

## chapter 16

# dialogue notations and design

## Dialogue Notations and Design

- Dialogue Notations
  - Diagrammatic
    - state transition networks, JSD diagrams, flow charts
  - Textual
    - formal grammars, production rules, CSP
- Dialogue linked to
  - the semantics of the system – what it does
  - the presentation of the system – how it looks
- Formal descriptions can be analysed
  - for inconsistent actions
  - for difficult to reverse actions
  - for missing actions
  - for potential miskeying errors

## what is dialogue?

- conversation between two or more parties
  - usually cooperative
- in user interfaces
  - refers to the *structure* of the interaction
  - syntactic level of human-computer 'conversation'
- levels
  - lexical – shape of icons, actual keys pressed
  - syntactic – order of inputs and outputs
  - semantic – effect on internal application/data

## structured human dialogue

- human-computer dialogue very constrained
- some human-human dialogue formal too ...

Minister: do you *man's name* take this woman ...  
Man: I do  
Minister: do you *woman's name* take this man ...  
Woman: I do  
Man: With this ring I thee wed  
(*places ring on womans finger*)  
Woman: With this ring I thee wed (*places ring ..*)  
Minister: I now pronounce you man and wife

## lessons about dialogue

- wedding service
  - sort of script for three parties
  - specifies order
  - some contributions fixed - "I do"
  - others variable - "do you *man's name* ..."
  - instructions for ring concurrent with saying words "with this ring ..."
- if you say these words are you married?
  - only if in the right place, with marriage licence
  - syntax not semantics

## ... and more

- what if woman says "I don't"?
- real dialogues often have alternatives:

Judge: How do you plead guilty or not guilty?  
Defendant: *either* Guilty or Not guilty

  - the process of the trial depends on the defendants response
- focus on normative responses
  - doesn't cope with judge saying "off with her head"
  - or in computer dialogue user standing on keyboard!

## dialogue design notations

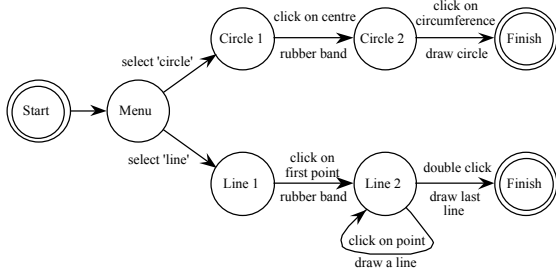
- dialogue gets buried in the program
- in a big system can we:
  - analyse the dialogue:
    - can the user always get to see current shopping basket
  - change platforms (e.g. Windows/Mac)
  - dialogue notations helps us to
    - analyse systems
    - separate lexical from semantic
- ... and before the system is built
  - notations help us understand proposed designs

## graphical notations

state-transition nets (STN)  
Petri nets, state charts  
flow charts, JSD diagrams

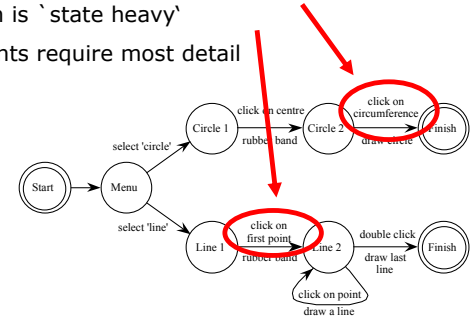
## State transition networks (STN)

- circles - states
- arcs - actions/events



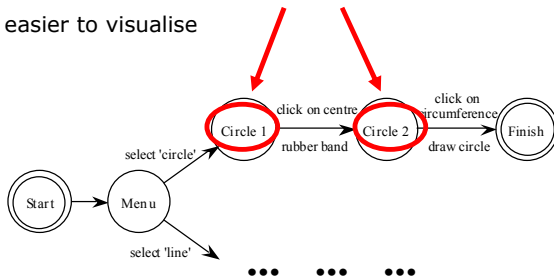
## State transition networks - events

- arc labels a bit cramped because:
  - notation is `state heavy`
  - the events require most detail



