Outline

- Announcements
- Components & Industry
- Student Presentation
- Standardization
- Student Presentation - MySQL Case
- MySQL
Class announcements

- Assignment 4 out today

- Thursday 2/24:
  - Eugene Ken Tsuchida Case: Akamai
  - Nicholas John Raimondi Bus Proj: Home Depot

- Reading for next class:
  - Ch. 15.3.1 - 15.3.3, 15.3.6, 15.4 of Messerchmitt (pp. 426-430, 432-437)
Today:
Timothy Lin - Costco
Andy Chan Case: MySQL
Feedback for Midterm

Histogram of Midterm

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's
28.50 64.38 75.50 73.47 83.00 94.00 1.00

Comments?
Components, Suppliers
A component implementation is encapsulated (although often configurable)

Component: A subsystem purchased “as is” from an outside vendor

(Alternative – building your own subsystem)
Supplier Types

- Three types of infrastructure/application suppliers:
  - Component Suppliers
  - Custom Subsystem Developers
  - System Integrators

- (Some suppliers are 2 or even 3 of above.)
Four possibilities (examples)

- **Application**
  - Microsoft Office
  - Microsoft Windows

- **Infrastructure**

- **Product**
  - Microsoft Office
  - Microsoft Windows

- **Service**
  - Hotmail
  - Internet DNS (Domain Name System)

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

- Bundled
  - An infrastructure provider bundles applications with their infrastructure
  - Example: AOL, telephony service providers

- Unbundled
  - A provider of an application service without providing an infrastructure service
  - Examples?
Examples of unbundled ASP model

- Web-based calendar (e.g. Yahoo, Google)
- Web-based email (e.g. Hotmail, Gmail)
- Web-based stock trading (e.g. Charles Schwab)
Unbundled ASP model

Advantageous to user

- Proven way to reduce installation, integration, and maintenance costs
- Contractual obligation for availability and quality
- Location independence

Slide adapted from slides for Understanding Networked Applications
By David G Messerschmitt. Copyright 2000. See copyright notice
Unbundled ASP model (con’t)

Advantages to supplier

- Ongoing revenue stream supporting upgrade and maintenance
- Usage-based revenue better aligned with user’s value proposition
- Opportunity for price discrimination, advertising revenue, etc.
Some pricing alternatives

Price discrimination?
Usage dependent?
Terms and conditions
  - fixed, leasing, per-use, subscription
  - warranty, service level agreements
Bundles
  - maintenance, support, releases, provisioning and operations
Who pays?
  - sometimes not the end user
Infrastructure acquisition

Infrastructure

- Build and operate
- Build but do not operate
- Do not build but operate
- Neither

Trend

Outsourced operations
System integrator
Service provider

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

Application

{ Develop internally  Buy as product  Contract development  Product w/ customization }

Software supplier

Outsource developer

Supplier, consultants
The changing industry structure
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

---

Slide adapted from slides for *Understanding Networked Applications*  
By David G Messerschmitt. Copyright 2000. See copyright notice
From stovepipe to layering

Data

Voice

Video

Many applications

Integrated Infrastructure
(Maybe broken into Additional layers.)

Application-dependent infrastructure

Application-independent infrastructure
Stovepipe vs. Integrated Infrastructure

- What are some examples of each?
  - Telephone network
  - Broadcast tv
  - Internet
  - Pc

- What are the advantages of each approach?
Two approaches for companies wishing to expand their product offerings

A company is *vertically integrated* when it makes rather than buys the subsystems in its products. Example: IBM

A *diversified* company produces products across different industry segments. Example: Compaq

Slide adapted from slides for *Understanding Networked Applications* by David G Messerschmitt. Copyright 2000. See copyright notice
Less Vertical Integration - More 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.

Slide adapted from slides for *Understanding Networked Applications* by David G Messerschmitt. Copyright 2000. See copyright notice.
Less Vertical Integration - More Diversification

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

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

Slide adapted from slides for *Understanding Networked Applications* by David G Messerschmitt. Copyright 2000. See copyright notice.
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
The Standardization Process

- Before something becomes a standard requires....
  - recognition of its need by a standards body/industry/government
  - commitment of monetary and human resources by participants

- Usually, is an ongoing process
  - Refinements/Extensions

Examples:

ISO: http://standards.iso.org/ittf/PubliclyAvailableStandards/
W3C: http://www.w3.org/
The Standardization Process

- Each organization participates in Working Groups/Committees of interest
  - Hold periodic meetings for debates/arguments/negotiation
  - When reaching a consensus, publish a RFC (Request for Comments) draft
  - Others can give feedback/Send comments etc.
  - The committee should answer to all comments and incorporate needed changes
  - Time-consuming process

- Results in extensive documentation and sometimes in system prototypes

- Usually standards evolve
  - Backward compatibility (e.g. MPEG)
  - Compatibility with existing standards (e.g. XQuery, XSLT based on XPath)
Some issues

- Slow and cumbersome process

- Once a standard is set
  - becomes possible source of industry lock-in; overcoming that standard requires a major advance
  - may lock out some innovation
Why do companies participate?

