# UNIT – VI
# HCI MODELS AND THEORIES

# Goal and Task Hierarchy Model

- Many models make use of the model of mental processing
- In which user achieves goals by solving sub goals in divide-and-conquer fashion
- Two such models are *GOMS, CCT, TAG,* etc.
- Consider an example, we want to generate a report on sales of introductory HCI textbooks
- To achieve this goal we will divide it into several sub goals

# Goal and Task Hierarchy Model

produce report

gather data

. find book names

. . do keywords search of names database

<<further subgoals>>

. . sift through names and abstracts by hand

<<further subgoals>>

. search sales database

<<further subgoals>>

layout tables and histograms

<<further subgoals>>

write description

<<further subgoals>>

# GOMS

- Invented by Card, Moran and Newell is
- Acronym for Goals, Operators, Methods & Selection
- Consists of these four elements
  - Goals - User's goals, describing what user wants to achieve
  - Operators - Lowest level of analysis. They are basic actions that user must perform in order to use system
  - Methods - There are several ways in which a goal can be split into sub goals
  - Selection – No random choice, but attempts to predict which methods will be used

# GOMS

- Create GOMS description of task of photocopying an article from a journal. Discuss the issue of closure in terms of your GOMS description

```
GOAL: PHOTOCOPY-PAPER
.     GOAL: LOCATE-ARTICLE
.     GOAL: PHOTOCOPY-PAGE repeat until no more pages
.     .     GOAL: ORIENT-PAGE
.     .     .       OPEN-COVER
.     .     .       SELECT-PAGE
.     .     .       POSITION-PAGE
.     .     .       CLOSE-COVER
.     .     GOAL: PRESS-COPY-BUTTON
.     .     GOAL: VERIFY-COPY
.     .     .       LOCATE-OUT-TRAY
.     .     .       EXAMINE-COPY
.     GOAL: COLLECT-COPY
.     .     LOCATE-OUT-TRAY
.     .     REMOVE-COPY (outer goal satisfied!)
.     GOAL: RETRIEVE-JOURNAL
.     .     OPEN-COVER
.     .     REMOVE-JOURNAL
.     .     CLOSE-COVER
```

# Linguistic Model

- User’s interaction with computer is often viewed in terms of language
- Several of the dialog notations on linguistic ideas
- BNF grammars are frequently used to specify the dialogs
- Models have been proposed for understanding user’s behavior and analyzing cognitive difficulty of the interface

# BNF

- Representative of linguistic approach is Reisner's use of Backus–Naur Form (BNF) rules to describe the dialog grammar
- This views the dialog at a purely syntactic level, ignoring semantics of the language
- BNF has been used widely to specify the syntax of computer programming languages
- Many system dialogs can be described easily using BNF rules

# BNF

- For example, imagine a graphics system that has line-drawing function

```
draw-line       ::= select-line + choose-points
                    + last-point
select-line     ::= position-mouse + CLICK-MOUSE
choose-points   ::= choose-one
                    | choose-one + choose-points
choose-one      ::= position-mouse + CLICK-MOUSE
last-point      ::= position-mouse + DOUBLE-CLICK-MOUSE
position-mouse  ::= empty | MOVE-MOUSE + position-mouse
```

# BNF

- Names in description are of 2 types: *non-terminals*, in lower case, and *terminals*, in upper case
- Terminals represent lowest level of user behavior, such as pressing key, clicking mouse button, etc.
- Non-terminals are higher-level abstractions
- Non-terminals are defined in terms of other non-terminals and terminals by a definition of form

*Name :: = Expression*

# BNF

- The ‘::=’ symbol is read as ‘is defined as’
- Only non-terminals may appear on the left of a definition
- The right-hand side is built up using two operators ‘+’ (sequence) and ‘|’ (choice)

# Physical and Device Models

- Keystroke-level model
  - Uses problem solving understanding as a basis for detailed predictions about user performance
  - It is aimed at unit tasks within interaction – execution of simple command sequences, typically taking no more than 20 seconds
  - Examples of this would be using a search and replace feature, or changing the font of a word
  - It does not extend to complex actions such as producing a diagram

