Class announcements

- Tuesday:
  - Messerschmitt Ch 18 (493-512)

- For Thursday 3/4:
  - Database assignment due

Student Presentation

Stovepipe vs. Integrated Infrastructure

**Stovepipe Architecture**

--- or ---

**Turnkey Solution**

- Single supplier provides all encompassing solution
- (complete with infrastructure)

**Integrated Infrastructure**

- Separate infrastructure that can support many applications

From stovepipe to layering

- Data
- Voice
- Video

Application-dependent infrastructure

Many applications

Integrated Infrastructure
(Maybe broken into Additional layers.)

Application-independent

Stovepipe vs. Integrated Infrastructure

- What are some examples of each?

- What are the advantages of each approach?
**Vertical Integration vs. Diversification**

- A company is **vertically integrated** when it makes rather than buys the subsystems in its products.
- A **diversified** company produces products across different industry segments.

**Vertical Integration vs. Diversification**

- Why do customers favor less vertical integration?
  - Prefer competition amongst component suppliers
  - Mix and match components
  - Reduced lock in
- Disadvantages??
  - Customer needs to integrate components from different suppliers.

**Vertical Integration vs. Diversification**

- Why do customers favor diversification?
  - Reduce coordination costs by having to deal with fewer suppliers.

**General Trend**

- Less Vertical Integration
- More Diversification
- Of course there are exceptions...

**Today’s supplier structure**

- Applications
- Frameworks and components
- Middleware
- Infrastructure (network, OS, software)
- Equipment (network, computers)
- Semiconductors, components

**Standardization**
Purpose of a standard?

- Allow products or services from different suppliers or providers to be interoperable

Scope of a standard

Included:

- interfaces (physical, electrical, information)
- architecture (reference model)
- formats and protocols (FAP)
- compliance tests (or process)

Excluded:

- implementation
- (possibly) extensions

Reference model

Decide decomposition of system

- where interfaces fail

Defines the boundaries of competition and ultimately industrial organization

- competition on the same side of an interface
- complementary suppliers on different sides
- hierarchical decomposition at the option of suppliers
- (possibly) optional extensions at option of suppliers

Some issues

Once a standard is set

- becomes possible source of industry lock-in: overcoming that standard requires a major (~10x?) advance
- may lock out some innovation

In recognition, some standards evolve

- IETF, CCITT (modems), MPEG
- backward compatibility

Types of standards

de jure

- Sanctioned and actively promoted by some organization with jurisdiction, or by government

de facto

- Dominant solution arising out of the market
- Voluntary industry standards body

Industry consortium

Common or best practice

Examples

de jure

- GSM, ISDN Telephone interface

de facto

- Microsoft Windows API (Application Programming Interface)
- Intel Pentium instruction set
- IEEE (Institute of Electrical and Electronic Engineers)
- IETF (Internet Engineering Task Force)
- W3C (World Wide Web Consortium)
- SET (Secure Electronic Transactions)
- Best practice
- Windows GUI
The changing process

- As technology and industry move more quickly, the global consensus standards activity has proven too unwieldy
  - e.g. ISO
- "New age" standards activities are more informal, less consensus driven, a little less political, more strategic, smaller groups
  - e.g. OMG, IETF, ATM Forum, WAP
- Programmable/extendible approaches for flexibility
  - e.g. XML, Java

Reasons for change

- From government sanction/ownership to market forces
  - Increasing fragmentation
  - Importance of time to market
- Greater complexity
  - Less physical/performance constraint for either hardware or software

Lock-in

- (Particularly open) standards reduce consumer lock-in
  - Consumers can mix and match complementary products
- Increase supplier lock-in
  - Innovation limited by backward compatibility
  - e.g. IP/TCP, x86, Hayes command set

Aside: Network Effects

- The value of owning some products goes up if lots of other people have it too.
  - Examples?
- This phenomenon is called "network effects"
- How do standards influence network effects?

Network effects

- Standards can harness network effects to the industry advantage
  - Revenue = (market size) x (market share)
- Increases value to customer
- Increases competition
  - Only within confines of the standard
  - But forces customer integration or services of a system integrator
**Why standards?**

de jure are customer driven to reduce confusion and cost
de facto standards are sometimes the result of positive feedback in network effects
Customers and suppliers like them because they
- increase value
- reduce lock-in
Governments like them because they
- promote competition in some circumstances
- May believe they can be used to national advantage

**Approaches**

- Consensus
  - ISO
- Collaborative design
  - MPEG
- Competitive "bake off"
  - IETF
- Coordination of vendors
  - OMG

**Open vs. Proprietary Standards**

- Open standard – a standard that is well documented, unencumbered by intellectual property rights and restrictions, and available to any vendor.
- What are the advantages?
- What are the disadvantages?

