

HUMAN-COMPUTER INTERACTION THIRD EDITION DIX FINLAY ABOARD BEALE

chapter 18

modelling rich interaction

HUMAN-COMPUTER INTERACTION

Modelling Rich Interaction

- status–event analysis
- rich environments in task analysis
- sensor-based systems

HUMAN-COMPUTER INTERACTION

status-event analysis

- **events** – things that happen
 - e.g. alarm bell, beeps, keystrokes
- **status** – things that are
 - e.g. screen display, watch face, mouse position
- unifying framework – system (formal analysis)
 - user (psychology & heuristics)
- time behaviour – detect delays, select feedback
- transferable phenomena
 - e.g. polling – active agent discovers status change

HUMAN-COMPUTER INTERACTION

rich set of phenomena

	events	status
input	keypress	mouse position
output	beep	display
internal	interrupt	document state
external	time	temperature

Most notations only deal with subset of these
e.g. STNs: event-in/event-out

- ⇒ may need awkward work-arounds

HUMAN-COMPUTER INTERACTION

rich set of behaviour

- actions:
 - state change at (user initiated) event
- status change events:
 - e.g. stock drops below re-order level
- interstitial behaviour:
 - between actions – e.g. dragging an icon

standard notations:

- usually,
- sometimes,
- never!

HUMAN-COMPUTER INTERACTION

Properties of events

- status change event
 - the passing of a time
- actual and perceived events
 - usually some gap
- polling
 - glance at watch face
 - status change becomes perceived event
- granularity
 - birthday – days
 - appointment – minutes

Design implications

- actual/perceived lag... matches application timescale?
- too slow
 - response to event too late e.g., power plant emergency
- too fast
 - interrupt more immediate task e.g., stock level low

Naïve psychology

- Predict where the user is looking
 - mouse – when positioning
 - insertion point – intermittently when typing
 - screen – if you're lucky
- Immediate events
 - audible bell – when in room (and hearing)
 - peripheral vision – movement or large change
- Closure
 - lose attention (inc. mouse)
 - concurrent activity

email delivery

The diagram shows a vertical timeline on the left labeled 'time'. Four vertical lines represent different system components: mailbox file status, mailtool agent, screen status, and user agent.

- 'mail arrives event' occurs on the mailbox file status line.
- 'mailtool notices event' occurs on the mailtool agent line.
- 'change icon event' occurs on the screen status line.
- 'user notices event' occurs on the user agent line.

 Arrows indicate the flow of information and events between these components.

email delivery (ctd)

This diagram is similar to the previous one but includes a 'mail arrives event' on the mailbox file status line and a 'mailtool notices event' on the mailtool agent line. It also shows a 'change icon event' on the screen status line and a 'user notices event' on the user agent line.

- Perceived event in minutes – not guaranteed
- alternative timescale
 - explicit examination – hours/days
 - audible bell – seconds
- but want minutes – guaranteed

screen button widget

screen button often missed, ... but, error not noticed

a common widget, a common error: Why?

Closure
mistake likely – concurrent action
not noticed – semantic feedback missed

Solution
widget feedback for application event
a perceived event for the user

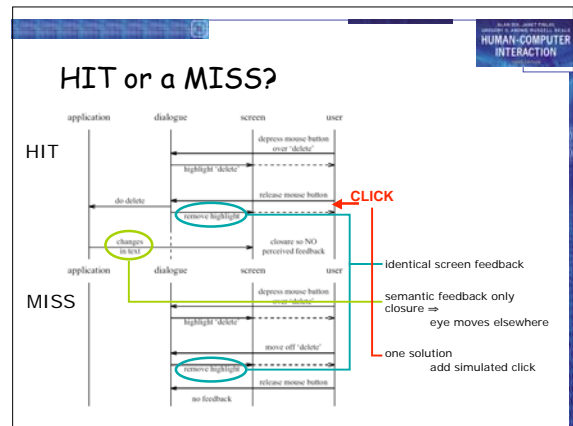
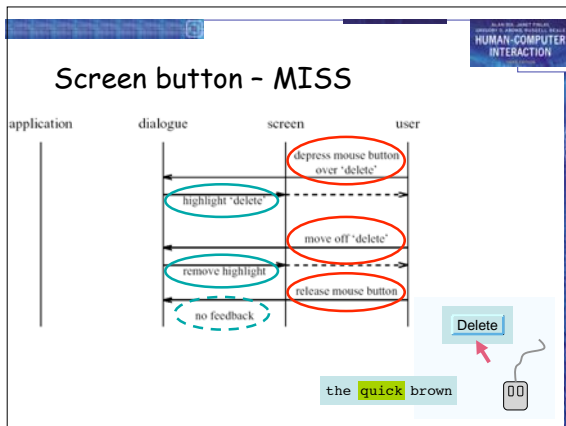
N.B. an expert slip – testing doesn't help

