ISM 50 - Business Information Systems

Lecture 15

Instructor: John Musacchio

UC Santa Cruz
November 12, 2009
Class announcements

- **Tuesday:**
  - Assignment 4 due
  - *Messerschmitt Ch 18* (493-512)

- **For Thursday 11/19:**
  - Database assignment due
Student Presentation
Another Interface Example:
Automatic teller machine (ATM)

What is the interface between this machine and the customer?
Steps

Define available actions

Define, for each higher level function, a protocol

- Single action or a finite sequence of actions
Interface building blocks

Message on screen or printed
- Menu of actions or returns from an action
- Touch selection of action

Keypad
- Input parameters to an action

Card reader
- Authentication, input parameters

Money output slot
- Returns money
Action: authentication

Parameters
- Identity (card in slot)
- Institution (card in slot)
- PIN (typed on keypad)

Internally, it contacts institution and matches against its database, institution noted for all subsequent actions (example of state)

Returns
- Screen message (“Invalid PIN” or menu of available actions)
Action: specify_account

Parameters

- Account (touch screen from menu of choices)

Internally, choice noted for all subsequent actions (another example of state)

Returns

- None
Action: amount

Parameters

- Dollars_and_cents (typed on keypad)

Internally, amount noted (another example of state)
Protocol: cash_withdrawal

What is the sequence of actions?
Protocol: cash_withdrawal

- authentication → failure
- choose objective → other objectives
- account → no accounts
- amount → balance exceeded!
More on layering

Slides modified from those by
David G. Messerschmitt
Example 1

Bob sends a letter to Alice

Bob

Envelope

US Postal Service

Shipping Container

ABC Airlines

Alice

Envelope

UK Royal Mail

Shipping Container
Interaction of layers

Layer above is a client of the layer below

Each layer provides services to the layer above...

....by utilizing the services of the layer below and adding capability

Layer below as a server to the layer above
Three types of software

Application

• Components and frameworks:
  What is in common among applications

• Infrastructure:
  Basic services (communication, storage, concurrency, presentation, etc.)
Major layers

- Applications
- Application frameworks and components
- Middleware
- Operating system
- Network
Data and information

Application
Deals with information

Assumes structure and interpretation

Infrastructure
Deals with data

Ignores structure and interpretation
Example 2

- Web server
- Web browser
- Screen
- HTML
- Application
  - File
  - Message
  - File system
- Operating system
- Network
  - Fragmentation
  - Collection of packets
  - Assembly
Package = file or message

Infrastructure deals with a package of data (non-standard terminology)
- collection of bits
- specified number and ordering

Infrastructure stores and communicates packages
- File for storage
- message for communication
Data integrity

Retain the
- values
- order
- number

of bits in a package
Data and information in layers

- The infrastructure should deal with data, or at most minimal structure and interpretation.
- The application adds additional structure and interpretation.
- This yields a separation of concerns.
Sometimes it is appropriate for the infrastructure to assume structure and interpretation for data

- to add capabilities widely useful to applications
- to help applications deal with heterogeneous platforms, where representations differ

At most, data types
Data and information

Application
Deals with information

Assumes structure and interpretation

Assumes standard data types

Infrastructure
Deals with data types
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.
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.
The Palm OS we are buying “off the shelf” and integrating into our architecture. The Palm OS is a component.
Other 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 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

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

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

<table>
<thead>
<tr>
<th>Product</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer installed and operated</td>
<td>Functionality provided over a wide-area</td>
</tr>
<tr>
<td>Often (but not necessarily) sold</td>
<td>network</td>
</tr>
<tr>
<td>or licensed at a fixed price</td>
<td>Often (but not necessarily) sold by</td>
</tr>
<tr>
<td></td>
<td>subscription</td>
</tr>
</tbody>
</table>

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

Infrastructure
- Equipment and/or software used by many applications

Applications
- Provide specific capabilities and features serving individual users.
Industry Structure (cont’d)
Four possibilities

Product | Service
--- | ---
Microsoft Office | Hotmail
Application | Internet DNS
Infrastructure | Personal computer

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

\[ \begin{align*}
\text{Build and operate} & \quad \text{Build but do not operate} & \quad \text{Do not build but operate} & \quad \text{Neither} \\
\end{align*} \]

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