

Interacção Humana com o Computador

Aula I I



Departamento de Informática
UBI 2024/2025

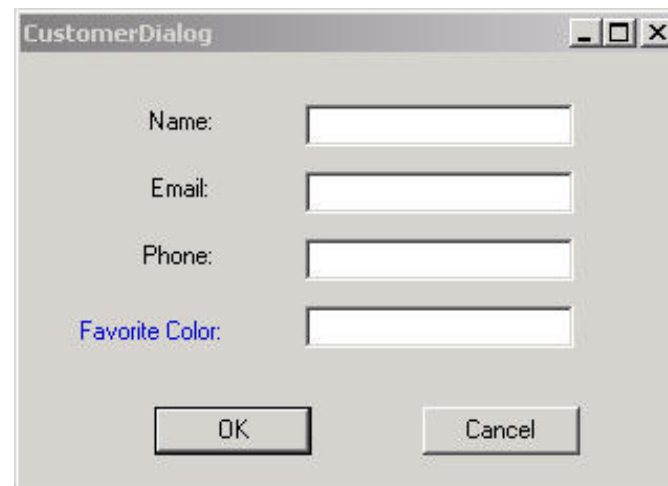
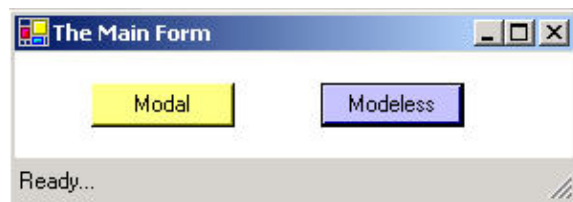
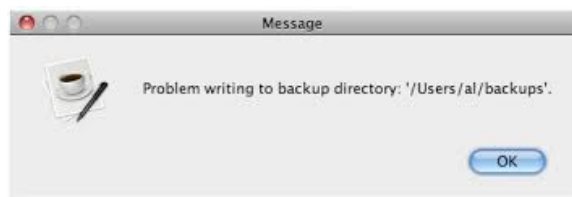
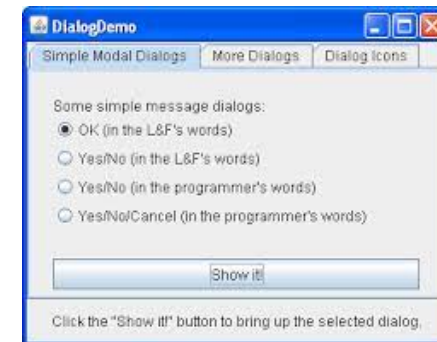
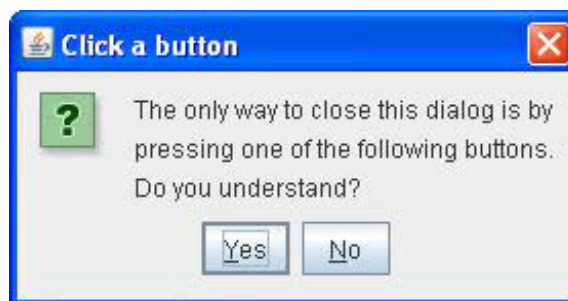
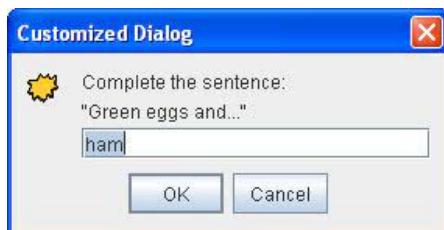
João Cordeiro
jpcc@ubi.pt



