ISM 50 - Business Information Systems
Lecture 12

Instructor: John Musacchio
UC Santa Cruz
November 1, 2005

Class announcements
- Assignment 4 out!
  Due next Tuesday (November 8th)
- Reading for Wednesday:
  - Chapter 11.1 - 11.2, 15.1-15.2
- Student Presentations Thursday
  - Keith Lucitt
  - Ryan Fargo

Student Presentations
- Melinda Hsieh
- Salvador Barrios

Suggested Research sources
- ABI/Inform
  - http://library.ucsc.edu/Zope/eresources/bytoo/l/ArticleDatabases
- 10K reports
  - http://www.morningstar.com/
  - Lookup company, click SEC filings

Architecture

Two ways to design a system
- System requirements
- Assembly from available components

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

Component: A subsystem purchased "as is" from an outside vendor

(Alternative – building your own subsystem)

A component implementation is encapsulated (although often configurable)

Component implementation

HHC Architecture

The Palm OS we are buying "off the shelf" and integrating into our architecture.
The Palm OS is a component.

HHC Application

Palm OS

Networking Infrastructure

Coordination With HHC Server

User Interface

Data Management

Interoperability

- Components are interoperable when they interact properly to achieve some desired functionality
- Increasingly component interoperability cannot be dependent on end-user integration
  - PC and peripherals
  - Enterprise, inter-enterprise, consumer applications
  - Role for standardization

Other Examples of components

Computer
Disk drive
Network
Network router
Operating system
Integrated circuit
Database management system

Why is a component implementation encapsulated?

Interoperability

Outsourcing

Outsourcing: A subsystem design is contracted to an outside vendor

Responsibility is delegated

Outsourcing

HHC Architecture

Suppose we choose to pay another firm to develop the user interface.
This is called Outsourcing.
Why would we do this?
Suppose we bring together all these subsystems and test them...

This is called **System Integration**

**System Integration**
- Bring together subsystems;
- make them work together;
- to achieve a goal.

**Requires**
- Testing
- Making modifications to
  - architecture and/or
  - subsystem implementation

---

Can System Integration be Outsourced?

- Of course!

---

**Supplier Types**

- Three types of suppliers:
  - Component Suppliers
  - Custom Subsystem Developers
  - System Integrators
- (Some suppliers are 2 or even 3 of above.)

---

**Two ways to sell Software**

- **Product**
  - Customer installed and operated
  - Often (but not necessarily) sold or licensed at a fixed price

- **Service**
  - Functionality provided over a wide-area network
  - Often (but not necessarily) sold by subscription

---

**Recall: Infrastructure and Applications**

**Infrastructure**
- Equipment and/or software used by many applications

**Applications**
- Provide specific capabilities and features serving individual users.
Four possibilities

<table>
<thead>
<tr>
<th>Product</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Office</td>
<td>Hotmail</td>
</tr>
</tbody>
</table>

Application Infrastructure

| Personal computer | Internet DNS |

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
  - warrantee, 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

### Application acquisition

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

### Stovepipe vs. Integrated Infrastructure

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

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

### From stovepipe to layering

- Data
- Voice
- Video

- Many applications

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

- Application-dependent infrastructure
- Application-independent

### 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...

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**

*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
- Industry consortium
- W3C/XML, SET
- Best practice
- 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/extendible 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

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 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 lock-in
- 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.

Midterm Results

- High: 97
- Low: 36

![Midterm Results Chart]