Human Computer Systems

Lecture : Design and Models of the User

Lecture Content: Design and Models of the User

- There are a range of models that can be used during the interface design process
- This lecture looks at 2 types of models, both are highly user-centred
  - User requirements modelling looks <u>outwards</u> at the larger human context
  - Cognitive Models focuses <u>inwards</u> on the individual user

1. User Requirements Modelling

- Often in software engineering requirements capture focuses on the functional requirements of the system (what it should do) - and less on the human issues such as usability & acceptability - User Requirements Modelling (URM) redresses this imbalance
- consider some models

URM - Socio-Technical Models

- Concerned with:
  - technical, social, organisational & human aspects of design
- User Skills & Task Match (USTM) and its form for small organisations called CUSTOM
- Open Systems Task Analysis (OSTA)
- Effective Technical & Human Implementation of Computer Systems (ETHICS)
- Dix... pp224-227

URM - ETHICS

- Effective Technical and Human Implementation of Computer-based Systems
- Developed to support participative development : organisational and social context of primary importance
- Addresses both technical design and human activity design
- 15 steps (or 6 stage 26 step) method

Ethics

- A methodology that embodies an ethical position.
- Devised by Enid Mumford
- A methodology based on the participative approach.
- It encompasses the socio-technical view that for a system to be effective the technology must fit closely with the social and organisational factors.
- This means that an improved quality of working life and enhanced job satisfaction of the users must be a major objective.

Objectives of the ETHICS Method

- To enable the future users of a new system to play a major role in its design. This involves a learning process and a set of simple diagnostic and socio-technical design tools.
- To ensure that new systems are acceptable to users because they both increase user efficiency and job satisfaction.
- To assist users to become increasingly competent in the management of their own organizational change so that this becomes a shared activity with the technical specialists and reduces the demand for scarce technical resources

Diagnosis of Needs

- Job satisfaction needs
  - So defined in ETHICS as the attainment of a good ‘fit’ between what employees are seeking from their work- their job needs, expectations and aspirations - and what they are required to do in their work.
  - the organisational job requirements which mould their experience.
  - Need a viable theory or framework for measuring job satisfaction.
  - One has been developed for the ETHICS method

Job satisfaction

- Defined as the achievement of a good fit between job needs and expectations and job experience.
  - The KNOWLEDGE ‘fit’
  - The PSYCHOLOGICAL ‘fit’
  - The EFFICIENCY ‘fit’
  - The TASK STRUCTURE ‘fit’
  - The ETHICAL (social value) ‘fit’

Setting objectives

- If the proposed system is a large one and incorporates a number of sub-systems that have to be integrated, the next, and most important, step is to set specific efficiency, effectiveness and job satisfaction objectives for the new system.
- Alternative design strategies can then be tested against these and the strategy that best fits the objectives is the one chosen for implementation.

Socio-technical Systems Design

Set human objectives (improved job satisfaction and quality of working life)

Specify human alternatives (work organisation, job design)

Set efficiency objectives (improved business efficiency)

Specify technical and administrative alternatives (hardware, software, work procedures, information flow)

Match as socio-technical alternatives

Rank in terms of ability of each alternative to meet human and efficiency objectives

Consider costs, resources, constraints

Select best socio-technical solution

URM - Soft Systems Methodology

- SSM takes a broader view- the organisation as a system of which the technology and the people are components
- 7 stages (a distinction is made between the ‘real world’ stages (1-2, 5-7) and the system stages (3-4)
- Dix pp227-229

URM - Soft Systems Methodology

- Evolved from social sciences general system theory
- Holistic approach seeing sytems development as inextricably linked to human activity
- Used to deal with “fuzzy, ill-defined problems”