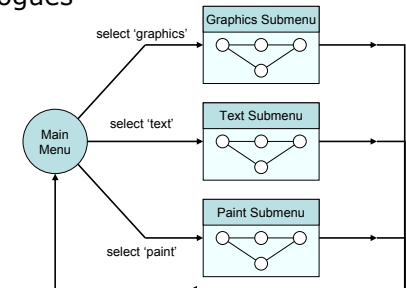
## State transition networks - states

- labels in circles a bit uninformative:
  - states are hard to name
  - but easier to visualise



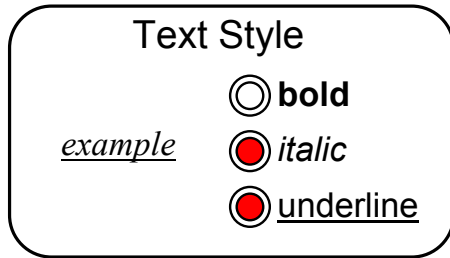
## Hierarchical STNs

- managing complex dialogues
- named sub-dialogues



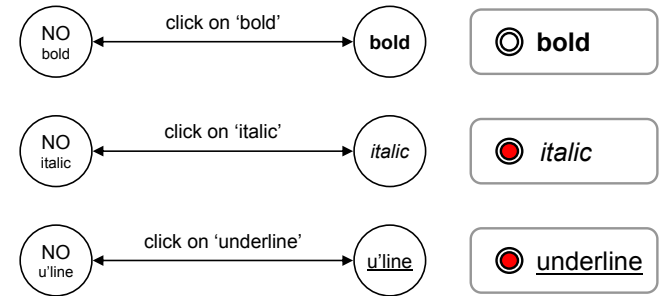
## Concurrent dialogues - I

### simple dialogue box



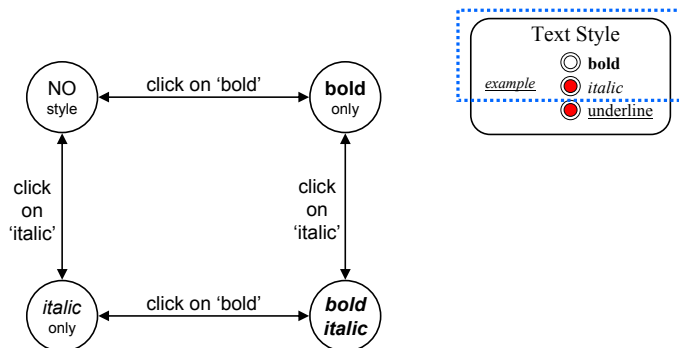
## Concurrent dialogues - II

### three toggles - individual STNs



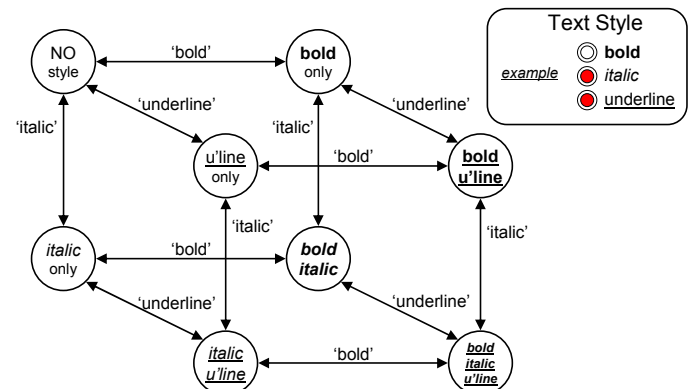
## Concurrent dialogues - III

### bold and italic combined



## Concurrent dialogues - IV

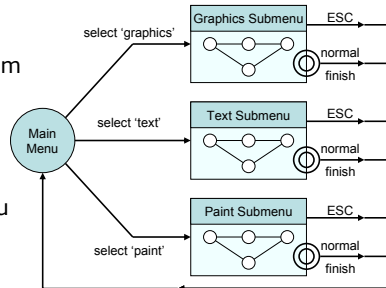
### all together - combinatorial explosion