# Physical and Device Models

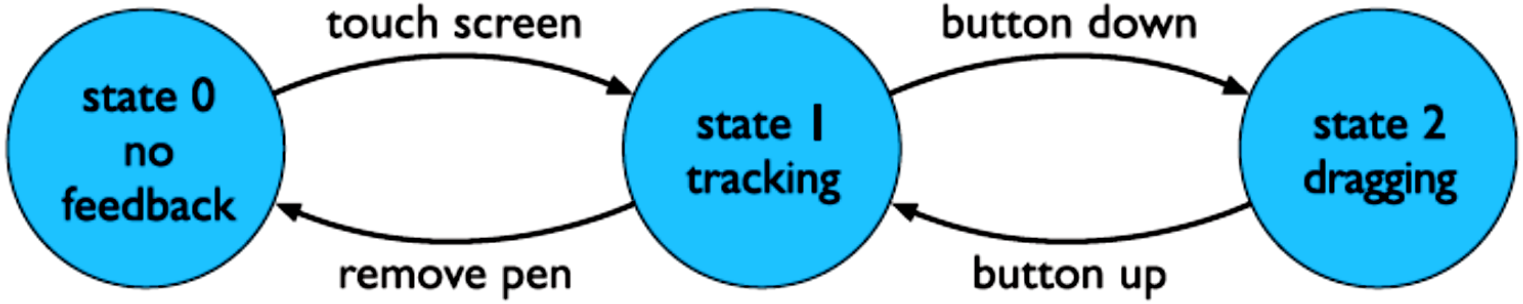
- Keystroke-level model
  - Assumption is that these more complex tasks would be split into subtasks (as in GOMS) before user attempts to map them into physical actions
  - The task is split into two phases:
    - **Acquisition** of the task, when the user builds a mental representation of the task;
    - **Execution** of the task using the system's facilities

# Physical and Device Models

| Operator | Remarks | Time (s) |
|---|---|---|
| **K** | Press key | |
| | good typist (90 wpm) | 0.12 |
| | poor typist (40 wpm) | 0.28 |
| | non-typist | 1.20 |
| **B** | Mouse button press | |
| | down or up | 0.10 |
| | click | 0.20 |
| **P** | Point with mouse | |
| | Fitts' law | $0.1 \log_2(D/S + 0.5)$ |
| | average movement | 1.10 |
| **H** | Home hands to and from keyboard | 0.40 |
| **D** | Drawing – domain dependent | – |
| **M** | Mentally prepare | 1.35 |
| **R** | Response from system – measure | – |

