The User Experience

Week 15

LBSC 671

Creating Information Infrastructures

Tonight

- Information architecture

- Human-Computer Interaction (HCI)

- Viruses and other nasty things
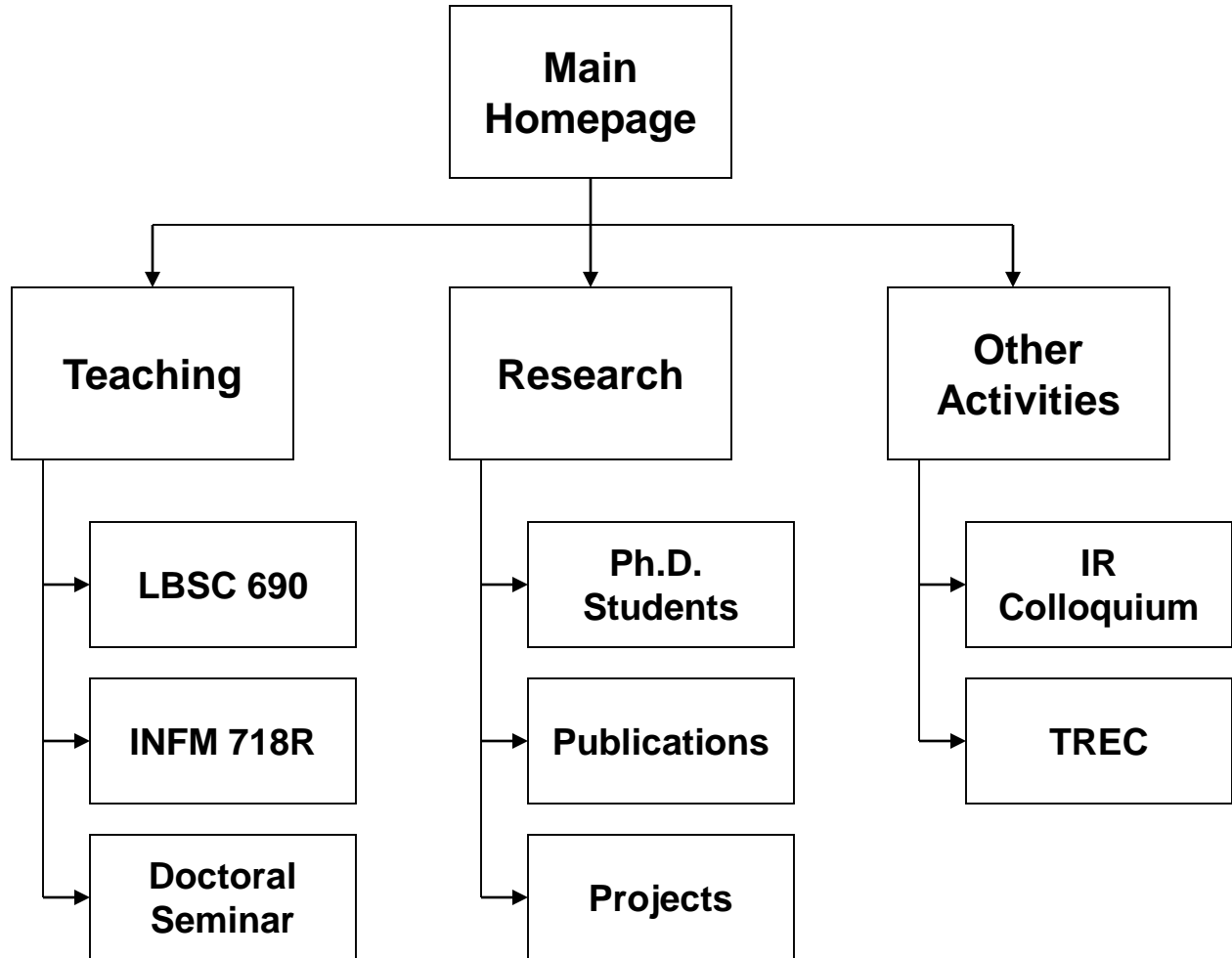
- Open-source software

Information Architecture

- The structural design of an "information space" to facilitate access to content

- Consists of at least two components:
  - Static design
  - Interaction design

Static Design

- Organizing Principles
  - Logical: inherent structure
  - Functional: by task
  - Demographic: by user

- Take advantage of metaphors
  - Organizational: e.g., e-government
  - Physical: e.g., online grocery store
  - Functional: e.g., cut, paste, etc.
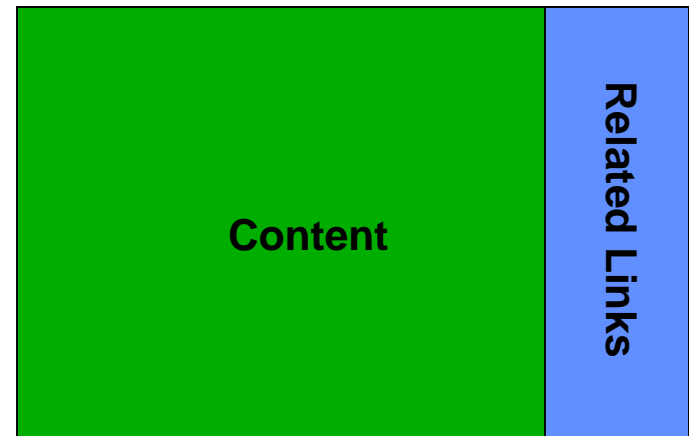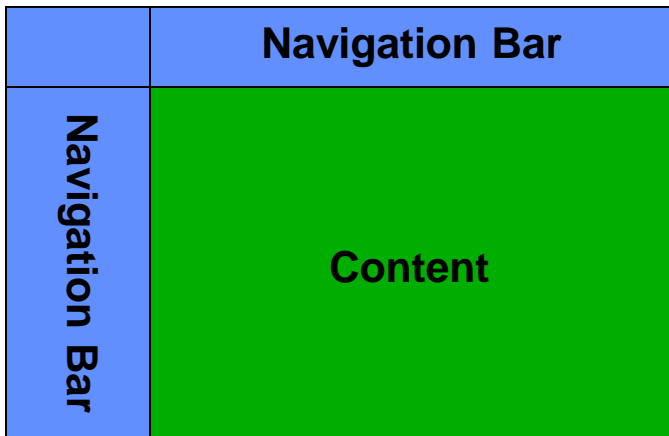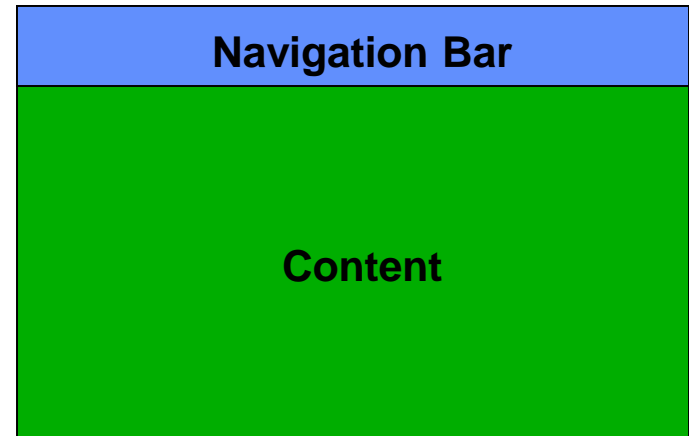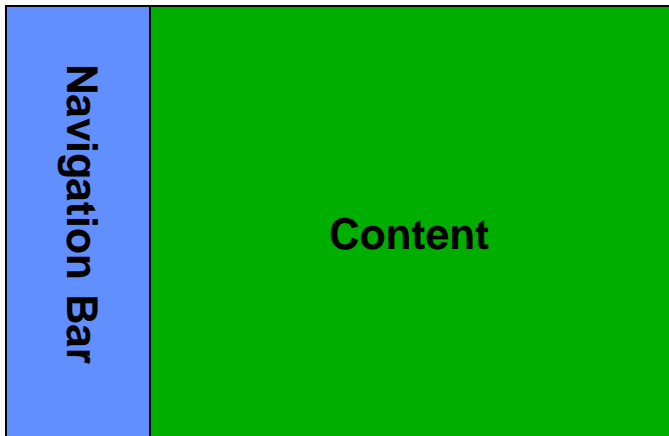 - Visual: e.g., octagon for stop

"Site Blueprint"

```
                    ┌──────────────┐
                    │    Main      │
                    │  Homepage    │
                    └──────┬───────┘
          ┌────────────────┼────────────────┐
          ▼                ▼                ▼
   ┌───────────┐    ┌───────────┐    ┌───────────┐
   │ Teaching  │    │ Research  │    │  Other    │
   │           │    │           │    │ Activities│
   └─────┬─────┘    └─────┬─────┘    └─────┬─────┘
```

Teaching	Research	Other Activities
LBSC 690	Ph.D. Students	IR Colloquium
INFM 718R	Publications	TREC
Doctoral Seminar	Projects	

Some Layout Guidelines

- **C**ontrast: make different things different
  - to bring out dominant elements
  - to create dynamism
- **R**epetition: reuse design throughout the interface
  - to create consistency
- **A**lignment: visually connect elements
  - to create flow
- **P**roximity: make effective use of spacing
  - to group related and separate unrelated elements

Screen Design: Use Grids

Grid Layout: NY Times

Grid Layout: NY Times

Interaction Design

- Chess analogy: a few simple rules that disguise an infinitely complex game

- The three-part structure
  - Openings: many strategies, lots of books about this
  - End game: well-defined, well-understood
  - Middle game: nebulous, hard to describe

- Information navigation has a similar structure!
  - Middle game is underserved

From Hearst, Smalley, & Chandler (CHI 2006)
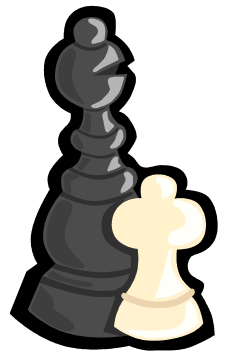
Opening Moves

Opening Moves

Middle Game

Middle Game

Navigation Patterns

- Drive to content

- Drive to advertisement

- Move up a level

- Move to next in sequence

- Jump to related

Moore's Law

computer performance

transistors
speed
storage

...