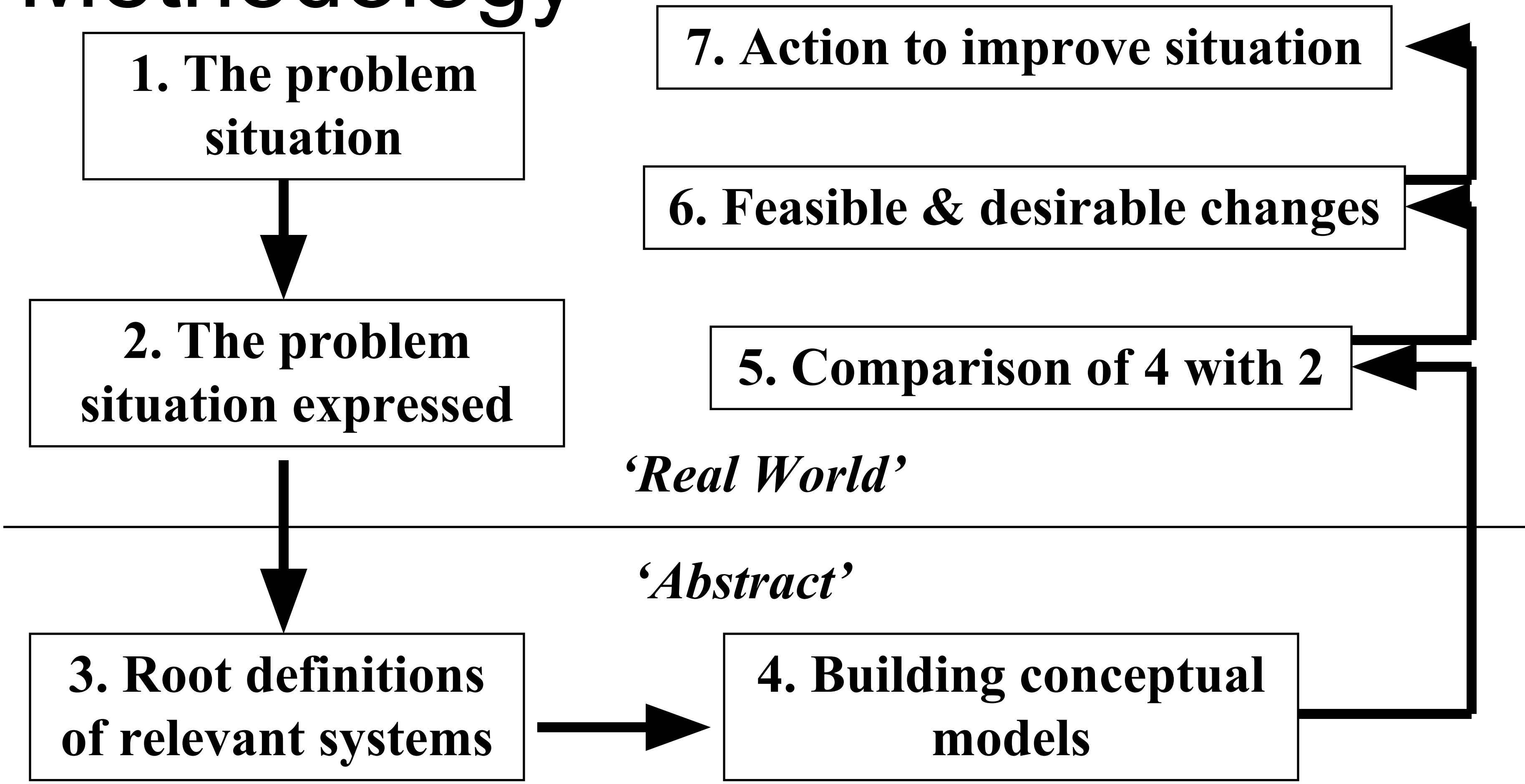
URM - Soft Systems Methodology

- Can be used as a front-end to more deterministic approaches to define the problem they will solve
- Main techniques used are:
  - Rich pictures: to capture views of the problem expressed by various stakeholders
  - CATWOE
  - Both used to generate “Root Definition”
  - Conceptual model

URM - CATWOE

- **Customers** – ones who benefit from the activity
- **Actors**: Ones who carry out the activities of the system
- **Transformation**: Conversion of input to output
- **Weltanschauung**: how the system is perceived from a particular perspective (assumptions about the system)
- **Owner** - one who causes the system to be developed, ,owns it or causes it to be terminated
- **Environment** – the world that influences the system but has no control over it.