# Physical and Device Models

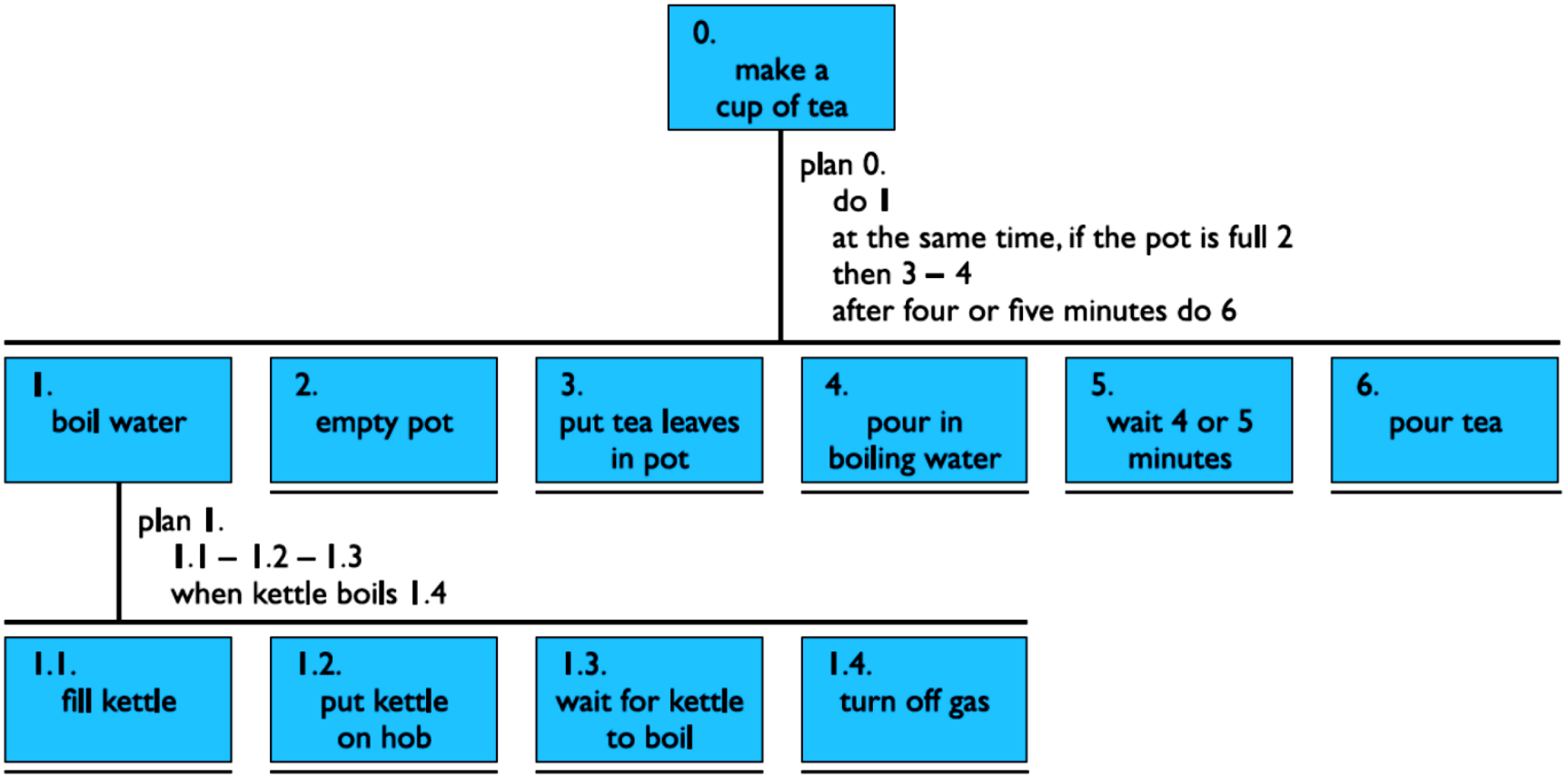
- Three-state model



# Hierarchical Task Analysis (HTA)

- Method of task decomposition
- A task is decomposed into several subtasks
- Outputs of HTA are *hierarchy* of tasks and subtasks
- Also *plans* describing in what order and under what conditions subtasks are performed
- Consider an example, task hierarchy for making a cup of tea
- The main task, ‘make a cup of tea’, is decomposed into six subtasks

# Hierarchical Task Analysis (HTA)



# Uses of Task Analysis

# Manuals and Tuition

**To make cups of tea**

boil water – *see page 2*
empty pot
make pot – *see page 3*
wait 4 or 5 minutes
pour tea – *see page 4*

– page 1 –

**Make pot of tea**
*once water has boiled*

warm pot
put tea leaves in pot
pour in boiling water

– page 3 –

# Requirements Capture and Systems Design

- Task analysis can be used to guide the design of new systems
- Requirements capture is the process of eliciting what a new system should do
- Task analysis is not a form of requirements capture as it refers to existing system, not planned system
- It includes many elements which are not part of the system

# Detailed Interface Design

- Top-level menus can be labeled after the top-level decomposition, and submenus after next level, etc.
- For this, tree may be first reduced to simple tree, thus guaranteeing that each object/action is under exactly one menu
- Alternatively, more complex trees allowing AND & OR as well as XOR branches can be used
- In this case, an object/action may be found by several paths through the menus

# Computer Mediated Communication