1950   1990   2030

Human Cognition

human performance

1950   1990   2030

1990
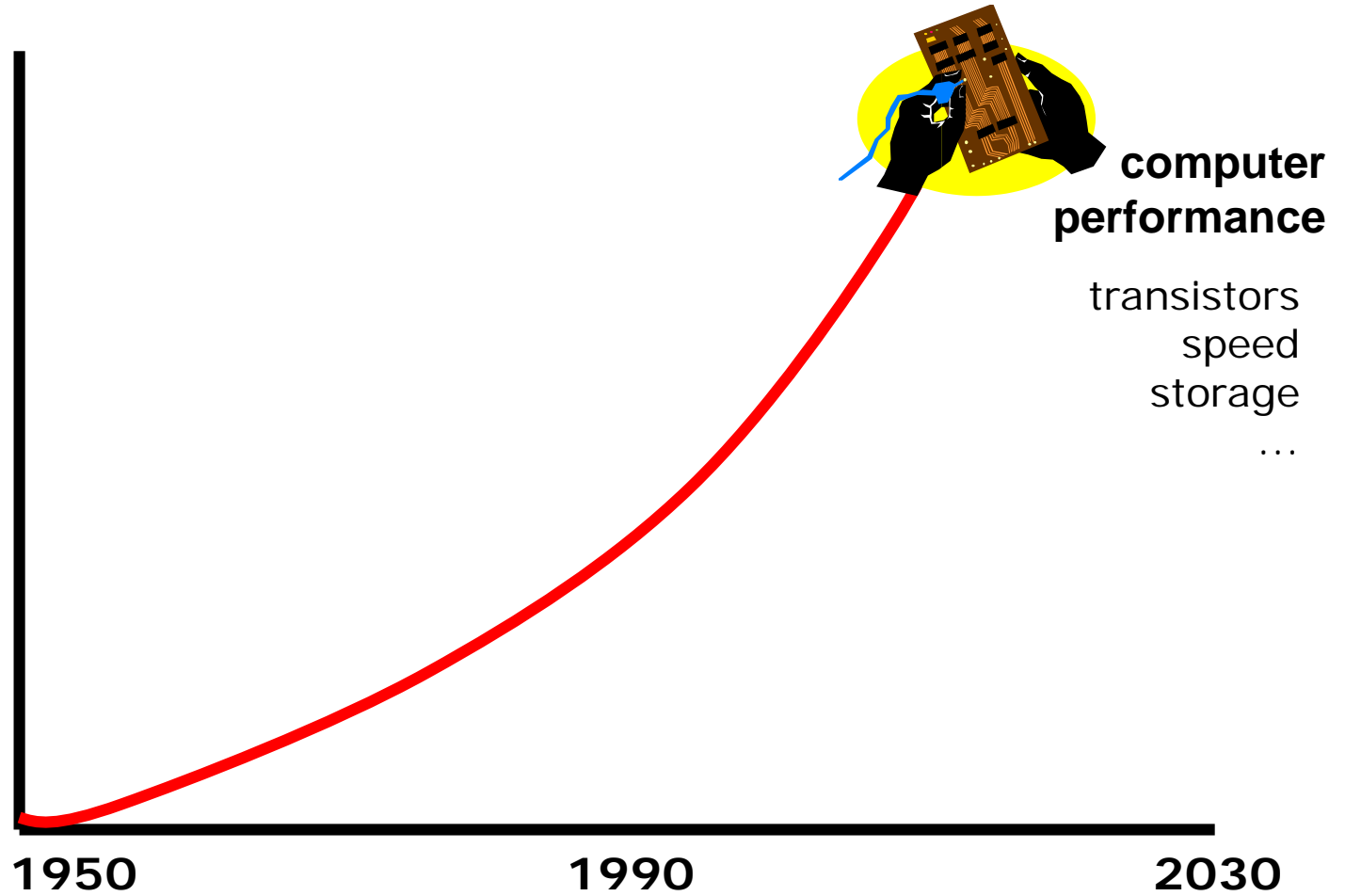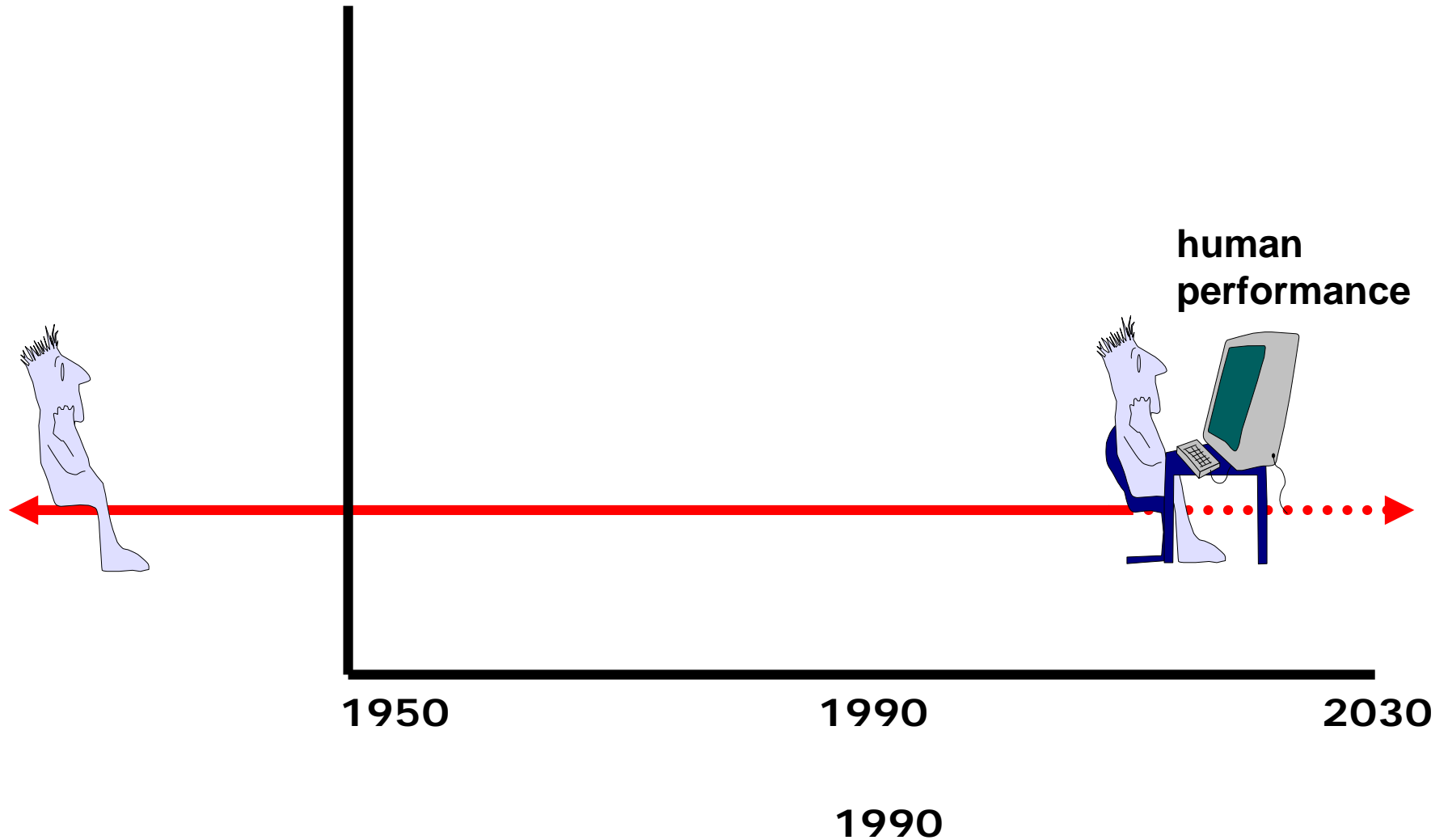
Human Computer Interaction

- A discipline concerned with the

Design

Implementation

Evaluation
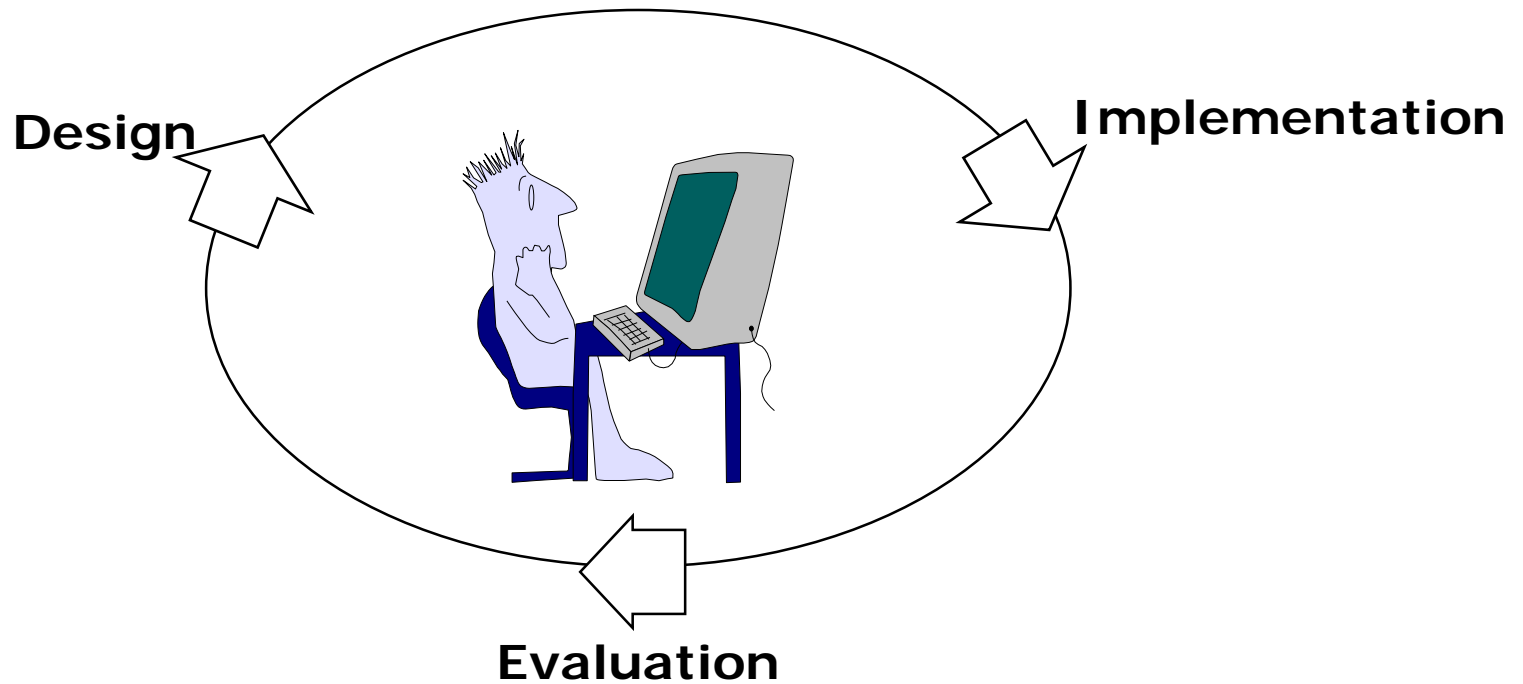
of interactive computing systems for human use

What are Humans Good At?

- Sense low level stimuli
- Recognize patterns
- Reason inductively
- Communicate with multiple channels
- Apply multiple strategies
- Adapt to changes or unexpected events

What are Computers Good At?