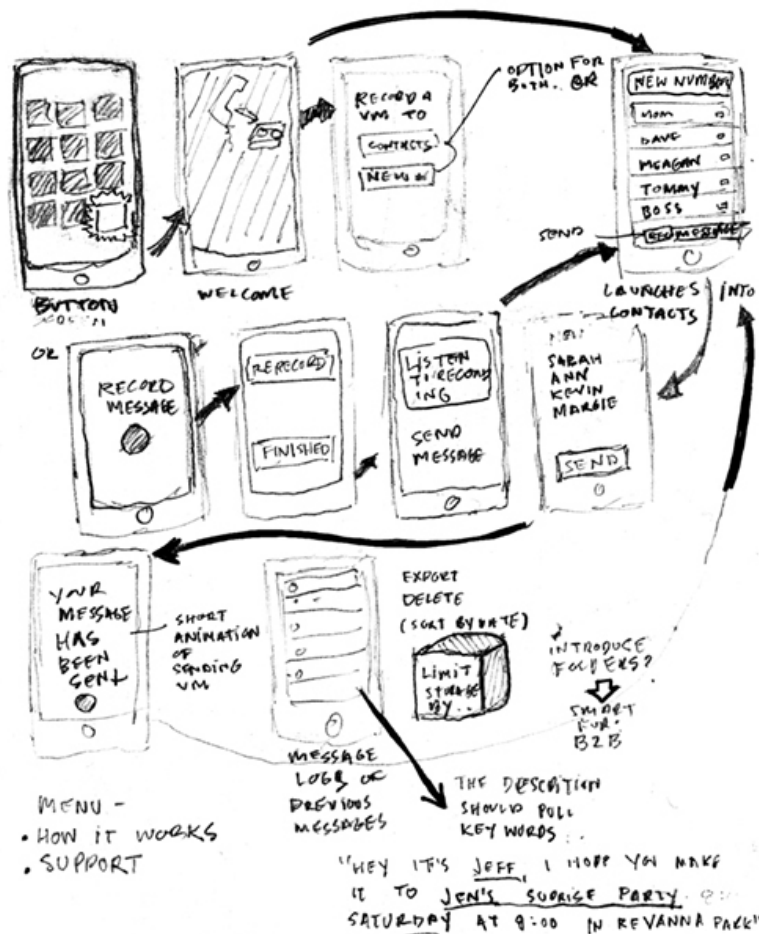
Chapter 16

Dialogue Notations and Design

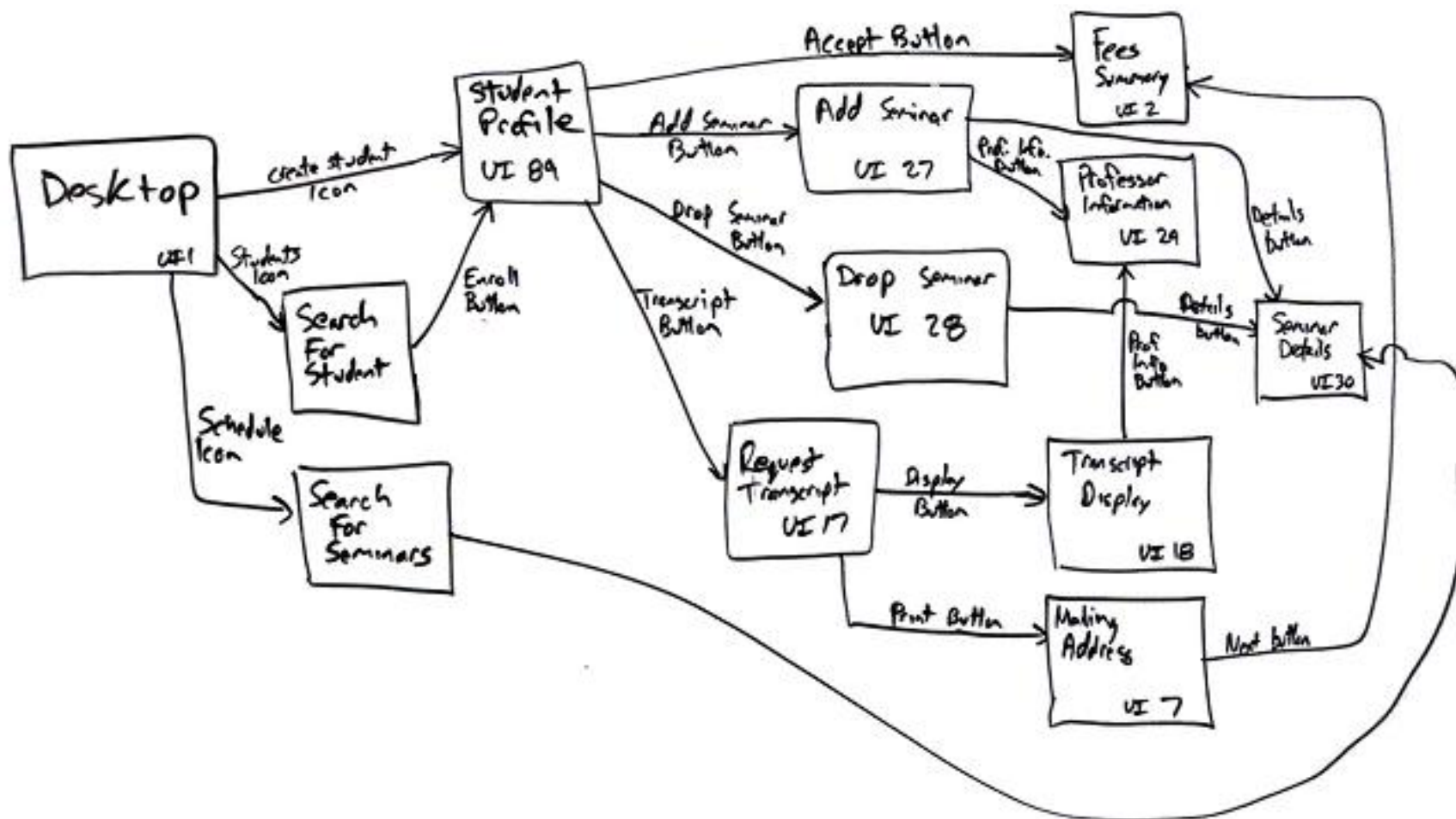
More than just dialog boxes



More than model navigation flow



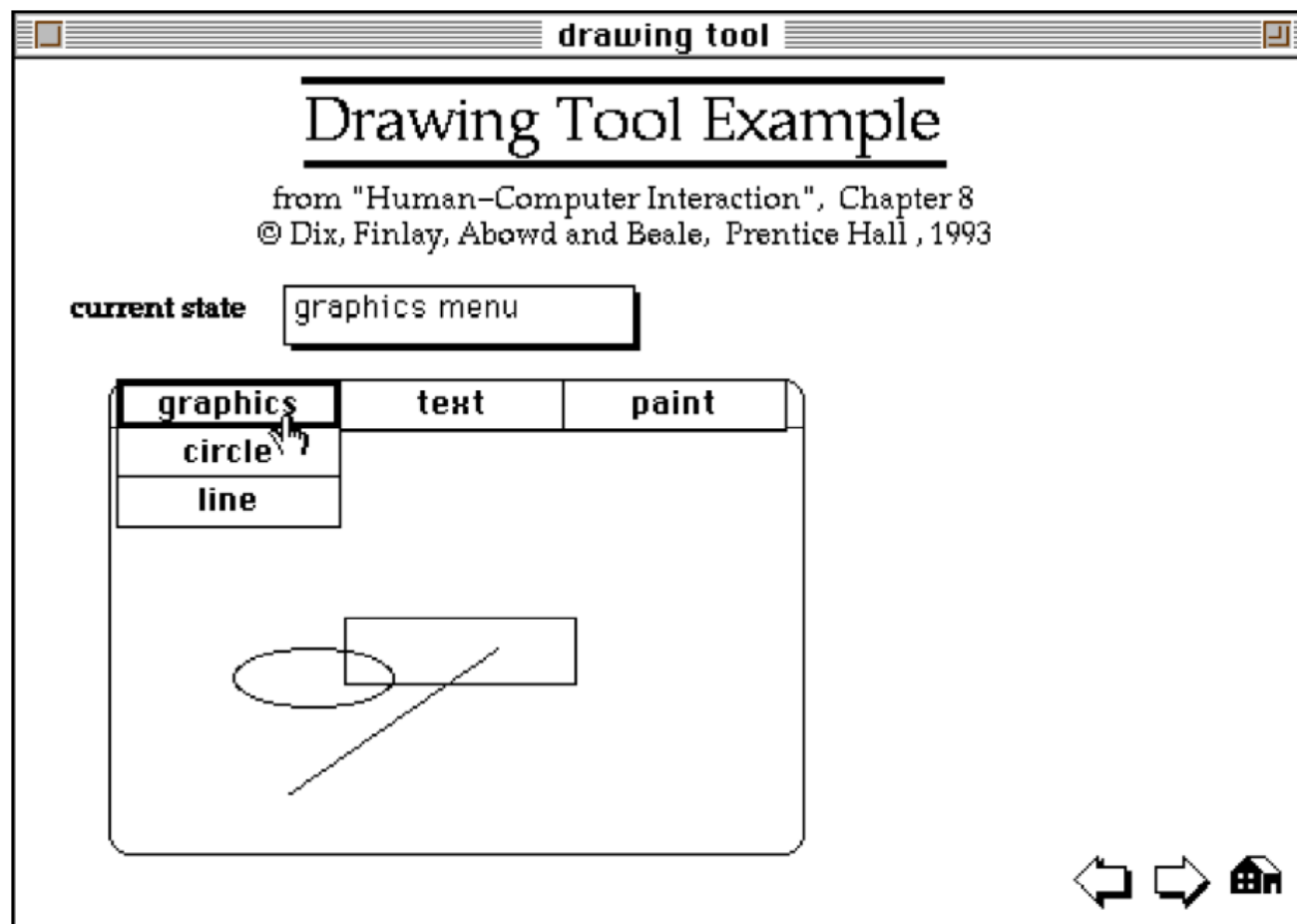
More than model navigation flow



Graphical Notations

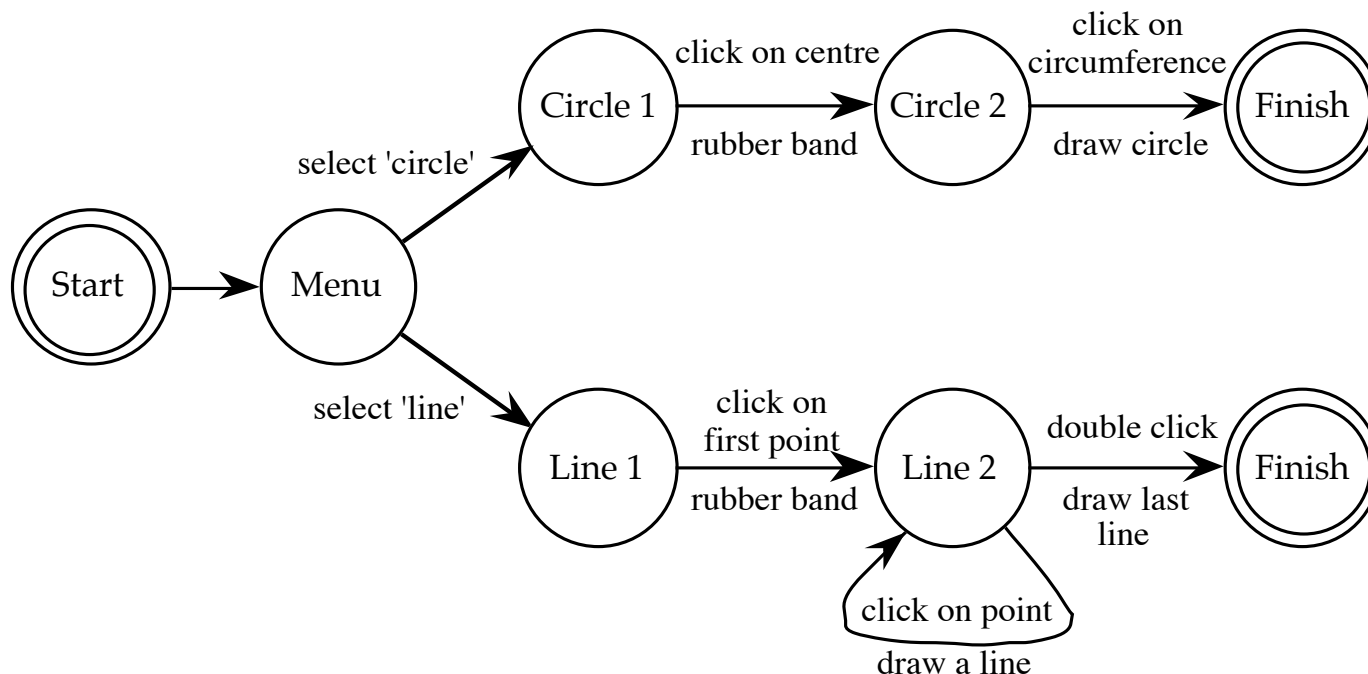
State-Transition Nets (STN)
Petri Nets, State Charts
Flow Charts, JSD diagrams

State Transition Networks (STN)

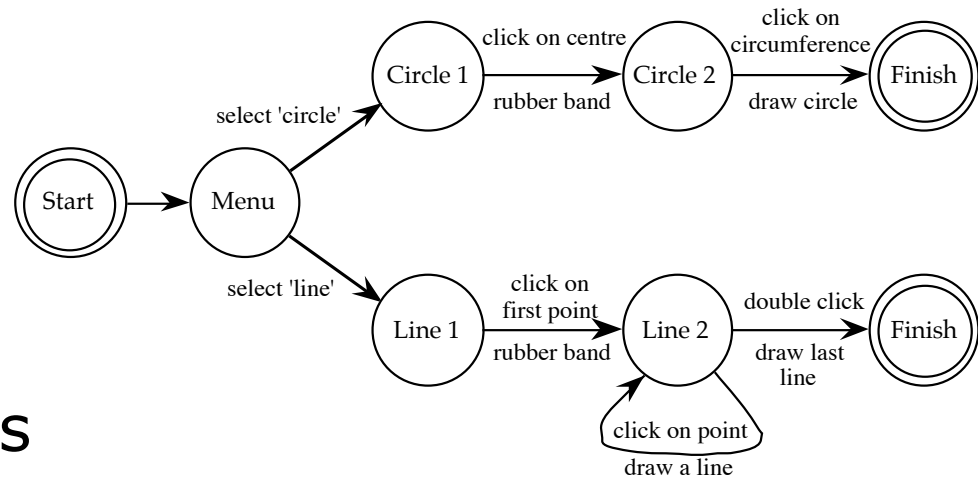


State Transition Networks (STN)

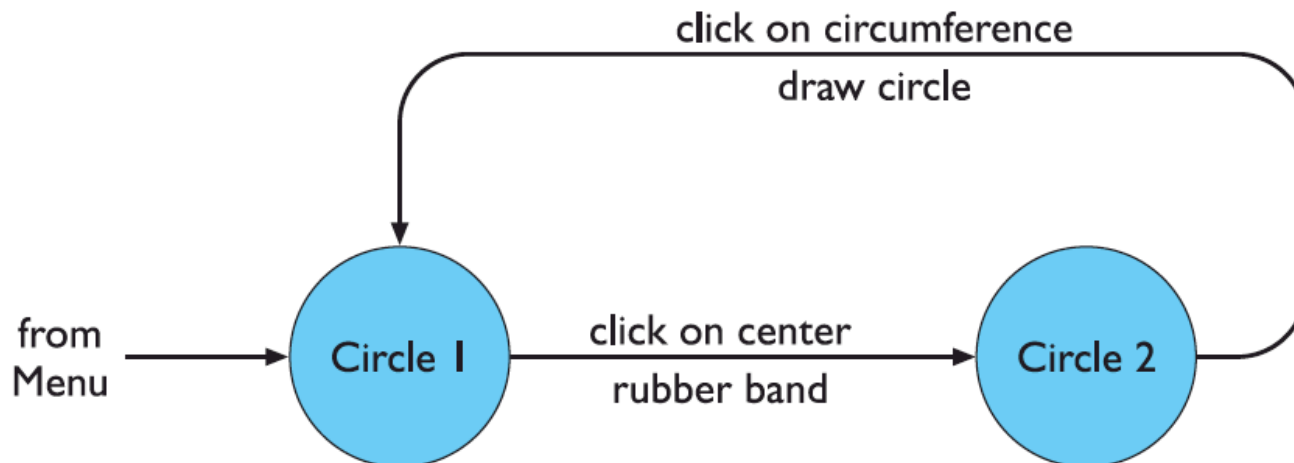
- circles - states
- arcs - actions/events



State Transition Networks (STN)

