Class announcements

- For Next Class -- Thursday November 9th
  - Assignment 4 due
  - Read: MySQL Database Case
  - Student Presenters:
    - Desiree Trundy (MySQL Database case)
    - Johan Stenberg (Business Paper)

Industry Structure (cont’d)

Four possibilities

<table>
<thead>
<tr>
<th>Product</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Office</td>
<td>Hotmail</td>
</tr>
<tr>
<td>Application</td>
<td></td>
</tr>
<tr>
<td>Infrastructure</td>
<td></td>
</tr>
<tr>
<td>Personal computer</td>
<td>Internet DNS</td>
</tr>
</tbody>
</table>

Application Service Provider

- 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?

Student Presentations

- Zhuo H Yang
  - New Story
- Emily Herrick
  - Business paper: Southwest Airlines
- Gabriela Arreguin
  - Business Paper: Charles Schwab
Examples of unbundled ASP model

- Yahoo: Web-based calendar
- Hotmail: Web-based email
- Schwab: Web-based stock trading

Unbundled ASP model

Advantageous to user

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

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

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

Outsourced operations
System integrator
Service provider

Trend

Application acquisition

- Develop internally
- Buy as product
- Contract development
- Product w/ customization

Trend
Software supplier
Outsource developer
Supplier, consultants

Outsourced operations
System integrator
Service provider
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

<table>
<thead>
<tr>
<th>Data</th>
<th>Voice</th>
<th>Video</th>
</tr>
</thead>
</table>

Many applications

Application-dependent infrastructure

Application-independent

---

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

Role of Venture Capital in Computing.
- Open interfaces allow small firms to contribute components without having to develop entire solution
- Fast decision making and no supplier lock-in.
- Other Advantages?

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 fall
- 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?

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

- Application I
  - Aggregation: accessing multiple databases
- Application II
  - Sharing: two or more applications accessing the same databases
**Relational table**

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

**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.

**Database operations**

```
<table>
<thead>
<tr>
<th>Departments</th>
<th>Employees</th>
</tr>
</thead>
<tbody>
<tr>
<td>“PROJECT”</td>
<td>“SELECT”</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Each operation results in a new table.

**Database Operations**

```
<table>
<thead>
<tr>
<th>Passengers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>Alice</td>
</tr>
<tr>
<td>Bob</td>
</tr>
<tr>
<td>Chris</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Departments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>Alice</td>
</tr>
<tr>
<td>Bob</td>
</tr>
<tr>
<td>Chris</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

**Fields, columns, attributes**

- Entries are simple data types or compositions of those types
  - Integer, string, etc.
**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.

---

**MySQL Quiz!!!**

Take out a sheet of paper, and answer these questions:

1) Who are the top three companies in the database business (in terms of sales revenue)?

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 student talk**

- Desiree Trundy

---

**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

<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>MySQL or other DB</td>
<td>Oracle or other DB</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 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?

**MySQL: market**

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

**MySQL: Growth Strategy**

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

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

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

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

- **Market not so big**

<table>
<thead>
<tr>
<th>Enterprise-wide data 90%</th>
<th>Small 20%</th>
<th>Medium 30%</th>
<th>Large 50%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Sites 10%</td>
<td>Microsoft</td>
<td>Oracle, IBM</td>
<td>MySQL</td>
</tr>
</tbody>
</table>

- **Reliability**
- **Scalability**
- **Support**
- **Longevity**

- **Cost**

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

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