Announcements

- **Final Exam December 7**
  - Thimann Lecture Hall 001 (this room)
  - 8am – 11am
  - Review Guide to be posted tonight.
Markup languages
A **markup language** describes the structure of a document

- Based on tags
- Tags denote structural elements like sections, subsections, figures, etc

Internationally standardized, so application independent
Example: HTML

<html>
<h1> Super Widget </h1>
<h2> Widgets Incorporated</h2>
<em>123456789</em>
<br>
<p>$300</p>
</html>

Super Widget

Widgets Incorporated

123456789

$300
Example: XML

Tags Emphasize what the things *mean* rather than how to *format* their Presentation.

```
<xml>
  <product>
    <model> Super Widget </model>
    <make> Widgets Incorporated </make>
    <sku> 123456789 </sku>
    <price> $300 </price>
  </product>
</xml>
```
XML in Ecommerce example

<xml>
  <product>
    <model> Super Widget </model>
    <make> Widgets Incorporated </make>
    <sku> 123456789 </sku>
    <price> $300 </price>
  </product>
</xml>

Stuff4U

<table>
<thead>
<tr>
<th>Product</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Super Widget</td>
<td>$300</td>
</tr>
<tr>
<td>Amazing Gadget</td>
<td>$500</td>
</tr>
</tbody>
</table>

Supplier

Retailer

Product info
From each Supplier sent in XML

Consumer
XML in ecommerce example 2

From each Supplier sent in XML product info, the Super Widget recognized and managed by SCM software.
Family lineage

- SGML
  - Standardized in mid 80s by ISO
  - Proposed in mid 90s

- HTML
  - Emphasizes formatting and presentation of documents
  - Introduced in Early 90s

- XML
  - Emphasizes structure of documents
  - Purpose- and industry-specific extensions
Domain Names
Domain Names

IP addresses are inconvenient for people

- 32 bits hard to remember
- 128 bits very hard to remember

Domain names

- e.g. argus.eecs.berkeley.edu

- Easier to remember than IP addresses

- However, we need some way of mapping domain names to IP addresses.
Domain Name System (DNS)

- Root Name Server
  - Berkeley Name Server
    - EECS Name Server
  - UCSC Name Server
    - SoE Name Server
Hierarchy in Addresses vs. Names

Addresses hierarchical in topology
- Maximize “wild cards” and distribute address administration

Names hierarchical in administration
- Single administered organizations often distributed topologically (e.g. ibm.com)
AA Case

Student Talk: Chi Hou Ip
American Airlines Case—Systems Operations and Control center (SOC)

- **Flight Dispatching**
  - Focus of case
  - Flight Path
  - Fuel Load
  - En route weather, problems
  - Each dispatcher assigned a geographic area

- **Load Planning**
  - Optimize loading of passengers and freight
  - Consider runway length, weather, plane type, etc.

- **Crew Scheduling**
  - Crews under strict regulations about amount of time can work
  - Certain crews can fly certain planes
  - Seniority
  - Positioning for future flights
Dispatch Automation Package

- **Flight tracking application**
  - View 1
    - List of all flights dispatcher responsible for
  - View 2
    - Dependencies of one flight on other flights.

