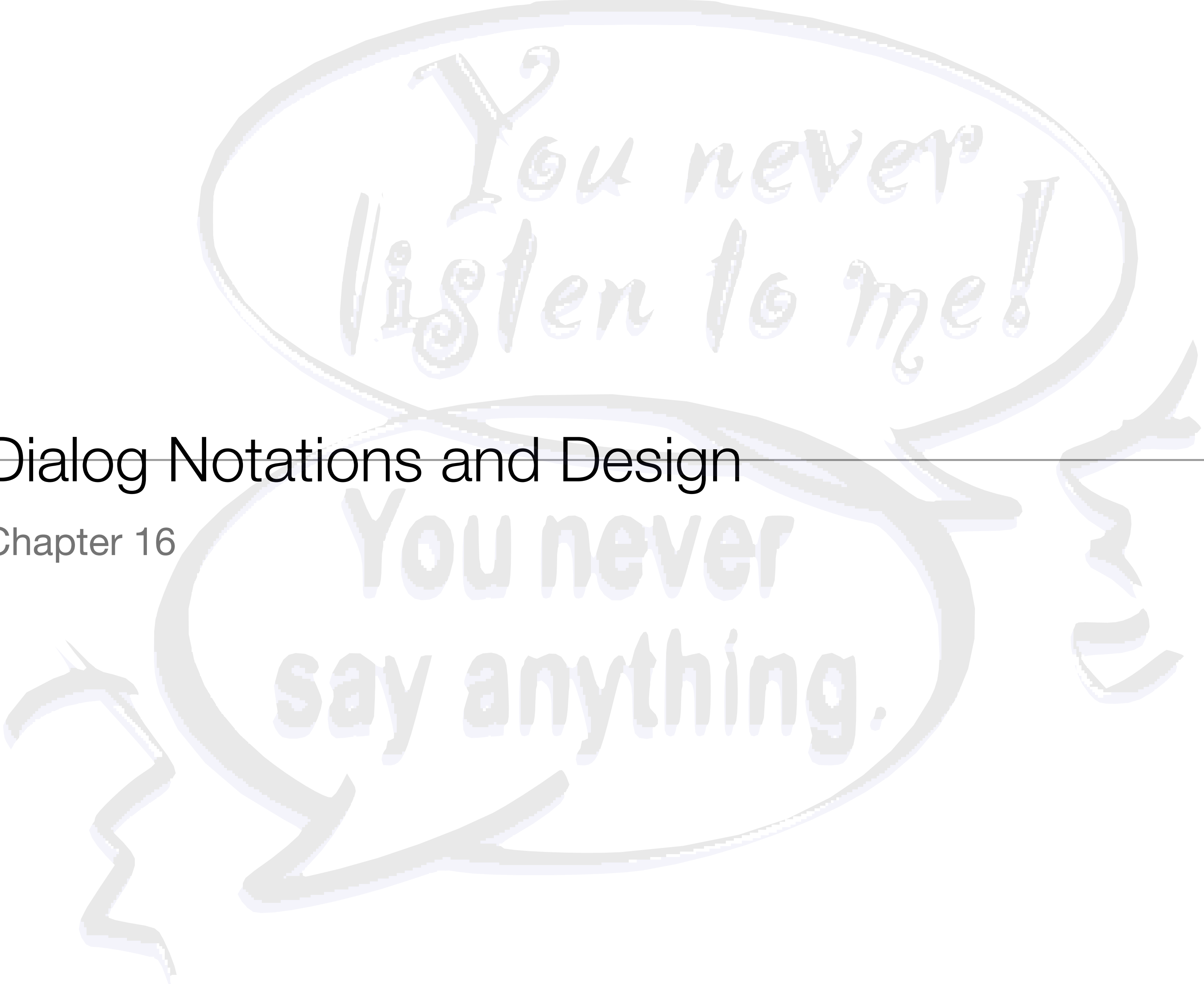
Dialog Notations and Design

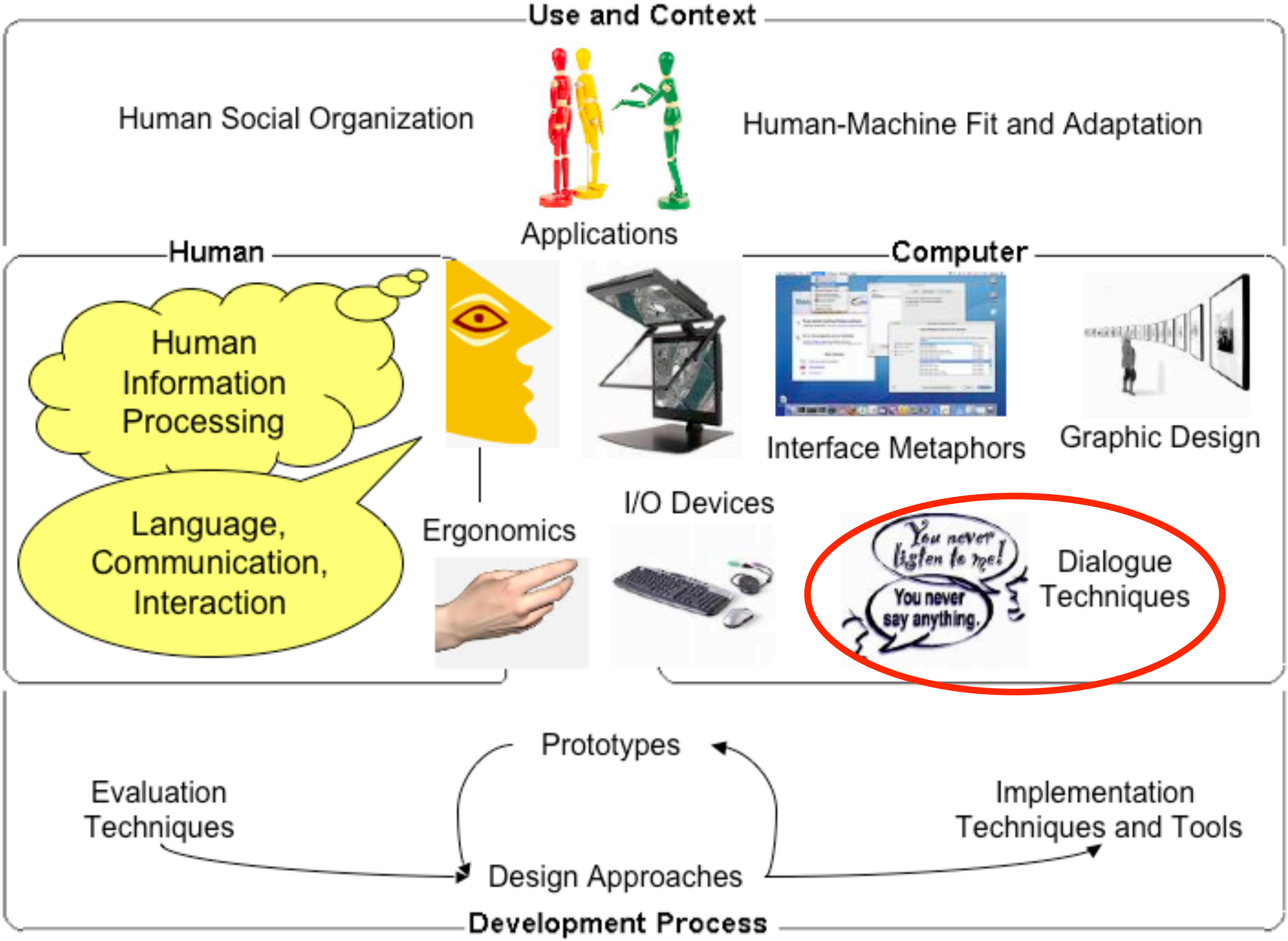
Chapter 16



These slides are based upon those from Dr. Vincent Ng and from the Dix et al. text.

