Human Abilities
“Design” in HCI

• Note: Differences with “design” in software engineering
  – Design in HCI = create a new concept
  – Design in SE = given concept, create an architecture/schema for the system being built

• Design includes two different aspects
  – Low level aspects of UI that help people interact more efficiently
  – High level representation of concepts in UI that help people understand and interact with software
“Design” in HCI

Both the widgets that instantiate the UI and the representation of information are informed by characteristics of people.
Understanding People

- **Movement**
  - Fitt’s Law
  - Steering Law

- **Memory**

- **Reasoning**
Movement

• Fitt’s Law
  \[ T = a + b \log_2 \left( \frac{A}{W} + 1 \right) \]

• Steering Law
  \[ T = a + b \int_c \left( \frac{1}{W(s)} \right) ds = a + b \left( \frac{A}{W} \right) \]
Design Implications – Fitts’ Law

Pop-up Linear Menu

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Today</td>
</tr>
<tr>
<td>Sunday</td>
</tr>
<tr>
<td>Monday</td>
</tr>
<tr>
<td>Tuesday</td>
</tr>
<tr>
<td>Wednesday</td>
</tr>
<tr>
<td>Thursday</td>
</tr>
<tr>
<td>Friday</td>
</tr>
<tr>
<td>Saturday</td>
</tr>
</tbody>
</table>

Pop-up Pie Menu

From Landay’s HCI slides
I’m still not sold on Pie menus
Design Implications – Fitts’ Law
Expect-K

- ..\Videos\uist.avi
- ..\Videos\ThickButtons  finger text input for touchscreen smartphones.flv
- ..\Videos\YouTube - iPhone Typing Demonstration.flv
Design Implications – Steering Law

- hierarchical menu item selection

\[
T_n = a + b \left( \frac{nh}{w} \right) + a + b \left( \frac{w}{h} \right) \\
= 2a + b \left( \frac{n}{x + x} \right) \text{ with } x = \frac{w}{h}
\]

- So \( T \) is minimal when \( x = \sqrt{n} \) or \( w = \sqrt{n} \times h \)
  - the greater the number of menu items there are, the greater the quotient \( w/h \) is

- Can be used to compare designs, i.e. linear hierarchical menus and hierarchical pie menus
Design Implications – Steering Law
Memory
Model Human Processor

Long-term Memory

Working Memory

Visual Image Store

Auditory Image Store

Perceptual Processor

Motor Processor

Cognitive Processor

Eyes

Ears

sensory buffers

Fingers, etc.
MHP basics

• Based on empirical data
  – years of basic psychology experiments in the literature

• Three interacting subsystems
  – perceptual, motor, cognitive

• Sometimes serial, sometimes parallel
  – serial in action & parallel in recognition
    • pressing key in response to light
    • driving, reading signs, & hearing at once

• Parameters
  – processors have cycle time (T) ~ 100-200 ms
  – memories have capacity, decay time, & type
Memory

• Three types of memory

1. Sensory memory
   Focusing attention transfers to

2. Short term (working) memory
   Practice/rehearsal transfers to

3. Long term memory
Sensory memory

• Short term buffers
• Different channels have different buffers:
  – Iconic memory for visual stimuli
  – Echoic memory for auditory stimuli
  – Haptic memory for touch
  – New information overwrites old information
• Existence demonstrated in a couple of ways:
  – After images
  – Direction from which sound emanates and recall of question you didn’t think you knew
• Collects information all the time
  – Need some way to filter
  – We do this by attention and focus
Short-term memory

- Think about a task like reading:
  - Need to keep info from first of a sentence in order to get meaning
  - Meaning is what’s stores, not words
  - Implies a need for temporary “working” storage
- Accessed rapidly: ~70ms
- Limited capacity
  - Two ways this has been tested:
    1. Lengths of sequences: 7 +/- 2 digits
    2. Free recall of info in any order
Short-term memory exercises

• Here is a sequence of numbers:
  – 2653797620853261823
Short-term memory exercises

• Here is a sequence of numbers:
  – 871 392 567 481 28 10 21 37
Short-term memory exercises

• Here is a sequence of numbers:
  – 871 392 567 481 28 10

• We remember best when information is “chunked”
Long-term memory

• Stores factual information, experiential knowledge, and rules of behavior
• Huge, if not unlimited
• Slow access time (100 ms)
• Two types:
  – Episodic memory
    • Our memory of events
  – Semantic memory
    • Structured record of facts
• Use rehearsal to move info from short term to long term memory
LTM Processes

• Getting info into long term memory:
  – How do I learn?
  – Optimizations include:
    • Total time hypothesis: Amount learned is proportional to time spent learning
    • Distribution of practice effect: Learning works best if spread out
  – Learning well includes understanding
    • Build models of information
    • Structure, familiarity, concreteness
    • Particularly for devices – why is a VCR hard to program?
LTM Processes