- **Message tracking**
  - e-mail to flight crew
<table>
<thead>
<tr>
<th>CRIT</th>
<th>C/P</th>
<th>F/A</th>
<th>A/C O</th>
<th>TYPE</th>
<th>FROM</th>
<th>FLT/DA</th>
<th>DPT</th>
<th>SKED</th>
<th>LAT</th>
<th>L AVL</th>
<th>LAT</th>
<th>S DLAY</th>
<th>QPAK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ht1</td>
<td>Ht1</td>
<td>495</td>
<td>SP60</td>
<td>1655</td>
<td>1459/11</td>
<td>OMA</td>
<td>1355</td>
<td>1406</td>
<td>ORD</td>
<td>1517</td>
<td>0</td>
<td>2 53</td>
<td></td>
</tr>
<tr>
<td>Orig</td>
<td>Orig</td>
<td>517</td>
<td>SP80</td>
<td>Thru</td>
<td>632/11</td>
<td>STL</td>
<td>1415</td>
<td>1413</td>
<td>DFW</td>
<td>1607</td>
<td>C</td>
<td>5 2 54</td>
<td></td>
</tr>
<tr>
<td>Orig</td>
<td>Orig</td>
<td>495</td>
<td>SP80</td>
<td>Thru</td>
<td>1226/11</td>
<td>DFW</td>
<td>1412</td>
<td>1417</td>
<td>BOS</td>
<td>1700</td>
<td>10</td>
<td>2105</td>
<td></td>
</tr>
<tr>
<td>345</td>
<td>345</td>
<td>110</td>
<td>7271</td>
<td>783</td>
<td>1640/11</td>
<td>DFW</td>
<td>1418</td>
<td>1446</td>
<td>ORD</td>
<td>1755</td>
<td>C</td>
<td>-15 2 54</td>
<td></td>
</tr>
<tr>
<td>991</td>
<td>991</td>
<td>495</td>
<td>SP80</td>
<td>361</td>
<td>189/11</td>
<td>LAX</td>
<td>1422</td>
<td>1426</td>
<td>ORD</td>
<td>1817</td>
<td>0</td>
<td>3 79</td>
<td></td>
</tr>
<tr>
<td>600</td>
<td>1012</td>
<td>517</td>
<td>SP80</td>
<td>901</td>
<td>632/11</td>
<td>STL</td>
<td>1435</td>
<td>1447</td>
<td>DFW</td>
<td>2007</td>
<td>C</td>
<td>5 3 32</td>
<td></td>
</tr>
<tr>
<td>444</td>
<td>444</td>
<td>279</td>
<td>SP80</td>
<td>1832</td>
<td>1216/11</td>
<td>DFW</td>
<td>1435</td>
<td>1445</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ht1</td>
<td>Ht1</td>
<td>464</td>
<td>7271</td>
<td>1655</td>
<td>1610/11</td>
<td>DFW</td>
<td>1439</td>
<td>1445</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Status: Last update time 16:03:04 GMT

Note: This figure is a reconstruction, all data is fictional.
View 2:
"Passenger Service System" (Does Reservations)

“Fare Pricing Complex” Responsible for updating Fares (3 mainframes)

“Flight Operating System” Maintains Critical Data 1-Mainframe

Dispatcher Workstations

File Servers (Cache FOS data every 2 min)
Flight Tracking Application

- Fall 90 - Built as prototype as a way for someone to teach himself OOP

- May 91 - OK to develop application

- Work divided (one person in charge of each)
  - User interface
  - data model
  - data exchange with FOS

- Nov 91 - production installation complete

- 2 months testing
Flight Tracking Application

- **Facts and Figures:**
  - written in Smalltalk
  - 210 classes
  - 2000 methods
  - 160000 lines of assembler code
  - 150000 object instances in memory at all times
Flight Tracking Application

- OOP + good architecture made 3 changes easier
  - Changed how flight was referenced, major change to data model
    - (1.5 weeks)
  - Introduced File servers to cache FOS data
    - (1 day, 4 weeks test)
  - Developed message queuing monitor
    - (1 wk, test 3 wk)
Flight Tracking Application

- **Good architecture allowed extensions later**
  - Feature to allow dispatcher to focus on very really late flights
  - Flight lock - stop flights to airport for bad weather
  - In flight fuel calculation
  - Geographical flight monitor
Did AA follow Application Lifecycle Model?

Fall 90 - Built as prototype as a way for someone to teach himself OOP

May 91 - OK to develop application

Work divided (one person in charge of each)
- User interface
- data model
- data exchange with FOS

Nov 91 - production installation complete

2 months testing

Extensions added Later

1) CONCEPTUALIZATION
2) ANALYSIS?
3) ARCHITECTURE
4) DEVELOPMENT
5) TESTING
6) DEPLOYMENT
7) OPERATION, EXTENSION, MAINTENANCE
Final Review

- Comprehensive
- More emphasis on 2\textsuperscript{nd} half of class.
IT History

What distinguished the Data Processing Era from the Micro-Era?

• What distinguished the Micro-Era from the Networking Era?

• Why did Data Processing Managers feel threatened in the Micro-Era?
Messerschmitt 2.3

- Data vs. Information

- Information Pull vs. Information Push
What is Porter’s Competitive model?
- 5 forces: ...

What are the Porter Strategies
- Primary – Cost Leadership, Differentiation
  - Why must a company do at least one of these?
- Supporting – Alliance, Growth, Innovation

Concepts:
- Switching costs, barrier to entry,
- business function, business process,
- business process reengineering,
- knowledge vs information, ...
Messerschmitt Ch 3

- ERP
- Decision Support Systems
- Electronic commerce
  - Inter-enterprise B2B
  - Consumer B2C
  - Inter Consumer C2C
  - Examples of each?
- Supply Chain Management
- Mass Customization
- Electronic Data Interchange
Information represented as data
Regeneration
Software Layering
  - OS, Middleware, Application

File System
Database Management System
Message, Packet,
internet, Internet, intranet, extranet
Cash Flow Analysis

- **Net Present Value**
  - What is it?
  - How do you Calculate it?