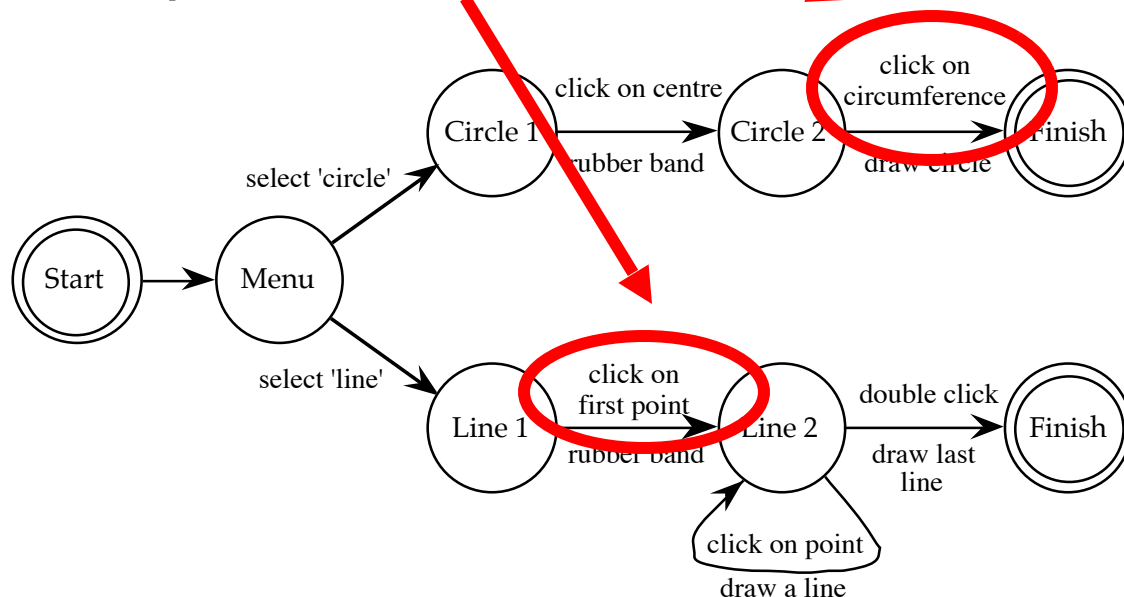


Draw several circles



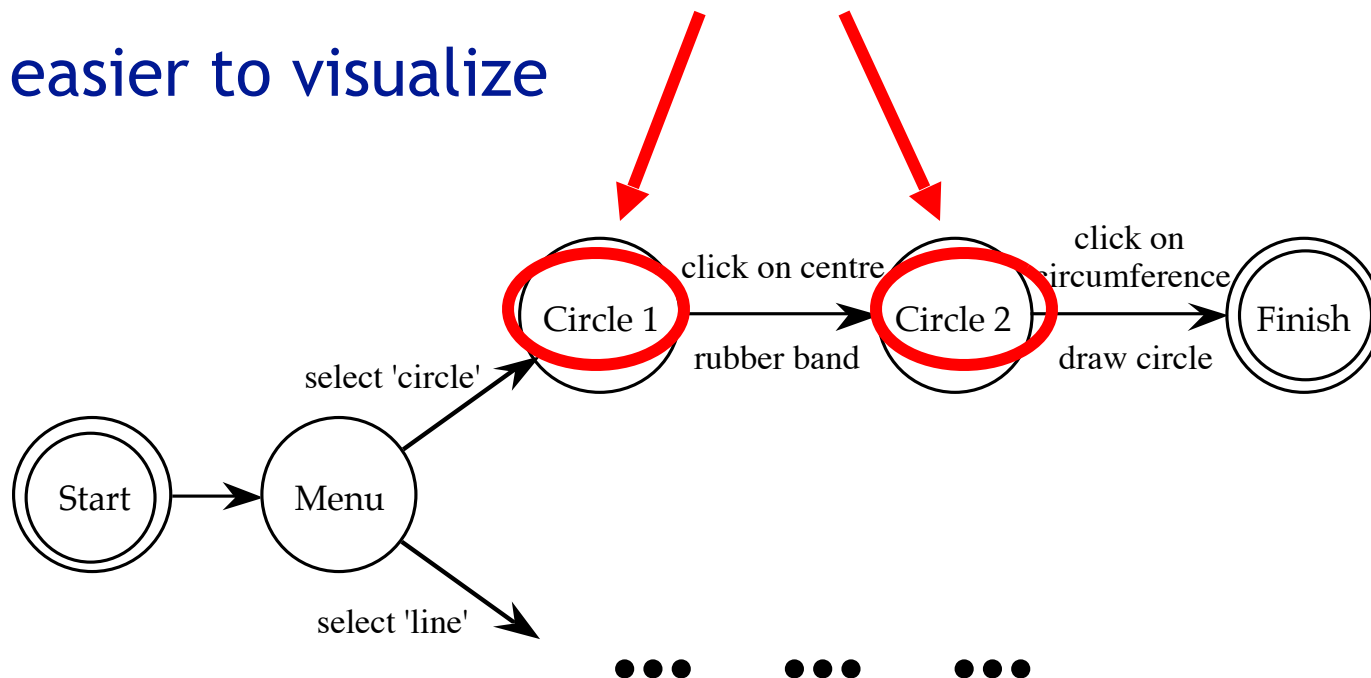
State Transition Networks - events

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



State Transition Networks - states

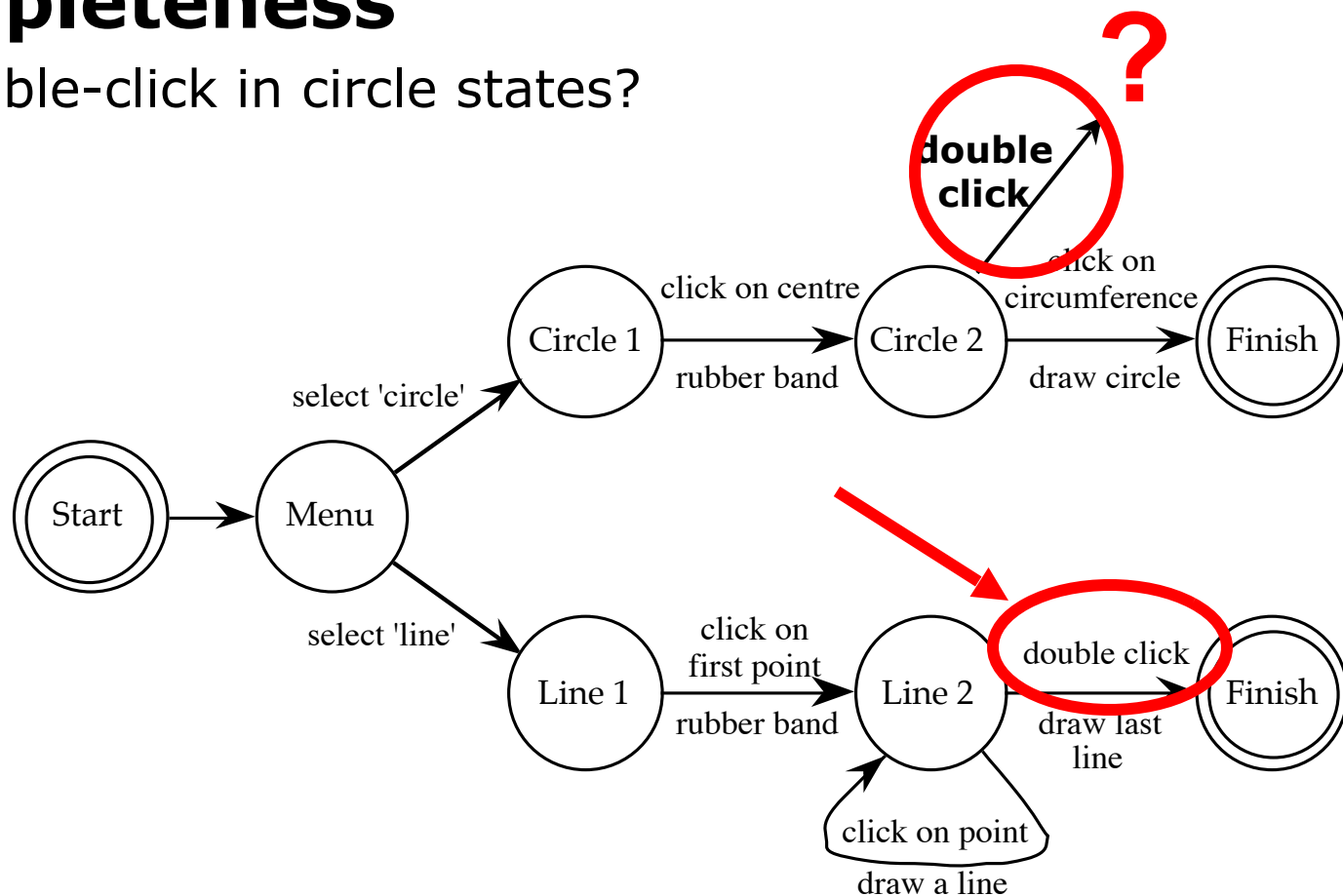
- labels in circles a bit uninformative:
 - states are **hard to name**
 - but **easier to visualize**



Checking properties (i)

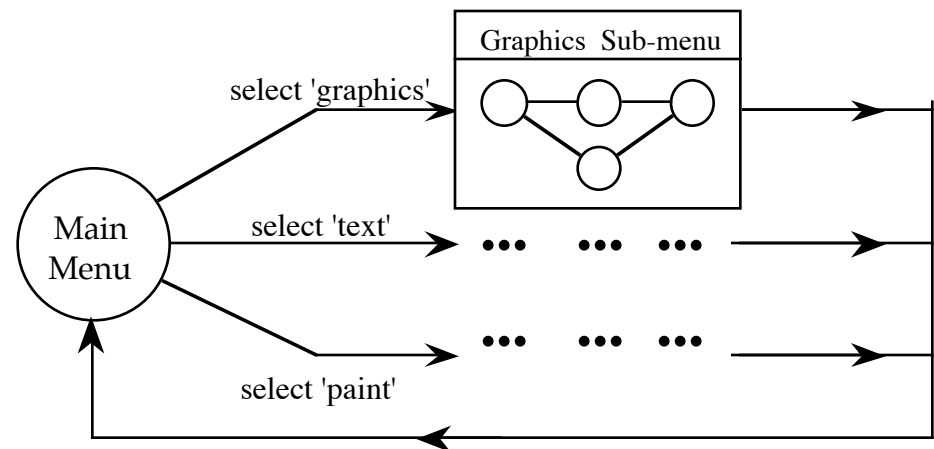
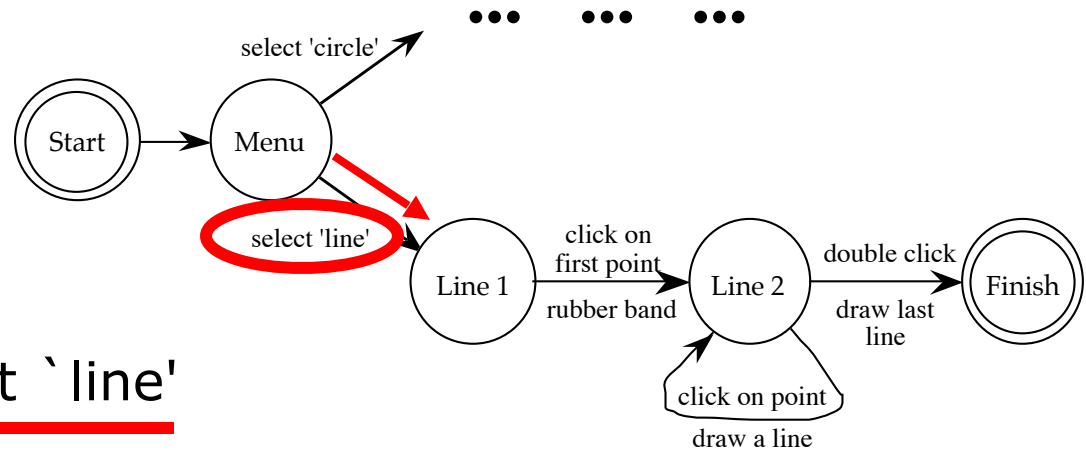
- **Completeness**

- double-click in circle states?



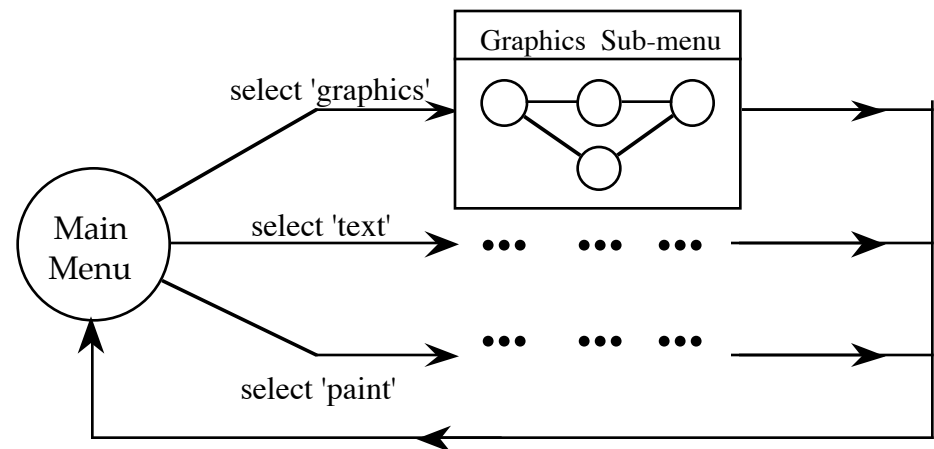
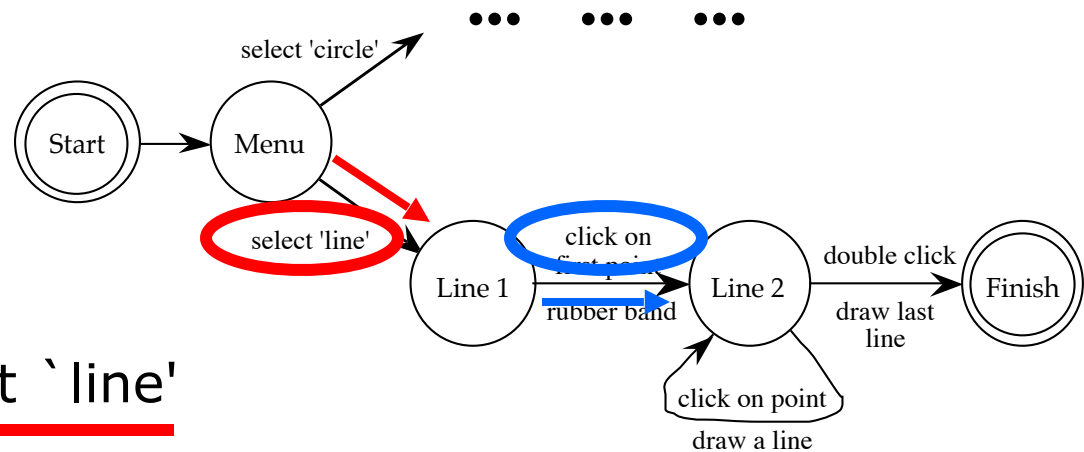
Checking properties (ii)

- **Reversibility:**
 - to reverse select 'line'



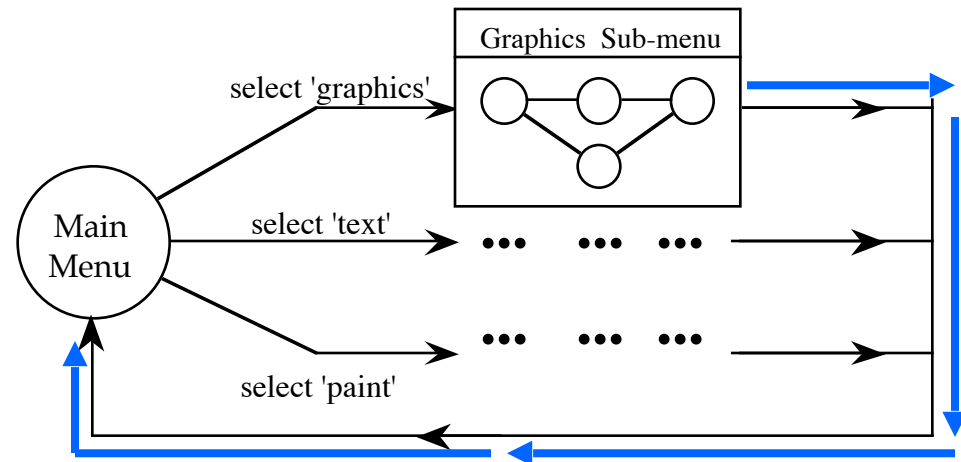
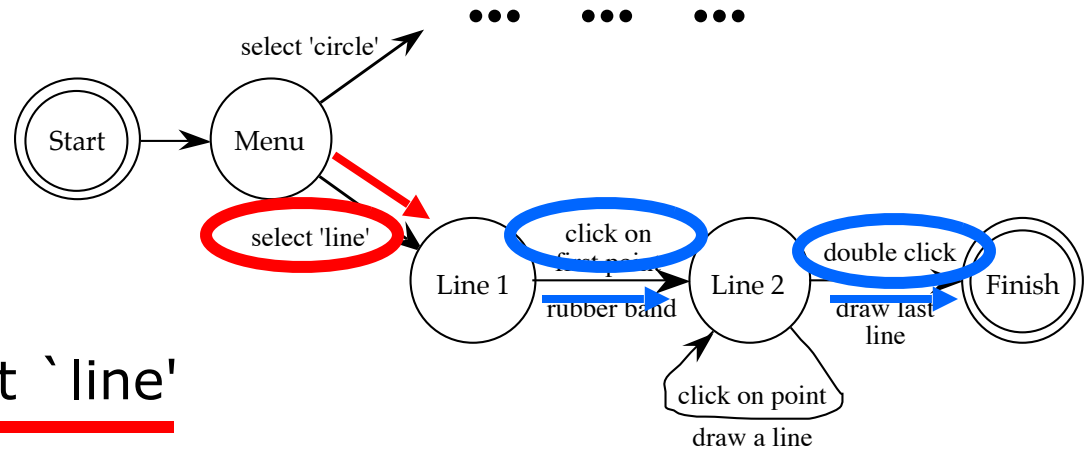
Checking properties (ii)

