Outline For Today

- Class Announcements
- Student Presentations
- Business Paper Suggested Sources
- Cash flows
- Computer Industry Structure
Class announcements

- Assignment 4 due Wednesday

- Reading for Wednesday:
  - Chapter 11.1, 11.2, 15

- Student Presentations Wednesday
  - Moya, Antonio Julian
  - Shen, David
Student Presentations
Suggested Research sources

- **ABI/Inform**
  - [http://library.ucsc.edu/Zope/eresources/bytool/ArticleDatabases](http://library.ucsc.edu/Zope/eresources/bytool/ArticleDatabases)

- **10K reports**
  - Lookup company, click SEC filings
Infrastructure and Applications

Infrastructure
- Equipment and/or software used by many applications

Applications
- Provide specific capabilities and features serving individual users.
Two ways to design a system

Decomposition from 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)

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

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

Why is a component implementation encapsulated?

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

- Components are interoperable when they interact properly to achieve some desired functionality.
- Increasingly component interoperability cannot be dependent on integration, or is dependent on end-user integration.
  - PC and peripherals
  - Enterprise, inter-enterprise, consumer applications
  - Role for standardization
Outsourcing: A subsystem design is contracted to an outside vendor.

Responsibility is delegated.
System integration

Architecture --> subsystem implementation an/or component purchase → system integration

Bring together subsystems and make them cooperate properly to achieve desired system functionality

- Always requires testing
- May require modifications to architecture and/or subsystem implementation

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

- Three types of suppliers:
  - Component Suppliers
  - Custom Subsystem Developers
  - System Integrators
- (Some suppliers are 2 or even 3 of above.)
- Examples?
The task of combining components, and custom designed subsystems to achieve a higher-level system goal is **system integration**.

- Three types of suppliers:
  - Component Suppliers
  - Custom Subsystem Developers
  - System Integrators

  (Some suppliers are 2 or even 3 of above.)
Buy vs. Make

- Can either buy or custom design a subsystem
- A sub-system that is purchased is called a component
  - Components most likely to be available when functionality is common, well defined, and accepted.
  - Components are often highly configurable
    - Example: ERP application
Two ways to sell

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
Four possibilities

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

Slide adapted from slides for *Understanding Networked Applications* by David G Messerschmitt. Copyright 2000. See copyright notice.
Examples of unbundled ASP model

- Telephony
- Web-based information access and commerce
- 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

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

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

Application

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

- When a single supplier provides an all encompassing application solution (complete with infrastructure) it is called
  - A *stovepipe architecture*
  - A *Turnkey Solution*

- Alternative: separate infrastructure that can support many applications
  - *Integrated infrastructure*

---

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

- A *diversified* company produces products across different industry segments.

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 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.
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

Infrastructure:
- Allow products or services from different suppliers or providers to be interoperable

Application:
- Enable applications to run across uncoordinated administrative domains
Scope of a standard

Included:

- architecture (reference model)
- interfaces (physical, electrical, information)
- 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
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