In this lecture

- You will learn
  - Reasons for formal dialog specification
  - Be able to use STNs and Regular Expressions to specify a dialog
  - Be able to conduct checks for completeness, reversibility and dangerous states with dialogs
  - Know about other methods for dialog specifications



Use and Context

Human Social Organization

Applications

Human-Machine Fit and Adaptation

Human

Human Information Processing

Language, Communication, Interaction

Ergonomics

Computer

Interface Metaphors

Graphic Design

I/O Devices

Dialogue Techniques

Prototypes

Evaluation Techniques

Design Approaches

Implementation Techniques and Tools

Development Process

what is dialogue?

- conversation between two or more parties
  - usually cooperative
- in HCI:
  - syntactic level of human–computer ‘conversation’
  - refers to the *structure* of the interaction
  - like a script, but have choices
- 3 levels:
  - lexical -- what shape, color of icons
  - syntactic <-- most user interfaces
  - semantic

structured human dialogue

- most human-human dialogue is unstructured
- human-computer dialogue very constrained
- some human-human dialogue formal too …

structured human dialogue

- most human-human dialogue is unstructured
- 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:
  - the process of the trial depends on the defendant’s response
- focus on normative responses
  - doesn’t cope with judge saying “off with her head”
  - or in computer dialogue user standing on keyboard!

… 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 defendant’s response
- focus on normative responses
  - doesn’t cope with judge saying “off with her head”
  - or in computer dialogue user standing on keyboard!

why use dialogue design notations?

- In a big system can we:
  - Analyze the dialogue:
    - Can the user always get to see current shopping basket
  - Change platforms (e.g. Windows/Mac)
  - Dialogue notations helps us to
    - Analyze systems
    - Separate lexical from semantic
- … and before the system is built
  - Notations help us understand proposed designs
- Hard to answer all that from looking at the program code!
  - Dialogue gets buried in the program logic

Dialog Designs

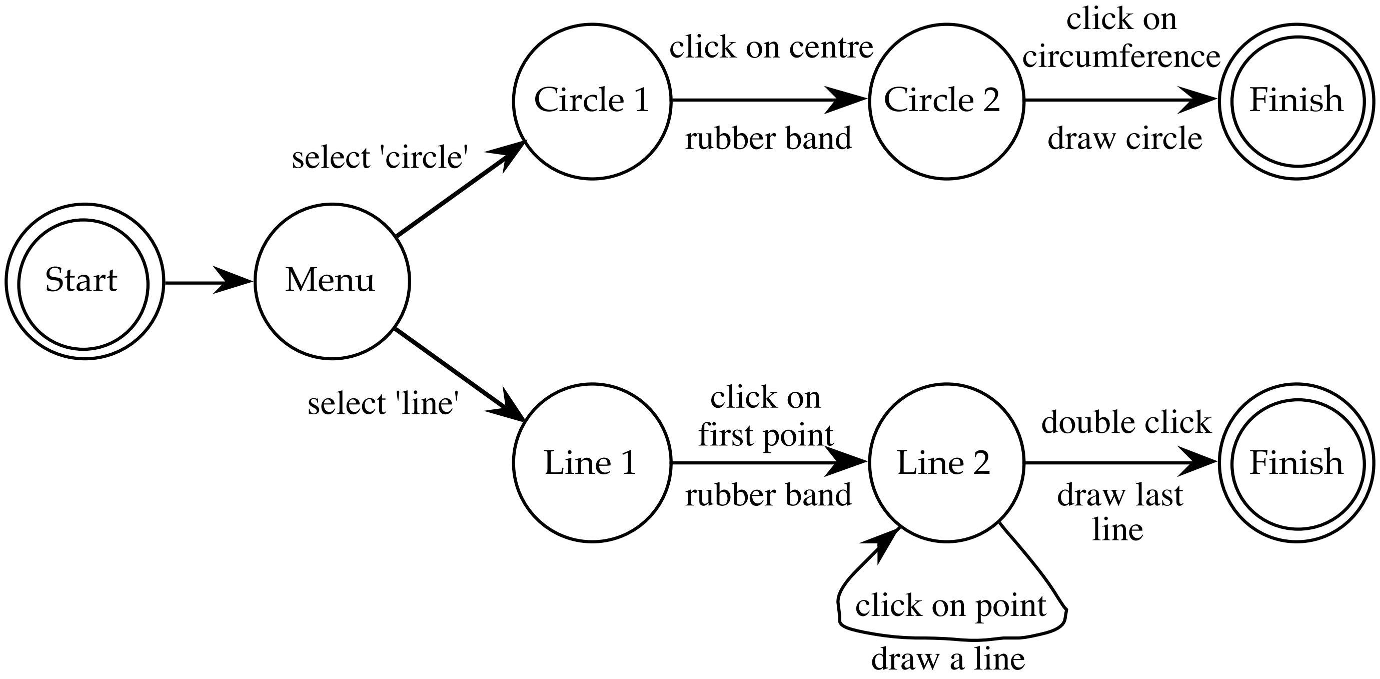
- Diagrammatic
  - **State transition networks**, JSD Diagrams, flowcharts, etc
- Textual
  - **BNFs, Formal Grammars**, Production Rules, CSP
- Linked to:
  - System semantics -- what it does
  - System Presentation -- how it looks

Diagrammatic Notations

- Heavily used, can see structure at a glance
- But problems with extensive or complex structures...