- Reversibility:
 - to reverse select 'line'
 - click



Checking properties (ii)

- Reversibility:
 - to reverse select 'line'
 - click - double click

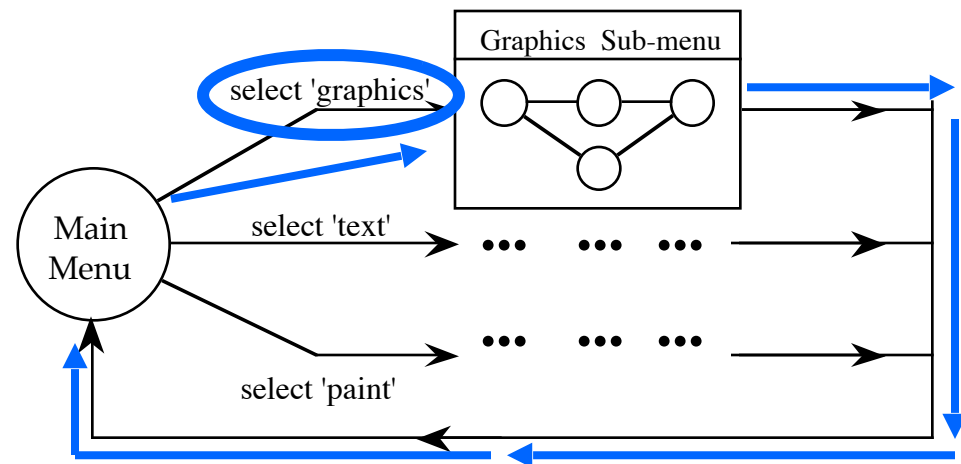
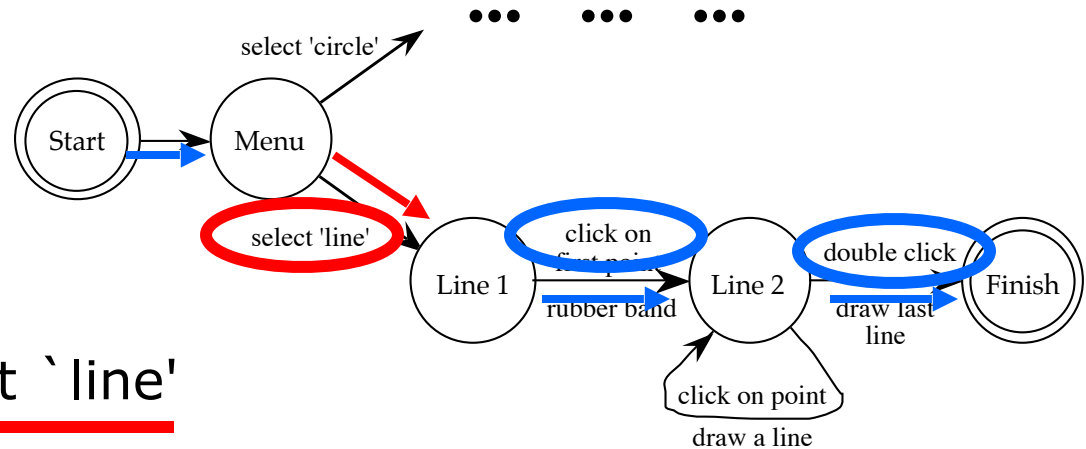


Checking properties (ii)

- Reversibility:

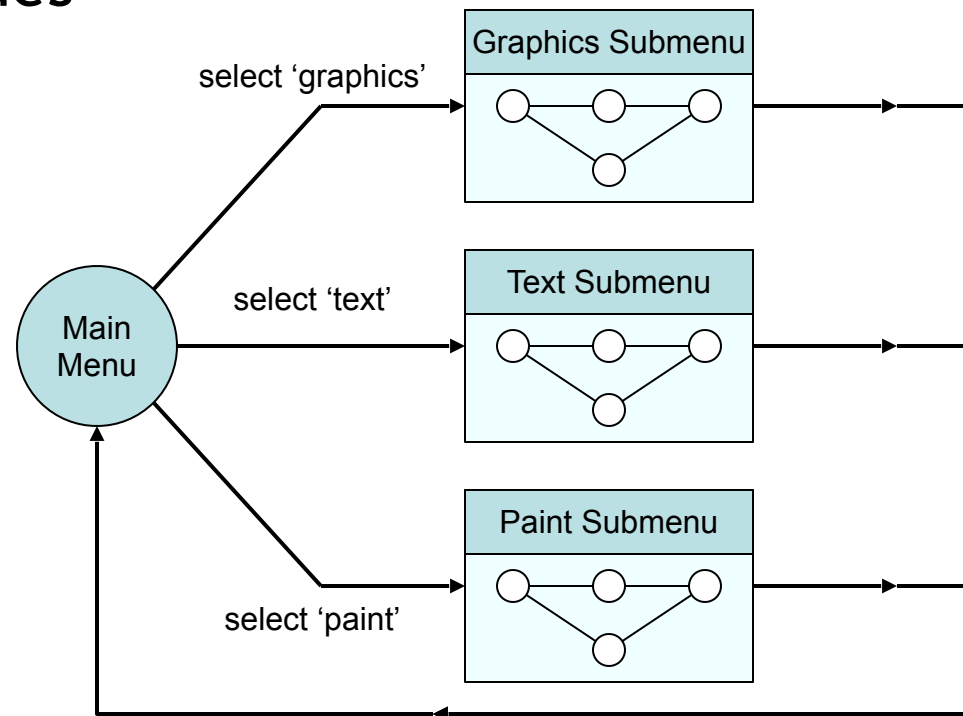
- to reverse select 'line'
- click - double click - select 'graphics'
- (3 actions)

- N.B. not undo



Hierarchical STNs

- Managing complex dialogues
- Named sub-dialogues



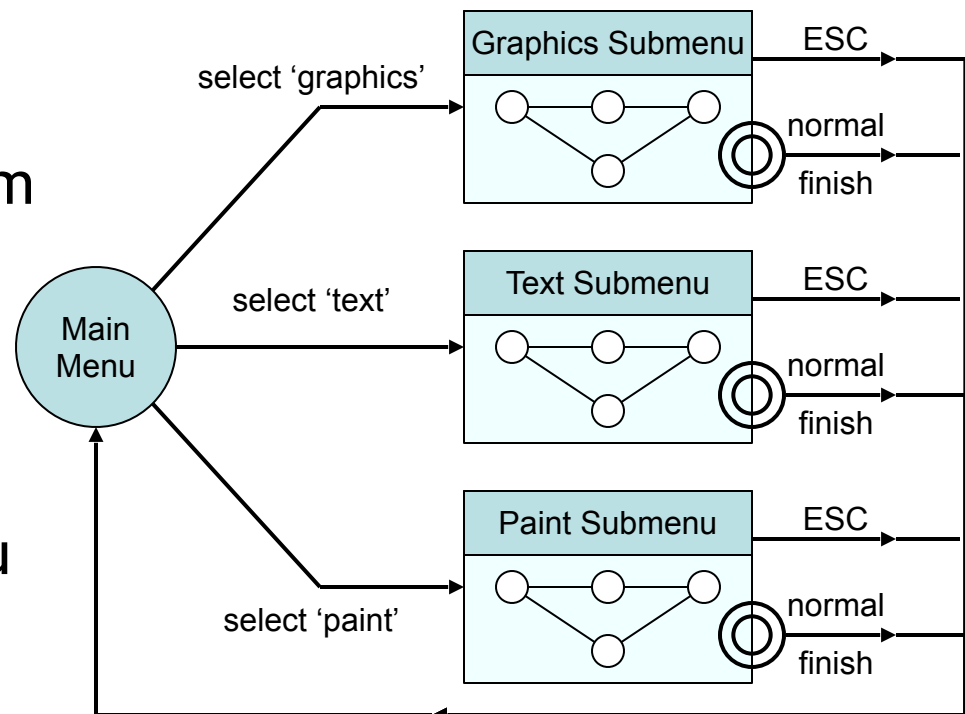
Escapes

- ‘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

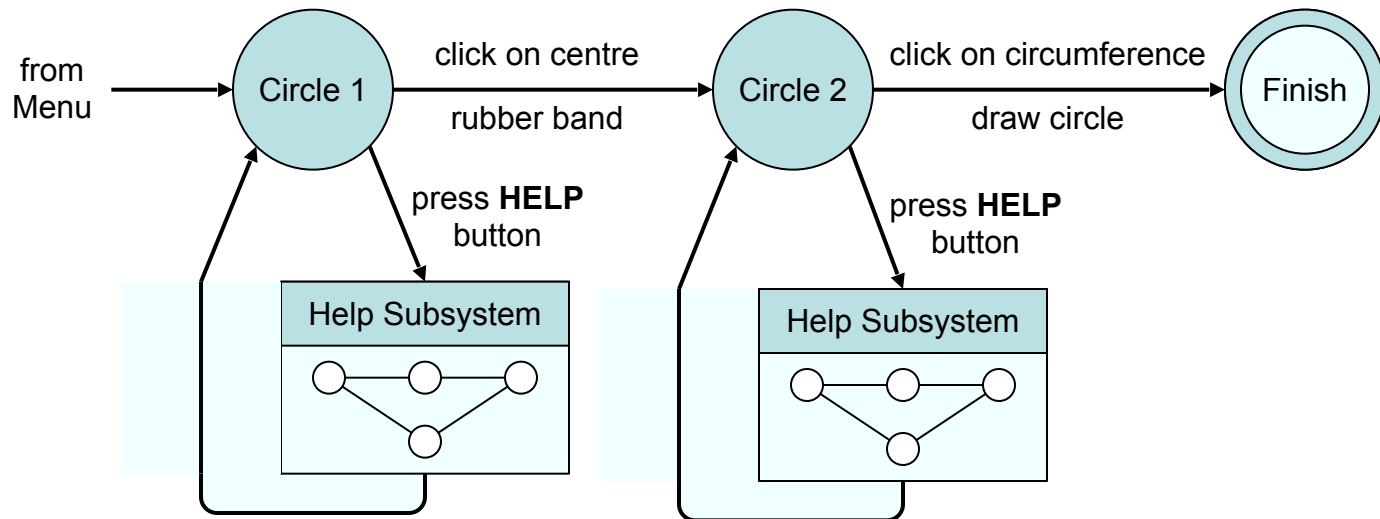
‘normal’ exit for
each submenu

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



State properties

- **Reachability**

- can you get anywhere from anywhere?
- and how easily

- **Reversibility**

- can you get to the previous state?
- but NOT undo

- **Dangerous states**

- some states you don't want to get to



Principles of robustness

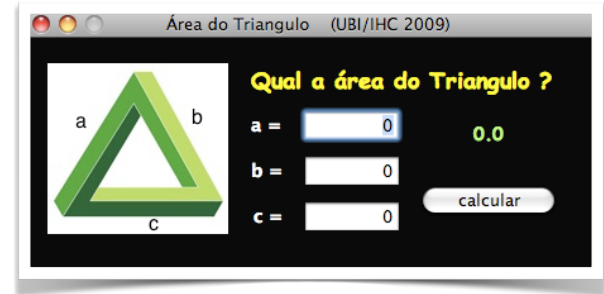


Observability

- ability of user to evaluate the internal state of the system from its perceivable representation
- 5 other principles: browsability; defaults; reachability; persistence; operation visibility

Vocal: email beep

Visual: signal persists



Recoverability

- ability of user to take corrective action once an error has been recognised
- **forward/backward** recovery;
commensurate effort

delete > rename



State properties

- **Reachability**

- can you get anywhere from anywhere?
- and how easily

- **Reversibility**

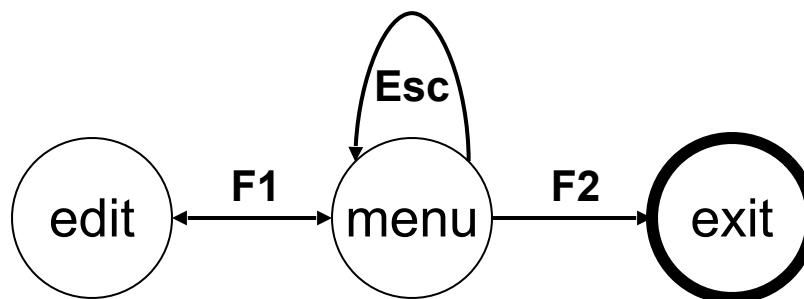
- can you get to the previous state?
- but NOT undo

