Human Computer Interaction

UNIT-2

Lecture 4:

Model-based Design

Mr. Nachiket Sainis / Reena Saini

B K Birla Institute of Engineering & Technology, Pilani

Lecture 4:
(CMN)GOMS

Objective

- In the previous lectures, we learned about the KLM.
- In KLM, we list the elementary (cognitive) steps or operators required to carry out a complex interaction task.
  - The listing of operators implies a linear and sequential cognitive behavior.
- In this lecture, we shall learn about another model in the GOMS family, namely the (CMN)GOMS
  - CMN stands for **C**ard, **M**oran and **N**ewell – the surname of the three researchers who proposed it

KLM vs (CMN)GOMS

- In (CMN)GOMS, a hierarchical cognitive (thought) process is assumed, as opposed to the linear thought process of KLM
- Both assumes error-free and ‘logical’ behavior
  - A logical behavior implies that we think logically, rather than driven by emotions

(CMN) GOMS – Basic Idea

- (CMN)GOMS allows us to model the the task and user actions in terms of four constructs (Goals, Operators, Methods, Selection rules)
  - **Goals**: represents what the user wants to achieve, at a higher cognitive level. This is a way to structure a task from cognitive point of view
  - The notion of Goal allows us to model a cognitive process hierarchically

(CMN) GOMS – Basic Idea

- **Operators**: elementary acts that change user's mental (cognitive) state or task environment. This is similar to the operators we have encountered in KLM, but here the concept is more general.

- **Methods**: these are sets of goal-operator sequences to accomplish a sub-goal

- **Selection rules**: sometimes there can be more than one method to accomplish a goal. Selection rules provide a mechanism to decide among the methods in a particular context of interaction

Operator in (CMN)GOMS

- As mentioned before, operators in (CMN) GOMS are conceptually similar to operators in KLM

- The major difference is that in KLM, only seven operators are defined. In (CMN)GOMS, the notion of operators is not restricted to those seven.

  - The modeler has the freedom to define any "elementary" cognitive operation and use that as operator.

Operator in (CMN)GOMS

- The operator can be defined
  - At the keystroke level (as in KLM)
  - At higher levels (for example, the entire cognitive process involved in “closing a file by selecting the close menu option” can be defined as operator)
  - (CMN)GOMS gives the flexibility of defining operators at any level of cognition and different parts of the model can have operators defined at various levels

Example 1

- Suppose we want to find out the definition of a word from an online dictionary. How can we model this task with (CMN)GOMS?

- We shall list the goals (high level tasks) first
  - Goal: Access online dictionary (first, we need to access the dictionary)
  - Goal: Lookup definition (then, we have to find out the definition)

Example 1 continue

- Next, we have to determine the methods (operator or goal-operator sequence) to achieve each of these goals
  - Goal: Access online dictionary
    - Operator: Type URL sequence
    - Operator: Press Enter
  - Goal: Lookup definition
    - Operator: Type word in entry field
    - Goal: Submit the word
      - Operator: Move cursor from field to Lookup button
      - Operator: Select Lookup
    - Operator: Read output

Example 1 explanation

- Notice the hierarchical nature of the model
- Note the use of operators
  - The operator “type URL sequence” is a high-level operator defined by the modeler
  - “Press Enter” is a keystroke level operator
  - Note how both the low-level and high-level operators co-exist in the same model

Example

- Note the use of methods
  - For the first goal, the method consisted of two operators
  - For the second goal, the method consisted of two operators and a sub-goal (which has a two-operators method for itself)

Another Example

- The previous example illustrates the concepts of goals and goal hierarchy, operators and methods

- The other important concept in (CMN)GOMS is the selection rules

- Suppose we have a window interface that can be closed in either of the two methods: by selecting the ‘close’ option from the file menu or by selecting the Alt key and the F4 key together. How we can model the task of “closing the window” for this system?

Example 2

- Here, we have the high level goal of “close window” which can be achieved with either of the two methods: “use menu option” and “use Alt+F4 keys”
  - This is unlike the previous example where we had only one method for each goal
- We use the “Select” construct to model such situations.

Example 2

Goal: Close window

- [Select Goal: Use menu method

  Operator: Move mouse to file menu
  Operator: Pull down file menu
  Operator: Click over close option

  Goal: Use Alt+F4 method

  Operator: Press Alt and F4 keys together]

The select construct implies that “selection rules” are there to determine a method among the alternatives for a particular usage context

Example2

- Example selection rules for the window closing task can be

  Rule 1: Select “use menu method” unless another rule applies
  Rule 2: If the application is GAME, select “use Alt+F4 method”

  The rules state that, if the window appears as an interface for a game application, it should be closed using the Alt+F4 keys. Otherwise, it should be closed using the close menu option.

Example 3

- GOAL: DELETE-FILE
- . GOAL: SELECT-FILE
- . . [select: GOAL: KEYBOARD-TAB-METHOD
- . . GOAL: MOUSE-METHOD]
- . . VERIFY-SELECTION
- . GOAL: ISSUE-DELETE-COMMAND
- . . [select*: GOAL: KEYBOARD-DELETE-METHOD
- . . . PRESS-DELETE
- . . . GOAL: CONFIRM-DELETE
- . . GOAL: DROP-DOWN-MENU-METHOD
- . . . MOVE-MOUSE-OVER-FILE-ICON
- . . . CLICK-RIGHT-MOUSE-BUTTON
- . . . LOCATE-DELETE-COMMAND
- . . . MOVE-MOUSE-TO-DELETE-COMMAND
- . . . CLICK-LEFT-MOUSE-BUTTON
- . . . GOAL: CONFIRM-DELETE

- . . GOAL: DRAG-AND-DROP-METHOD
- . . . MOVE-MOUSE-OVER-FILE-ICON
- . . . PRESS-LEFT-MOUSE-BUTTON
- . . . LOCATE-RECYCLING-BIN
- . . . MOVE-MOUSE-TO-RECYCLING-BIN
- . . . RELEASE-LEFT-MOUSE-BUTTON]
-
- *Selection rule for GOAL: ISSUE-DELETE-COMMAND
- If hands are on keyboard, use KEYBOARD-DELETE-METHOD,
- else if Recycle bin is visible, use DRAG-AND-DROP-METHOD,
- else use DROP-DOWN-MENU-METHOD

Steps for Model Construction

- A (CMN)GOMS model for a task is constructed according to the following steps
  - Determine high-level user goals
  - Write method and selection rules (if any) for accomplishing goals
  - This may invoke sub-goals, write methods for sub- goals
  - This is recursive. Stop when operators are reached

Use of the Model

- Like KLM, (CMN)GOMS also makes quantitative prediction about user performance
  - By adding up the operator times, total task execution time can be computed
- However, if the modeler uses operators other than those in KLM, the modeler has to determine the operator times

Use of the Model

- The task completion time can be used to compare competing designs
- In addition to the task completion times, the task hierarchy itself can be used for comparison
  - The deeper the hierarchy (keeping the operators same), the more complex the interface is (since it involves more thinking to operate the interface)

Model Limitations

- Like KLM, (CMN)GOMS also models only skilled (expert) user behavior
  - That means user does not make any errors
- Can not capture the full complexity of human cognition such as learning effect, parallel cognitive activities and emotional behavior