ISM 50 - Business Information Systems

Lecture 13

Instructors: Mary Doyle and Geoff Ryder
UC Santa Cruz
May 13, 2008

Announcements
- Database project is out. New due date: June 1
- Sign up for one of the database project help sessions on one of the sheets before or after class
- Assignment 4 is out, (due Wednesday 5/20)
- Can pick up old assignments before or after class, or in OH
- For Monday, read:
  - MySQL case study, in the reader
- Today’s student presenters:
  - Matthew Villarreal, news story
  - Xuanyu Pang, news story
- Presenter for Monday:
  - Mario Demerling, MySQL case study

Today’s Class
- Student presentations
- Guest speaker: Ken Moll, Cisco Systems
- Business paper proposals and research
- Architecture: Modularity and Layering
- Computer and Communications Industries, Chapter 7 (continued)
- Algorithms and Protocols, Ch. 11
- (if time) Database Management, Ch. 15

Student Presentations

Matthew Villarreal

Xuanyu Pang

UCSC ISM 50
Primer on Cisco Systems
Business and Technology
May 13, 2009

Guest Speaker:
Ken Moll

BSEE-MIT, MBA-Berkeley Haas School of Business
Manager, Backbone Router ASIC Design Group

Cisco Overview
- 65,000 employees, world wide
- Revenue about $32B this year
- Known for router and switch products
- Now moving into “adjacent markets” with services, communications, data center, and even consumer products
- 133 Acquisitions
Routers

- Cisco’s first product, originally designed to bridge networks from different vendors
- Now almost all networks run TCP/IP, so router’s function has changed:
  - Core internet routing
  - Gateway to internet in enterprise networks
  - Residential internet connection
- Device IP address assigned by network administrator
- Can route between devices with different physical interface types
- Can support networks with billions of end points

Switches

- Steer traffic within a closed network
- Larger networks have thousands of endpoints
- Devices identified by MAC address, assigned by device manufacturer
- Devices typically have the same physical interface type
- Networks relative easy to set up and administer

Routers versus Switches

- Switches steer traffic based on end point devices’ fixed address
  - Non-scalable mechanism, like sending letters without a zip code
- Routers steer traffic based on devices’ assigned “geographic” network address
  - Logical address groupings, like ucs.edu and att.net
  - Hierarchical network grouping allow distributed routing
  - Routers can send traffic between devices with different physical interface types
- Summary: Switching is fast and simple, but not as scalable as routing

Cisco’s “Market Adjacencies”

- Strategy: Use existing customer relationships and core competencies to move into different markets
- Usually timed with a “market disruption”
- Recent examples:
  - IP Telephony
  - Telepresence
  - Webex conferencing
  - Data center

IP Telephony

- Strategy: Use IP networking expertise to enter the enterprise IP telephony market
- Disruption: evolution from TDM to packet based voice
- Acquire and internally develop the call manager, phones, and voice gateways
- Triggers upgrade cycle of higher capability switches and routers
- Over 7 million phones sold

Telepresence

- Reinvention of video conferencing
  - Conference room metaphor
  - “It’s all about the experience.”
- Disruptions: Dis-incentives to business travel
  - 9/11, SARS, Fuel Costs, Economic problems
- Triggers both customer and service provider equipment upgrades
Conferencing: Webex, and “Software as a Service”

- Unified conferencing: voice, video, document sharing, and other "collaboration tools"
- Platform for partners to develop and distribute applications, such as call center and customer support

Data Center: Virtualization and Cloud Computing

- "The next big thing."
- Single network to integrate servers, storage, and network connections
- Scalable “computing on demand”
- Migration to virtual machines reduces hardware costs, administration costs, and power consumption

Porter’s Five Forces, Cisco 3750

- **Barriers to Entry**:
  - Technical sales force
  - ASIC expertise
  - Large software team
  - Feature velocity prevents commoditization
- **Competition**:
  - Cisco, HP, 3Com, Nortel
  - Market leaders develop custom ASICs and software
  - Leverage customer relationships to shape the market
- **Buyers’ Power**:
  - Many customers & few suppliers
  - Fortune 1000 enterprises
  - Government and education service providers
- **Suppliers’ Power**:
  - Suppliers are both partners and competitors. (Broadcom and Marvell)
- **Substitutes**:
  - Lower cost unmanaged switches
  - Commodity copies of older generation products
- **Rivalry**:
  - Heavy use of Oracle consultants despite desire for no customization
  - Extreme control over suppliers: marquis account
  - IT as a strategic advantage
  - Investment versus cost

