formal methods in HCI

a success story

problem

- context
  - mid 80s
  - local authority DP dept
- transaction processing
  - vast numbers of users
  - order processing, pos systems etc.
  - COBOL!
- existing programs ... didn’t work
TP physical architecture

what happens

user edits form
message goes to TP engine
passed to application module
which processes the message
and prepares new screen
which is sent to the user
....
structure of programs

why?

program is trying to work out what is happening!

- standard algorithm
  - program counter implicit
- TP, web, event-based GUI
  - need explicit dialogue state
mixed up state

- many users – one application module
  
  user A starts multi-screen search list
  application stores value ‘next_record’
  user B starts multi-screen search list
  application overwrites value ‘next_record’
  user A selects ‘next screen’ ...
  ... and gets next screen of B’s search!

state is hard

- recent MSc course
  - CS and psych
  - exercise – state of 4 function calculator
    - difficult for both

- why?
  - in real life state is implicit
  - in standard CS state is implicit too!
solution?

- flowchart!
- not of program ... but of dialogue
- a formal dialogue specification!

```
Delete  D1
Please enter employee no.: ___

 Delete  D2
Name: Alan Dix
Dept: Computing
delete? (Y/N):__
Please enter Y or N

Delete  D3
Name: Alan Dix
Dept: Computing
Delete? (Y/N):__
Please enter Y or N

Delete  D4
Name: Alan Dix
Dept: Computing
Delete? (Y/N):__
Please enter Y or N

Finish
```
details ...

- miniature screen sketch

- minimal internal details
and then …

- hand transformation to boiler plate code
- store ‘where next’ for each terminal
  - in ‘session’ data
- code starts with big ‘case’
- do processing
- set new ‘where next’ … send screen

lessons

- useful – addresses a real problem!
- communication – mini-pictures and clear flow easy to talk through with client
- complementary – different paradigm than implementation
- fast pay back – quicker to produce application (at least 1000%)
- responsive – rapid turnaround of changes
- reliability – clear boiler plate code less error-prone
- quality – easy to establish test cycle
- maintenance – easy to relate bug/enhancement reports to specification and code
states in web applications (inter alia)

• persistent (deep) state
  – the things you are really interested in
    e.g. an order, images in an album, facebook profile

• temporary interaction state
  – part-created or part-edited objects
    e.g. new cracker setting decoration, joke, etc.

• placeholder in interaction
  – current state/mode/location in dialogue

pretty clear in database or similar

less clear and often confused

where state is stored

(a) cookies
(b) hidden form variables
  – `<input type="hidden" name="sid" value="A378F9E6B2" />`
(c) encoded in URL
(d) session variables
  – session/transaction id stored using (a), (b), or (c)
(e) database or persistent store
  – as (d) needs identifier or token
what goes where?

interaction issues:
- multiple windows on same app?
- privacy and security
- 'back' button? ... and bookmarks
danger of 'resubmitting' same data

implementation issues
- 'garbage collecting' session data
- restarting server

often 'hacked' not considered

lesson – clearly specify the state
... and only then map into implementation!

dialogue too ... the way it often is ...

initial prototype linear
A then B then ...

at most one path
for each link/button

user clicks
link

code to
produce
page A

user clicks
link

code for
page B
update
and fetch
data

DB
... and the way it goes ...

initial prototype linear

but ...

later add validation

if validation fails

page B code needs

to produce something
‘like’ page A

... and it just keeps

getting worse

separate dialogue and presentation

model dialogue

– take into account both
user and system branches

three kinds of code

– gather data for page
access database
but no updates except logging

– format page
only code to produce HTML
maybe use template engine

– post-page process actions
validation checks
update persistent data
choose next page to show

view / presentation
dialogue & semantics