- **Dangerous states**

- some states you don't want to get to

Dangerous States

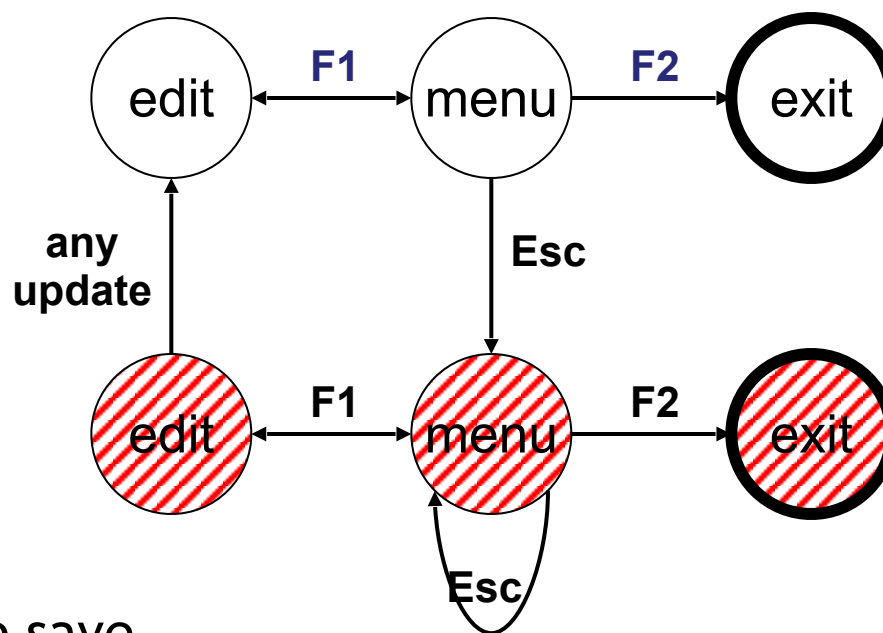
- Word processor: two modes and exit
 - F1 - changes mode
 - F2 - exit (and save)
 - Esc - no mode change



but ... Esc resets autosave

Dangerous States (ii)

- Exit with/without save \Rightarrow dangerous states
- Duplicate states - semantic distinction



F1-F2 - exit with save

F1-Esc-F2 - exit with no save

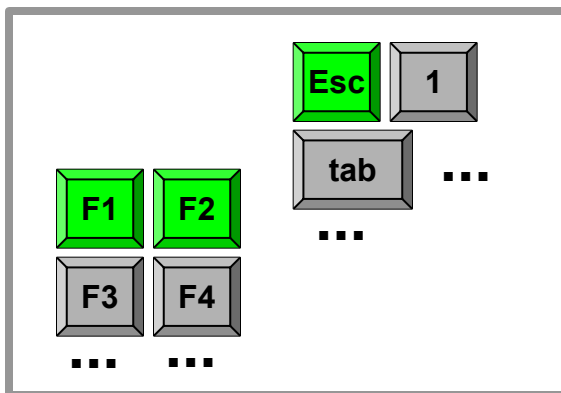
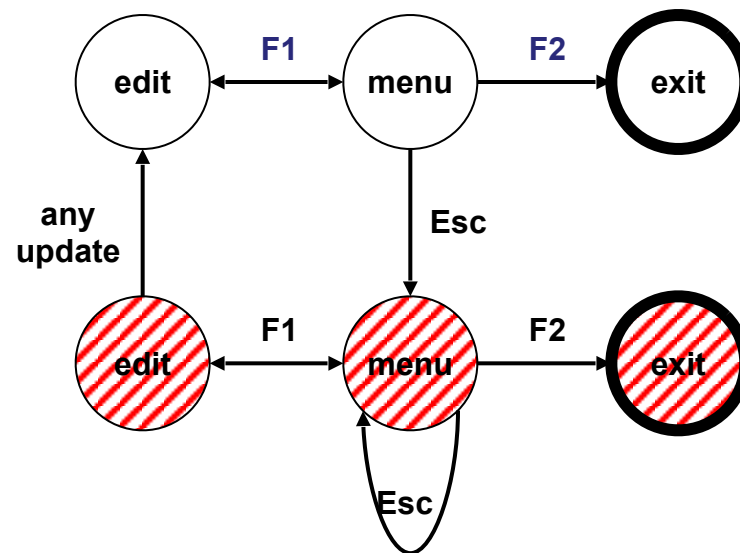
Lexical Issues

- visibility
 - differentiate modes and states
 - annotations to dialogue
- style
 - command - verb noun
 - mouse based - noun verb
- layout
 - not just appearance ...

layout matters

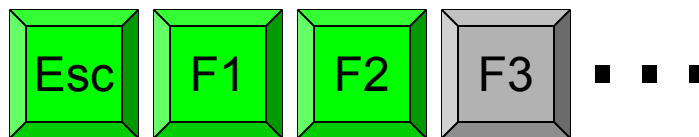
- Word processor - dangerous states

- Old keyboard - OK



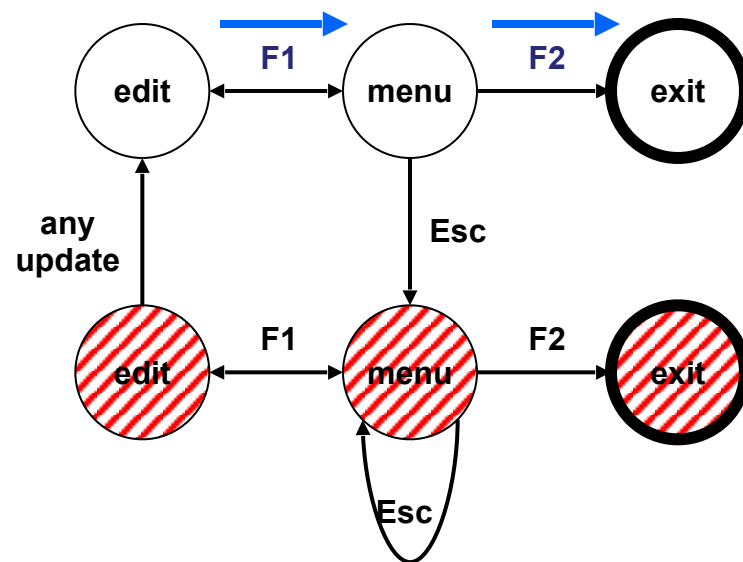
layout matters

- **but** with new keyboard layout



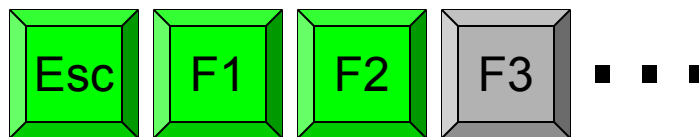
intend F1-F2 (save)

finger catches Esc



layout matters

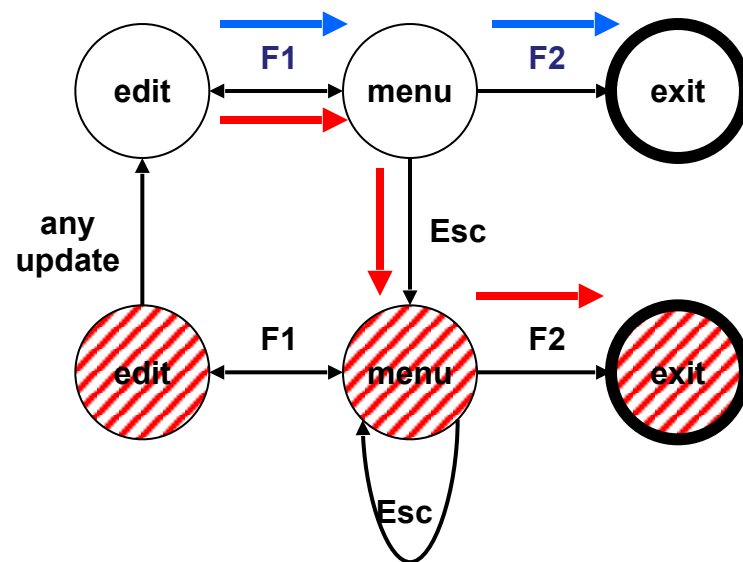
- **but** with new keyboard layout



intend **F1-F2** (save)

finger catches Esc

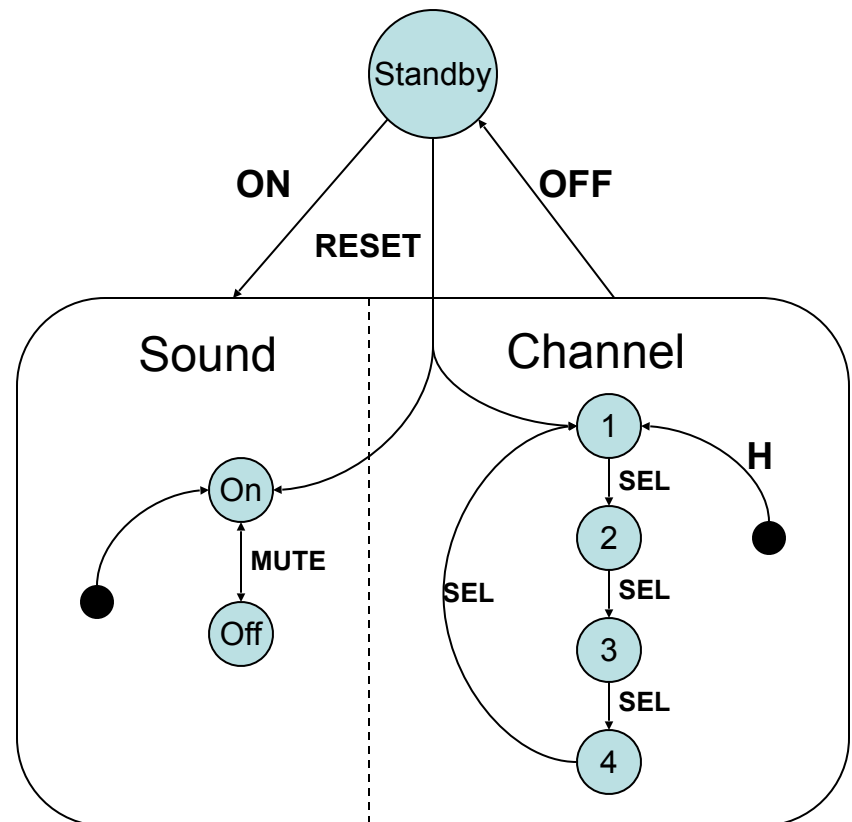
F1-Esc-F2 - **disaster!**



State Charts

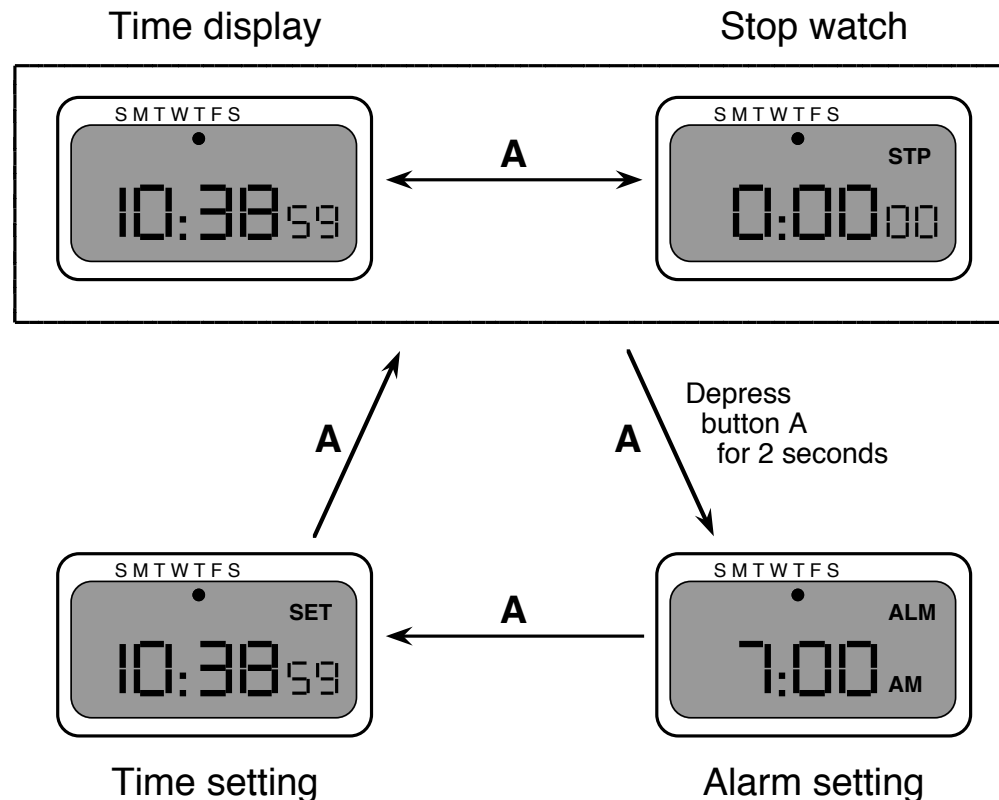
- Used in UML
- Extension to **STN**
 - hierarchy
 - concurrent sub-nets
 - escapes
 - OFF always active
 - history
 - link marked H goes back to last state on re-entering sub-dialogue