Observations on Cisco ERP Case

- **Self imposed urgency**
  - RFP written in ten days!
- **Customers drove features**
  - Specified no change in existing processes
  - Attempt to make new system function like old system
  - Heavy use of Oracle consultants despite desire for no customization
- **Extreme control over suppliers**: marquis account
  - **IT as a strategic advantage**
  - Investment versus cost

Text, Chapter 7: Computer and Communications Industries (Continued from Monday)

...this is closely related to our discussion of Chapters 4, 5, and 6: these industries have shaped themselves around the architectural concepts we’ve been talking about.
Recall four possibilities among applications vs. infrastructure; and products vs. service:

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

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

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

Trend

Outsourced operations
System integrator
Service provider

Application acquisition

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

Trend

Software supplier
Outsource developer
Supplier, consultants

Stovepipe vs. Integrated Infrastructure

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

- **Integrated Infrastructure**
  - Separate infrastructure that can support many applications

Application-dependent infrastructure
Application-independent infrastructure

From stovepipe to layering

- Many applications
  - Integrated Infrastructure
    - (Maybe broken into Additional layers.)

Application
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 it's 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...

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?

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 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

Examples

- Industry consortium
- Common or best practice

Examples?

- de jure
  - GSM, ISDN Telephone interface
- de facto
  - Hayes command set, Windows API, Pentium instruction set, Ethernet
  - Voluntary industry standards body
  - OMG/Corba, IAB/IETF, IEEE
  - W3C/XML, SET
  - 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/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
Greater complexity
  - Less physical/performance constraint for either hardware or software

Old giving way to the new

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

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 standards 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 lockin
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 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.

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 turn**
  - Throw dice
  - Move token number of squares indicated on dice

- **Yes**
  - Move to "jail" square
  - Land on "go to jail"?

- **No**
  - Do not move; follow policies for square (like "pay rent")

- **Finish turn**

**Algorithm building blocks**

- **Start**
  - Action

- **Decision**
  - Action

- **Test**
  - Action

- **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**
  - One-turn algorithm
- **Player 2**

**Application and infrastructure**

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

**Example:**

- HEADQUARTERS
  - Airline Intranet
  - Airline Datserver
- HHC
  - HHC Server
  - Wireless Link
  - HEADQUARTERS
  - HHC

**Programming languages support these three building blocks**
Layered Protocols Example

Three simple protocols

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

Send - Receive

Client

send

Server

receive

Request - Response

Client

request

Server

response

Send - Acknowledge

Client

send

Server

acknowledge

Example: HTTP (Hyper Text Transfer Protocol)

1. User activates URL
2. HTTP request
3. Browser displays document (if HTML) or invokes “helper application”
4. HTTP response (embedded document)
**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>
</tr>
</thead>
<tbody>
<tr>
<td>Employee</td>
</tr>
<tr>
<td>Name</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**

Each operation results in a new table
Multiple tables

Database Operations

Fields, columns, attributes

Object-relational database

Benefits of ORDBMS

Markup languages
**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>
  <h1>Super Widget</h1>
  <h2>Widgets Incorporated</h2>
  <em>123456789</em>
  <p>$300</p>
</html>
```

**Example: XML**

```
<xml>
  <product>
    <model>Super Widget</model>
    <make>Widgets Incorporated</make>
    <sku>123456789</sku>
    <price>$300</price>
  </product>
</xml>
```

Tags Emphasize what the things *mean* rather than how to *format* their Presentation.

**XML in Ecommerce example**

Supplier Stuff4U

```
<xml>
  <product>
    <model>Super Widget</model>
    <make>Widgets Incorporated</make>
    <sku>123456789</sku>
    <price>$300</price>
  </product>
</xml>
```

Product info From each Supplier sent in XML

Retailer Stuff4U

```
<xml>
  <product>
    <model>Amazing Gadget</model>
    <make>XYZ Manufacturing</make>
    <sku>987654321</sku>
    <price>$500</price>
  </product>
</xml>
```

Consumer

Super widget recognized and managed by SCM software.

**XML in Ecommerce example 2**

Supplier XYZ Manufacturing

```
<xml>
  <product>
    <model>Amazing Gadget</model>
    <make>XYZ Manufacturing</make>
    <sku>987654321</sku>
    <price>$500</price>
  </product>
</xml>
```

Product info From each Supplier sent in XML

**Family lineage**

- **SGML**: Standardized in mid 80s by ISO
- **HTML**: Proposed in mid 90s
- **XML**: Emphasizes structure of documents
- **SGML**: Emphasizes formatting and presentation of documents

- **XYZ Manufacturing**
  - Introduced in Early 90s
  - Super widget recognized and managed by SCM software.