Screen-button - HIT

The diagram shows a sequence of events between four entities: application, dialogue, screen, and user.

- The user performs a 'depress mouse button over 'delete'' action.
- The screen highlights 'delete'.
- The dialogue highlights 'delete'.
- The application performs 'do delete'.
- The dialogue removes the highlight.
- The screen removes the highlight.
- The user releases the mouse button.
- The application changes the text in the dialogue.
- The screen shows 'closure so NO perceived feedback'.

 A screenshot at the bottom shows the text 'the | brown fox' with a 'Delete' button highlighted over the word 'brown'.



rich contexts

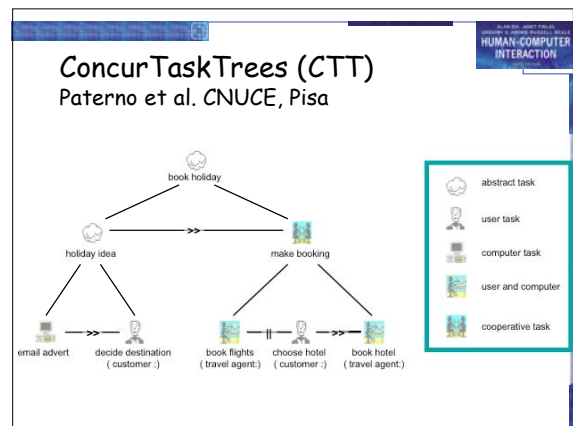
the problem

- task models
 - formal description
- situatedness
 - unique contexts
- ethnography
 - rich ecologies

bringing them together?

collaboration

- already in several notations
 - e.g. CTT, GTA
- add artefacts too?



Groupware Task Analysis

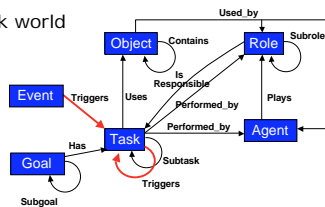
GTA

- conceptual framework, tools, elicitation techniques

rich model of task world

rich ontology

- human roles for collaboration
- physical and electronic objects



information

pre-planned cognitive model
goal → action

situated action
environment → action

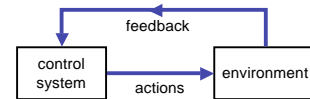
control

- open loop control
 - no feedback
 - fragile

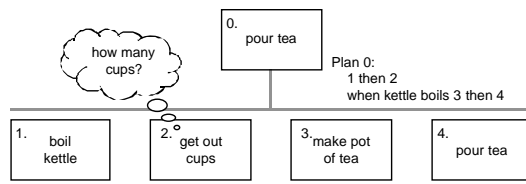


control

- open loop control
 - no feedback
 - fragile
- closed loop control
 - uses feedback
 - robust



adding information



adding information (ctd)

information required when

- subtask involves input (or output)
- some kind of choice (how to know what to do)
- subtask repeated (but iterations unspecified)

sources of information

- part of existing task (e.g. phone number entered)
- user remembers it (e.g. recall number after directory enquiry)
- on device display (e.g. PDA address book, then dial)
- in the environment
 - pre-existing (e.g. phone directory)
 - created in task (e.g. write number down on paper)

GUI easy (lots of space) mobile/PDA need to think

triggers

process – what happens and order

```

    graph LR
      A[get post from pigeon hole] --> B[bring post to desk]
      B --> C[open post]
  