URM - Soft Systems Methodology



URM - Soft Systems Methodology

- Question:

  The government of a small african country is considering implementing a system that will require schools to administer standard assessment tests for pupils at ages 7, 11, 14 and 16. The results will be published nationally so that comparisons between schools can be made.Identify the stakeholders in this system.

URM - Soft Systems Methodology

- Stakeholders:

  teachers, parents, children, educationalists, government officials, prospective parents, employers, and anyone interested in the education of children

URM - Soft Systems Methodology

- Question:

  Produce a root definition for the school system using the CATWOE elements

URM - Soft Systems Methodology

- Solution:

<u>Clients</u> - teachers, schoolchildren, parents, government, educationalists

<u>Actors</u> - teachers, schoolchildren, assessment material producers, government

<u>Transformation</u> - from input of children and assessment material to produce figures showing scores for tests for each school in each age group

URM - Soft Systems Methodology

Weltanschauung - government view that such tests are a feasible & desirable method of comparisons

Owners - government

Environment - schools in the education system

URM - Participatory Design

- Design in the workplace, incorporating the user as an individual and as a member of the design team
- users are active collaborators in design process
- users viewed as experts in the in the work context so should be allowed to contribute actively to the design

URM - Participatory Design

- 3 characteristics
  - aims to improve the work environment
  - involves collaboration - user involved in design team
  - iterative - design subject to revision at each stage
- originated in Scandinavia - now promoted in law and accepted work practices there - not widely practiced elsewhere(often due to cost & time involved)

URM - Participatory Design

- Brainstorming
- Storyboarding
- Workshops
- Pencil & Paper exercises

2. Cognitive Models

- Representation of users as they interact with an interface
- model some aspect of user's understanding, knowledge, intention or processing
- level of representation differs from technique to technique - from high level goals and the results of problem-solving activities to descriptions of motor-level activities eg keystrokes, mouse clicks etc

Cognitive Models

- Look at 3 categories
  - Hierarchical representations of user’s task and goal structure
  - Linguistic & grammatical models
  - Physical & device-level models

Goal & Task Hierarchies

- Many models use a model of mental processing in which the user achieves goals by solving sub-goals in a divide-and-conquer fashion
- look at 2 models GOMS & CCT (Cognitive Complexity Theory)

GOMS Summary

- **Goals**: what the user wants to achieve.
  - A memory point to return to on error
  - A point of evaluation
- **Operators**: lowest level of **<u>this</u>** analysis
  - System affected, eg "click minimise"
  - User affected, eg scan task bar

GOMS Summary

- **Methods**: Identify sub-goals
  - eg Maximise window (so that it can be seen)
  - Choices of Method
- **Selection**: choice of Method with rules
  - eg "Press Terminate button Method" OR "File-Exit Method"

GOMS Summary

- ***Needs:***
  - Detailed understanding of the domain
  - Understanding of users' problem solving strategies
- ***Predictions:***
  - Goal stacking indicates short term memory needs
  - Refinement of rules based on objective samples
  - Prediction of user commands
  - Prediction of time to use

GOMS Summary

- ***Limitations***:
  - Detailed analysis
  - Experts focus
  - Doesn't deal with errors
  - Choice of level (granularity)
  - Often used post-hoc

Cognitive Complexity Theory

- Begins with GOMS
- provides more predictive power
- 2 parallel descriptions
  - users goals - based on GOMS-like hierarchy, expressed using production rules
  - computer system (device)

Production Rules

- If ...condition ....then ....action
- if condition is true then the production rule is fired
- set of ‘style’ rules for novices i.e. Test constantly all rules in working memory to check for feedback

Goal hierarchies - closure

- E.g early ATMs gave customers money before returning card
- many customers left cards behind despite messages on-screen
- ‘closure’ problem
- user’s main goal is satisfied (get money) and user doesn’t complete or close subtasks

Cash Machine Example

GOAL : GET-MONEY

GOAL : USE-ATM

INSERT-CARD

ENTER-PIN

ENTER-AMOUNT

COLLECT-MONEY

<< outer goal satisfied - goal stack popped >>

COLLECT-CARD - sub-goal missed

Cash Machine Example

- Banks now return card before money dispensed
- Can make general rule - no higher level rule satisfied until all sub-goals satisfied
- but not easy to predict when user will consider goal satisfied e.g. Collect card and forget money!