- **Rate of Return**
Client Server Computing

- What is a client-server architecture?
- What is a three-tier client server architecture?
- What is a peer to peer architecture?
- What is a thin client?
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
- Support multiple applications with common data
- Protect critical data
- Decouple data administration and application administration

Shared data
Modularity and Layering

- **Modularity**
  - Why is Modularity a good way to architect things?
    - Separation of concerns
    - Interoperability
    - Reusability
- **Is Layering the same as Modularity?**
- **What is Hierarchy?**
- **What are Interfaces?**
  - External view of a subsystem, defines how other subsystems can interact with it. Often done using, actions, parameters, and returns.
- **What is Abstraction?**
  - Hide irrelevant detail at the interface of a subsystem
  - Helps manage complexity
- **What is Encapsulation?**
  - Enforce abstraction, by making subsystem details inaccessible at the interface
Computers and Communication Industries

- Infrastructure vs Application
- Decomposition vs Assembly
- System Integration
  - What is it?
  - Can it be outsourced?
- Stovepipe vs Integrated infrastructure
  - Examples? What are the tradeoffs of each approach?
- Standardization
  - Why are they needed?
  - Why do companies participate?
  - Can a reference model be part of a standard?
  - Should an implementation be part of a standard?
  - What’s an open standard?
  - What are network effects?
Data Sharing

- What is a DBMS?
- What is a relational database?
- SQL
- Markup Languages
  - What is the difference between HTML and XML?
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.
Applications and the Organization

- What are the advantages and disadvantages of
  - Buying off the shelf
  - Outsourcing
  - Making yourself

- What are the 7 steps in the application lifecycle model?
Application Architecture

- What is Object Oriented Programming
  - What are methods?
  - What are attributes?
  - What are instances?
- What are software components?
- What are component assembly tools?
- What are software frameworks?
Networks

- What are the 7 OSI Layers?
- Ethernet belongs to which layer?
- What is the difference between a MAC address and a IP Address?
- What is the difference between Time Division Multiplexing and Statistical Multiplexing?
- How does routing work on the Internet?
- How is an Internet Packet sent over an Ethernet?
- What does TCP do?
Link and Network Layer Interaction

1. **MAC Header**
2. **Ethernet Frame**
3. **Payload**
   - **IP Header**
   - **IP Payload**

- **Host A**
- **Host B**
- **Host C**

- **Ethernet Hub**
- **Router**

**128.114.60.202**

**MAC address**: 00-A4-B7-34-57-23

- Strip MAC header off frame.
- Forward IP packet based on Routing table.
Frito-Lay

What were the main reasons why Frito Lay deployed the HHC?

What changes in marketing strategy did Frito-Lay believe the HHC data would help enable?

How might the HHC project change Frito-Lay’s competitive position with its direct competitors, new entrants, and its customers?
What are some of the actions that Cisco took that contributed to the successful deployment of ERP?

What mistakes did Cisco make?

What are the most important lessons that another company that wants to deploy ERP could learn from Cisco’s experience?
Alibris

How and why did Alibris plan to change Interloc’s revenue model?

How did Alibris plan to avoid becoming dis-intermediated?

Why did Alibris abandon Thunderstone software, and why did it choose to switch Oracle?

What made Alibris’ IT challenge particularly difficult compared to what other ecommerce companies faced?
Sun Case

What are the drawbacks of having thin-client vs. a traditional fat-client?

What selfish reasons does Sun have for advocating a thin-client model? Why does Microsoft prefer maintaining the dominance of the fat-client model?

What is Java, and what advantages does it have over other languages?

Since the case was written, which OS has come to dominate the web-server market?
MySQL Case

Who are the three biggest suppliers of DBMS? What competitive advantages does MySQL have in the Web Site data segment?

Why would large enterprises prefer database software from one of the three major DBMS providers, over MySQL’s cheaper product?

• What is a General Public License (GPL)? Why were MySQL’s customers willing to pay for the product, when they could get the product for free under a GPL?
What is a Content Distribution Network (CDN)? What does it provide over ordinary web caching?

Where did Akamai locate its servers? What barriers to entry existed for a new entrant to build a CDN to compete with Akamai?

Did Akamai choose to market its products with a direct sales force or through distribution partners? What are the advantages of each choice?

Why did Akamai’s marketing strategy have to change when they transitioned from the Free Flow product to the Edge Suite product?
American Airlines Case

How did the dispatch automation package assist the flight dispatchers?

What were some of the benefits of good architecture and Object Oriented Programming in the context of the flight dispatch automation package?