• Forgetting
  – Decay or interference:
    • Decay:
      – Theory that over time, information degrades
      – Actually plotted logarithmic scale
    • Interference:
  – Now a debate about whether forgetting ever happens or if it’s a retrieval problem
    • Old information breaking through
    • Tip of tongue phenomenon
LTM Processes

• Recall vs. recognition
  – Recall
    • Information is reproduced from memory
  – Recognition
    • Presentation of information cues us to fact we’ve seen this before
  – Should stress recognition over recall
    • Why?
LTM Processes

• Recall vs. recognition
  – Recall
    • Information is reproduced from memory
  – Recognition
    • Presentation of information cues us to fact we’ve seen this before
  – Should stress recognition over recall
    • Provide strong cues for recall if used
Reasoning
Reasoning

• Deductive
  – Uses logic to derive conclusions from premises
    • If it is Friday then she will go to work.
    • It is Friday, therefore she will go to work

• Inductive
  – Generalizes from cases we have seen
    • Can disprove simply by producing counter-example.
    • Scientific method.

• Abductive
  – Reasons about causes from events
    • I pressed a button and the window closed.
Reasoning

• Problem Solving
  – Gestalt theory
    • Restructuring and insight to perform productive problem solving
      – Pendulum example
  – Problem space theory
    • Problem solving looks at problem space as state space and moves from initial to goal state using operators
      – Math example
  – Using analogy
    • Solving novel problems involves mapping previous knowledge – analogical mapping
      – Medical example
Problem solving - continued

• Characteristics of experts
  – Chess – don’t consider more moves, consider better ones.
  – Reading diagrammatic notations (grouping)

• Better encoding of knowledge as skill increases
  1. General purpose rules (slow)
  2. Rules specific to task
  3. Rules are tuned to boost performance
Mental Models – enabling problem solving

• A model of how device works
• Based on cognitive psychology
• Consider ATM card
  – What information does it contain?
• Problem with mental models is that term is over-used
  – Any argument about need for understanding of the device or application
Mental Models

• Debate about the importance ...
  – For example, “success of a computer system is almost totally controlled by how well it fits into user’s work practice” – Stephen J. Payne (Mental Models researcher)

• But, an understanding of differing theories of models can help you understand user’s problem solving approach

• Differing theories as to how mental models are formed
Mental Models: Theories

• Naïve physics
  – Mechanics or electricity

• Problem spaces
  – Accomplish tasks by searching a space of possible actions

• Representational artifacts
  – Reading text versus understanding meaning

• Homomorphisms
  – Directions by reading a map versus from someone with experience with Toronto
Mental Models – differing theories

- Mental Models as Naïve physics
  - People understand the physical world based on their (imperfect) understanding of mechanics or electricity
  - Theorize about the physical world based on their mental model of the world
  - Mental models look at systems in the large
  - HCI often concerned with discrete phenomena
  - ATM cards ... Study by Payne showing that discrete behaviors can be explained by models even if overall system is poorly understood
Mental Models – differing theories

• Mental models as problem spaces
  – Methods for achieving tasks
  – Problem solving involves searching a problem space of possible states
  – Skilled behavior involves remembering sequences of states to accomplish tasks
  – Problem: perfect skill is never reached
    • Always some aspect of search to solve problems
  – Behavior is either problem solving or learned
    • Learned behavior is either skill-based (controlled)/rule-based (automatic)
  – Examples of this include learning reverse-polish notation
    • (1+2)*3 => 1 2 + 3 * (rote procedures) vs. stack representation (model: an operator is entered, top of stack is popped)
Mental Models – differing theories

• Models as representational artifacts
  – Reading text -> understanding the meaning
  – To know what’s on these slides, you don’t need to remember the text
  – To search and find something in these slides, you need to remember the text
  – Payne proposes a “yoked state space” for software
    • Using software requires some representation of domain of software
      – User’s goals are states in the domain
    • Using software also requires knowledge of the operations to transform states
      – This is the “device space”
    • These spaces have to be connected for user
Mental Models – differing theories

• Mental models as homomorphisms
  – Basically, the model is an analog
    • A verbal description of a picture vs. the picture itself
    • Picture is much more constrained.
  – Consider a map vs. experience with Toronto
    • People who have extensively studied the map vs. people who have lived here for a long time
    • Both can give good directions based on experience
    • Contrast with new-comers with no map
  – People with experience with software develop a cognitive map of the software
Take-Away Points

1. Know what people are going to do with your software and **how** they will do their task

2. People have characteristics and limitations
   - Biological in nature
   - Seeing, touch, movement, and thinking must all be considered.

- Overall, the design of software should consider characteristics of individuals