- Sense stimuli outside human's range
- Calculate quickly and accurately
- Store large quantities and recall accurately
- Respond rapidly and consistently
- Perform repetitive actions reliably
- Work under heavy load for an extended period

Synergy

- Humans do what they are good at

- Computers do what they are good at

- Strengths of one cover weakness of the other

Types of Applications

- Life critical
  - Low error rate first and foremost
  - Justifies an enormous design and testing effort
- Custom Commercial
  - Speed and error rate
- Office and Home
  - Easy learning, high user satisfaction, low cost
- Creative
  - User needs assessment is very challenging

User Characteristics

- Physical
  - Anthropomorphic (height, left handed, etc.)
  - Age (mobility, dexterity, etc.)
- Cognitive
- Perceptual
  - Sight, hearing, etc.
- Personality
  - Including cultural factors

Modeling Interaction

Human

Mental Models

Task   System

Sight
Sound

Hands
Voice

Software Models

Task   User

Keyboard
Mouse

Display
Speaker
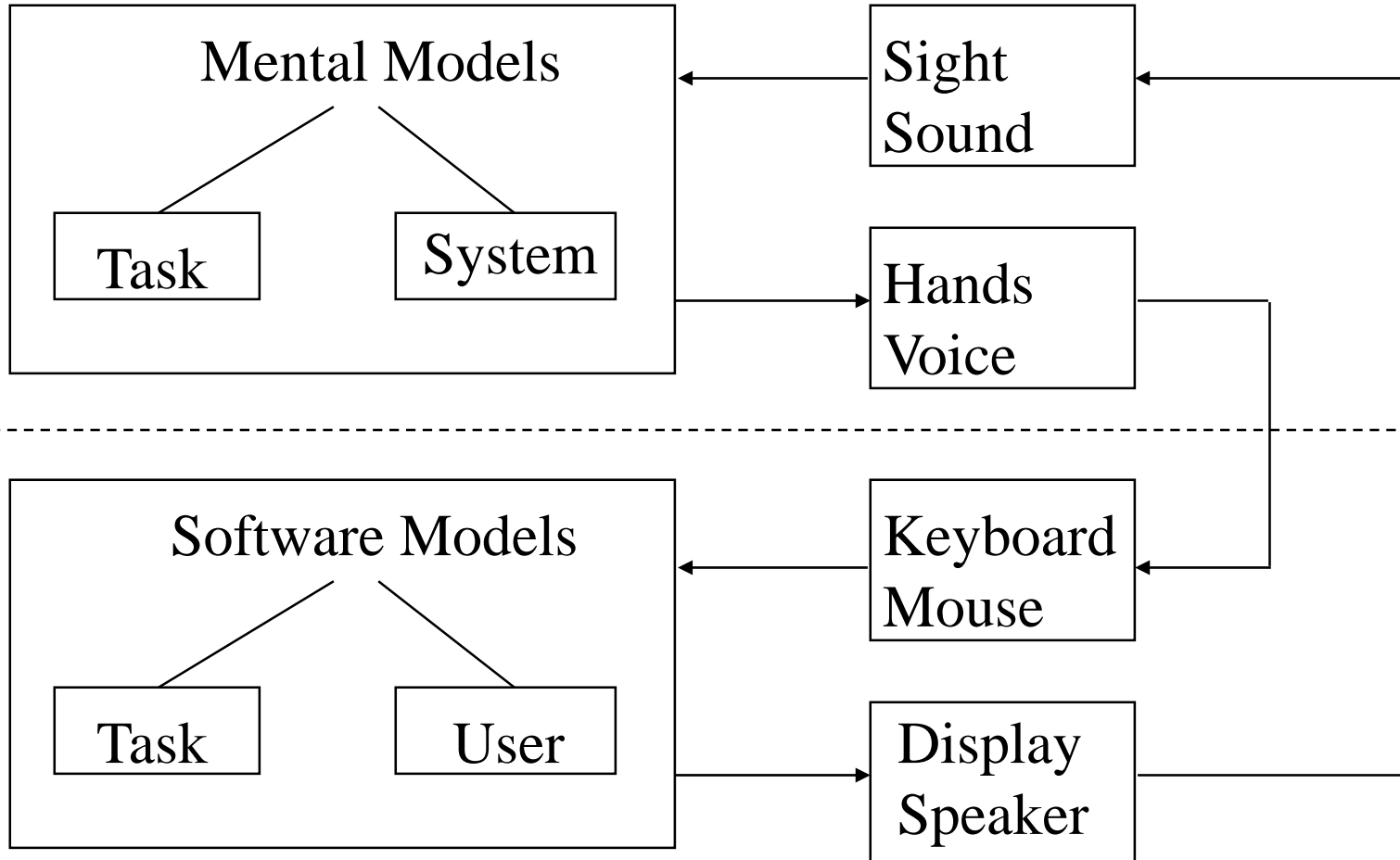
Computer

Discussion Point: Mental Models

- As a user, what do you need to know about a machine in order to interact with it effectively?

Mental Models

- How the user <u>thinks</u> the machine works
  - What actions can be taken?
  - What results are expected from an action?
  - How should system output be interpreted?
- Mental models exist at many levels
  - Hardware, operating system, and network
  - Application programs
  - Information resources

Stages of Interaction

Goals

Intention - - - - - → **Expectation** ← - - - - - **Evaluation**

Conceptual
Model

Selection   **Interpretation**

Execution   **Perception**
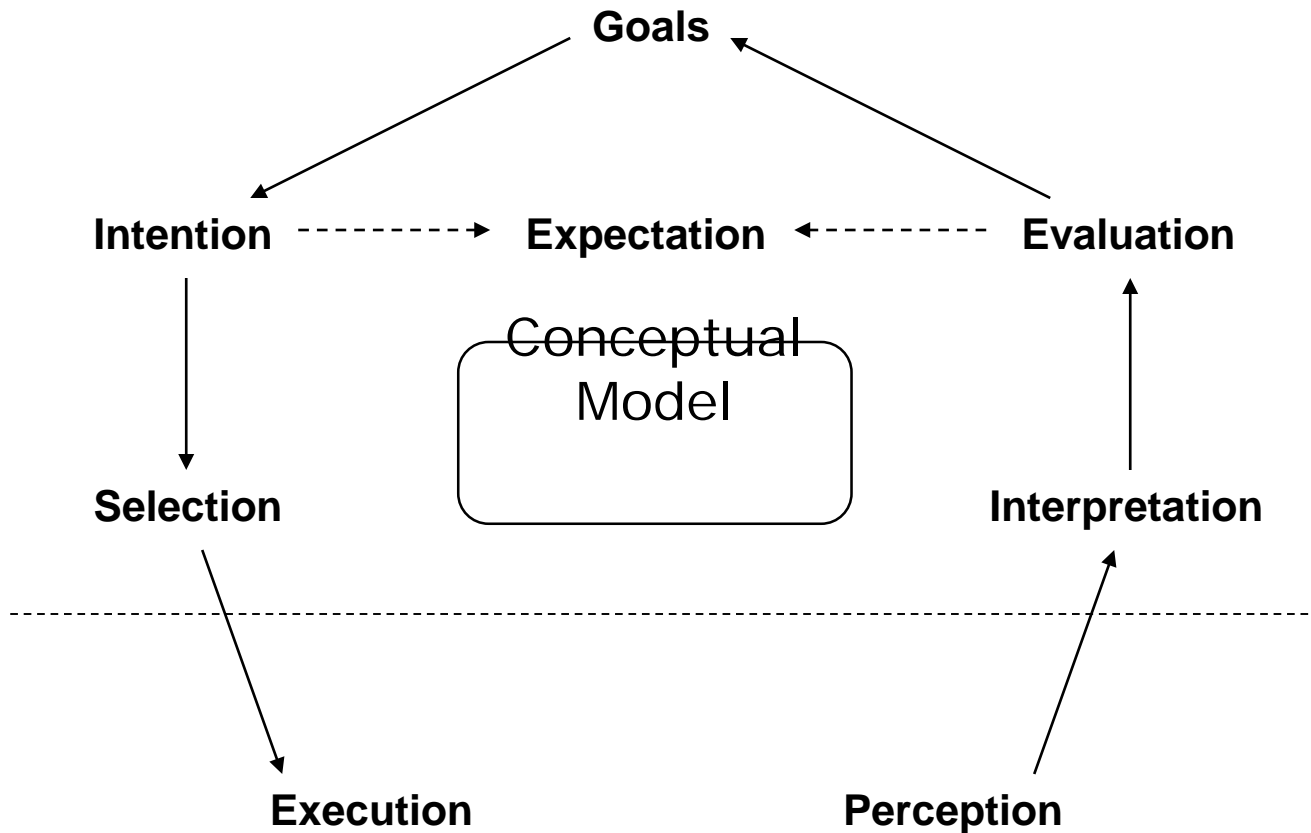
The GOMS Perspective

- **G**oals
  - What the user is trying to achieve

- **O**perators
  - What capabilities the system provides

- **M**ethods
  - How those capabilities can be used

- **S**election strategies
  - Which method to choose in a specific case

Input Devices

- Text
  - Keyboard, optical character recognition
  - Speech recognition, handwriting recognition
- Direct manipulation
  - 2-D: mouse, trackball, touch pad, touch panel
  - 3-D: wand, data glove
- Remote sensing
  - Camera, speaker ID, head tracker, eye tracker

Keyboard

- Produces character codes
  - ASCII: American English
  - Latin-1: European languages
  - UNICODE: (nearly) Any language
    - Pictographic languages need "entry methods"
- Keyboard shortcuts help with data entry
  - Different conventions for standard tasks abound
- VT-100 standard" functions are common
  - Differing layouts can inhibit usability

