ISM 50 - Business Information Systems

Lectures 14 & 15

Instructor: John Musacchio

Guest Instructor: Patrick Mantey

UC Santa Cruz

November 7, 2006 (Tuesday)

November 9, 2006 (Thursday)
For Next Class -- Thursday November 9th

- Assignment 4 due
- Read: MySQL Database Case
- Student Presenters:
  - Desiree Trundy (MySQL Database case)
  - Johan Stenberg (Business Paper)
Student Presentations

- Zhuo H Yang
  - New Story

- Emily Herrick
  - Business paper: Southwest Airlines

- Gabriela Arreguin
  - Business Paper: Charles Schwab
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>Internet DNS</td>
</tr>
<tr>
<td>Infrastructure</td>
<td>Personal computer</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?
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

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
- warrantee, service level agreements

Bundles
- maintenance, support, releases, provisioning and operations

Who pays?
- sometimes not the end user

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

Infrastructure

\{\text{Build and operate} \quad \text{Build but do not operate} \quad \text{Do not build but operate} \quad \text{Neither}\}

Trend

\{\text{Outsourced operations} \quad \text{System integrator} \quad \text{Service provider}\}

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

Application

\[ \begin{align*}
\text{Develop internally} & \quad \text{Buy as product} & \quad \text{Contract development} & \quad \text{Product w/ customization} \\
\text{Trend} & \quad \text{Software supplier} & \quad \text{Outsource developer} & \quad \text{Supplier, consultants}
\end{align*} \]
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

Many applications

Integrated Infrastructure (Maybe broken into Additional layers.)

Application-independent

Data Voice Video

Application-dependent infrastructure
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.

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

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

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

\textit{de jure}
- Sanctioned and actively promoted by some organization with jurisdiction, or by government

\textit{de facto}
- Dominant solution arising out of the market
- Voluntary industry standards body

Industry consortium
Common or best practice

Examples?

---

Slide adapted from slides for \textit{Understanding Networked Applications}  
By David G Messerschmitt. Copyright 2000. See copyright notice
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

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

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

Record

Field
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

Each operation results in a new table.

“PROJECT”

“SELECT”

Departments

Employees
### Database Operations

#### Passengers

<table>
<thead>
<tr>
<th>Name</th>
<th>Dept ID</th>
</tr>
</thead>
<tbody>
<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>

#### Departments

<table>
<thead>
<tr>
<th>Dept Name</th>
<th>Dept ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engineering</td>
<td>1</td>
</tr>
<tr>
<td>Sales</td>
<td>2</td>
</tr>
</tbody>
</table>

#### JOIN

<table>
<thead>
<tr>
<th>Name</th>
<th>Dept ID</th>
<th>Dept Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>1</td>
<td>Engineering</td>
</tr>
<tr>
<td>Bob</td>
<td>1</td>
<td>Engineering</td>
</tr>
<tr>
<td>Chris</td>
<td>2</td>
<td>Sales</td>
</tr>
<tr>
<td>Year</td>
<td>City</td>
<td>Accommodation</td>
</tr>
<tr>
<td>------</td>
<td>----------</td>
<td>------------------</td>
</tr>
<tr>
<td>2002</td>
<td>Oakley</td>
<td>Bed &amp; Breakfast</td>
</tr>
<tr>
<td>2002</td>
<td>Oakley</td>
<td>Resort</td>
</tr>
<tr>
<td>2002</td>
<td>Oakland</td>
<td>Bed &amp; Breakfast</td>
</tr>
<tr>
<td>2002</td>
<td>Oakland</td>
<td>Resort</td>
</tr>
<tr>
<td>2002</td>
<td>Berkeley</td>
<td>Camping</td>
</tr>
<tr>
<td>2002</td>
<td>Berkeley</td>
<td>Bed &amp; Breakfast</td>
</tr>
<tr>
<td>2002</td>
<td>Berkeley</td>
<td>Resort</td>
</tr>
<tr>
<td>2002</td>
<td>Albany</td>
<td>Camping</td>
</tr>
<tr>
<td>2002</td>
<td>Albany</td>
<td>Bed &amp; Breakfast</td>
</tr>
<tr>
<td>2003</td>
<td>Oakley</td>
<td>Bed &amp; Breakfast</td>
</tr>
<tr>
<td>2003</td>
<td>Oakley</td>
<td>Resort</td>
</tr>
<tr>
<td>2003</td>
<td>Oakland</td>
<td>Bed &amp; Breakfast</td>
</tr>
<tr>
<td>2003</td>
<td>Oakland</td>
<td>Resort</td>
</tr>
<tr>
<td>2003</td>
<td>Berkeley</td>
<td>Camping</td>
</tr>
<tr>
<td>2003</td>
<td>Berkeley</td>
<td>Bed &amp; Breakfast</td>
</tr>
<tr>
<td>2003</td>
<td>Berkeley</td>
<td>Resort</td>
</tr>
<tr>
<td>2003</td>
<td>Albany</td>
<td>Camping</td>
</tr>
<tr>
<td>2003</td>
<td>Albany</td>
<td>Bed &amp; Breakfast</td>
</tr>
</tbody>
</table>

- Entries are simple data types or compositions of those types
  - Integer, string, etc.
Object-relational database

<table>
<thead>
<tr>
<th>Column</th>
<th>SQL</th>
<th>Data Type</th>
</tr>
</thead>
</table>

- 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.
mySQL Case
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

- Which companies are competitors?
- Which are complimeners?
- 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?
<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></td>
<td></td>
<td></td>
<td>Scalability</td>
</tr>
<tr>
<td>Web Sites</td>
<td>My SQL</td>
<td></td>
<td></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>

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

- 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.
**MySQL: Growth Strategy**

- 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

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

- **Reasons:**
  - + 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.