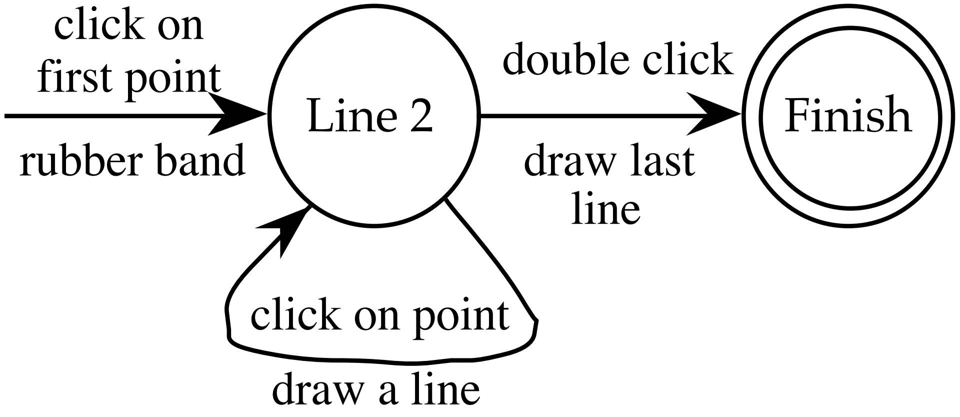
State Transition Networks (STNs)

- Circles: States -- where system is waiting for next input (unless we're at finish)
- Arcs: Actions/Events -- transitions between states, labelled with the user action that triggers the transition and the system response



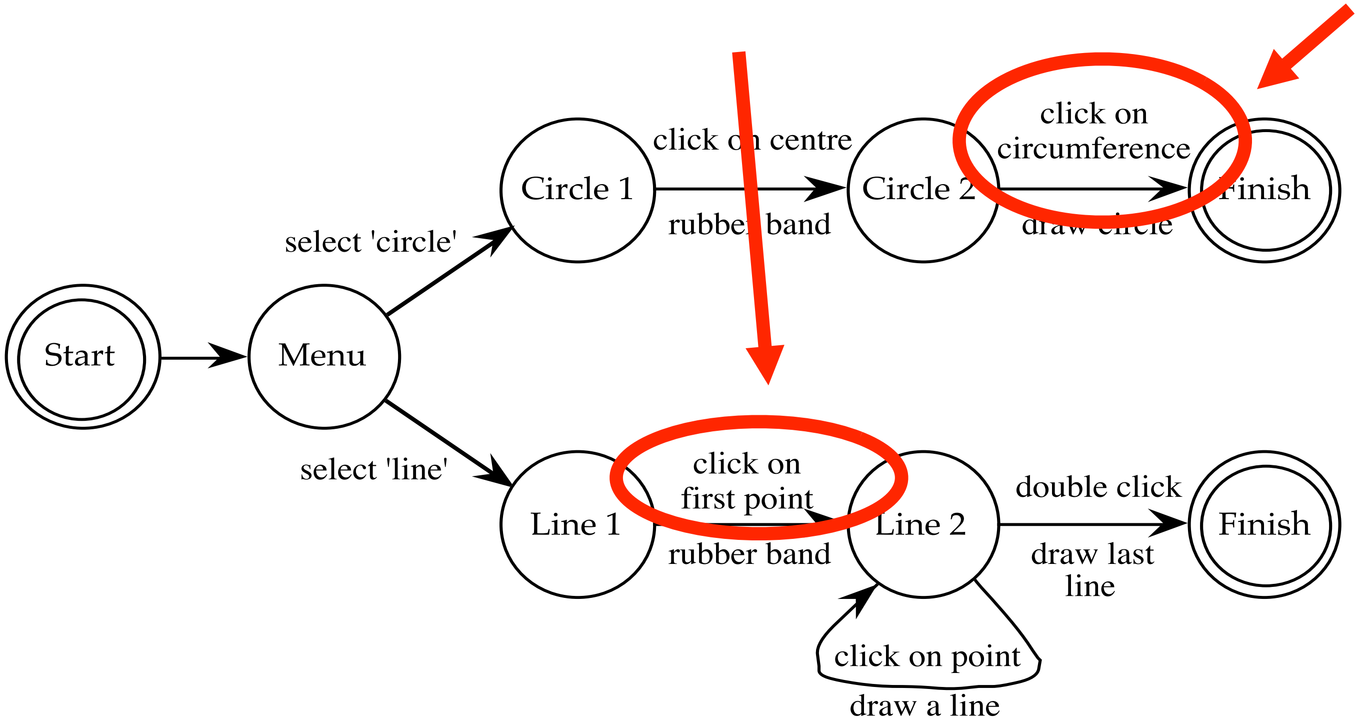
State Transition Networks (STNs)

- Multiple choices by user can be illustrated
- Iteration --> one or more states



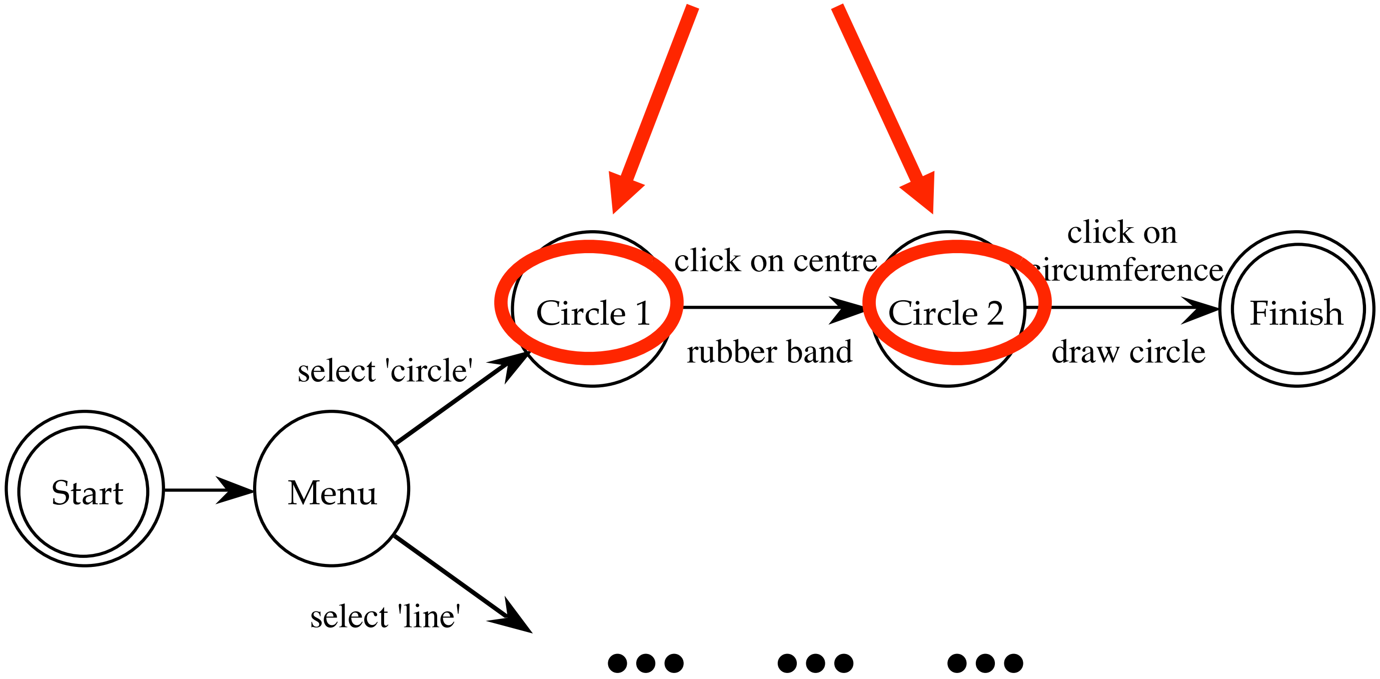
STNs -- Events

- Arc labels a bit cramped because:
  - Notation is “state heavy”
  - But events require most details



