ISM 50 - Business Information Systems

Lecture 15

Instructor: John Musacchio
UC Santa Cruz
November 18, 2008
Class announcements

Thursday:
- Assignment 4 due
- Messerschmitt Ch 18 (493-512)
- Talks
  - Jack O’Neil
  - Kian Fattahi

For Tuesday 11/25:
- Database assignment due
Student Presentation

David Kuepfer (Amazon)
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

Trend

- Software supplier
- Outsource developer
- Supplier, consultants
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
Standardization

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

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

Slide adapted from slides for 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

<table>
<thead>
<tr>
<th>Employee</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Address</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”
# 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.