```

triggers

process – what happens and order
triggers – when and why

```

    graph LR
      T1[first thing in the morning] --- A[get post from pigeon hole]
      A --> B[bring post to desk]
      T2[holding post] --- B
      B --> C[open post]
      T3[at coffee time] --- C
  
```

common triggers

- immediate
 - straight after previous task
- temporal
 - at a particular time
- sporadic
 - when someone thinks of it!
- external event
 - when something happens, e.g. phone call
- environmental cue
 - something prompts action ... artefacts

artefacts

- ethnographic studies
- as shared representation
- as focus of activity
- act as triggers, information sources, etc.

9.37	BTN	180	BRITANNIA BAL770 5423	300	CREW 9.25
			M/B737/C T420	EGGW 1A2 1B3 1B4 EGAA	

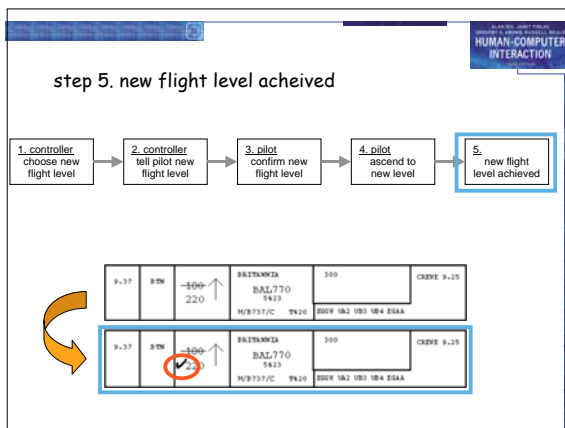
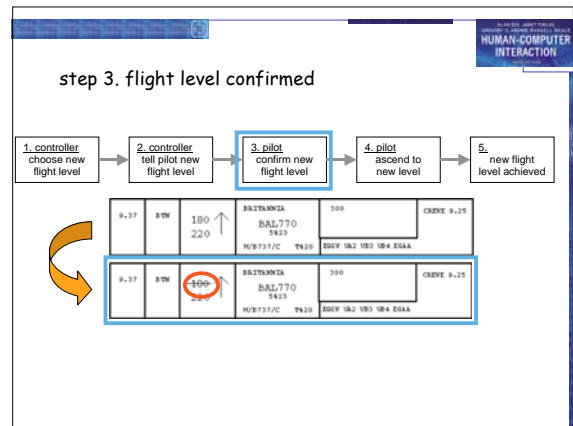
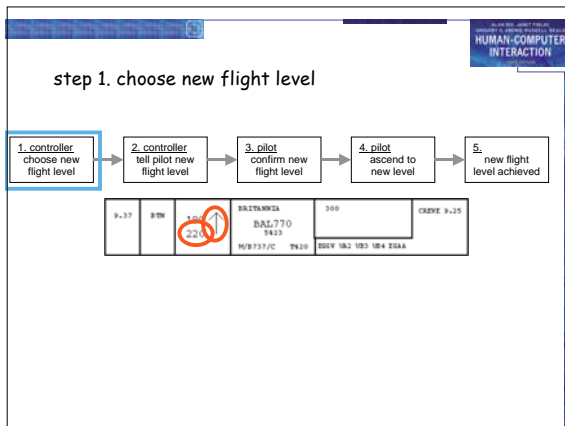
placeholders

- knowing where you are in a process
 - like a program counter
- coding:
 - memory
 - explicit (e.g. to do list)
 - in artefacts

where are you?

```

    graph LR
      1[1. controller choose new flight level] --> 2[2. controller tell pilot new flight level]
      2 --> 3[3. pilot confirm new flight level]
      3 --> 4[4. pilot ascend to new level]
      4 --> 5[5. new flight level achieved]
  
```



tracing placeholders

a form of information, may be ...

- in people's heads
 - remembering what to do next
- explicitly in the environment
 - to-do lists, planning charts, flight strips, workflow
- implicitly in the environment
 - location and disposition of artefacts

electronic environments ...