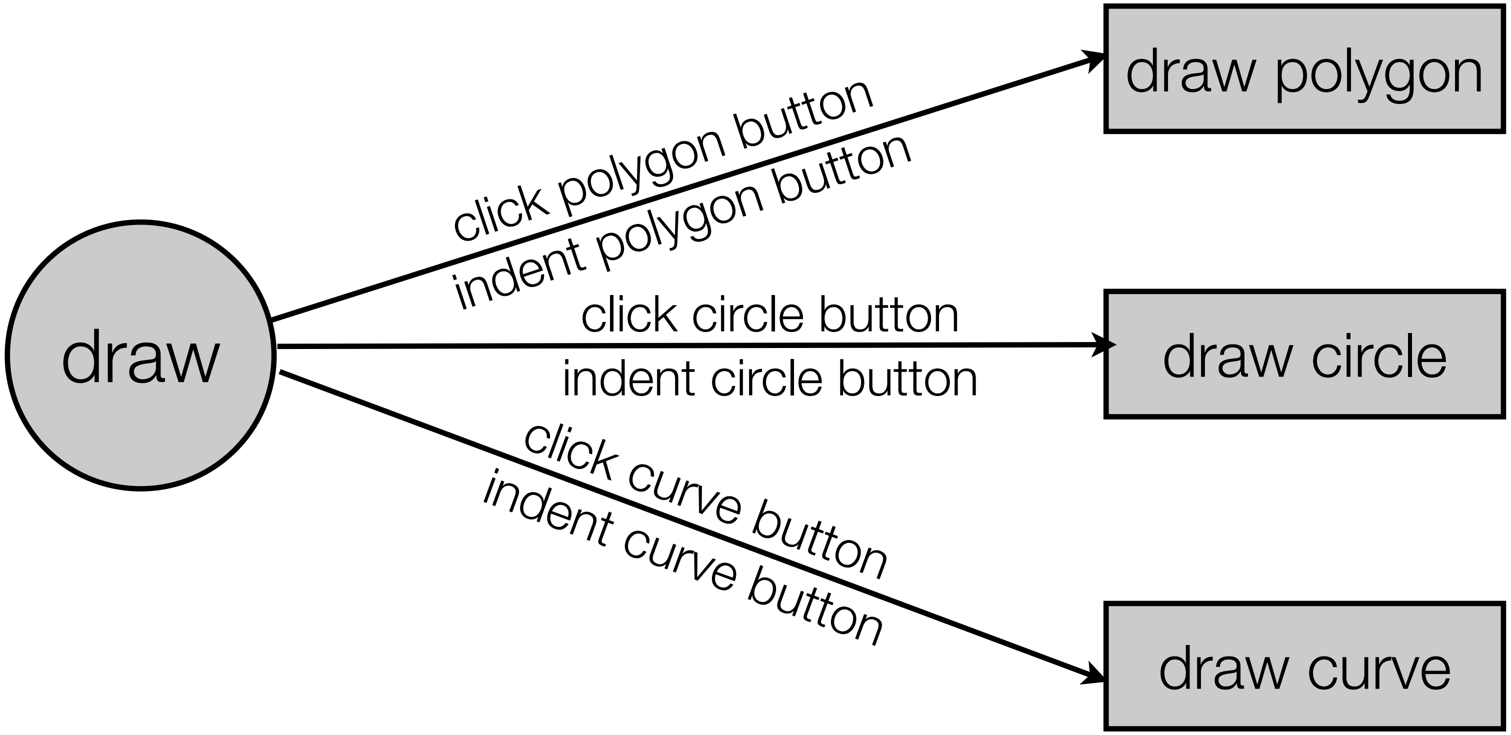
STNs -- States

- Labels in circles a bit uninformative
    - States are hard to name
    - But easier to visualize



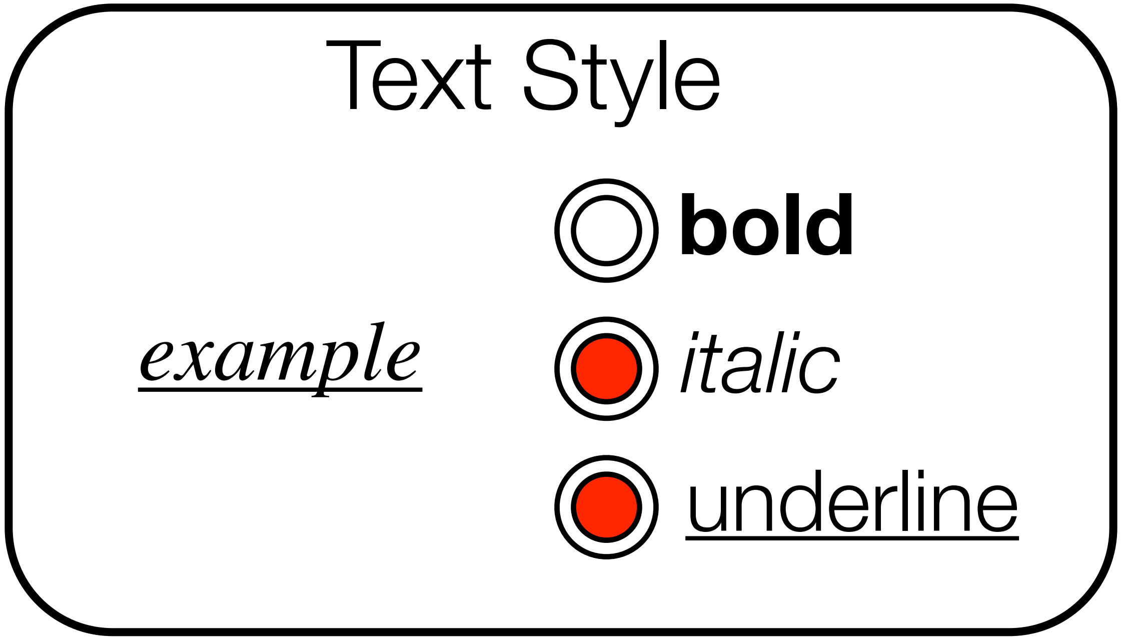
Hierarchical STNs

- Really just an STN inside another STN
  - Named sub-dialogs
- Essential for managing complex dialogs



Concurrent Dialogs

- What if several things happen simultaneously?
- e.g. Simple dialog box for text formatting



