Cognitive models

Overview

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

Cognitive models

They model aspects of user:

- understanding
- knowledge
- intentions
- processing

Common categorisation:

- Competence
- 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

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)

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: ICONISE-WINDOW
  [select
    GOAL: USE-CLOSE-METHOD
      MOVE-MOUSE-TO-WINDOW-HEADER
      POP-UP-MENU
      CLICK-OVER-CLOSE-OPTION
    GOAL: USE-L7-METHOD
      PRESS-L7-KEY]
```

For a particular user:

`Rule 1:` Select `USE-CLOSE-METHOD` unless another rule applies.

`Rule 2:` If the application is `GAME`, select `L7-METHOD`.

CCT

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 contents of 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 $\texttt{LINE} = 5$
$\texttt{COLUMN} = 23$.

Four rules would model inserting a space:

SELECT-INSERT-SPACE
INSERT-SPACE-MOVE-FIRST
INSERT-SPACE-DOIT
INSERT-SPACE-DONE

```
(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)))
```

When fired, adds to working memory

```
(GOAL insert space)
(NOTE executing insert space)
(LINE 5)
(COLUMN 23)
```

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?

Simple extensions possible (e.g., closure)

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)

BNF

Very common notation from computer science

A purely syntactic view of the dialogue

Terminals lowest level of user behaviour

`CLICK-MOUSE, MOVE-MOUSE`

Nonterminals ordering of terminals; higher level of abstraction

`select-menu, position-mouse`

Example of BNF

Basic syntax:

$$nonterminal ::= expression$$

An expression contains terminals and nonterminals combined in sequence (+) or as alternatives (|).

$$
\begin{array}{lll}
draw\_line & ::= & select\_line + choose\_points + \\
 & & \quad last\_point \\
select\_line & ::= & pos\_mouse + CLICK\_MOUSE \\
choose\_points & ::= & choose\_one \\
 & | & choose\_one + choose\_points \\
choose\_one & ::= & pos\_mouse + CLICK\_MOUSE \\
last\_point & ::= & pos\_mouse + DBL\_CLICK\_MOUSE \\
pos\_mouse & ::= & NULL \\
 & | & MOVE\_MOUSE + pos\_mouse
\end{array}
$$

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

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

$$\begin{array}{lcl} copy & ::= & \texttt{cp} + filename + filename \\ & | & \texttt{cp} + filenames + directory \\ move & ::= & \texttt{mv} + filename + filename \\ & | & \texttt{mv} + filenames + directory \\ link & ::= & \texttt{ln} + filename + filename \\ & | & \texttt{ln} + filenames + directory \end{array}$$

No BNF measure could distinguish between this and a less consistent grammar in which

$$\begin{array}{lcl} link & ::= & \texttt{ln} + filename + filename \\ & | & \texttt{ln} + directory + filenames \end{array}$$

Consistency in TAG (cont'd)

In TAG, this consistency of argument order can be made explicit using a parameter, or *semantic feature* for file operations.

Feature	Possible values
Op	$\{ copy, move, link \}$

$$
\begin{aligned}
file\_op[Op] ::= \; & command[Op] + \\
& filename + filename \\
| \; & command[Op] + \\
& filenames + directory
\end{aligned}
$$

$$
\begin{aligned}
command[Op = copy] &::= \texttt{cp} \\
command[Op = move] &::= \texttt{mv} \\
command[Op = link] &::= \texttt{ln}
\end{aligned}
$$

Other uses of TAG

Users *existing knowledge*

Congruence between features and commands

These are modelled as *derived rules*

Physical and device models

Based on empirical knowledge of human motor system

User's task: acquisition then execution.

These only address execution

Complementary with goal hierarchies

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

KLM

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.

$$T_{execute} = T_K + T_P + T_H + T_D + T_M + T_R$$

Example

GOAL: ICONISE-WINDOW
  [select
    GOAL: USE-CLOSE-METHOD
      MOVE-MOUSE-TO-WINDOW-HEADER
      POP-UP-MENU
      CLICK-OVER-CLOSE-OPTION
    GOAL: USE-L7-METHOD
      PRESS-L7-KEY]

Assuming hand starts on mouse:

`USE-L7-METHOD`

Operator	T (sec)
H[to kbd]	0.40
M	1.35
K[L7 key]	0.28
Total	2.03

`USE-CLOSE-METHOD`

Operator	T (sec)
P[to menu]	1.1
B[LEFT down]	0.1
M	1.35
P[to option]	1.1
B[LEFT up]	0.1
Total	3.75

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.

Level of granularity

Exploratory interaction versus planning