Linguistic Models

- BNF (Backus-Naur Form)
- dialog grammar
- ignores semantics of language
- widely used to describe programming languages & dialogs

Linguistic Models - BNF

- Example: Polygon line drawing
- points to drawing area - double click to indicate last point of polygon

Linguistic Models - BNF

```
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
```

Linguistic Models - BNF

- Notes:
- terminals UPPERCASE (lowest level of user behaviour eg click mouse)
- non-terminals lowercase
- ::= reads as ‘ is defined as’
- + (sequence) | (choice)
- empty - no action
- position-mouse & choose-points recursive - ie call themselves so any number of moves

Analysing BNF descriptions of interface

- Count rules - the more rules the more complicated the interface
- count + & | operators - penalise long, complex singles rules - split into smaller less-complicated ones

Physical & Device Models

- Keystroke-level Model (KLM)
  - aimed at unit tasks within interaction
  - execution of simple command sequences
  - typically 20 secs
  - eg search & replace, change size etc
  - split into subtasks (like GOMS) before mapping to physical actions
  - can be thought of as low-level GOMS

Keystroke-level Model (KLM)

- 5 physical motor operators
- 1 mental operator
- 1 system response operator

Keystroke-level Model (KLM)

K	Keystroking, actually striking keys, including SHIFT etc
B	Pressing mouse button
P	Pointing, move mouse to target
H	Homing, switch hand between mouse & keyboard
D	Drawing lines with mouse
M	Mentally prepare for physical action
R	System response which may be ignored if user does not wait for it

Keystroke-level Model (KLM)

- Executing tasks
  - interleaving occurrences of various operators
  - eg correct typing error & return to previous position

Keystroke-level Model (KLM)

1	move hand to mouse	**H**[mouse]
2	position mouse after bad character	**PB**[LEFT]
3	return to keyboard	**H**[keyboard]
4	delete character	**MK**[DELETE]
5	type correction	**K**[char]
6	reposition insertion point	**H**[mouse]**MPB**[LEFT]

Keystroke-level Model (KLM)

- Model predicts time taken during execution phase by adding time estimates
- Card, Moran & Newell

Keystroke-level Model (KLM)

Operator	Remarks	Time (secs)
K	Key press -	
	Good typist (90 WPM)	0.12
	Poor typist (40 wpm)	0.28
	Non-typist	1.20
B	Point button press	
	Down or up	0.10
	Click	0.20

Keystroke-level Model (KLM)

Operator	Remarks	Time (secs)
P	Point with mouse	
	Fitts' law	0.1 $\log_2$
	Average movement	1.10
H	Homing hands to & from keyboard	0.40
D	Drawing – domain dependent	-
M	Mentally prepare	1.35
R	Response from system - measure	-

Keystroke-level Model (KLM)

- Times taken are obtained from empirical data
- keying obviously depends on typing skill
- pressing a mouse is usually quicker than typing
- more accurate predictions if mouse button presses **(B)** separated from keystrokes **(K)**

Keystroke-level Model (KLM)

- Pointing times can be calculated by Fitts law and thus depends on size & position of target <u>or</u> use average time
- drawing times depend on the number and length of lines drawn and is domain specific
- homing & mental preparation are assumed to be constant

Keystroke-level Model (KLM)

- Mental operator is the most complex part of KLM
- user is assumed to have decided what to do, and how to do it
- mental preparation is just the slight pause to recall what to do next
- complicated heuristics to decide where to put **M** operators, but depends on level of chunking

Keystroke-level Model (KLM)

- Physical operator times all depend on the level of skill of the user, likewise mental operators depend on expertise of user, so consider type of user before using KLM
- power of KLM lies in ability to make comparisons of key tasks - faster & cheaper than lengthy experiments/ tests
- can also look at times to decide which parts of system to training on - teach fastest method etc

Summary

- Cognitive architectures i.e. how the user performs the cognitive processing involved in carrying out a task
- GOMS - divide & conquer, split into subgoals
- CCT - production rules - stored in long term memory, ‘matched’ with short term memory to decide which rules to fire
- KLM - values for motor & mental operators