DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF TORONTO

CSC428F/2514F

HUMAN-COMPUTER INTERACTION

Lecture 10

INTERACTIVE SYSTEMS ANALYSIS AND DESIGN; TASK MODELS

Ronald Baecker
Professor of Computer Science,
Electrical and Computer Engineering, and Management
University of Toronto

10.1 Systems analysis and design

How to use understanding of user needs and user activities and user psychology in the development of an interactive system

Description of user needs and activities -->
- Analysis and abstraction of user task (activity) model -->
- Synthesis or enhancement of solution to support tasks -->
- Analysis of solution in terms of task model -->
- Resynthesis of new solution --> etc.

We call this iterative process *systems analysis and design*

Introduction today, see also a software engineering course for a somewhat different perspective on the same issue

10.2 Basing systems analysis on task analysis

Task analysis is a description of:
- What work does the user do?
- How does he/she do this work?
  - What roles do tools play in the work?
  - What roles do collaborators play in the work?
- Why are things done they way they are done?

Why do a task analysis?
- Insights into how people accomplish their work
- Insights into why people work the way they do
- Insights into what a system should do
- Insights into how a system could work
- Insights into how to improve upon the current way of work

Understanding how the task is done
- Inputs and outputs
- Operational procedures
- Decision points, planning, problem solving
- Use of tools
- Use of collaborators
- Task requirements
  - Criticality
  - Speed, accuracy, quality
  - Frequency and sequencing of execution
  - Flexibility and discretion: pace, priority, procedure
  - Dependency on, relationships to other tasks
- Task context — physical, perceptual, cognitive, environmental, health and safety

10.3 Models in task analyses

Hierarchic task models
Concurrent process models
Data flow models
Object models
Scenarios as models
Task activity models (cognitive models)
Meta-models

10.4 Hierarchic task models

Representation of task a sequence of subtasks
Subtasks may in turn be represented as sequences of subsubtasks, etc.
Mail order example shown in Figure 10.1
See N&L for description of how to use node labeling to deal with variability of sequencing within this kind of notation

Figure 10.1 A hierarchic model of a mail order task (N& L, 1995, Fig. 6.7, 118)

10.5 Concurrent process models

What if subtasks are done concurrently as well as sequentially?
- We can represent them as concurrent processes
- Air traffic control example shown in Figure 10.2

Figure 10.2 Representing air traffic control activities as concurrent processes (N& L, 1995, Fig. 4.7, 77)

10.6 Data flow models

Focus on modelling data — its forms, flow, interrrelationships
- Data flow diagrams identify where data is processed or stored and the paths along which it flows
- We identify data inputs and outputs and processing and storage components

A mail order system is shown in Figure 10.3
- Entities, e.g., organizations, are shown as boxes
- Individual processes are shown as circles
- Data stores are shown as pairs of parallel lines
- Data flows are shown as arrows

10.7 Object models

Representing tasks as objects (processes and data)
- Tasks represented by classes, subtasks by subclasses
- Clear relationships between model of tasks and implementation of systems in languages such as C++, Java, Smalltalk

10.8 Scenarios as models

Specific scenarios also function as models of tasks
- Advantage: Cognitively accessible
- Disadvantage: Need to generalize

Police detective task example shown in Figure 10.4

Directory assistance example from lecture 5 shown in Fig. 10.5

Figure 10.3 Data flow diagram of a mail order system (N& L, 1995, Fig. 6.12,127)

Figure 10.4 Activities of one day's work by a police detective (N& L, 1995, Fig. 5.7,102)

Figure 10.5 An example of a directory assistance call, focusing on operator's knowledge work (Muller, et al., 1995, p. 132)

10.9 Task activity models

Carrying out tasks can be thought of in terms of procedures
- Hierarchic decomposition: tasks and subtasks (10.4)
- Sequential execution of a series of steps
- Repetition of an action, a subtask
- Conditional execution of an action

Thus tasks can be represented or described in terms of formal notations, task description languages

Example: Cognitive modelling using the GOMS notation of Card, Moran, and Newell
- Set of *Goals* that need to be achieved in order to complete a task
- Set of *Operators* that users are skilled at and which must be used to complete the task
- Set of *Methods* for achieving the goals, which are sequences of subgoals and operators
- Set of *Selection rules* for choosing among a goal's competing methods
- Example: Figures 10.6

Later in term, we will examine special cases of GOMS modelling (keystroke models) that will allow us to predict the time an expert user will take to execute a task using a system, providing he/she uses a method without error

Figure 10.6 A hierarchic GOMS description of a manuscript editing task with a line-oriented text editor (BGBG, 1995, p. 591, from Card and Moran, Fig. 4)

10.10 A meta-model: Norman's task performance model

A framework for understanding how users' actions relate to their goals and to the system they use

Cycle of interaction (Figure 10.7) consists of:

Establishing the goal
Forming the intention
Specifying the action sequence
Executing the action (motor activities)
Perceiving the system state (perceptual activities)
Interpreting the state
Evaluating the system state with respect to the goals

An application of this meta-model to modelling a page layout task appears in Figure 10.8

Fig. 10.7 Norman's seven-stage model of interaction (N&L, 1995, Fig. 13.16, 343)

Figure 10.8 Applying Norman's model to the task of fitting a memo on one page (N&L, 1995, Fig. 3.8, 60)

10.11 From task models to requirements & specifications

Eventually, we must take these understandings of user tasks embedded in task models and map them in to system requirements documents and functional specifications

How to do this is discussed in CSC408, Software Engineering

10.12 Users in systems analysis and design

Clearly, we can build better task models if the users who do the tasks can be involved in the process

Three approaches
- User-centred design
- Human-centred design
- Participatory design

10.13. Participatory design

Evolved in Scandinavia through alliances between computer scientists and trade unionists (see also Enid Mumford's*socio-technical design* in England)

Users must be involved in design from the outset, not merely as clients or consultants or informants, but as equal participants and co-designers

Both the process and end result of designing and implementing information systems are intensely political, so this must be recognized and dealt with systematically and openly if the real political conflicts are to be dealt with effectively

"...I claim the importance of rethinking the design process to include structures through which ordinary people at their workplace more democratically can promote their own interests. I also claim the importance of rethinking the use of descriptions in design, and of developing new design methods that enable users of new or changed computer artifacts to envision their future use situation and to express all their practical competence and creativity in designing their future." (Ehn, in Dunlop & Kling, p. 297)

Envisioning future use situations to ground new designs — Expressing practical competence and creativity by thinking about tasks in terms of metaphors — Next lecture.....