**Why companies participate**

- Pool expertise in collaborative design
  - e.g. MPEG
- Have influence on the standard
  - Get technology into the standard
  - Proprietary, with expectation of royalties
  - Non-proprietary
  - Reduced time to market

**Standards applied to Business Processes?**

- Can you standardize business processes?
  - Yes!
    - ISO 9000
    - A set of standardized business processes for Quality Management
    - Supports TQM (Total Quality Management)
  - RosettaNet
    - A set of standardized business processes, and accompanying standardized data interfaces/formats for conducting e-business

**Databases**

by

David G. Messerschmitt
Databases

Treat data as a separate asset
- May be shared by multiple applications
Provide protection and integrity features appropriate to mission-critical data
- Access control
- Integrity constraints
- Persistence
- etc.

Two capabilities

SQL interface

- SQL (Structured Query Language)
- Presents single abstract interface to the application logic
  - For manipulating, and extracting data from database
- Standardized, not vendor specific
- Encapsulates various internal details
  - Data partitioning and replication
  - Host mapping
  - File representation
  - etc.

Relational table

<table>
<thead>
<tr>
<th>Table</th>
<th>Employee</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Address</td>
<td>Dept</td>
</tr>
<tr>
<td>Record</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yield</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Database operations

Each operation results in a new table

Multiple tables

“PROJECT” “SELECT”
Database Operations

<table>
<thead>
<tr>
<th>Passengers</th>
<th>Departments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Dept ID</td>
</tr>
<tr>
<td>Alice</td>
<td>1</td>
</tr>
<tr>
<td>Bob</td>
<td>1</td>
</tr>
<tr>
<td>Chris</td>
<td>2</td>
</tr>
</tbody>
</table>

JOIN

Object-relational database

* A column can store object instances of a given class rather than data of a given simple or compound data type
* Because of the table structure, SQL can be extended to this case
* Standard SQL queries can be extended to methods returning simple data types
* Many other good ideas

Benefits of ORDBMS

Extension: manage arbitrarily complex data types
Migration: preserve and extend existing databases
Preserve SQL interface
- OR extensions in latest standard
All the benefits/experience of earlier databases
- Access control, data integrity, persistence, etc.
Killer app: Behind Web/CGI
- Images, video, audio, animation, applets, etc.

Definition

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
title='Super Widget'
Widgets Incorporated

123456789
$100
```

**Example: XML**

```
<product>
  <model>Super Widget</model>
  <maker>Widgets Incorporated</maker>
  <sku>123456789</sku>
  <price>$100</price>
</product>
```

**XML in Ecommerce example**

![XML in Ecommerce example](image)

**XML in ecommerce example 2**

