Task Analysis

HCI Lecture 5

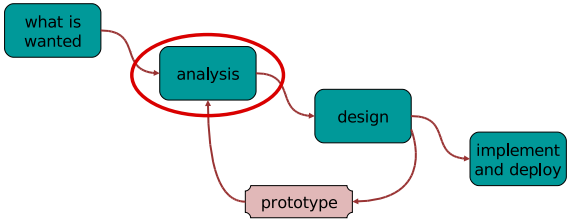
David Aspinall

Informatics, University of Edinburgh

5th October 2007

Outline

Focus on Analysis



- Interaction design driven by what is wanted
- Analysis of tasks and knowledge informs:
  - functionality and objects offered in interface;
  - organisation (layout, grouping, navigation).

Task Analysis

- **Task Analysis** is the study of the way people perform their jobs. Aim is to determine:
  - what they do
  - what things they use
  - what they must know
- Task analysis gathers both *declarative* and *procedural* knowledge
  - Declarative: objects and relationships
  - Procedural: task sequences, goals and subgoals
  - Also dependencies and constraints
- Originally a tool for writing training manuals, now used more widely in business process analysis
- Emphasises users+existing tasks, rather than desired system as in **systems analysis**
- Emphasises observable behaviour and whole job, rather than internal mental state and "unit" tasks as in **cognitive models**

Example Task: Cleaning House

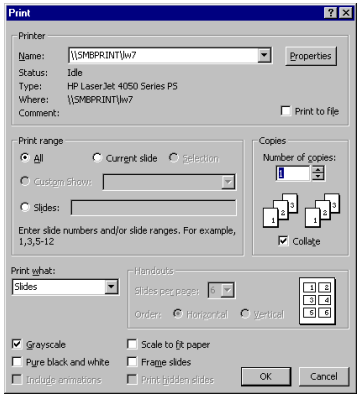
To clean the house:

get the vacuum cleaner out;
fix the appropriate attachments;
clean the rooms;
when the dust bag gets full, empty it;
put the vacuum cleaner and tools away.

We must know about:

- vacuum cleaners, their attachments, dust bags, cupboards, rooms.

Example Layout



- Items related by proximity and boundaries
- Layout suggests order, but doesn't impose it

Approaches

There are many different approaches, notations and techniques.

- **Task decomposition**
  - splitting task into (ordered) subtasks
- **Knowledge-based techniques**
  - what the user knows about the task
  - and how it is organised
- **Entity/object based analysis**
  - relationships between objects, actions and the people who perform them
  - gardener *digs* soil *using* spade
  - cf database design
  - not covered further here

General Method

The general method for Task Analysis is:

- observe
- collect unstructured lists of words and actions
- organize using notation or diagrams

Task Decomposition

- Aims:
  - describe the actions people do
  - structure them within task subtask hierarchy
  - describe order of subtasks
- Variants:
  - **Hierarchical Task Analysis (HTA)**
    – the most common
  - **ConcurTaskTrees (CTT)**, by Paternò (2000)
    – uses LOTOS temporal operators
- Procedural task knowledge elicitation techniques:
  - Observation, re-enactment
  - Ask about procedures and triggers (pre-conditions)
  - "What happens if X goes wrong?"
  - Sorting steps into appropriate orders

Textual HTA

Hierarchy description

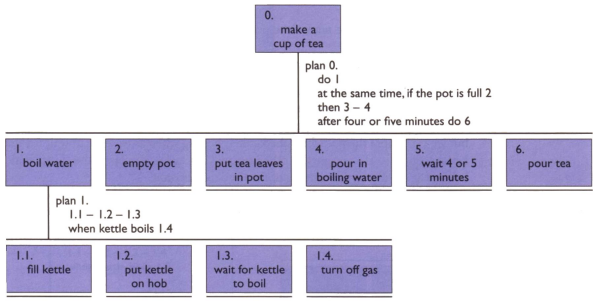
0. clean the house
   1. get the vacuum cleaner out
   2. get the appropriate attachment
   3. clean the rooms
      3.1 clean the hall
      3.2 clean the living rooms
      3.3 clean the bedrooms
   4. empty the dust bag
   5. put vacuum cleaner and attachments away

Plans

Plan 0: do 1, 2, 3, 5 in order;
when dust bag full, do 4

Plan 3: do 3.1, 3.2, 3.3 in any order, as needed

Diagrammatic HTA



[Dix et al, p. 515]

Refinement

How to check or improve the initial HTA?
Some heuristics are:

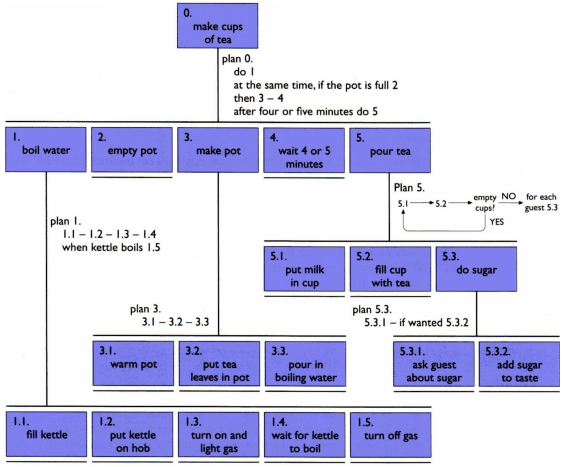
paired actions where is "turn on gas"?

restructure generate task "make pot"

balance is "pour tea" simpler than "make pot"?

generalise make one cup . . . or more

Refined HTA for making tea



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.

