<tr>
<td>2003</td>
<td>Berkeley</td>
<td>Bed &amp; Breakfast</td>
<td>4560</td>
</tr>
<tr>
<td>2003</td>
<td>Berkeley</td>
<td>Resort</td>
<td>419000</td>
</tr>
<tr>
<td>2003</td>
<td>Albany</td>
<td>Camping</td>
<td>7650</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
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
Communication between intranets encrypted.
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**
Client - Server Computing
Client Server Example

Client

“I want to see www.google.com”

Server

<html><head><meta http-equiv="content-type" content="text/html; charset=UTF-8"><title>Google</title><style>!--body,td,a,p,.h{font-family:arial,sans-serif;}
 .h{font-size: 20px;}
 .q{color:#0000cc;}
//-->

...
3-Tier Client Server Architecture example

Client

Balance $0.50

Application Server

What is Bob’s balance?

$0.50

Clicks, keystrokes

What is Bob’s balance?

Shared data

Shared data

$0.50
3-Tier Client Server Architecture example

Client

Application Server

Web Server

Common Gateway Interchange

Application Logic

Shared data
3-Tier Client Server Architecture example

Client

Application Server

Web Server

Common Gateway Interchange

Application Logic

What is Bob’s Balance?

Database Management System (DBMS)

Database

Shared data
3-Tier Client Server Architecture example

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

What is Bob’s Balance?

Database Management System (DBMS)
## 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**

**Client**
- Accept instructions from user
- Make requests of server
- Display responses of server

**Application Server**
- Takes inputs from client
- Decides what to be done next
- Decides what shared data to access and manipulates it
- Processes shared data

**Shared data**
- Support multiple applications with common data
- Protect critical data
- Decouple data administration and application administration
Book distribution centers

Customers

books4u.com

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
Slide adapted from slides for *Understanding Networked Applications* by David G Messerschmitt. Copyright 2000. See copyright notice.
Peer to peer

Slide adapted from slides for *Understanding Networked Applications* by David G Messerschmitt. Copyright 2000. See copyright notice.
Peer to Peer

- What is peer to peer good for?
Sun Case
Sun N-tier case

- What does Sun make?
  - Workstations
  - Servers
  - Software
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

Exhibit 1  Three-tier Architecture

Tier One

Client Applets
Client Hardware

Tier Two

Applets
App Server

Tier Three

Database

JDBC

These had to be managed locally. If code needed to be updated, each app server had to be shut down, updated and rebooted.

HTTP
RMI

Client Applets
Client Hardware

JDBC: Stands for Java Database Connectivity. It is a programming interface that lets Java applications access a database via the SQL language; RMI: Stands for Remote Method Invocation. It is the method by which a remote Java object from one location can be invoked from other Java virtual machines. HTTP: Stands for HyperText Transport Protocol. It is the communications protocol used to connect to servers on the World Wide Web.
Sun N-tier

Asia

Tier Four

Tier Three

High latency servlet talks back & forth

United States

Tier Two

Webtop Server

1st time Servlet sent

Webtop Server

1st time, applet sent

Tier One

Client

Client

Webtop Server

Webtop Server

Webtop Server

Webtop Server

Client

Client

Client

Europe

app server talks to central database

If bug found, change code here. Next time, corrected applet is sent down
Sun N-Tier

**Step 1:** The user logs into his client and calls down an application. This message is sent to the Application Server.

**Step 2:** An initial applet is sent to the client. At the same time a servlet is sent to the Webtop Server.

**Step 3:** The applet talks back and forth with the Webtop Server via the LAN.

**Step 4:** As new data is received (i.e., a new customer's name) the App Server communicates with the database to update that information.

**Remote:** The database and App Server communicate with the Webtop Server via a WAN.

**Local:** The Webtop Server and client communicate via a LAN.

**Exhibit 3** How the N-tier Architecture Works
Sun’s Performance

Net Revenue

Net income

Net Revenue

Net income
Sun’s Performance

The chart shows the trend of total assets and long-term debt for Sun from 1994 to 2007. The vertical axis represents the value in thousands, with values ranging from 0 to 20,000. The years 1994 to 2007 are labeled along the horizontal axis.
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
Today

- **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...*