## escapes

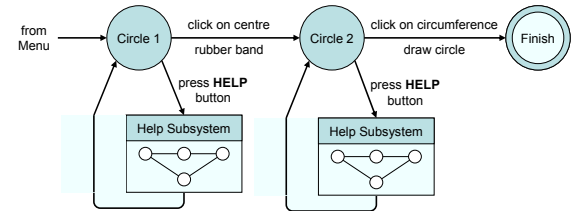
- 'back' in web, escape/cancel keys
  - similar behaviour everywhere
  - end up with spaghetti of identical behaviours
- try to avoid this

e.g. on high level diagram  
'normal' exit for each submenu  
plus separate escape arc active 'everywhere' in submenu



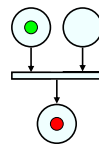
## help menus

- similar problems
  - nearly the same everywhere
  - but return to same point in dialogue
  - could specify on STN ... but very messy
  - usually best added at a 'meta' level

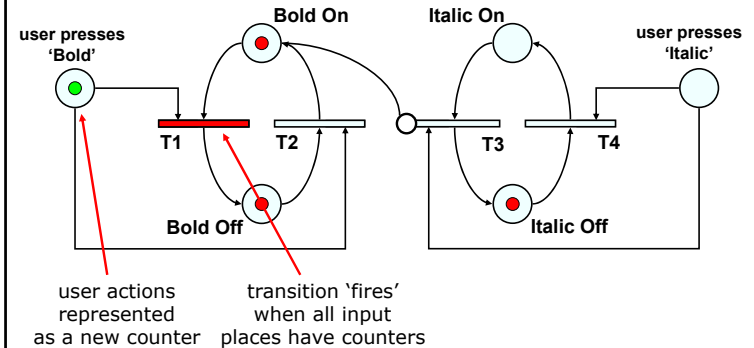


## Petri nets

- one of the oldest notations in computing!
- flow graph:
  - places - a bit like STN states
  - transitions - a bit like STN arcs
  - counters - sit on places (current state)
- several counters allowed
  - concurrent dialogue states
- used for UI specification (ICO at Toulouse)
  - tool support - Petshop

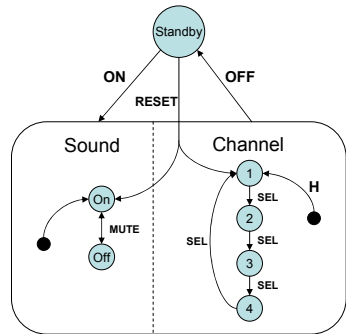


## Petri net example



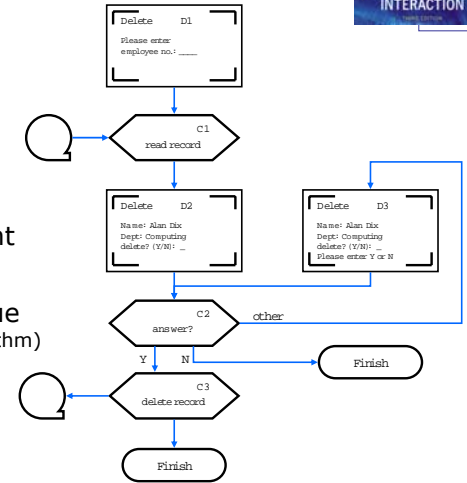
## State charts

- used in UML
- extension to STN
  - hierarchy
  - concurrent sub-nets
  - escapes
  - OFF always active
- history
  - link marked H goes back to last state on re-entering subdialogue



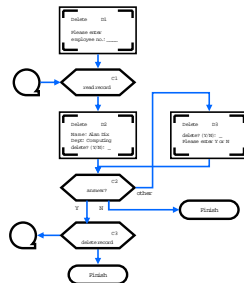
## Flowcharts

- familiar to programmers
- boxes
  - process/event
  - not state
- use for dialogue (not internal algorithm)



## it works!

- formal notations – too much work?
- COBOL transaction processing
  - event-driven – like web interfaces
  - programs structure ≠ dialogue structure
- used dialogue flow charts
  - discuss with clients
  - transform to code
  - systematic testing
  - 1000% productivity gain
- formalism saves time!!



## JSD diagrams

- for tree structured dialogues
  - less expressive
  - greater clarity