Design Example: QWERTY Keyboard
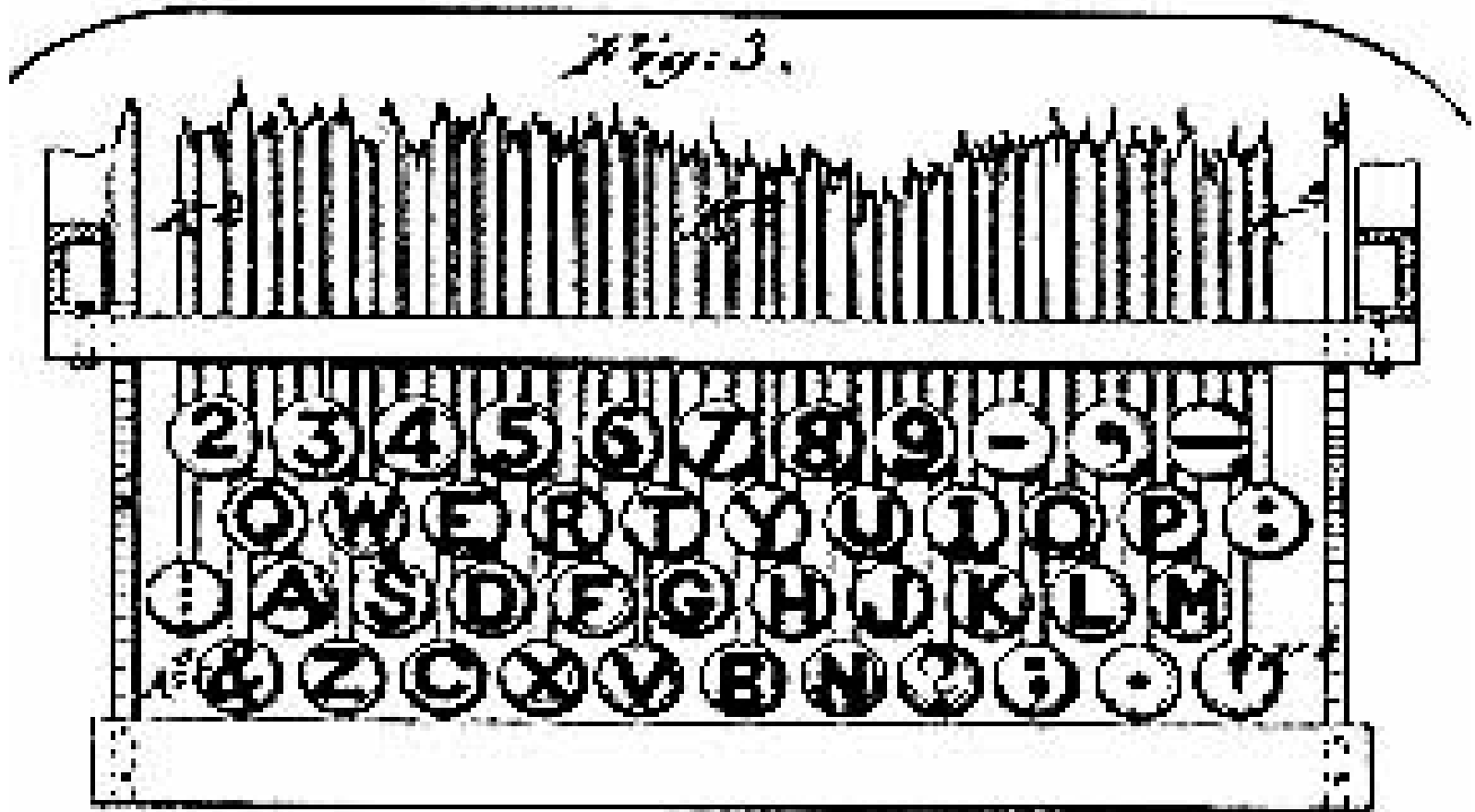
Dvorak Keyboard

Dvorak Keyboard Layout

From http://www.mwbrooks.com/dvorak/

2-D Direct Manipulation

- Match control actions with on-screen behavior
    - Use a cursor for visual feedback if needed
- Rotary devices
    - Mouse, trackball
- Linear devices
    - Touch pad, touch screen, iPod shuttle, joystick
- Rate devices
    - Laptop eraserhead

Modeling Interaction

Human

```
┌──────────────────────────────┐        ┌──────────────┐
│       Mental Models          │ ◄───── │ Sight        │ ◄───
│         ╱      ╲              │        │ Sound        │
│  ┌────────┐  ┌────────┐       │        └──────────────┘
│  │  Task  │  │ System │       │        ┌──────────────┐
│  └────────┘  └────────┘───────┼──────► │ Hands        │
│                              │        │ Voice        │
└──────────────────────────────┘        └──────────────┘
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
┌──────────────────────────────┐        ┌──────────────┐
│      Software Models         │ ◄───── │ Keyboard     │ ◄──
│         ╱      ╲              │        │ Mouse        │
│  ┌────────┐  ┌────────┐       │        └──────────────┘
│  │  Task  │  │  User  │       │        ┌──────────────┐
│  └────────┘  └────────┘───────┼──────► │ Display      │
│                              │        │ Speaker      │
└──────────────────────────────┘        └──────────────┘
```
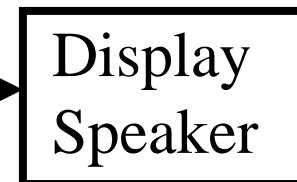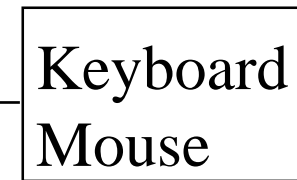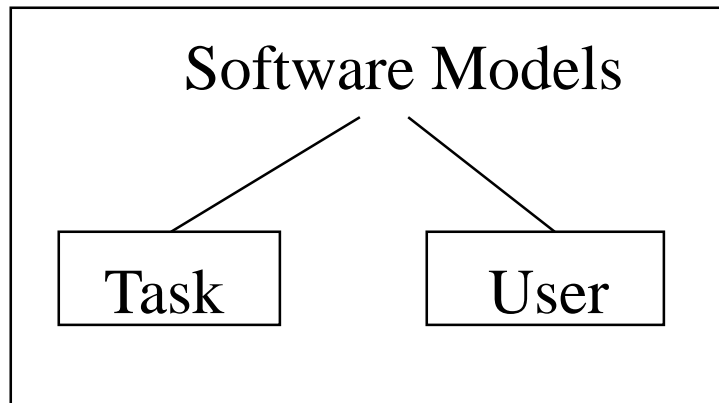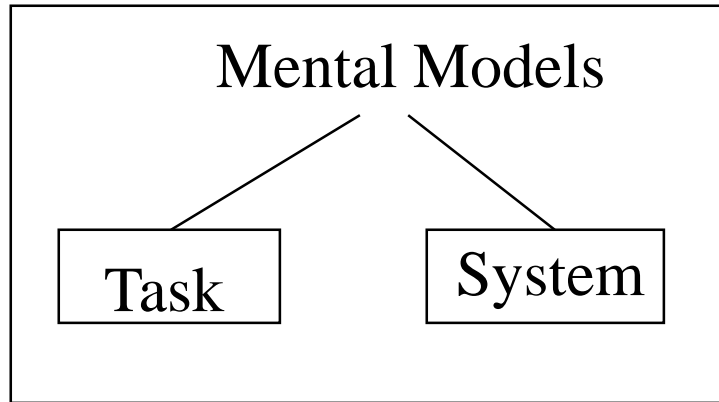
Computer

Human Senses

- Visual
  - Position/motion, color/contrast, symbols
- Auditory
  - Position/motion, tones/volume, speech
- Haptic
  - Mechanical, thermal, electrical, kinesthethic
- Olfactory
  - Smell, taste
- Vestibular

Computer Output

- Image display
  - Fixed view, movable view, projection
- Acoustic display
  - Headphones, speakers, within-ear monitors
- Tactile display
  - vibrotactile, pneumatic, piezoelectric
- Force feedback
  - dexterous handmaster, joystick, pen

Computer Output

- Inertial Display
    - Motion-based simulators
- Olfactory Display
    - Chemical (requires resupply)
- Locomotive display
    - Stationary bicycle, treadmill, ... (trip hazards)
- Temperature Display

Four Stages of Interaction

- Forming an **intention**
  - "What we want to happen"
  - Internal mental characterization of a goal
  - May comprise sub-goals (but rarely well planned)
  - For example, "write e-mail to grandma"

- **Selection** of an action
  - Review possible actions and select most appropriate
  - For example, "use Outlook to compose e-mail"

Four Stages of Interaction

- **Execution** of the action
  - Carry out the action using the computer
  - For example, "double-click Outlook icon"

- **Evaluation** of the outcome
  - Compare results with expectations
  - Requires perception, interpretation, and incremental evaluation
  - For example, "did Outlook open?"

Interaction Styles

- Graphical User Interfaces (GUI)
  - Direct manipulation (2D, 3D)
  - Menus
- Language-based interfaces
  - Command line interfaces
  - Interactive voice response systems
- Virtual Reality (VR)
  - Direct manipulation
- Ubiquitous computing

WIMP Interfaces

- **W**indows
  - Spatial context

- **I**cons
  - Direct manipulation

- **M**enus
  - Hierarchy

- **P**ointing devices
  - Spatial interaction

GUI Components

- Windows (and panels)
  - Resize, drag, iconify, scroll, destroy
- Selectors
  - Menu bars, pulldown lists
- Buttons
  - Labeled buttons, radio buttons, checkboxes
- Icons (images)
  - Select, open, drag, group

Direct Manipulation

- Select a metaphor
  - Desktop, CD player, map, …

- Use icons to represent conceptual objects
  - Watch out for cultural differences

- Manipulate those objects with feedback
  - Select (left/right/double click), move (drag/drop)

Visual Affordance

- The perceived and actual fundamental properties of the object that determine how it could be used
  - Appearance indicates how the object should be used
    - Chair for sitting
    - Table for placing things on
    - Knobs for turning
    - Slots for inserting things into
    - Buttons for pushing
- Complex things may need explaining but simple things should not
  - When simple things need instructions, design has failed
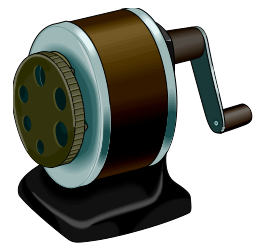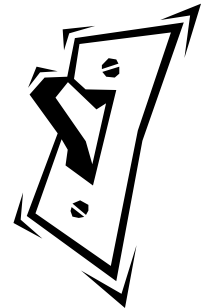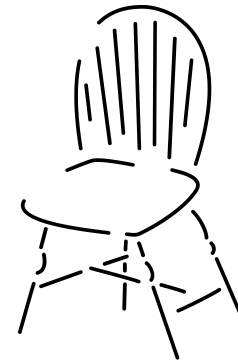
Visible Constraints: Date Entry

Causality

- The thing that happens right after an action is assumed by people to be caused by that action
  - "Feedback"
- False causality
  - Incorrect effect
    - Invoking unfamiliar function just as computer hangs
    - Causes "superstitious" behaviors
  - Invisible effect
    - Command with no apparent result often re-entered repeatedly
      - For example, mouse click to raise menu on unresponsive system