Text Style

example

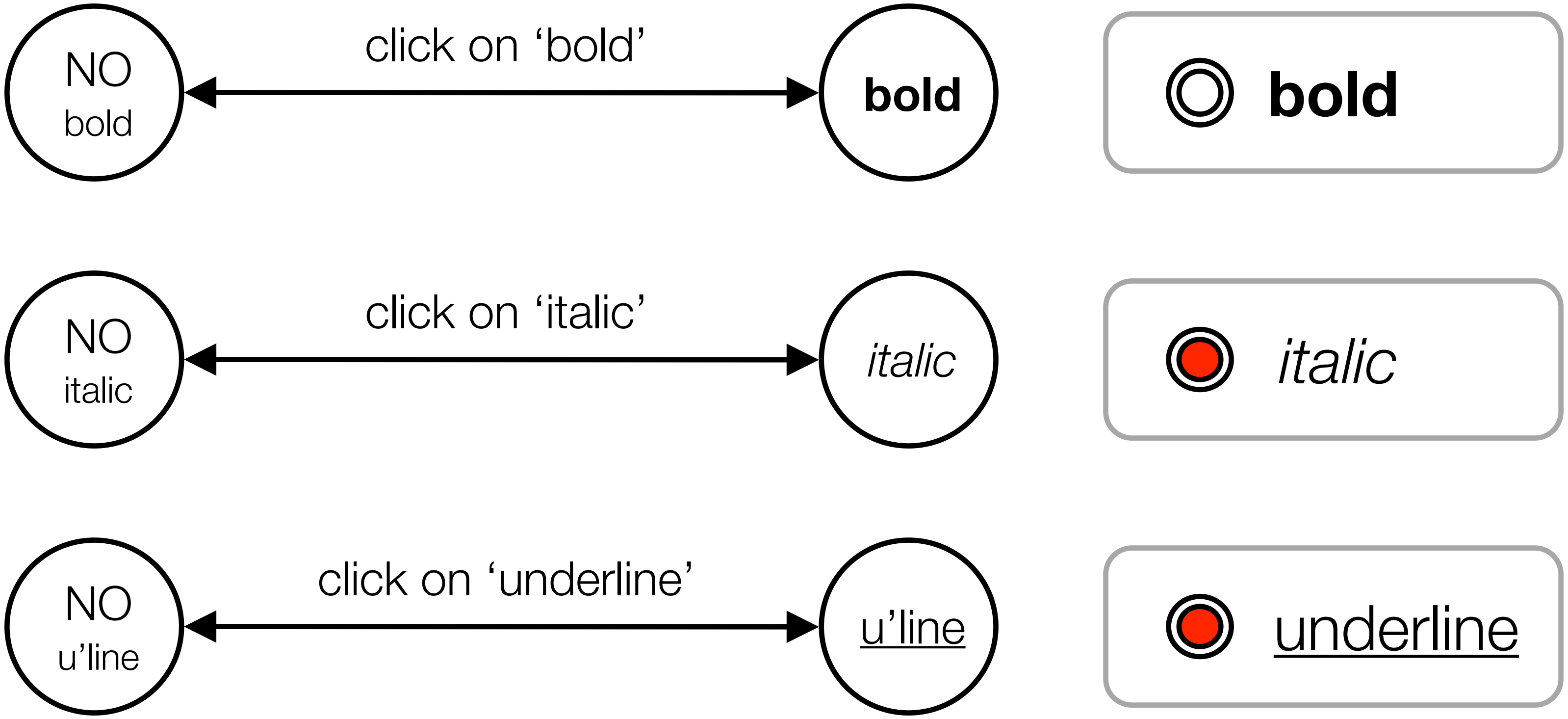
○ **bold**

● *italic*

● underline

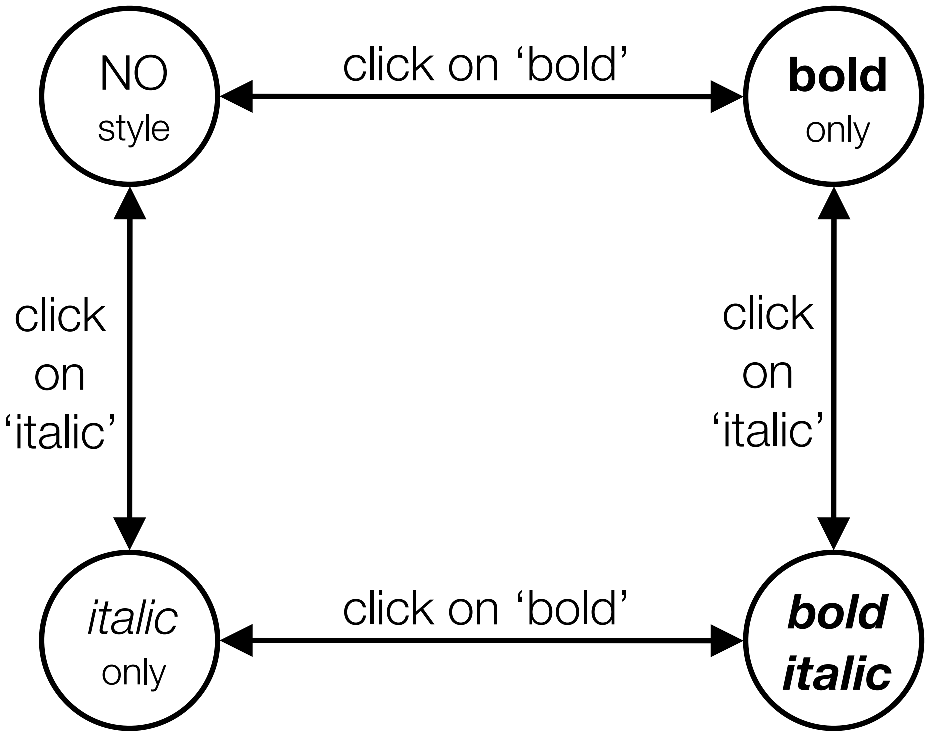
Concurrent Dialogs

- Three toggles: Three STNs



Concurrent Dialogs

- Concurrent means “together”...


NO style

bold only

italic only

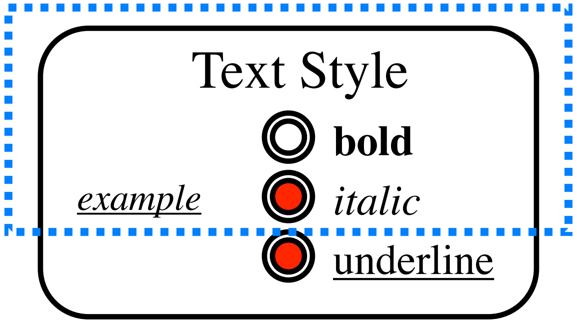
bold italic

click on ‘bold’

click on ‘bold’