TV Control with 5 button:
ON, OFF, RESET, MUTE, SEL



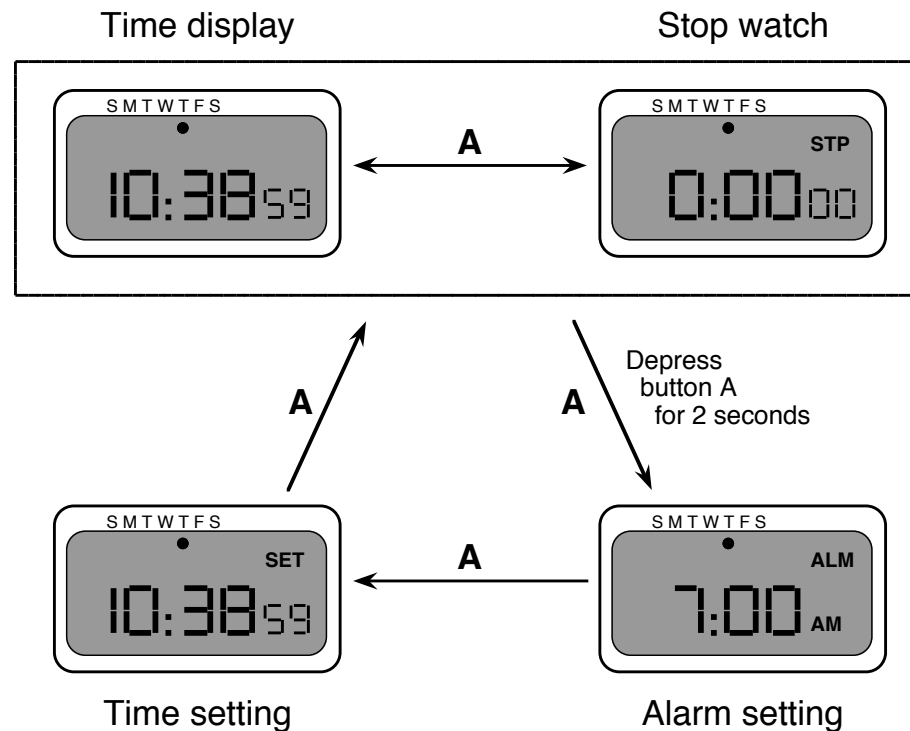
Digital watch - User Instructions

- two main modes
- limited interface
- 3 buttons
- button A
changes mode



Digital watch - User Instructions

- dangerous states
 - *guarded*
... by two second hold
- completeness
 - distinguish depress A and release A
 - what do they do in all modes?

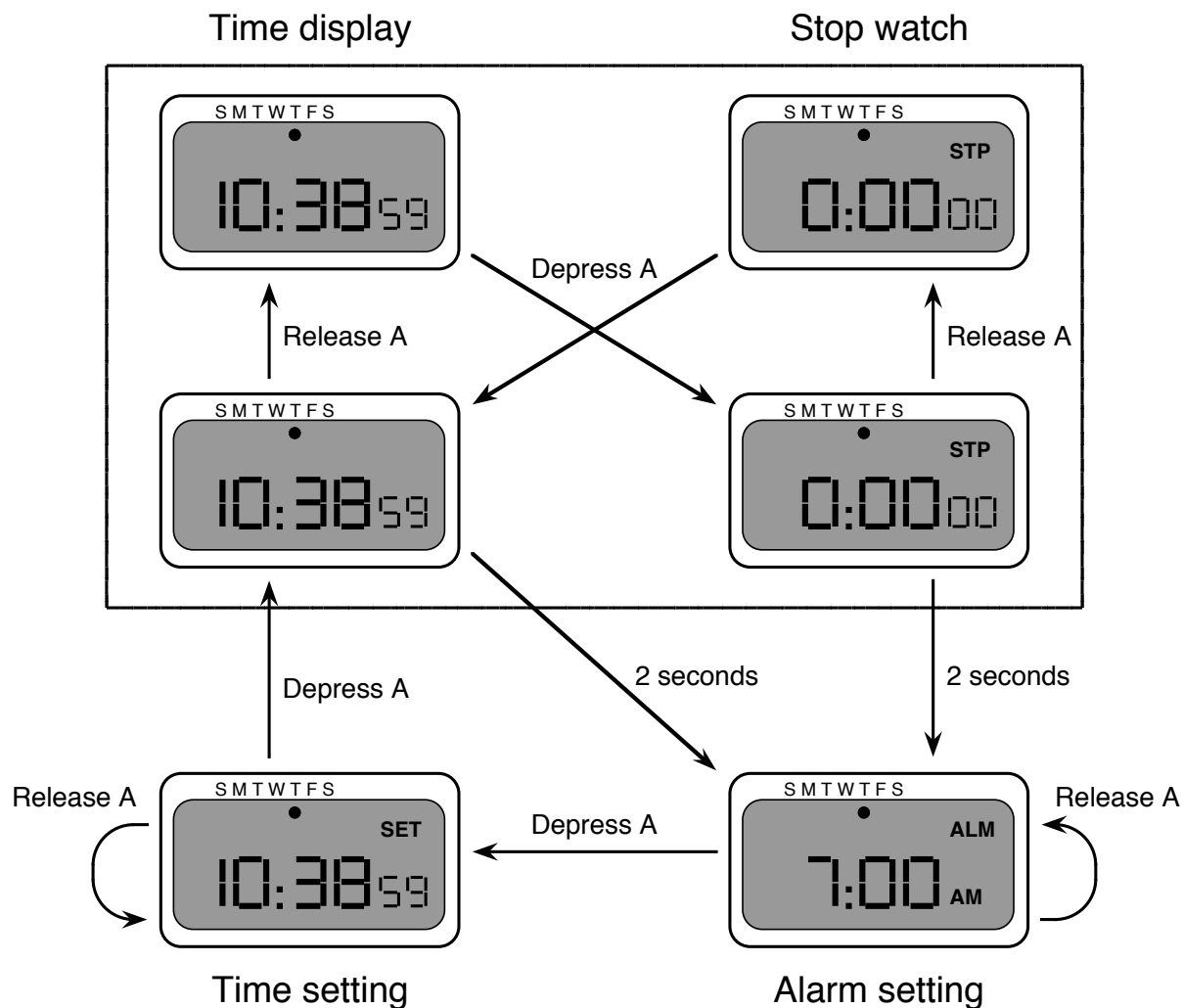




Digital watch - Designers instructions

and ...

that's just
one button



Concurrent dialogues - I

Example: a simple dialogue box

Text Style

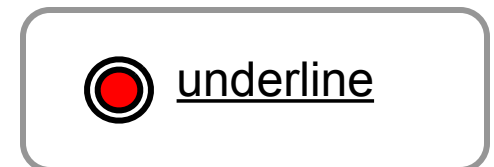
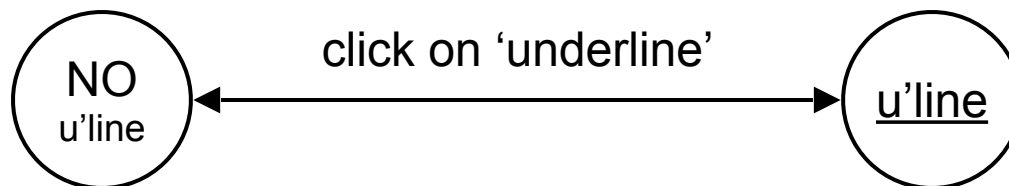
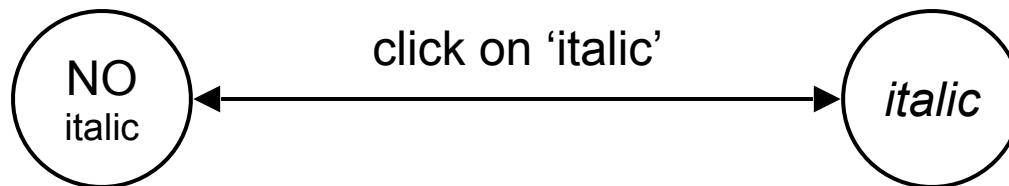
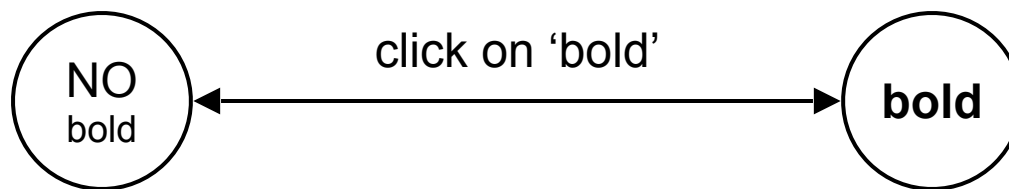
☐ **bold**

example ☒ *italic*

☒ underline

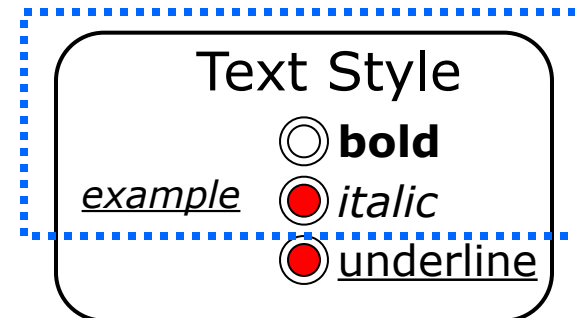
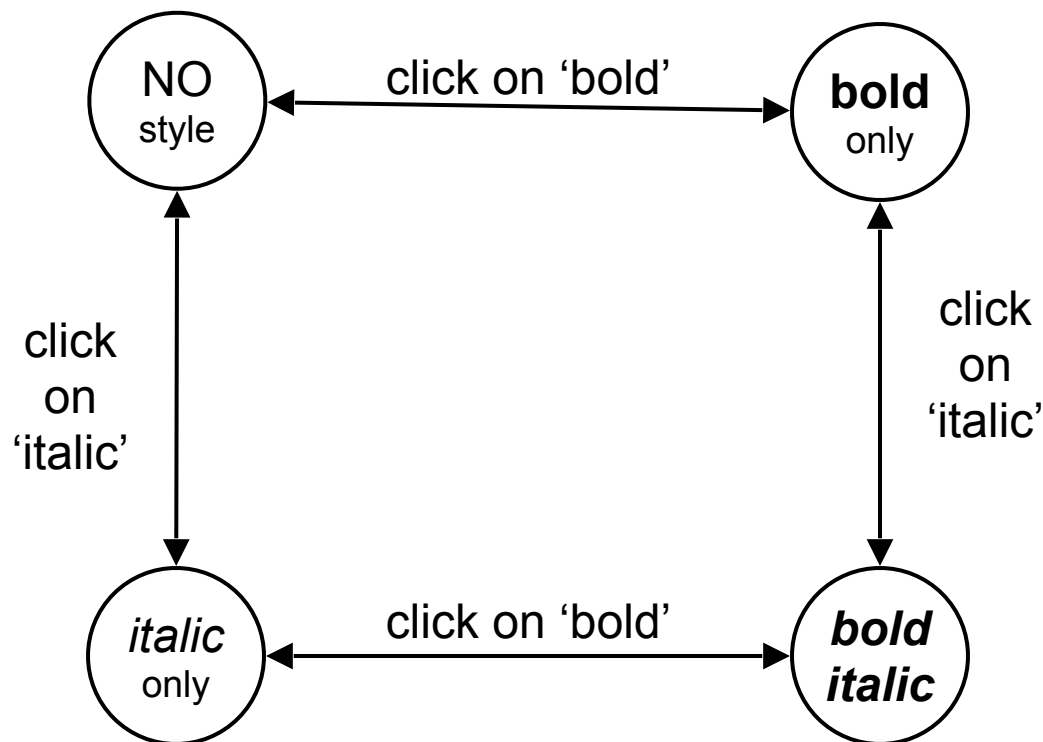
Concurrent dialogues - II

