HCI Lecture 5:
Conceptual models: task analysis
Barbara Webb

Key points:
- Motivation: obtaining conceptual models, understanding the user, filling out guidelines...
- Different approaches
  - Knowledge-based/Cognitive
  - Task decomposition
  - Entity-relationship description
- Applications and Issues

Motivation

- A good way to approach design is to have a conceptual model: how the designer wants the user to think system works
- A good starting point for the conceptual model is to understand how the user understands the task

"assume knowing something about how users approach and carry out tasks will aid software designers when making design decisions which will ultimately affect computer system usefulness and usability" (Johnson & Johnson, 1991)

Motivation

- This needs to get specific: 'task' can be defined as the activities required to obtain a particular goal using a particular device
  - E.g. buying a book at a shop vs. online
  - Using an online book shop to buy a gift vs. buy a textbook

- Note also that many design guidelines (lecture one) are implicitly task and user specific:
  - "Speak the user's language" – so what is their language? What terms are used in the task domain, and how are they related?
  - "Be consistent" – with which other systems? Is the new design replacing an existing system, and will procedures transfer?
  - "Give appropriate feedback" - what is appropriate for this group of users executing this task? What information transfers to and from the user are necessary or desirable?

Motivation

- Task analysis attempts to address these issues
- Can be seen as a specific focus within requirements analysis, which normally comprises:
  - Functional specification of what system will do
  - Data requirements and data flow
  - Usability requirements
- Task analysis: understanding and representing what the user has to do when interacting with the system
  - May involve functions, data and usability issues
- Aim: to provide a 'rigorous structured characterisation' or 'complete and explicit' description of user activity
- Reality: more often a guiding framework
Approaches to task analysis

- There are many approaches to task analysis, which emphasise different aspects of the conceptual model.
- ‘Knowledge-based’:
  - focus on hierarchical structuring of concepts relevant to the task
  - what does the user know, and how is it organised (in long term memory)?
  - Closely related is ‘Cognitive Task Analysis’ (CTA) i.e. understanding user’s internal mental states during a task
- ‘Task decomposition’:
  - focus on ordering of actions to achieve goals
  - Recursive sub-division into subtasks
  - Specifying plans
- ‘Entity-relationship’ (c.f. UML) - relations between objects, actions and agents/actors (see Dix section 15.5 for further discussion)

Knowledge-based

- Two well known methods for description are Task Knowledge Structure (TKS) and Task descriptive hierarchy (TDH)
- E.g. of TDH taxonomy (see Dix, sect. 15.4)

```
kitchen item
/   shape XOR
|   XOR dished mixing bowl, casserole, saucepan
|   soup bowl, glass
|   flat plate, chopping board, frying pan
|   function
|   preparation mixing bowl, plate, chopping board
|   cooking frying pan, casserole, saucepan
|   dining XOR
|   for food plate, soup bowl, casserole
|   for drink glass
```

Task decomposition

- Best known method is Hierarchical Task Analysis (HTA)
- Start from overall goal, e.g. clean the house
- Break down into numbered subgoals, e.g.
  0. clean the house
     1. get the vacuum cleaner out
     2. clean the rooms
        2.1 clean the hall
        2.2 clean the living rooms
        2.3 clean the bedrooms
     3. empty the dust bag
     4. put vacuum cleaner and attachments away
- Describe order of subtasks, e.g.
  Plan 0: do 1, 2, 4 in order; when dust bag full, do 3
  Plan 2: do 2.1, 2.2, 2.3 in any order, as needed
Task decomposition

- Refining the HTA: check and improve the decomposition
- Some heuristics are:
  - **Check for paired actions**: where is “turn on gas”?
  - **Restructure**: generate task “make pot”
  - **Balance**: is “pour tea” simpler than “make pot”?
  - **Generalise**: make one cup ... or more
  - **Sub-operations should be mutually exclusive and collectively exhaustive**
- **N.B.** Important that task decomposition is an iterative process that goes back to check decomposition with user

### Diagrammatic notation

#### Plan 0.
- **0.** make a cup of tea

#### Plan 1.
1.1 – 1.2 – 1.3
when kettle boils 1.4

1.1. fill kettle
1.2. put kettle on hob
1.3. wait for kettle to boil
1.4. turn off gas

#### Plan 1.
1.1 – 1.2 – 1.3
when kettle boils 1.4

1.1. fill kettle
1.2. put kettle on hob
1.3. wait for kettle to boil
1.4. turn off gas

#### Plan 2.
2. empty pot
3. put tea leaves in pot
4. pour in boiling water
5. wait 4 or 5 minutes
6. pour tea

#### Plan 3.
3.1 – 3.2 – 3.3

3.1. warm pot
3.2. put tea leaves in pot
3.3. pour in boiling water

#### Plan 5.
5.1 – 5.2 – 5.3

5.1. put milk in cup
5.2. fill cup with tea
5.3. do sugar

#### Plan 6.
6.1 – 6.3

6.1. ask guest about sugar
6.2. add sugar to taste

### Types of plan

- **sequence** 1.1 then 1.2 then 1.3
- **optional** if the pot is full 2
- **wait** when kettle boils, do 1.4
- **cycles** do 5.1 5.2 while there are still empty cups
- **parallel** do 1; at the same time ...
- **discretionary** do any of 1.3.1, 1.3.2 or 1.3.3 in any order

Most plans use several of these.

Waiting can be considered:
- a task — for “busy” waits, e.g. making tea
- part of the plan — end is the event, e.g. email reply received
Task decomposition

When is decomposition complete?

- When reach ‘actions’, i.e. task that user can execute without problem solving (but note this may differ for different users)
- Above ‘device specific’ implementation details (but note shape of task is often device dependent)
- Suggested heuristic is to stop when probability of error multiplied by cost of error is below threshold

- See also ‘ConcurTaskTrees’ (CTT) – expands HTA with more explicit temporal relationships, additional task attributes.

Specific applications

Have introduced task analysis as a route to obtain good conceptual model but can apply more directly:

- taxonomies suggest menu layout
- object/action lists suggest interface objects
- task frequency guides default choices
- task sequences guide dialogue design
- task structure guides organisation of help menus/manuals,
- domain description clarifies learning requirements

Issues

- There are many different tools; can be useful to try several complementary approaches
  - Knowledge-based methods tend to be bottom-up, HTA is top-down
- Also valuable to use multiple sources of evidence in constructing and refining the analysis:
  - Knowledge elicitation techniques (interviews, listing, sorting)
  - Observation of users engaged in task (existing or hypothetical)
  - Questioning about options, error recovery
  - Collecting artifacts, e.g. documentation, items used
- Most methods still lack a ‘direct translation’ from the task analysis to the system design
  - Recent interest in relating task analysis methods to UML, i.e. taking a user rather than system focus in creating use-cases, activity and sequence diagrams, class diagrams etc.
  - See e.g. Nobrega et. al. (2005)

References


- See also:
  Dix et. al. Chapter 15