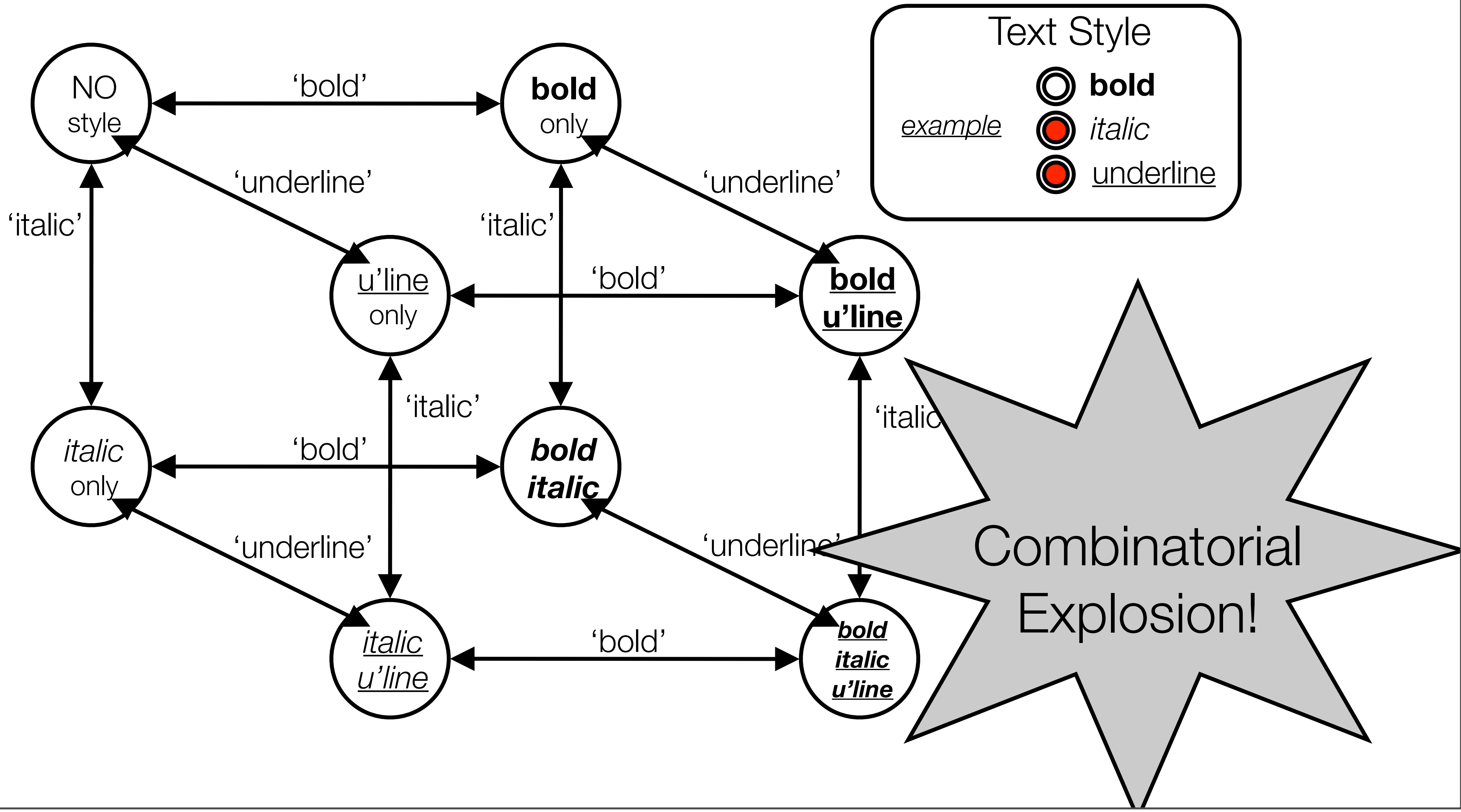
click on ‘italic’

click on ‘italic’



Concurrent Dialogs

- Put all possible combinations together...



Combinatorial Explosion!

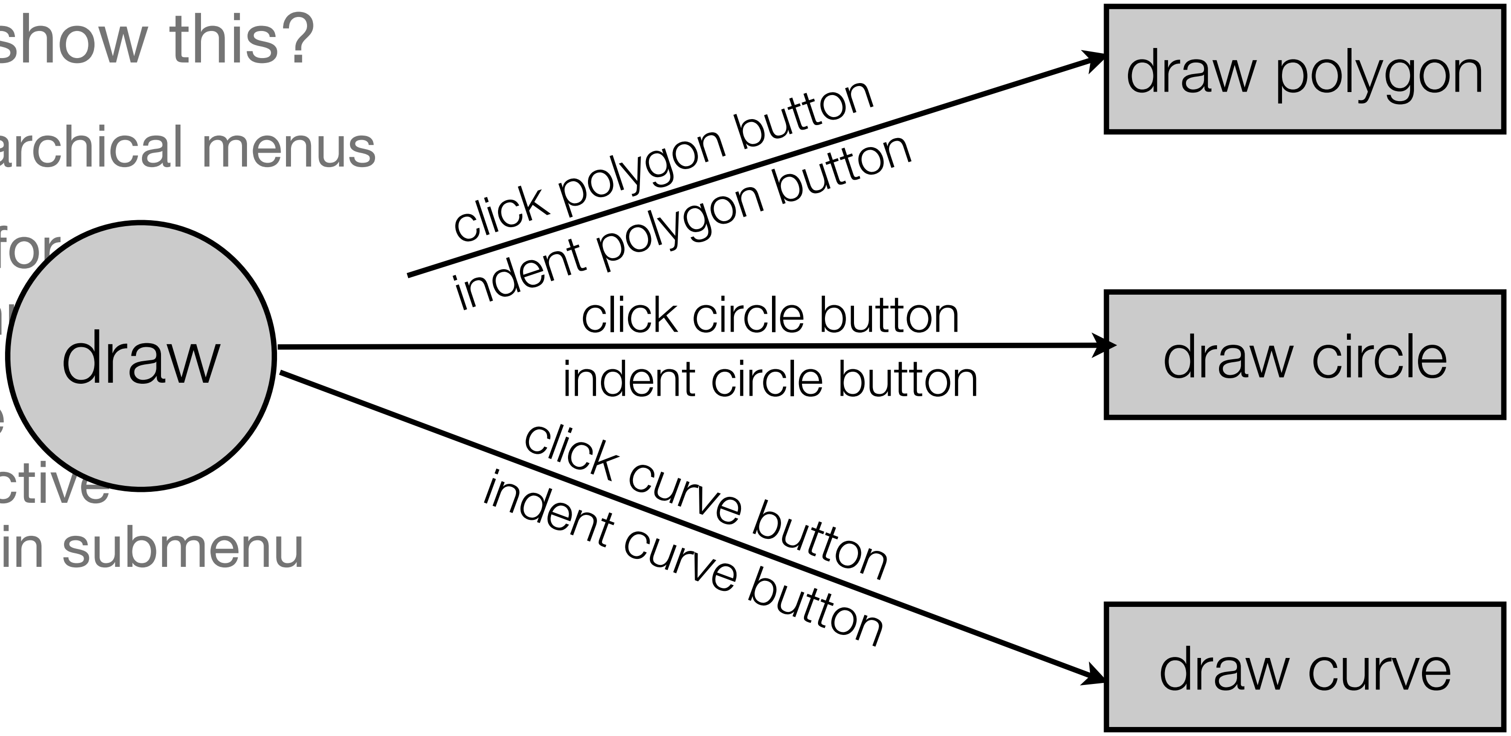
Escape/Undo

- ‘back’ in web, escape/cancel keys
  - Similar behavior everywhere
  - End up with spaghetti of identical behaviors!
- How do we show this?