three toggles - individual STNs



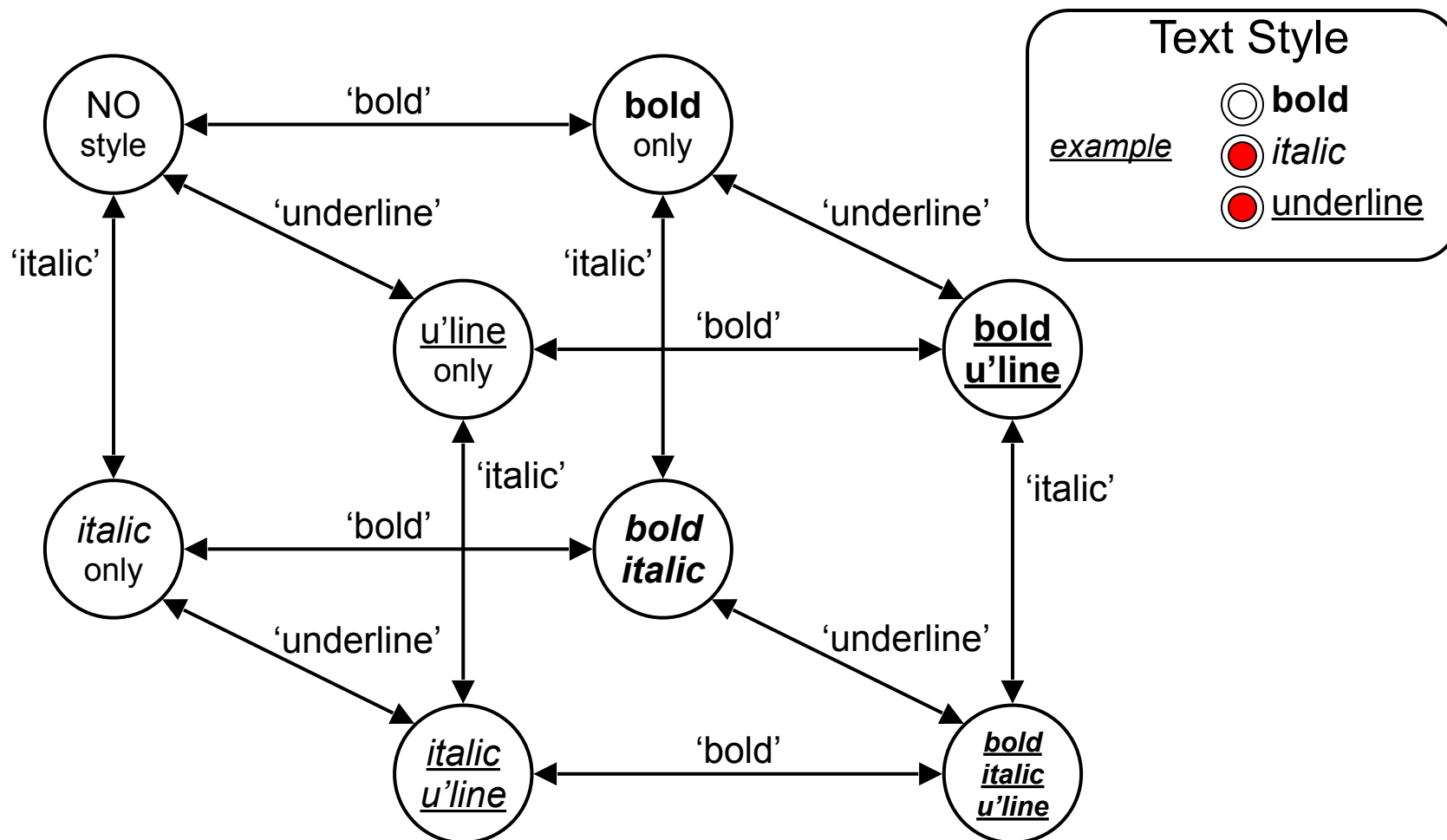
Concurrent dialogues - III

bold and italic combined



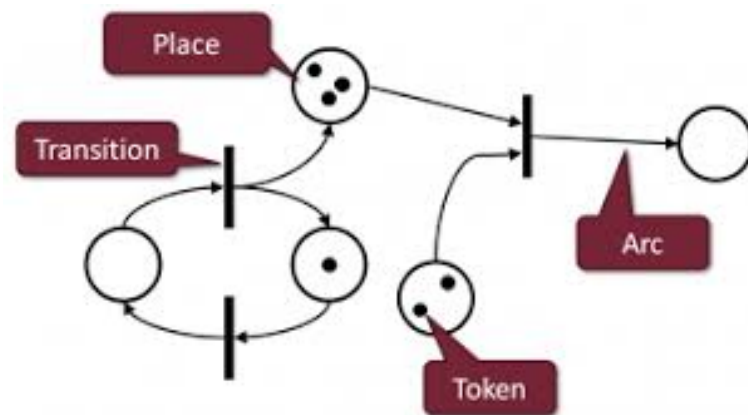
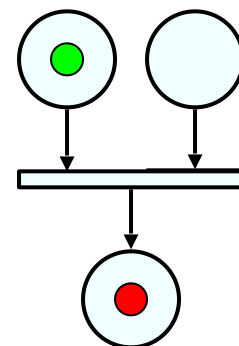
Concurrent dialogues - IV

all together - combinatorial explosion

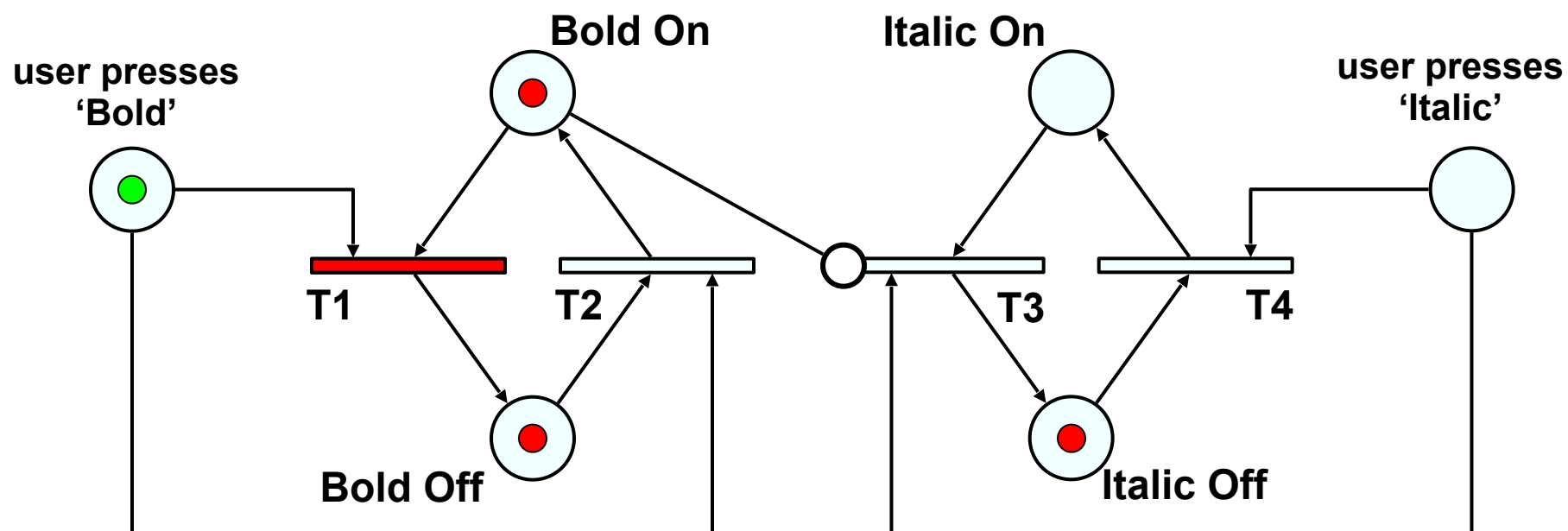


Petri Nets

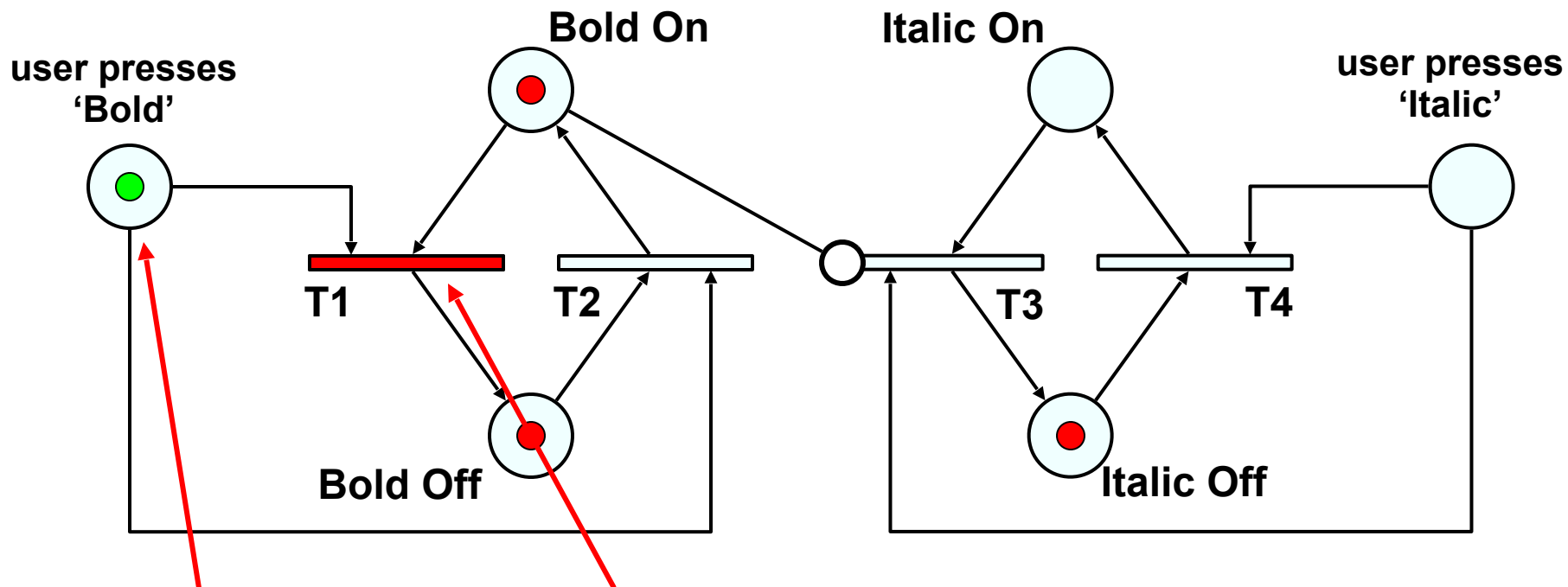
- One of the oldest notations in computing!
- Flow graph:
 - places - a bit like STN states
 - transitions - a bit like STN arcs
 - counters - sit on places (current state)
- Several counters allowed
 - **concurrent** dialogue states
- Used for UI specification
(ICO at Toulouse)
 - tool support - *Petshop*
- Reasoning about **concurrent** activities.



