chapter 12

Cognitive models

• goal and task hierarchies
• linguistic
• physical and device
• architectural

Cognitive models

• They model aspects of user:
  – understanding
  – knowledge
  – intentions
  – processing

• Common categorisation:
  – Competence vs. Performance
  – Computational flavour
  – No clear divide
Goal and task hierarchies

- Mental processing as divide-and-conquer
- Example: sales report
  - produce report
  - gather data
  - find book names
  - do keywords search of names database
  - further sub-goals
  - sift through names and abstracts by hand
  - further sub-goals
  - search sales database - further sub-goals
  - layout tables and histograms - further sub-goals
  - write description - further sub-goals

goals vs. tasks

- goals – intentions
  - what you would like to be true
- tasks – actions
  - how to achieve it
- GOMS – goals are internal
- HTA – actions external
  - tasks are abstractions

Issues for goal hierarchies

- Granularity
  - Where do we start?
  - Where do we stop?
- Routine learned behaviour, not problem solving
  - The unit task
- Conflict
  - More than one way to achieve a goal
- Error
Techniques

- Goals, Operators, Methods and Selection (GOMS)
- Cognitive Complexity Theory (CCT)
- Hierarchical Task Analysis (HTA) - Chapter 15

GOMS

Goals – what the user wants to achieve
Operators – basic actions user performs
Methods – decomposition of a goal into subgoals/operators
Selection – means of choosing between competing methods

GOMS example

GOAL: CLOSE-WINDOW
- [select GOAL: USE-MENU-METHOD
  - MOVE-MOUSE TO FILE-MENU
  - PULL-DOWN FILE-MENU
  - CLICK OVER CLOSE-OPTION
  - PRESS CONTROL-W-KEYS]

For a particular user:
Rule 1: Select USE-MENU-METHOD unless another rule applies
Rule 2: If the application is GAME, select CTRL-W-METHOD
Cognitive Complexity Theory

- Two parallel descriptions:
  - User production rules
  - Device generalised transition networks
- Production rules are of the form:
  if condition then action
- Transition networks covered under dialogue models

Example: editing with vi

- Production rules are in long-term memory
- Model working memory as attribute-value mapping:
  - (GOAL perform unit task)
  - (TEXT task is insert space)
  - (TEXT task is at 5 23)
  - (CURSOR 8 7)
- Rules are pattern-matched to working memory,
  e.g., LOOK-TEXT task is at %LINE %COLUMN is true, with LINE = 5 COLUMN = 23.

Four rules to model inserting a space

- Active rules:
  SELECT-INSERT-SPACE
  DESERT-SPACE-MOVE-FIRST
  DESERT-SPACE-GO
  DESERT-SPACE-LEFT

- Now working memory
  (GOAL insert space)
  (NOTE executing insert space)
  (LINE 5) (COLUMN 23)

- SELECT-INSERT-SPACE matches current working memory

  SELECT-INSERT-SPACE
  IF (AND (TEST-GOAL perform unit task)
  (TEST-TEXT task is insert space)
  (NOT (TEST-GOAL insert space))
  (NOT (TEST-NOTE executing insert space)))
  THEN ( (ADD-GOAL insert space)
  (ADD-NOTE executing insert space)
  (LOOK-TEXT task is at %LINE %COLUMN)))
Notes on CCT

• Parallel model
• Proceduralisation of actions
• Novice versus expert style rules
• Error behaviour can be represented
• Measures
  – depth of goal structure
  – number of rules
  – comparison with device description

Problems with goal hierarchies

• a post hoc technique
• expert versus novice
• How cognitive are they?

Linguistic notations

• Understanding the user’s behaviour and cognitive difficulty based on analysis of language between user and system.
• Similar in emphasis to dialogue models

• Backus–Naur Form (BNF)
• Task–Action Grammar (TAG)
Backus-Naur Form (BNF)

- Very common notation from computer science
- A purely syntactic view of the dialogue
- **Terminals**
  - lowest level of user behaviour
  - e.g. CLICK-MOUSE, MOVE-MOUSE
- **Nonterminals**
  - ordering of terminals
  - higher level of abstraction
  - e.g. select-menu, position-mouse