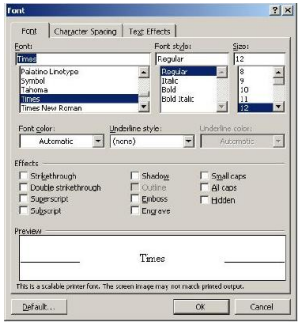
Waiting can be considered:

- a task — for “busy” waits, e.g. making tea
- part of the plan — end is the event, e.g. email reply received

Knowledge Based Analyses

- Aim to understand knowledge required for a task
  - provide training material, how-to manuals;
  - take advantage of common knowledge across tasks.
- Focus on:
  - objects used in task
  - actions performed
- Use **taxonomies**:
  - represent levels of abstraction
  - organisation (grouping) depends on purpose
- Declarative knowledge elicitation techniques:
  - established convention, existing documentation
  - asking users to list objects; card sorting
  - structured interviews, listing nouns and verbs

Laddering



1. Start subject off with a seed item: **type faces**
2. Move around task domain knowledge using prompts:
   - To move down: *Can you give examples of type faces?*
   - To move across: *What alternatives are there to type faces for changing the appearance of text?*
   - To move up: *What have Times Roman, Helvetica in common?*

Car Control Taxonomy

motor controls

steering		*steering wheel*, *indicators*
engine	direct	*ignition*, *accelerator*, *foot brake*
	gearing	*clutch*, *gear stick*
lights	external	*headlights*, *hazard lights*
	internal	*courtesy light*
wash/wipe	wipers	*front wipers*, *rear wipers*
	washers	*front washers*, *rear washers*
	front	*front wipers*, *front washers*
	rear	*rear wipers*, *rear washers*
heating		*temperature control*, *air direction*, *fan*, *rear screen heater*
parking		*hand brake*, *door lock*
radio		*numerous!*

Task Descriptive Hierarchy (TDH)

- **Task Analysis for Knowledge Description** (TAKD) uses three types of branches in TDH taxonomies:
  - **XOR** — object in exactly one branch
  - **AND** — object must be in both
  - **OR** — can be in one, many or none

wash/wipe **AND**

function **XOR**	wipers	*front wipers, rear wipers*
	washers	*front washers, rear washers*
position **XOR**	front	*front wipers, front washers*
	rear	*rear wipers, rear washers*

Larger TDH example

```
kitchen item AND
/____shape XOR
/    |____dished  mixing bowl, casserole, saucepan,
/    |            soup bowl, glass
/    |____flat    plate, chopping board, frying pan
/____function OR
     {____preparation    mixing bowl, plate, chopping board
     {____cooking    frying pan, casserole, saucepan
     {____dining XOR
          |____for food   plate, soup bowl, casserole
          |____for drink  glass
```

N.B. / | { indicates branch type; operator names AND, XOR, OR not usually used

Examining TDH taxonomies

- The **uniqueness rule**: can the diagram distinguish every object?
  - If not, may consider adding extra classifiers
- Taxonomies for actions are similar, e.g.

```
kitchen job XOR
|____ preparation  beating, mixing
|____ cooking frying, boiling, baking
|____ dining  pouring, eating, drinking
```

- Compare taxonomies to restructure/find omissions
- Objects often more easily observed than tasks!
- Notice: TDH decomposes by similarity, HTA by how-to.

Abstraction and cuts

- After producing detailed taxonomy, we can *cut* to yield an abstract view.
- e.g. cutting above shape and below dining, plate becomes:
  `kitchen item/function{preparation,dining}/`
- This is a term in the **Knowledge Representation Grammar** (KRG)
- Composite KRG term: *beating in a mixing bowl* is

```
kitchen job(preparation) using
    a kitchen item/function{preparation}/
```

- Terms and sequences in KRG may provide tools for further analysis.

Applying Task Analysis

- For documentation: **How To** manual
  - useful for novices
  - assumes all tasks known
- Requirements capture and design
  - lifts focus from system to use
  - suggests candidates for automation
  - may uncover user's conceptual model
- Detailed interface design
  - taxonomies suggest menu layout
  - object/action lists suggest interface objects
  - task frequency guides default choices
  - existing task sequences guide dialogue design

Task analysis can be continually iterated to improve and enhance.

Exercise

1. Investigate the use of **cluster analysis** to classify objects into groups.
   - Cluster analysis works by constructing a matrix of object similarities using a distance measure.
   - The distance measure may be a psychological (user-driven) estimate of relatedness determined by experiment or interview.
2. Derive a taxonomy from existing menu and dialogue layouts in a common application (e.g., word processor, image editor)
3. Perform a cluster analysis on the basic tasks and objects, and examine whether it agrees with the application layout.

References


Fabio Paternò. *Model-Based Design and Evaluation of Interactive Applications*.
Springer-Verlag, 2000.

See also:

- Dix et al, Chapter 15, and further reading recommendations there.
- More about CTT at `http://giove.cnuce.cnr.it/ConcurTaskTrees.html`