e.g. Use hierarchical menus

‘normal’ exit for
each submenu

plus separate
escape arc active
‘everywhere’ in submenu



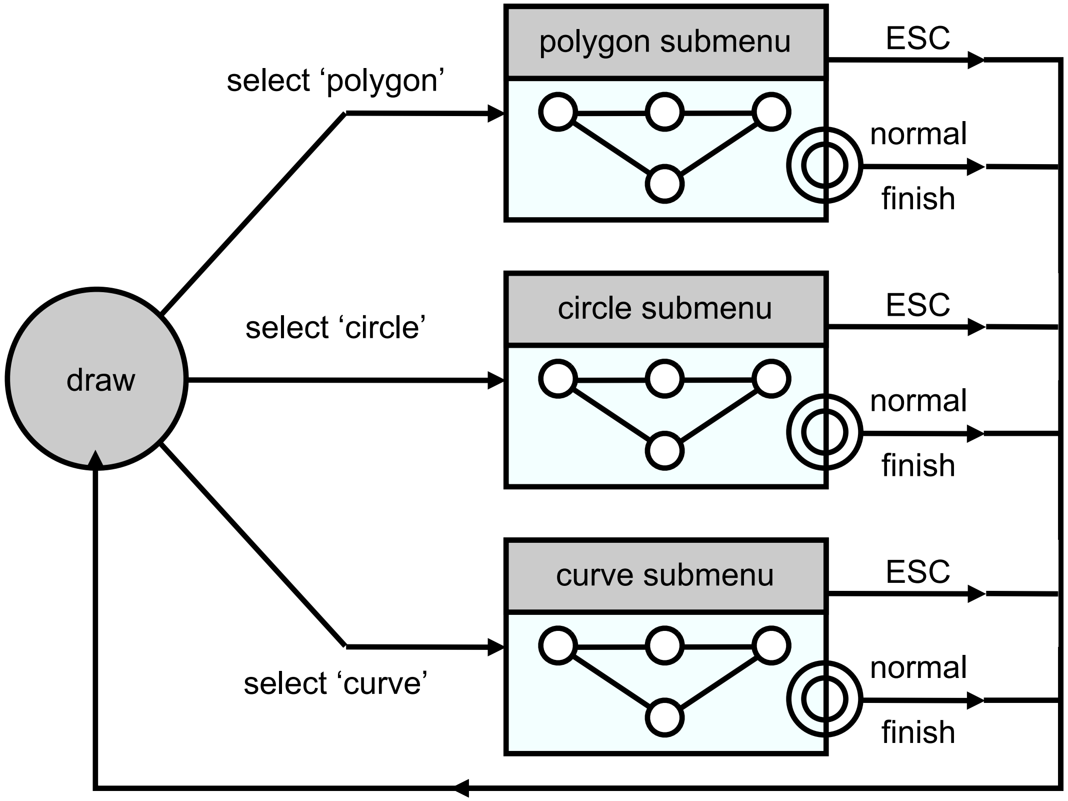
Escape/Undo

- ‘back’ in web, escape/cancel keys
  - Similar behavior everywhere
  - End up with spaghetti of identical behaviors!
- How do we show this?

  e.g. Use hierarchical menus

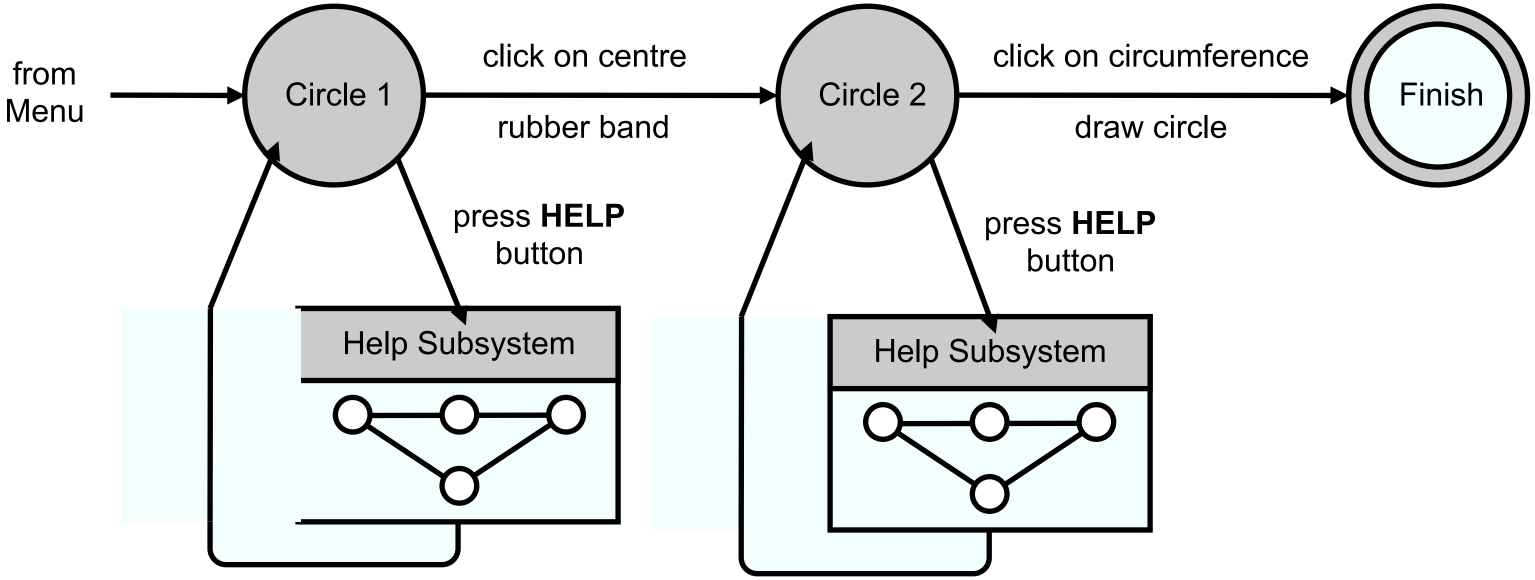
  ‘normal’ exit for each submenu

  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 -- subdialog hanging off every state