Causality: An Example

Effects visible only after Exec button is pressed
- Ok does nothing!
- Awkward to find appropriate color level
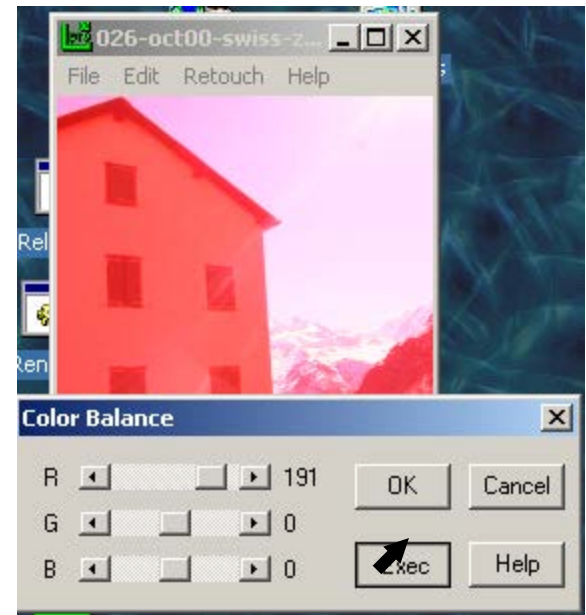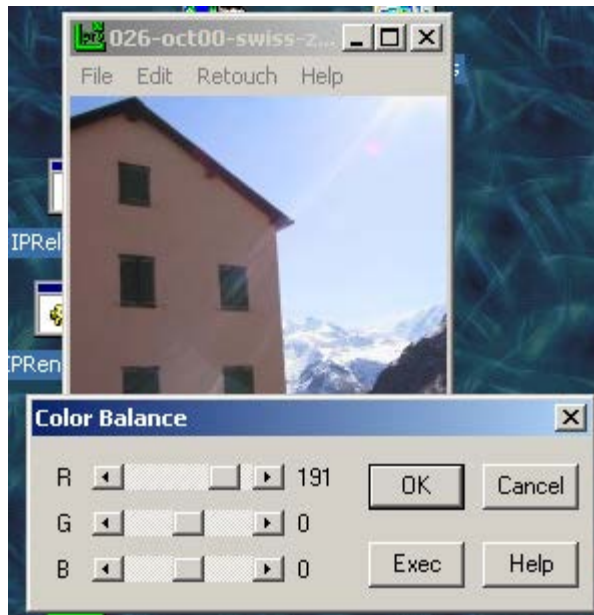
Transfer Effects

- People transfer expectations from similar objects
  - Positive: prior learning applies to new situation
  - Negative: prior learning conflicts with new situation
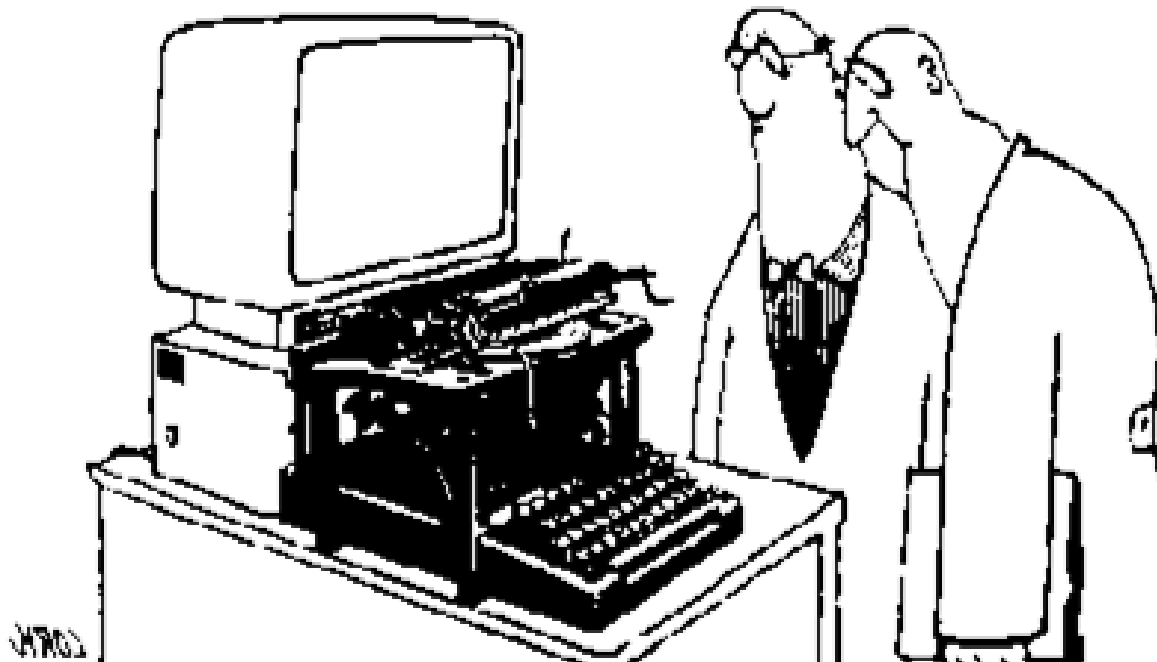
Positive and Negative Transfer

"First we thought the PC was a calculator. Then we found out how to turn numbers into letters with ASCII — and we thought it was a typewriter. Then we discovered graphics, and we thought it was a television. With the World Wide Web, we've realized it's a brochure." — Douglas Adams

Caller: Hello, is this Tech Support?"
Tech: Yes, it is. How may I help you?
Caller: The cup holder on my PC is broken and I am within my warranty period. How do I go about getting that fixed?
Tech: I'm sorry, but did you say a cup holder?
Caller: Yes, it's attached to the front of my computer.
Tech: Please excuse me if I seem a bit stumped, it's because I am. Did you receive this as part of a promotional, at a trade show? How did you get this cup holder? Does it have any trademark on it?
Caller: It came with my computer, I don't know anything about a promotional. It just has '4X' on it.

At this point the Tech Rep had to mute the call, because he couldn't stand it. The caller had been using the load drawer of the CD-ROM drive as a cup holder, and snapped it off the drive.

Cultural Associations

- Because a trashcan in Thailand may look like this:

- A Thai user is likely to be confused by this image popular in Apple interfaces:

- Sun found their email icon problematic for some American urban dwellers who are unfamiliar with rural mail boxes.
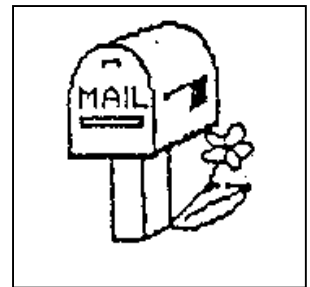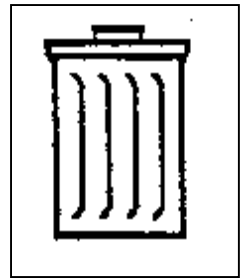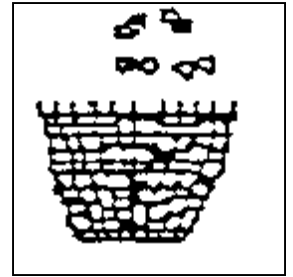
Population Stereotypes/Idioms

- People learn "idioms" that work in a certain way
  - Red means danger
  - Green means safe
- Idioms vary in different cultures
  - Light switches
    - America: down is off
    - Britain: down is on
  - Faucets
    - America: counter-clockwise on
    - Britain: counter-clockwise off
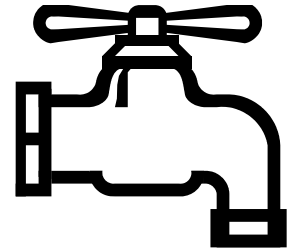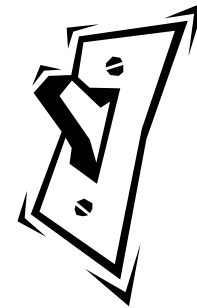
Spreadsheets: Direct Manipulation

	A	B	C	D	E	F
1	Relevant	Topic	ASR	Human	Human total	% Human
2	12	1188	0.0001	0.2104	0.2105	0%
3	17	1330	0.0029	0.7492	0.7521	0%
4	12	1630	0.0056	0.7041	0.7097	1%
5	46	1259	0.0048	0.354	0.3588	1%
6	22	1446	0.0062	0.3928	0.399	2%
7	15	1187	0.0221	0.9375	0.9596	2%
8	79	1663	0.0077	0.318	0.3257	2%
9	25	1424	0.0073	0.248	0.2553	3%
10	49	1508	0.0008	0.0202	0.021	4%
11	5	1148	0.0032	0.0694	0.0726	4%
12	10	1345	0.0337	0.4506	0.4843	7%
13	68	1181	0.0816	0.8026	0.8842	9%
14	43	1310	0.0533	0.456	0.5093	10%
15	23	14312	0.0428	0.3534	0.3962	11%
16	185	1269	0.0347	0.2478	0.2825	12%
17	10	1628	0.1003	0.6609	0.7612	13%
18	7	1173	0.0023	0.0148	0.0171	13%
19	46	1159	0.0006	0.0037	0.0043	14%
20	33	1286	0.0325	0.1737	0.2062	16%
21	61	1225	0.1189	0.538	0.6569	18%
22	40	1192	0.1932	0.6325	0.8257	23%
23	139	1623	0.0911	0.2852	0.3763	24%
24	14	1279	0.0944	0.2665	0.3609	26%
25	13	1332	0.0114	0.031	0.0424	27%
26	28	1414	0.2159	0.3767	0.5926	36%
27	610	1551	0.281	0.4189	0.6999	40%
28	43	1605	0.2884	0.4108	0.6992	41%
29	41	1311	0.0399	0.0164	0.0563	71%
30	28	1179	0.535	0.1593	0.6943	77%
31	Average		0.079714	0.355255172	0.434968966	0.176015
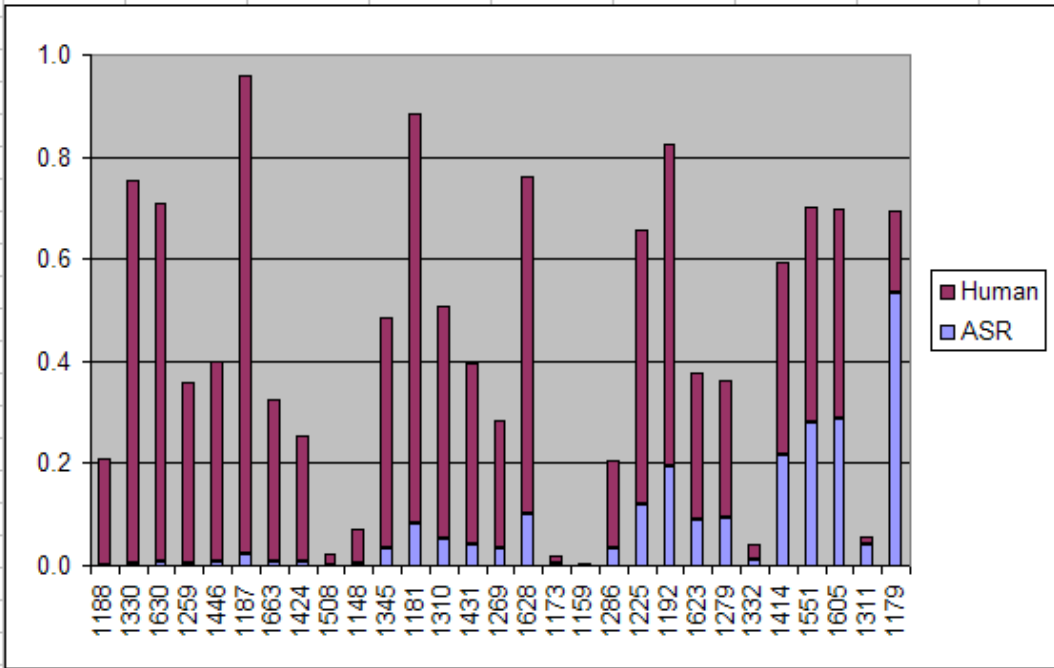
Menus

- Conserve screen space by hiding functions
  - Menu bar, pop-up

- Can hierarchically structured
  - By application's logic
  - By convention (e.g., where is the print function?)