Petri net example



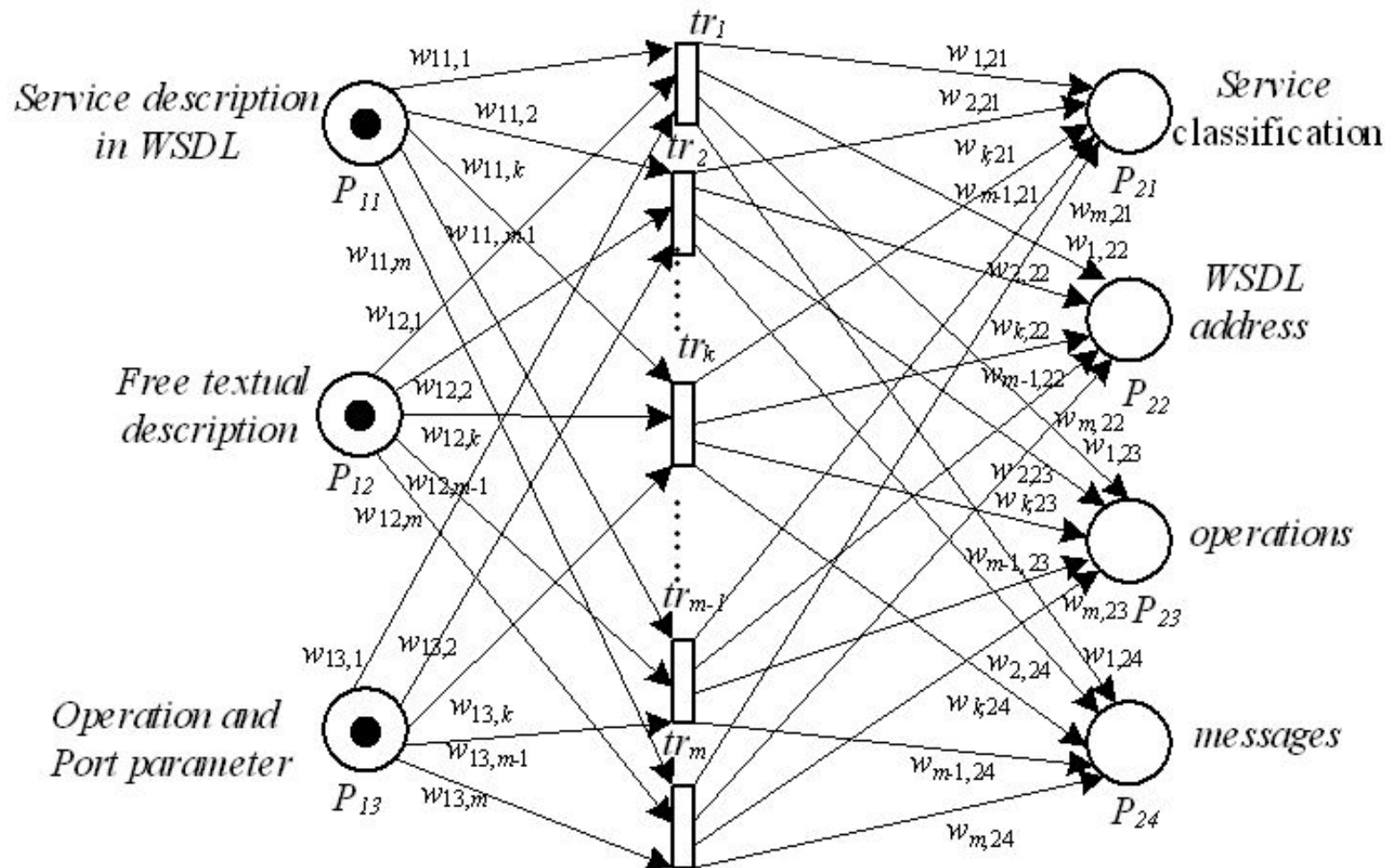
Petri net example



user actions
represented
as a new counter

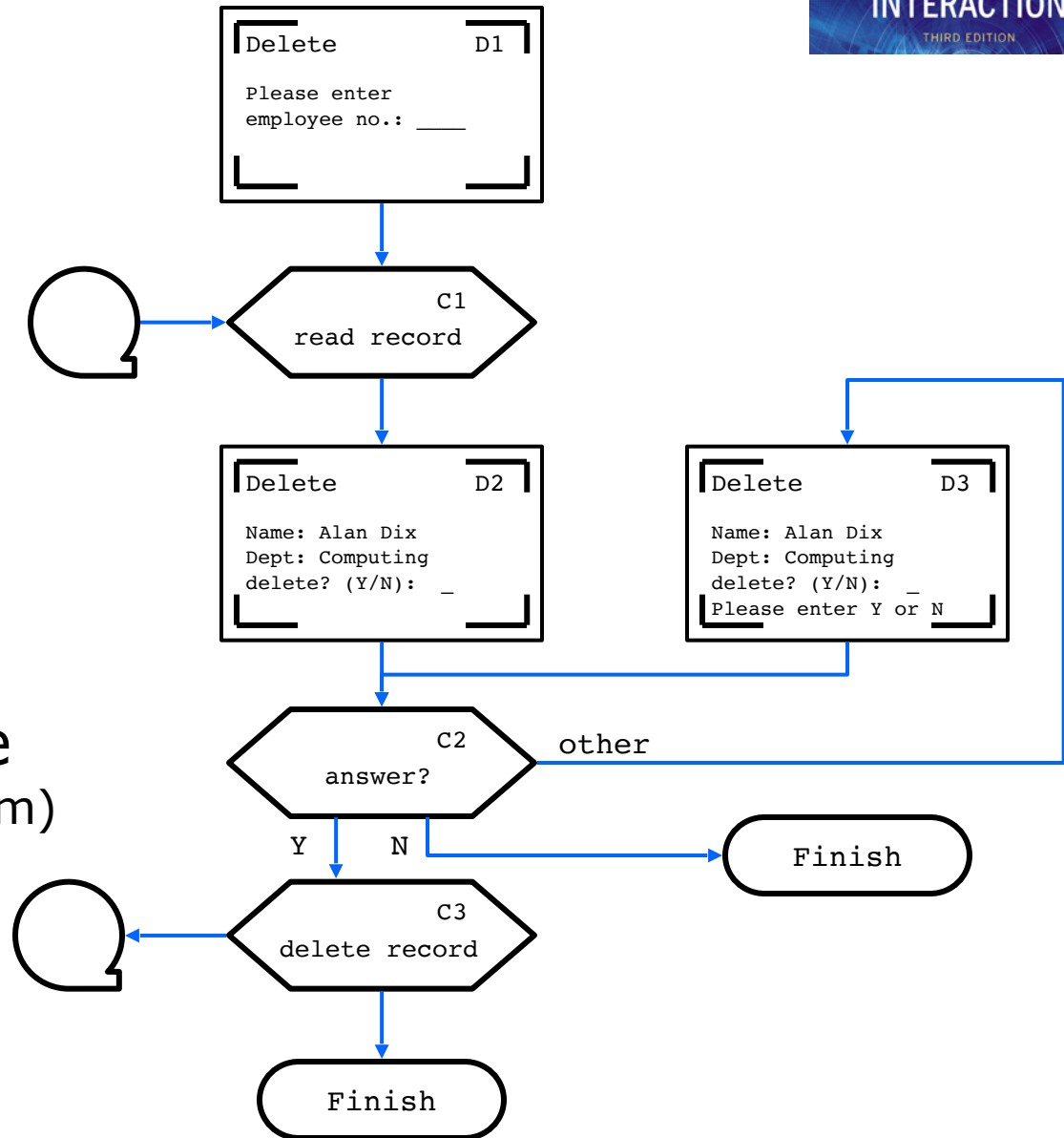
transition 'fires'
when all input
places have counters

Petri net example



Flowcharts

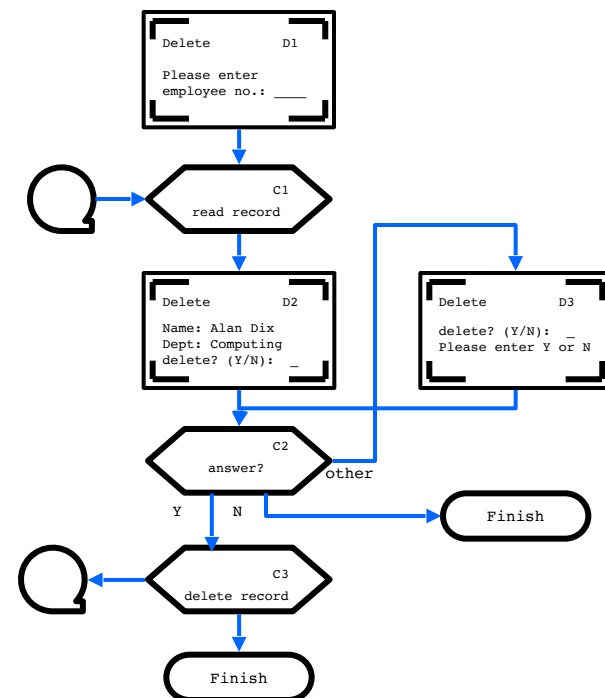
- Familiar to programmers
- Boxes
 - process/event
 - not state
- Use for dialogue (not internal algorithm)



Case study - it works!

Dix et Al. (2004), Page 561

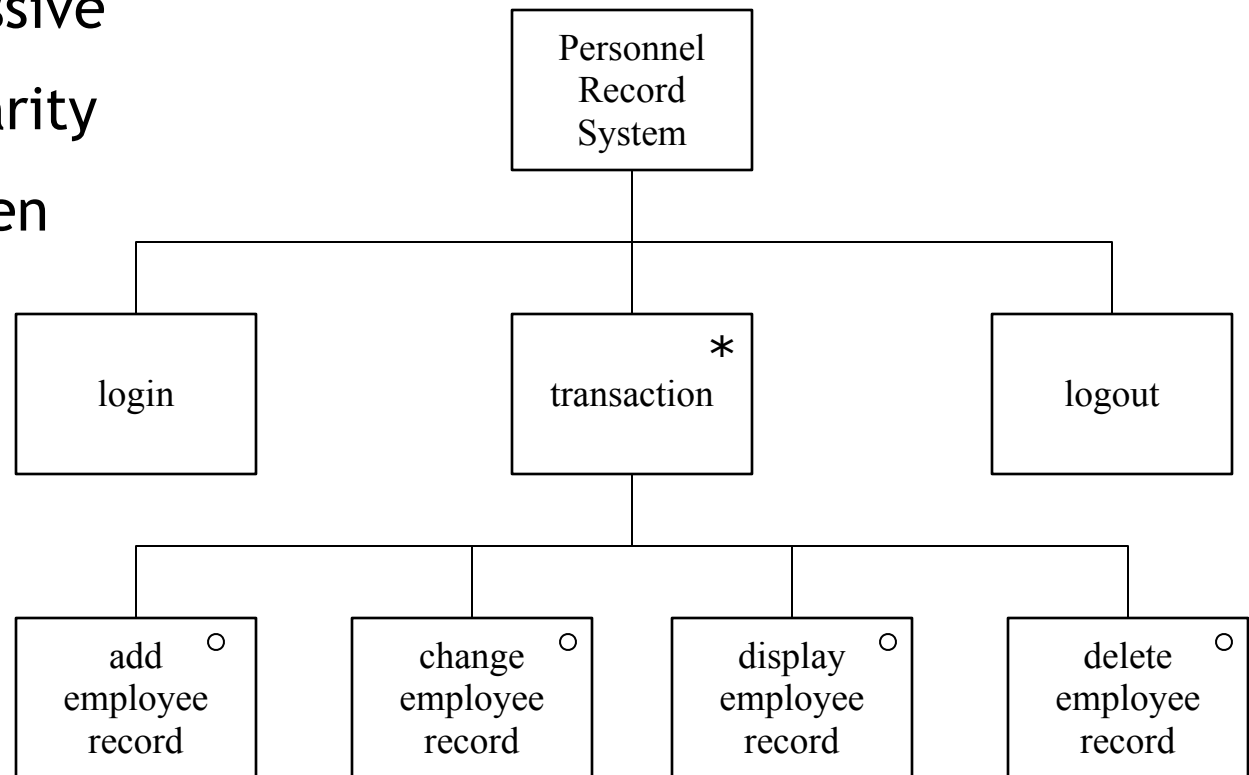
- Formal notations - too much work?
- COBOL transaction processing
 - event-driven - like web interfaces
 - programs structure
≠ dialogue structure
- Used dialogue flow charts
 - discuss with clients
 - transform to code
 - systematic testing
 - 1000% productivity gain
- Formalism saves time !!!



JSD — Jackson Structured Design diagrams

- For tree structured dialogues

- less expressive
- greater clarity
- menu-driven



Textual Notations

grammars
production rules
CSP and event algebras

Textual - Grammars

- Regular expressions

`sel-line click click* dble-click`

- compare with JSD

- same computational model
- different notation

- Backus-Naur Form (BNF)

```
expr ::= empty
      | atom expr
      | '(' expr ')' expr
```

- more powerful than regular exp. or STNs
- Still NO concurrent dialogue

Backus-Naur Form (BNF)

- Very common notation from computer science
- A purely syntactic view of the dialogue
- Terminals
 - lowest level of user behavior
 - e.g. CLICK-MOUSE, MOVE-MOUSE
- Nonterminals
 - ordering of terminals
 - higher level of abstraction
 - e.g. select-menu, position-mouse

Example of BNF

- **Basic syntax:**
 - nonterminal ::= expression
- **An expression**
 - contains terminals and nonterminals
 - combined in sequence (+) or as alternatives (|)

```
draw line    ::=  select line + choose points + last point
select line  ::=  pos mouse + CLICK MOUSE
choose points ::=  choose one | choose one + choose points
choose one   ::=  pos mouse + CLICK MOUSE
last point   ::=  pos mouse + DBL CLICK MOUSE
pos mouse    ::=  NULL | MOVE MOUSE+ pos mouse
```

Production rules

- Unordered list of rules:

if *condition* **then** *action*

- condition based on state or pending events
 - every rule always potentially active
- Good for concurrency
 - Bad for sequence