TIM 50 - Business Information Systems

Lecture 12

Instructor: John Musacchio

UC Santa Cruz

May 8, 2014
Class Announcements

- Database Assignment 1 due TODAY
- Database Assignment 2 soon to be posted
  - Due 5/27
- Business paper draft due 5/20
Business paper proposals --
ISSUES

- CITE SOURCES IN TEXT!
  - Non Obvious facts and figures
    - Southwest airlines Cost per Available Seat Mile in 2009 was 12.3 cents.... CITE?
    - Netflix is a large company that seeks to maximize profits.... CITE?
Business paper proposals --

**ISSUES**

- Proofread!
- Industry identification in Porter model
  - “Netflix is in the entertainment industry”
  - Too broad? Too narrow?
- **IT section**
  - IT is important to Amazon.... They have a web site and lots of servers....
    - Enough detail?
IEEE style

In the body of the text you cite sources with a number in square brackets [1].

WWW

Basic Format:


Example:


Chicago system

- In the Chicago author date system, you cite with a parenthetical reference with author name date, and sometimes page (Smith 2002, 102).


Protocol

In addition to atomic actions, an interface may define protocols

- Protocol == finite sequence of actions required to achieve a higher level function

- One action can be shared by multiple protocols

- Multiple modules may participate in a protocol
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
Internal functionality
Returns
**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

Internal functionality

Returns
**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)*

**Returns**
- Success or failure (state dependent, for example for a withdraw failure when dollars_and_cents exceeds balance)
Protocol: cash_withdrawal

What is the sequence of actions?
Protocol: cash_withdrawal

1. authentication → failure
2. choose objective → other objectives
3. account → no accounts
4. amount → balance exceeded!
More on layering

by
David G. Messerschmitt
Goals

Understand better

- how layering is used in the infrastructure
- how it contains complexity
- how it coordinates suppliers
- how it allows new capabilities to be added incrementally
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 as a server to the layer above
Layering builds capability incrementally by adding to what exists.
Data and information

Application
Deals with information

Assumes structure and interpretation

Infrastructure
Deals with data

Ignores structure and interpretation
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.
In the simplest case, the infrastructure deals with a package of data (non-standard terminology):

- collection of bits
- specified number and ordering

The objective of the infrastructure is to store and communicate packages while maintaining data integrity.

File for storage, message for communication
Data integrity

Retain the

- values
- order
- number

of bits in a package
Example 1

Bob sends a letter to Alice

Bob

Envelope

US Postal Service

Shipping Container

ABC Airlines

Alice

Envelope

UK Royal Mail

Shipping Container
Example 2
Example 3: Network Infrastructure Expanded

- Seatback Application
  - Linux OS
  - TCP transport layer
  - WiFi Link Layer
  - WiFi Physical Layer
- Passenger Information
- Airplane Server
  - Linux OS
  - TCP transport layer
  - WiFi Link Layer
  - WiFi Physical Layer
- Networking Infrastructure
  - Radio Signals
  - Packets

Message flow:
- Seatback Application → Passenger Information
- Passenger Information → Airplane Server
- Airplane Server → Seatback Application
Information in the infrastructure

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
Computer & Comm. Industry Structure
Two ways to design a system

Decomposition from system requirements

Assembly from available components

Available components

System requirements

Requirements

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
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
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

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.
Four possibilities

<table>
<thead>
<tr>
<th>Product</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Office</td>
<td>Hotmail</td>
</tr>
<tr>
<td>Application</td>
<td>Internet DNS</td>
</tr>
<tr>
<td>Infrastructure</td>
<td></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.
Application Service Provider

- Two types
  - Bundled
    - An infrastructure provider bundles applications with their infrastructure
      - Example: Comcast, 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
- gmail: 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

Slide adapted from slides for Understanding Networked Applications
By David G Messerschmitt. Copyright 2000. See copyright notice
Unbundled ASP model (cont’d)

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.

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