HCI Lecture 3

Task Analysis for
User Interface Analysis
& Design

Analysing, specifying and designing the task

- understand the *task domain*
- design appropriate model/metaphor
- determine the set of tasks to be performed
- determine
  - what controls to supply
  - ***how*** tasks map to controls (menu choices, etc)
  - appropriate set of ***atomic actions***

Task analysis – tools and techniques

- a range of methods and techniques used in
  - task analysis
  - task specification
  - task design & documentation
- separation of stages (analysis .. design) not always clear
- techniques applicable to both
- and to interface evaluation

Task domain – the user's point of view

- user is (usually) only interested in the task domain
- user *uses* the system to perform tasks
- user has no interest in the system *per se*
- system should be *transparent*, invisible

Achieving transparency

- system models task domain
- system metaphors drawn from task domain
- tasks, actions, vocabulary based on task domain
  - dtp terms based on printing
  - CAD terms based on engineering

Determine set of tasks

- methods for requirements capture vary
  - in-house/custom development
  - 'shrink-wrap' software
- domain experts
- users
- you – as author/designer

Modelling the task

- for analysis/design/evaluation
- validating requirements for client approval
- provide documentation/models of interface actions
- must be accessible, non-technical
  - story-boards
  - state transition diagrams
  - prototyping
- iteration

Examples

- Dix et al, Human Computer Interaction, 2nd edition, pp 208-9
- IBM designers' account of a system development process
  - 1984 Olympic Message System
- paper by Gould et al in Preece & Keller, Human-Computer Interaction

Storyboards

- pictures of interface sequences
  - like a cartoon
  - derived from film industry
- screen by screen
- effective for checking design decisions/look & feel before prototyping
- not necessarily reliable (think architectural models)

Prototyping

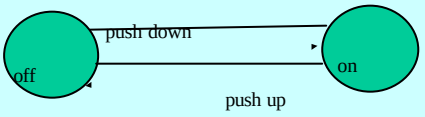
- software/mock-ups
- throw-away/evolutionary
- limited functionality
- test risk areas/ specific aspects of interface

State transition diagrams

- also known as
  - state charts
  - state diagrams
  - transition diagrams
- derived from methods for documenting real-time control & event-driven systems
- effective for documenting user interface states
- many different graphical conventions & notations

State transition diagram

- shows *states* the system can be in
- shows *transitions* between one state and the next
- and what action/event effects a transition
- state shown by **node**, transition shown by **arc**



light switch

Example

- greenhouse climate control system
- temperature sensor, adjusts heating, ventilation and watering
- 4 possible states

1. heat off, vent open, spray on
2. heat off, vent open, spray off
3. heat off, vent closed, spray off
4. heat on, vent closed, spray off

ATM – partial std

state: ready	transition: card in
request PIN	enter PIN
validating	validated
show menu	select cash
request amount	amount in
request confirm	confirm
show message	card eject
show message	card collect
present cash	cash collect

State transition diagrams for interface documentation

- nodes represent all the states the user sees
  - screens, displays, pop-ups, dialog boxes
- arcs represent the user actions which result in a change in state
- all possible user actions can be shown – each with their own arc leading to a new state
- or different routes to the same state

Advantages of state transition diagrams (1)

- for evaluation
  - assess structure/logic/consistency
- at analysis/requirements capture/validation
  - functionality
    - all functions planned for
    - sequence of operations appropriate
  - implementation independent
    - can be documented with or without implementation detail

Advantages (2)

- in design use
  - check implementation of functionality
  - check navigation
    - dead ends – there should be an exit from all states
    - error recovery
  - check sign-posting
- clearer than a verbal description

Disadvantages of state transition notation

- diagrams can quickly become complex and confusing
- may need a few attempts to get clear layout which illustrates system logic
- decompose diagrams when they become too full

Next week - some methods for task analysis

- Conceptual, semantic, syntactic, lexical
- GOMS - Goals, Operators, Methods, Selection Rules
- Task-Action grammars
- Object-Action Interface model

A note on task analysis methods

- These methods are not interchangeable equivalents
- They do different things/are good at different things
- Sometimes used in combination
- May apply both in task analysis and task design

For seminars & next week

- see schedule for reading on
  - HTA
  - GOMS
  - layered analysis (conceptual, semantic, syntactic, lexical)