- Tradeoff between breadth and depth
  - Too deep $\Rightarrow$ can become hard to find things
  - Too broad $\Rightarrow$ becomes direct manipulation

Dynamic Queries

- What to do when menus become too deep
  - Merges keyboard and direct manipulation

- Select menu items by typing part of a word
  - After each letter, update the menu
  - Once the word is displayed, user can click on it

- Example: Windows help index

Language-Based Interfaces

- Command Entry
  - Compact and flexible
  - Powerful in the hands of expert users
  - Difficult for novices to learn

- Natural Language
  - Intuitive and expressive
  - Ambiguity makes reliable interpretation difficult

"Seamless Interfaces"

- Informative feedback
- Easy reversal
- User in control
  - Anticipatable outcomes
  - Explainable results
  - Browsable content
- Limited working memory load
  - Query context
  - Path suspension
- Alternatives for novices and experts
  - Scaffolding

Evaluation Measures

- Time to learn

- Speed of performance

- Error rate

- Retention over time

- Subjective satisfaction

Evaluation Approaches

- Extrinsic vs. intrinsic

- Formative vs. summative

- Human subjects vs. simulated users

- Deductive vs. abductive

Evaluation Examples

- ## Direct observation
  - Evaluator observes users interacting with system
    - in lab: user asked to complete pre-determined tasks
    - in field: user goes through normal duties
  - Validity depends on how contrived the situation is

- ## Think-aloud
  - Users speak their thoughts while doing the task
  - May alter the way users do the task

- ## Controlled user studies
  - Users interact with system variants
  - Correlate performance with system characteristics
  - Control for confounding variables

Summary

- HCI design starts with user needs + abilities
  - Users have a wide range of both
- Users must understand their tools
  - And these tools can learn about their user!
- Many techniques are available
  - Direct manipulation, languages, menus, etc.
  - Choosing the right technique is important

Denial of Service Attacks

- Viruses
  - Platform dependent
  - Typically binary

- Flooding
  - Worms
  - Zombies
  - Chain letters

Viruses

- 1988: Less than 10 known viruses

- 1990: New virus found every day

- 1993: 10-30 new viruses per week

- 1999: 45,000 viruses and variants

Worms

- Self-reproducing program that sends itself across a network
  - Virus is dependent upon the transfer of files
  - Worm spreads itself

- SQL slammer worm (January 25, 2003) claimed 75,000 victims within 10 minutes
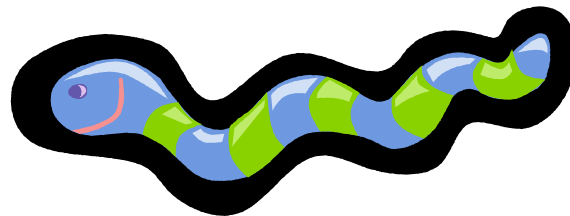
Viruses

- Computer programs able to attach to files
- Replicates repeatedly
  - Typically without user knowledge or permission
- Sometimes performs malicious acts

"On the Internet, nobody knows you're a dog."

Authentication

- Used to establish identity
- Two types
  - Physical (Keys, badges, cardkeys, thumbprints)
  - Electronic (Passwords, digital signatures)
- Protected with social structures
  - Report lost keys
  - Don't tell anyone your password
- Use SSH to defeat password sniffers

Good Passwords

- Long enough not to be guessed
  - Programs can try every combination of 5 letters
- Not in the dictionary
  - Programs can try every word in a dictionary
  - every proper name, pair of words, date, every …
- Mix upper case, lower case, numbers
- Change it often
- Reuse creates risks
  - Abuse, multiple compromise

Authentication Attacks

- Guessing
- Brute force
- Impersonation
- "Phishing"
- Theft

Symmetric Key Encryption

Same key used both for encryption and decryption
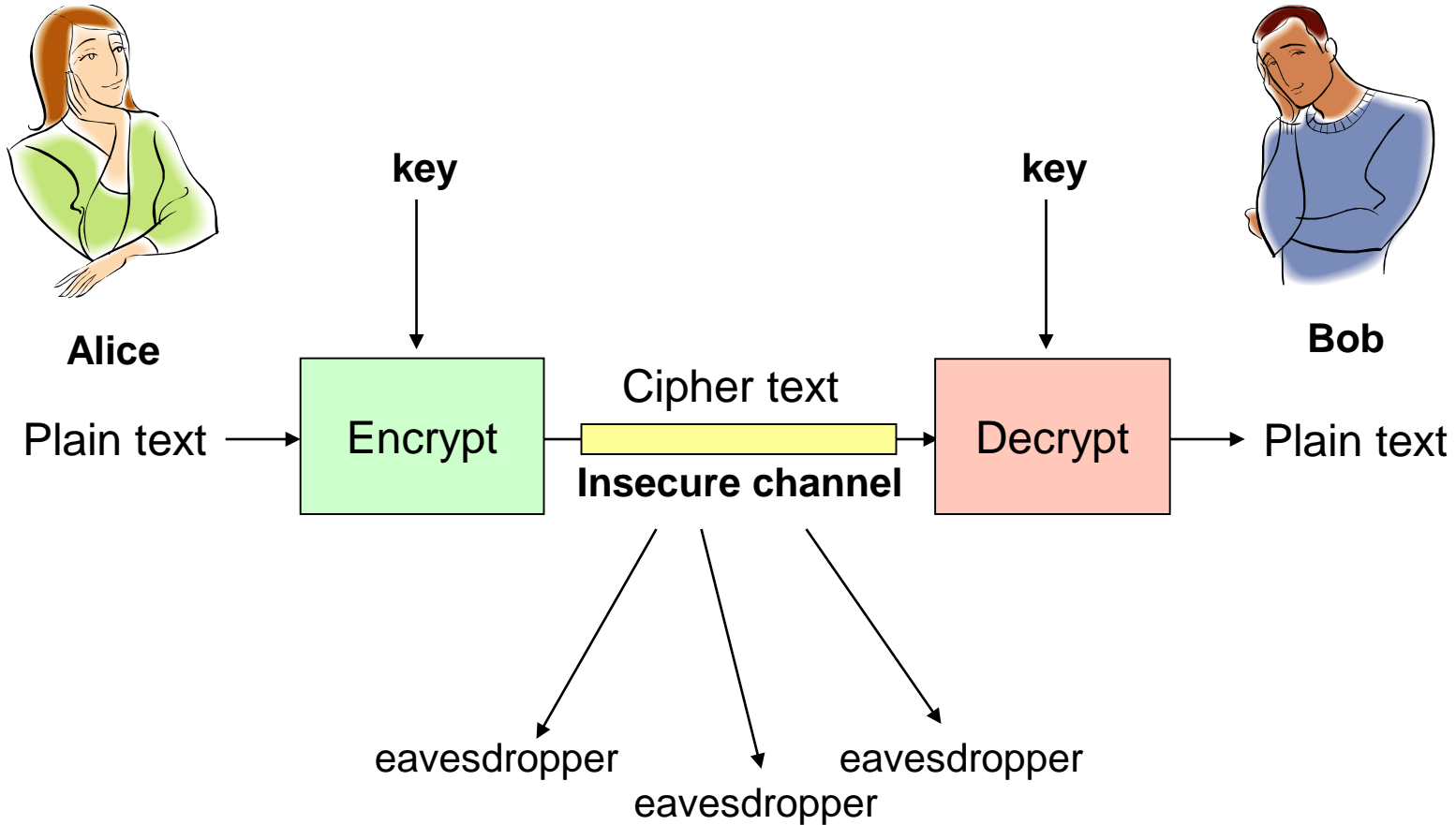
Asymmetric Key Encryption