```
XYZ Manufacturing
```

**Family lineage**

- SGML: Standardized in mid 80s by ISO
- HTML: Emphasizes formatting and presentation of documents
- XML: Emphasizes structure of documents
- Purpose and industry-specific extensions

**Break!**
MySQL Quiz!!!

Take out a sheet of paper and answer these questions:

1) What does MySQL make?

2) In 2003, MySQL formed an alliance with what major enterprise software company?

3) According to the case, what operating system was leading the open source software movement?
   a) Linux  b) Windows  c) MAC OS  d) DOS

MySQL

What does MySQL make?

How Successful is MySQL?
- Visibility: Fortune magazine, more mentions on www
- Reaction from giants
- Revenue growth 2001 700k, 2002 6.2m, 2003 10m
- Good performance reviews
- Recent SAP alliance
- But Market share tiny
  - $10 million out of $10 billion market!

Why Success?
- Good Technology
- Large DBMS's forgotten with features most don't need
- Innovative OSS model

MySQL

How does OSS work?

Two Types of License:
- GPL
  - Free
  - No Support
  - Any software that uses MySQL as a module must itself be made GPL

Commercial License
- Support
- Could be distributed with non-open source software
- Not Free:
  - MySQL Classic $250, Pro $495 (for ~ 50 users)
  - Compare to:
    - MSFT $1800 single price for 50 users
    - IBM $30000 single price for 50 users
    - Oracle $40000 single price for 50 users

Aside: DBs in different software stacks

<table>
<thead>
<tr>
<th>General Software Stack</th>
<th>ERP Software Stack</th>
<th>Web Application Software Stack</th>
<th>Banking Software Stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Middleware</td>
<td>Oracle or MySQL, IBM, etc.</td>
<td>Apache Web Server</td>
<td>Proprietary Banking App.</td>
</tr>
<tr>
<td>Operating System</td>
<td>MS Windows or other OS</td>
<td>Linux or other OS</td>
<td>IBM z/OS or other OS</td>
</tr>
</tbody>
</table>

- Which companies are competitors?
- Which are complainers?
- Which are both?
mySQL

- Which segments of market is mySQL strong in?
  - Large Companies or Small Companies?
  - Web applications or Critical Enterprise data?

- Why would a major enterprise want to pay so much more for an Oracle or IBM DB?

**My SQL: market**

<table>
<thead>
<tr>
<th></th>
<th>Small 20%</th>
<th>Medium 30%</th>
<th>Large 50%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enterprise</td>
<td>Microsoft</td>
<td>Oracle IBM</td>
<td></td>
</tr>
<tr>
<td>web data</td>
<td></td>
<td>Reliability</td>
<td>Support</td>
</tr>
<tr>
<td>90%</td>
<td></td>
<td>Scalability</td>
<td></td>
</tr>
<tr>
<td>Web Sites</td>
<td>My SQL</td>
<td>Cost</td>
<td></td>
</tr>
<tr>
<td>10%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

How should mySQL grow in order to meet its stated goal of getting to $100 million in revenue?

**My SQL: Growth Strategy**

- Lack of Brand identity in this segment
- mySQL lacks the organization to offer support
- Large enterprises have high switching costs

- Not a big enough market to reach stated $100 million goal.

- Many of these customers already using mySQL with websites
- Last emphasis on global organization
- Leverage SAP alliance
- Up against Microsoft

- builds on existing brand and strengths
- Market not so big

- Not a big enough market to reach stated $100 million goal.
### Algorithms and protocols

Adapted from David G. Messerschmitt

### Algorithm

**Specified sequence of steps that**
- accomplish a designated task
- in a finite number of steps

**Representation:**
- simple algorithm: flowchart
- complicated algorithm: program

### Example: one turn at monopoly

```
Start 

Throw die 

Move taken number of squares indicated on die 

Land on "Jail" square? 

Yes 

No 

Move to "Jail" square? 

Yes 

No 

Do not move; follow policy for square (like "pay rent").

Finish turn.
```

### Algorithm building blocks

- **Start**
- **Action**
- **Decision**
- **Test**
- **Sequence**
- **Selection**
- **Loop**

### Protocol

- Distributed algorithm ...
- Realized by two or more modules to coordinate their actions or accomplish some shared task
- **Module interoperability requires a protocol**
  - Prescribed order of method invocations
  - Part of interface documentation

### Monopoly players protocol

```
Player 1
Player 2
```

```
One-turn algorithm
```

```
This is a protocol interaction diagram
```

```
Time
```
Application and infrastructure

The application defines its own application-level protocols.
Internally, the network uses protocols to implement the services it provides.

Example:

Layered Protocols Example

Three simple protocols

One-way message: send-receive
Two-way interaction: request-response
Push: publish-subscribe

Send - Receive

Request - Response
Send - Acknowledge

Locating things

by
David G. Messerschmitt

Example: HTTP (Hyper Text Transfer Protocol)

Three ways of locating things

Name
- "Joe Blie"
Address
- "1299 Hearst St, Berkeley, CA"
Reference
- "Postmaster of Berkeley CA"

Name
- Symbolic (character string) representation
- Easy for people to remember or guess
- Identifies, but
- Does not locate directly
  - Distinction important for mobile entities
- Not unique: entities can have more than one name (called aliases)

Hierarchical names

Hierarchy makes names easier to remember or guess
Host domain names:
- "info.sims.berkeley.edu"
  - designates administrative hierarchy
File names:
- "c:\My Documents\Docs\Resume.doc"
  - designates folder hierarchy
**Address**

- Route or path to entity
  - is directly specified, or
  - can be inferred
- Independent of who or what is locating entity
- Topological specification

**Example**

![Diagram of a grid with a path from one point to another](image)

Path from to is (R,D,D,D,R,R,R,R)

Is (R,D,D,D,R,R,R,R) an address?
No! — not an address, because it depends on starting point

**Reference**

Abstract representation of an entity

Interaction is with representation
- infrastructure arranges redirection to actual entity
- especially appropriate for things that move

Example
- A cell phone number is a reference.
- A wired phone number is an address.

**Name services**

1. Name
2. Address or reference
3. Interaction