Example of BNF

- Basic syntax:
  - nonterminal ::= expression
- An expression
  - contains terminals and nonterminals
  - combined in sequence (+) or as alternatives (|)

```
draw line ::= select line + choose points + last point
select line ::= pos mouse + CLICK MOUSE
choose points ::= choose one   |   choose one + choose points
choose one ::= pos mouse + CLICK MOUSE
last point ::= pos mouse + DBL CLICK MOUSE
pos mouse ::= NULL   |   MOVE MOUSE + pos mouse
```

Measurements with BNF

- Number of rules (not so good)
- Number of + and | operators
- Complications
  - same syntax for different semantics
  - no reflection of user's perception
  - minimal consistency checking
Task Action Grammar (TAG)

- Making consistency more explicit
- Encoding user’s world knowledge
- Parameterised grammar rules
- Nonterminals are modified to include additional semantic features

Consistency in TAG

- In BNF, three UNIX commands would be described as:
  
  \begin{align*}
  \text{copy} & ::= \text{cp} + \text{filename} + \text{filename} | \text{cp} + \text{filenames} + \text{directory} \\
  \text{move} & ::= \text{mv} + \text{filename} + \text{filename} | \text{mv} + \text{filenames} + \text{directory} \\
  \text{link} & ::= \text{ln} + \text{filename} + \text{filename} | \text{ln} + \text{filenames} + \text{directory}
  \end{align*}

- No BNF measure could distinguish between this and a less consistent grammar in which
  
  \begin{align*}
  \text{link} & ::= \text{ln} + \text{filename} + \text{filename} | \text{ln} + \text{directory} + \text{filenames}
  \end{align*}

Consistency in TAG (cont’d)

- Consistency of argument order made explicit using a parameter, or semantic feature for file operations
- Feature Possible values
  
  \begin{align*}
  \text{Op} & = \text{copy}; \text{move}; \text{link}
  \end{align*}

- Rules
  
  \begin{align*}
  \text{file-op}[\text{Op}] & ::= \text{command}[\text{Op}] + \text{filename} + \text{filename} \\
  & \quad | \text{command}[\text{Op}] + \text{filenames} + \text{directory} \\
  \text{command}[\text{Op} = \text{copy}] & ::= \text{cp} \\
  \text{command}[\text{Op} = \text{move}] & ::= \text{mv} \\
  \text{command}[\text{Op} = \text{link}] & ::= \text{ln}
  \end{align*}
### Other uses of TAG

- User’s existing knowledge
- Congruence between features and commands
- These are modelled as derived rules

### Physical and device models

- The Keystroke Level Model (KLM)
- Buxton’s 3-state model

- Based on empirical knowledge of human motor system
- User’s task: acquisition then execution.
  - these only address execution
- Complementary with goal hierarchies

### Keystroke Level Model (KLM)

- lowest level of (original) GOMS
- six execution phase operators
  - Physical motor:  K - keystroking
    P - pointing
    H - homing
    D - drawing
  - Mental       M - mental preparation
  - System       R - response

- times are empirically determined.
  \[ T_{execute} = TK + TP + TH + TD + TM + TR \]
KLM example

**GOAL: ICONISE-WINDOW**

(select)

1. **GOAL: USE-CLOSE-METHOD**
   - MOVE-MOUSE-TO FILE-MENU
   - PULL-DOWN-FILE-MENU
   - CLICK-ON-CLOSE-OPTION
2. **GOAL: USE-CTRL-W-METHOD**
   - PRESS-CONTROL-W-KEY

- compare alternatives:
  - USE-CTRL-W-METHOD vs.
  - USE-CLOSE-METHOD
- assume hand starts on mouse

<table>
<thead>
<tr>
<th>Method</th>
<th>Use-Close-Meth</th>
<th>Use-CTRL-W-Meth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move to menu</td>
<td>0.1</td>
<td>0.40</td>
</tr>
<tr>
<td>Left down</td>
<td>1.35</td>
<td>1.35</td>
</tr>
<tr>
<td>Press Alt-Tab</td>
<td>0.28</td>
<td>0.28</td>
</tr>
<tr>
<td>Total</td>
<td>3.75 s</td>
<td>2.03 s</td>
</tr>
</tbody>
</table>

Architectural models

- All of these cognitive models make assumptions about the architecture of the human mind.
- Long-term/Short-term memory
- Problem spaces
- Interacting Cognitive Subsystems
- Connectionist
- ACT

Display-based interaction

- Most cognitive models do not deal with user observation and perception
- Some techniques have been extended to handle system output (e.g., BNF with sensing terminals, Display-TAG) but problems persist
- Exploratory interaction versus planning