Different keys used for encryption and decryption

Bob's public key

Bob's private key

Alice

Bob

Plain text → Encrypt → Cipher text → Decrypt → Plain text

Insecure channel

eavesdropper

eavesdropper

eavesdropper
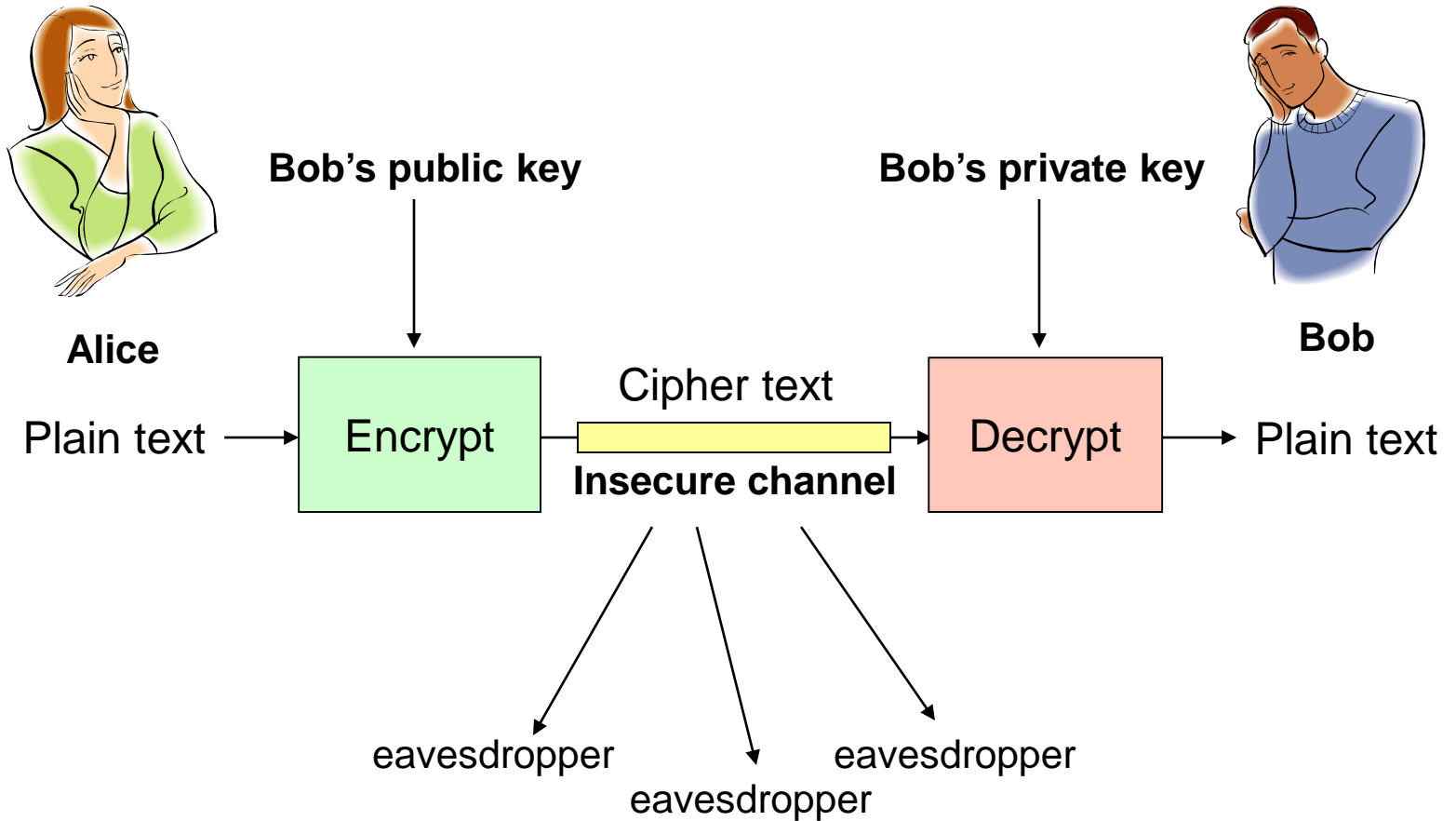
Asymmetric Key Encryption

- Key = a large number (> 1024 bits)
  - Public key: known by all authorized encoders
  - Private key: known only by decoder
- One-way mathematical functions
  - "Trapdoor functions"
    - Like mixing paint (easy to do, hard to undo)
  - Large numbers are easy to multiply, hard to factor
- Importance of longer keys
  - Keys < 256 bits can be cracked in a few hours
  - Keys > 1024 bits presently effectively unbreakable

RSA "Public Key" Encryption

Until 1997 – Illegal to show this slide to non-US citizens!
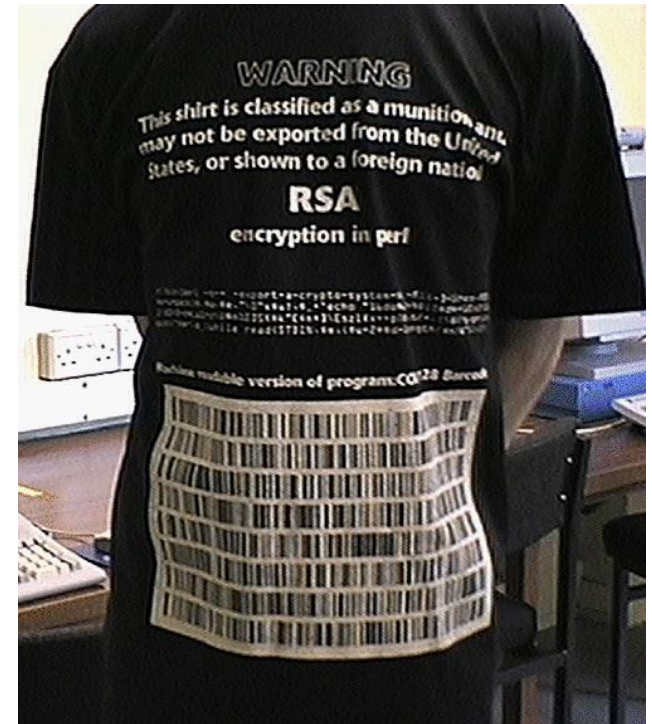
Trojan Horse

- Malicious program with undesired capabilities
  - Log key strokes and sends them somewhere
  - Create a "back door" administrator logon

- Spyware: reports information about your activity without your knowledge

- Doesn't (necessarily) replicate
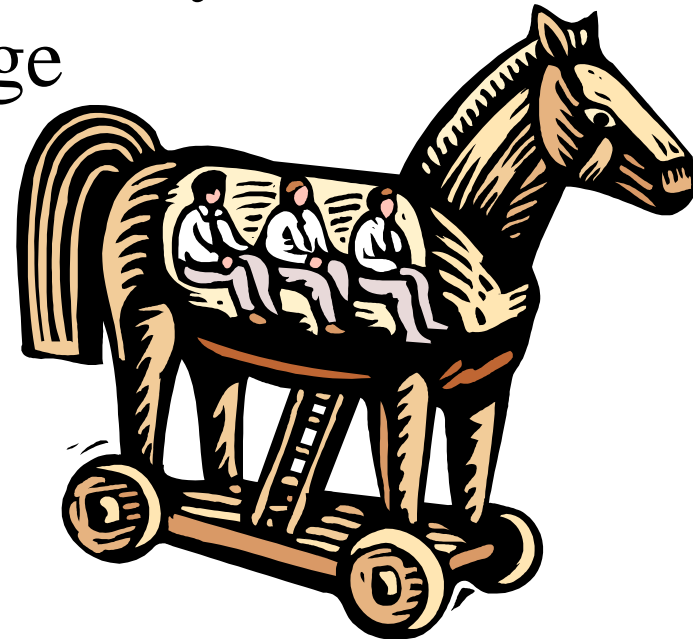
Real-Time Local Surveillance

- Built-in features of standard software
  - Browser history, outgoing email folders, etc.

- "Parental control" logging software
  - ChatNANNY, Cyber Snoop, FamilyCAM, …

- Personal firewall software
  - ZoneAlarm, BlackIce, …

Real-Time Centralized Surveillance

- Proxy server
  - Set up a Web server and enable proxy function
  - Configure all browsers to use the proxy server
  - Store and analyze Web server log files

- Firewall
  - Can monitor all applications, not just the Web

Forensic Examination

- Scan for files in obscure locations
    - Find-by-content for text, ACDSee for pictures, …

- Examine "deleted" disk files
    - Norton DiskDoctor, …

- Decode encrypted files
    - Possible for many older schemes

Integrity

- How do you know what's there is correct?
  - Attribution is invalid if the contents can change

- Access control would be one solution

- Encryption offers an alternative

Digital Signatures

- Alice "signs" (encrypts) with her private key
  - Bob checks (decrypts) with her public key

- Bob knows it was from Alice
  - Since only Alice knows Alice's private key

- Non-repudiation: Alice can't deny signing message
  - Except by claiming her private key was stolen!

- Integrity: Bob can't change message
  - Doesn't know Alice's Private Key

Key Management

- Pubic announcement of public key
  - e.g., append public key to the end of each email
  - But I can forge the announcement

- Establish a trusted "certificate authority"
  - Leverage "web of trust" to authenticate authority
  - Register public key with certificate authority