- Influence the standard
- Gain expertise and implement prototypes
  - Faster time to market than competitors
- Gain intelligence about competitors
  - That might be part of the standardization body as well
- May benefit financially through patent protection and royalties
  - Maintaining ownership of proprietary technology
- Many companies contribute their expertise to design something bigger
Types of standards

- **de jure**
  - Sanctioned and actively promoted by some standardization body, or by government

- **de facto**
  - Standard practice
  - Dominant solution arising out of the market, OR
  - Recommended by voluntary industry standards body

- Examples?
Examples

de jure

- GSM (global for mobile communication),
- ISDN (Integrated Services Digital Network) Telephone interface

de facto

- Windowed GUI
- Java
- Internet protocols

Voluntary industry standards body

- IEEE (Institute of Electrical and Electronic Engineers)
- IETF (Internet Engineering Task Force)

Industry consortium

- W3C (World Wide Web Consortium)
- SET (Secure Electronic Transactions)
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. W3C, IETF, WAP
- Programmable/extensible approaches for flexibility
  - e.g. XML, Java
Reasons for change

- From government sanction/ownership to market forces
  - Increasing fragmentation
  - Importance of time to market

Slide adapted from slides for *Understanding Networked Applications* by David G Messerschmitt. Copyright 2000. See copyright notice.
Open vs. Proprietary Standards

- **Open standard** - a standard that is well documented, unencumbered by intellectual property rights and restrictions, and available to any vendor
  - e.g. Internet protocols

- What are the advantages?

- What are the disadvantages?
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.
  - BPEL (Business Process Execution Language)
    - An XML-based language for the formal specification of business processes and business interaction protocols
The role of Venture Capital in Computing

- **Start-Ups**: Open interfaces allow small firms to contribute components without having to develop entire solution

- **High risk for VCs**

- **Diversification**
  - Each VC funds multiple start-ups
  - Each start-up funded by multiple VCs

- Is this model successful? For the start-up? For the VC? For the customers? Why?
Why standards?

- *de jure* are customer-driven to reduce confusion and cost
- *de facto* are sometimes the result of positive feedback in current practices

Customers like them because they
- Improve features and quality
- Reduce costs
- Reduce consumer lock-in

Suppliers like them because they
- Expand their market
- Increase revenues/profits
Open Standards

- **Reduce consumer lock-in**
  - Consumers can mix and match complementary products
  - Improved features and quality/Reduced costs

- **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?
mySQL Case
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 bloated 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 $3150 single proc for 50 users
      - IBM $33000 single proc for 50 users
      - Oracle $40000 single proc for 50 users
Aside: DB’s in different software stacks

- General Software Stack
  - Operating System
  - Middleware
  - Application

- ERP Software Stack
  - SAP
    - Or Oracle, Axtapa, etc.
  - Oracle
    - Or MySQL, IBM, etc
  - MS Windows
    - Or other OS

- Web Application Software Stack
  - MySQL
    - Or other DB
  - Linux
    - Or other OS

- Banking Software Stack
  - Proprietary Business Logic
  - Apache Web Server
  - Proprietary Banking App.
  - Oracle
    - Or other DB
  - IBM z/OS
    - Or other OS

- Which companies are competitors?
- Which are complimenters?
- 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?

- How should MySQL proceed? What are the advantages/disadvantages?
### 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</td>
<td>IBM</td>
</tr>
<tr>
<td>wide data 90%</td>
<td></td>
<td>Reliability</td>
<td>Scalability</td>
</tr>
<tr>
<td>Web Sites 10%</td>
<td>MySQL</td>
<td>Cost</td>
<td></td>
</tr>
</tbody>
</table>

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

Figure Adapted from “Teaching Note for MySQL Open Source Database,” 6/1/04, Stanford GSB.
- Lack of Brand identity in this segment
- MySQL lacks the organization to offer support
- Large enterprises have high switching costs

Figure Adapted from “Teaching Note for MySQL Open Source Database,” 6/1/04, Stanford GSB.
My SQL: Growth Strategy

<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</td>
<td>IBM</td>
</tr>
<tr>
<td>wide data</td>
<td></td>
<td>Reliability</td>
<td>Scalability</td>
</tr>
<tr>
<td>90%</td>
<td></td>
<td>Support</td>
<td>Longevity</td>
</tr>
<tr>
<td>Web Sites</td>
<td>My SQL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10%</td>
<td>Cost</td>
<td>Stay Put?</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

Figure Adapted from “Teaching Note for MySQL Open Source Database,” 6/1/04, Stanford GSB.
My SQL: Growth Strategy

- Many of these customers already using MySQL with websites
- Less emphasis on global organization
- Leverage SAP alliance
- Up against Microsoft.

<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></td>
<td>Oracle</td>
</tr>
<tr>
<td>wide data</td>
<td></td>
<td></td>
<td>IBM</td>
</tr>
<tr>
<td>90%</td>
<td></td>
<td></td>
<td>Reliability</td>
</tr>
<tr>
<td>Web Sites</td>
<td>My SQL</td>
<td>Maybe?</td>
<td>Scalability</td>
</tr>
<tr>
<td>10%</td>
<td>Cost</td>
<td></td>
<td>Support</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Longevity</td>
</tr>
</tbody>
</table>

Figure Adapted from “Teaching Note for MySQL Open Source Database,” 6/1/04, Stanford GSB.
## My SQL: Growth Strategy

<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></td>
<td>Oracle IBM</td>
</tr>
<tr>
<td>Wide Data</td>
<td></td>
<td></td>
<td>Reliability, Scalability, Support, Longevity</td>
</tr>
<tr>
<td>Web Sites</td>
<td>My SQL Cost</td>
<td></td>
<td>Maybe?</td>
</tr>
<tr>
<td>10%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

Figure Adapted from “Teaching Note for MySQL Open Source Database,” 6/1/04, Stanford GSB.
Approaches

Consensus
- ISO

Collaborative design
- MPEG

Competitive “bake off”
- IETF

Coordination of vendors
- OMG