- fewer affordances for artefacts
- danger for careless design!

papers tidy or skewed
letter open or closed

low intention and sensor-based interaction

car courtesy lights


- turn on
 - when doors unlocked/open
- turned off
 - after time period
 - when engine turned on

driver's *purpose* is to get into the car
incidentally the lights come on

HUMAN-COMPUTER INTERACTION

Pepys

- Xerox Cambridge (RIP)
- active badges
- automatic diaries



Allan's *purpose* to visit Paul's office
incidentally diary entry created

HUMAN-COMPUTER INTERACTION

MediaCup

- cup has sensors
 - heat, movement, pressure
- broadcasts state (IR)
- used for awareness
 - user is moving, drinking, ...




Han's *purpose* to drink coffee
incidentally colleagues are aware

HUMAN-COMPUTER INTERACTION

shopping cart

- goods in shopping cart analysed
 - e.g. Amazon books
- used to build knowledge about books
 - people who like X also like Y
- used to give you suggestions
 - "you might like to look at ...", "special offer ..."




my *purpose* to buy a book
incidentally shown related titles

HUMAN-COMPUTER INTERACTION

onCue

- 'intelligent' toolbar
 - appropriate intelligence
 - make it good when it works
 - don't make it hard if it doesn't
- analyses clipboard contents
- suggests things to do with it



user's *purpose* to copy text elsewhere
incidentally alternative things to do

HUMAN-COMPUTER INTERACTION

the intentional spectrum

intentional ↑ *press* light switch

expected ↓ walk into room *expecting* lights to switch on

incidental ↓ walk into room ... *unknown to you*
... air conditioning increases

HUMAN-COMPUTER INTERACTION

fluidity

intentional

expected

incidental

co-option
users explicitly use behaviour
e.g. open door for lights

comprehension
users notice, form model
then rely on behaviour

interaction models

- intentional cycle
 - Norman execution/evaluation loop
- some exceptions
 - multiple goals, displays, opportunistic
- guidelines
 - feedback, transparency

cognition

- physical things (inanimate)
 - directness of effect
 - locality of effect
 - visibility of state
- computational things (also animate)
 - complex effects
 - non locality of effect
 - distance – networks; time – delays, memory
 - large hidden state

cognition

- understanding
 - innate intelligences
 - physical, natural/animal, social, physiological
 - rational thought
 - imagination
- interfaces
 - GUI, VR, AR, tangible
 - recruit physical/tangible intelligence
 - ubicomp, ambient, incidental
 - ???

homunculi, haunted houses

designing incidental interaction

- need richer representations
 - of the world, of devices, of artefacts
 - wider ecological concerns
- two tasks
 - purposeful task – for interpretation
 - supported task – for actions

issues and process

- no accepted methods but ... general pattern
- uncertainty
 - traditional system due to errors
 - sensor-based intrinsic to design
 - uncertain readings, uncertain inference
 - usually control non-critical aspects of environment
- process ... identify
 - input – what is going to be sensed
 - output – what is going to be controlled
 - scenarios – desired output and available input

designing a car courtesy light

<ul style="list-style-type: none"> available input <ul style="list-style-type: none"> –door open, car engine desired output <ul style="list-style-type: none"> –light! identify scenario <ul style="list-style-type: none"> 0 don't care +, ++, ... want light –, ––, ... don't want it legal requirements <ul style="list-style-type: none"> light off whilst driving safety <ul style="list-style-type: none"> approaching car?? 	<ol style="list-style-type: none"> deactivate alarm 0 walk up to car 0 key in door – open door & take key + get in ++ close door 0 adjust seat + find road map ++ look up route +++ find right key + key in ignition – start car 0 seat belt light flashes 0 fasten seat belt + drive off –
---	--

safe? light advertises presence

illegal to drive with interior light on

implementation

- sensors not used for original purpose
 - open architectures, self-discovering, self-configuring
- privacy
 - internet—enables kettle broadcasts to the world!
- context
 - inferring activity from sensor readings – status not event
- data filtering and fusion
 - using several sensors to build context
- inference
 - hand-coded or machine-learning
- must be used
 - control something (lights) or modify user actions (TV on)

architectures for sensor-based systems?