Certificate Authority

Certificate Authority

Bob's public key

Bob's public key
Bob's private key

Alice

Bob

Plain text → Encrypt → Cipher text (Insecure channel) → Decrypt → Plain text

eavesdropper

eavesdropper

eavesdropper
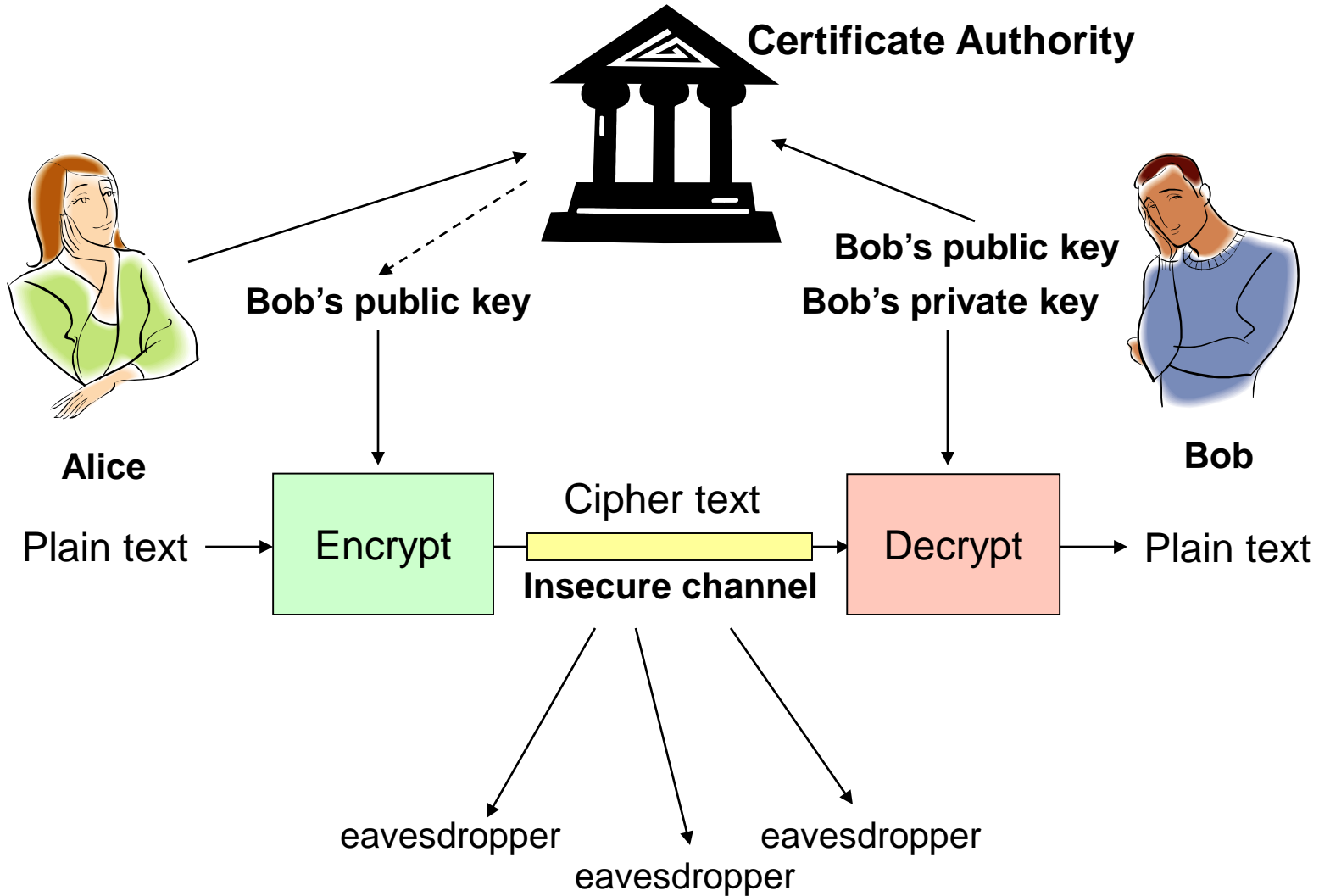
Certificates: Example
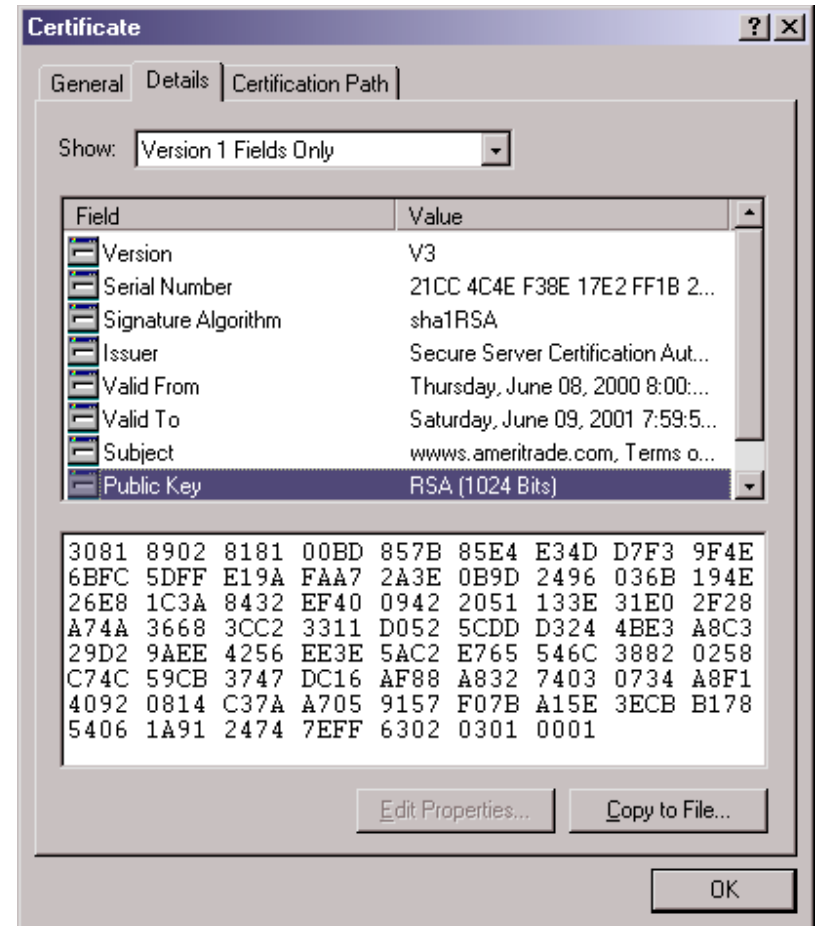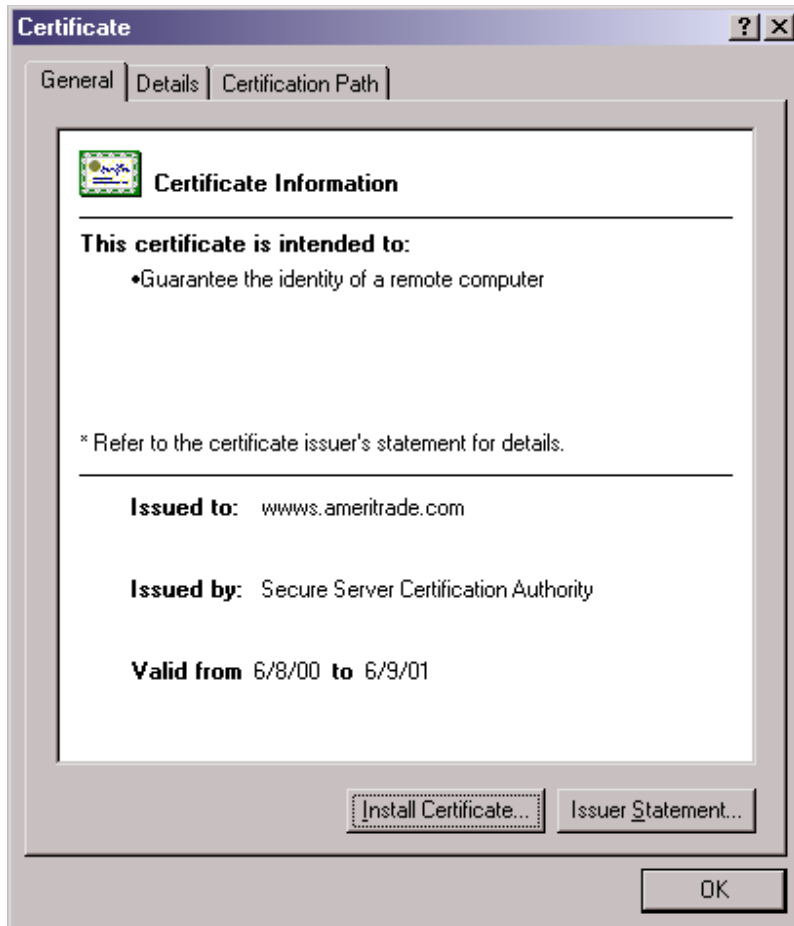
Practical Tips

- Keep anti-virus software current

- Keep software "patches" current

- Change default settings

- Be wary of anything free

Total Cost of Ownership

- Planning
- Installation
  - Facilities, hardware, software, integration, migration, disruption
- Training
  - System staff, operations staff, end users
- Operations
  - System staff, support contracts, outages, recovery, …

Total Cost of Ownership
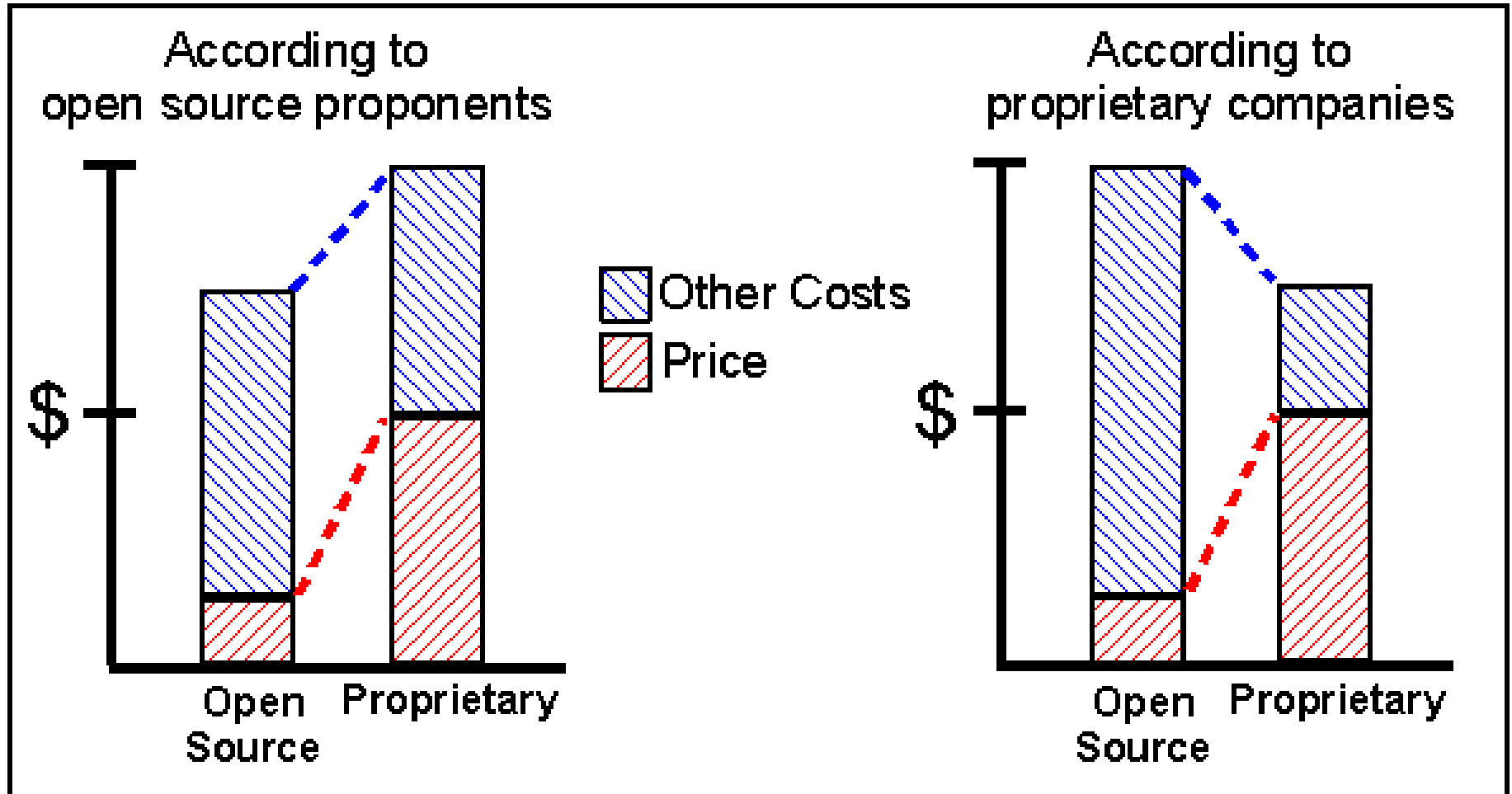
Some Examples

	Proprietary	**Open Source**
Operating system	Windows XP	Linux
Office suite	Microsoft Office	OpenOffice
Image editor	Photoshop	GIMP
Web browser	Internet Explorer	Mozilla
Web server	IIS	Apache
Database	Oracle	MySQL

Open Source "Pros"

- More eyes $\Rightarrow$ fewer bugs
- Iterative releases $\Rightarrow$ rapid bug fixes
- Rich community $\Rightarrow$ more ideas
  - Coders, testers, debuggers, users
- Distributed by developers $\Rightarrow$ truth in advertising
- Open data formats $\Rightarrow$ Easier integration
- Standardized licenses

Open Source "Cons"

- Communities require incentives
  - Much open source development is underwritten
- Developers are calling the shots
  - Can result in feature explosion
- Proliferation of "orphans"
- Diffused accountability
  - Who would you sue?
- Fragmentation
  - "Forking" may lead to <u>competing</u> versions
- Little control over schedule

Open Source Business Models

- ## Support Sellers

   Sell distribution, branding, and after-sale services.

- ## Loss Leader
   Give away the software to make a market for proprietary software.

- ## Widget Frosting

   If you're in the hardware business, giving away software doesn't hurt.

- ## Accessorizing
   Sell accessories:
   books, compatible hardware, complete systems with pre-installed software

Iron Rule of Project Management

- You can control any **<u>two</u>** of:
  - Capability
  - Cost
  - Schedule

- Open source software takes this to an extreme

Estimating Completion Time

- Rules of thumb
  - 1/3 specification
  - 1/6 coding
  - 1/2 test planning, testing, and fixing!

- Add time for coding to learn as you go, but don't take time away from the other parts!
  - Reread the section on "gutless estimating" if you are tempted