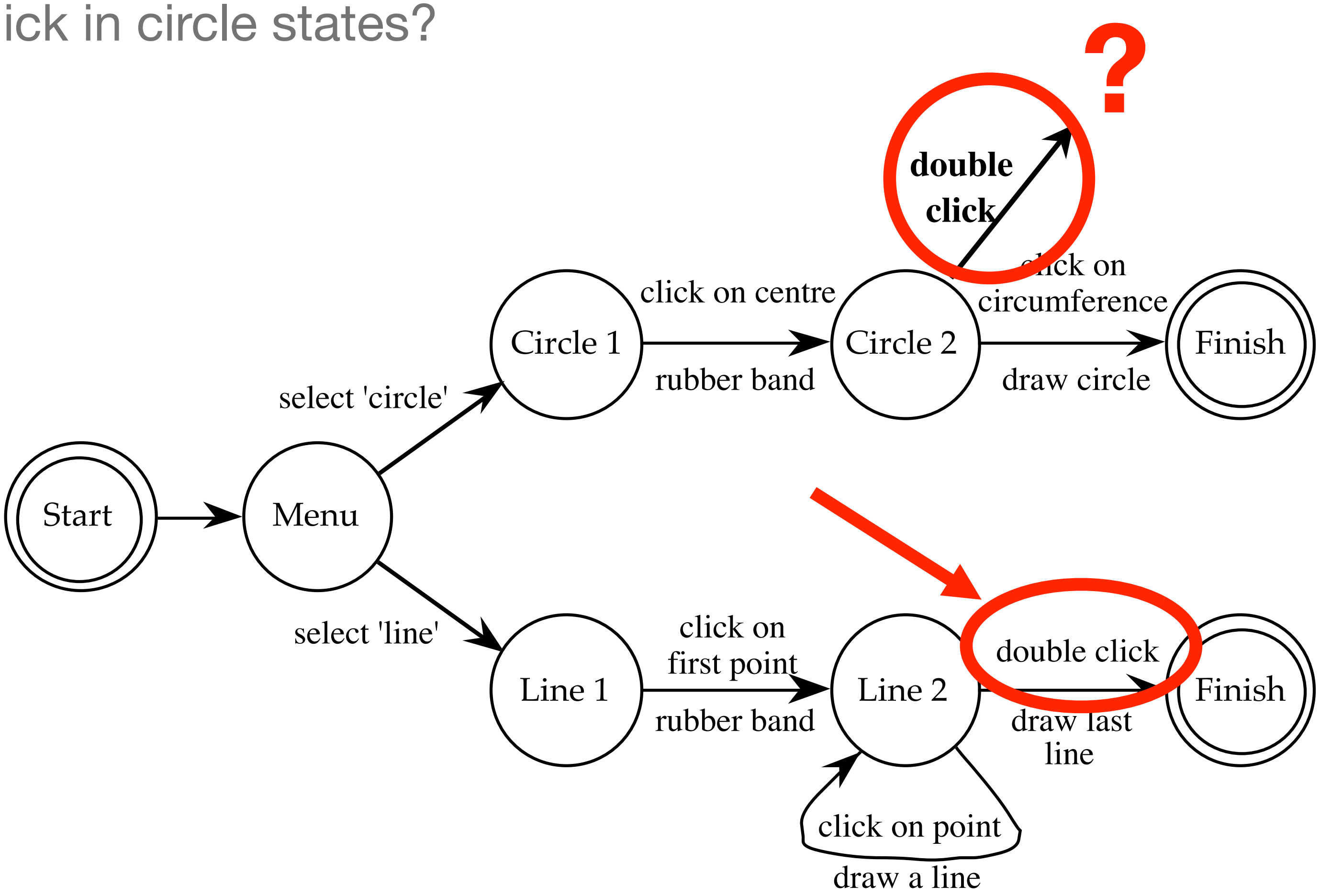


Action Properties

- Completeness
  - Missed arcs -- is there a possibility that we might get to an "unknown" state?
  - Unforeseen circumstances
    - e.g. what if user clicks on drawing surface while at the main menu?
- Determinism
  - One action, one state --> one result
  - Find several arcs with the same label coming out of the same state.
- Consistency
  - Same action in different circumstances == same effect?
    - e.g. tab key when entering text or navigating a dialog

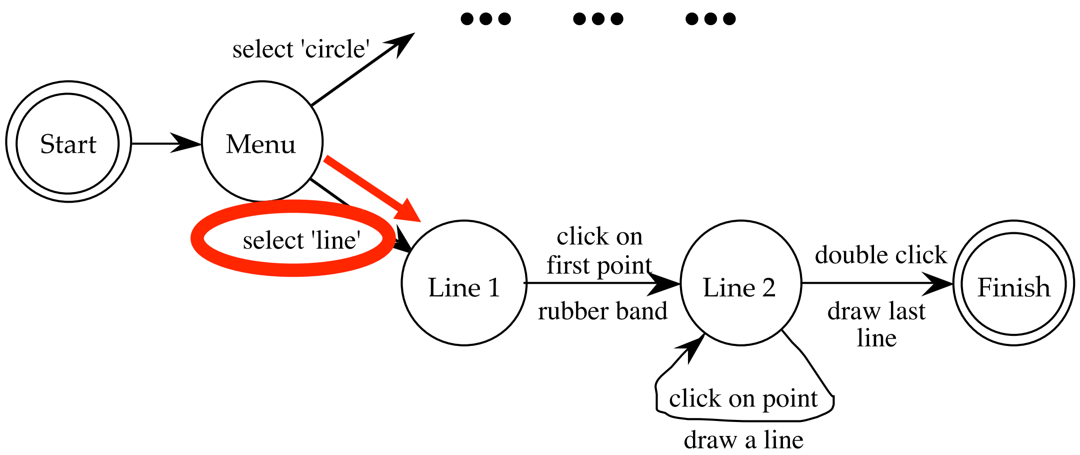
Checking Properties

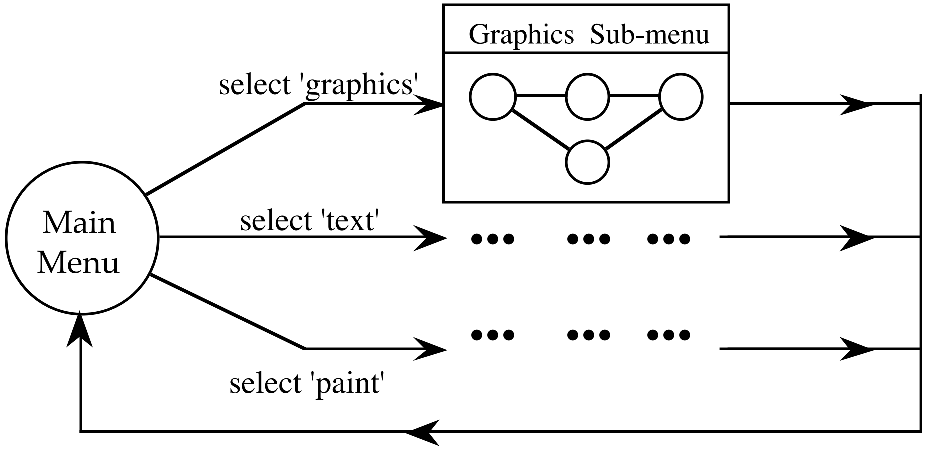
- Completeness
    - Double-click in circle states?



Checking Properties

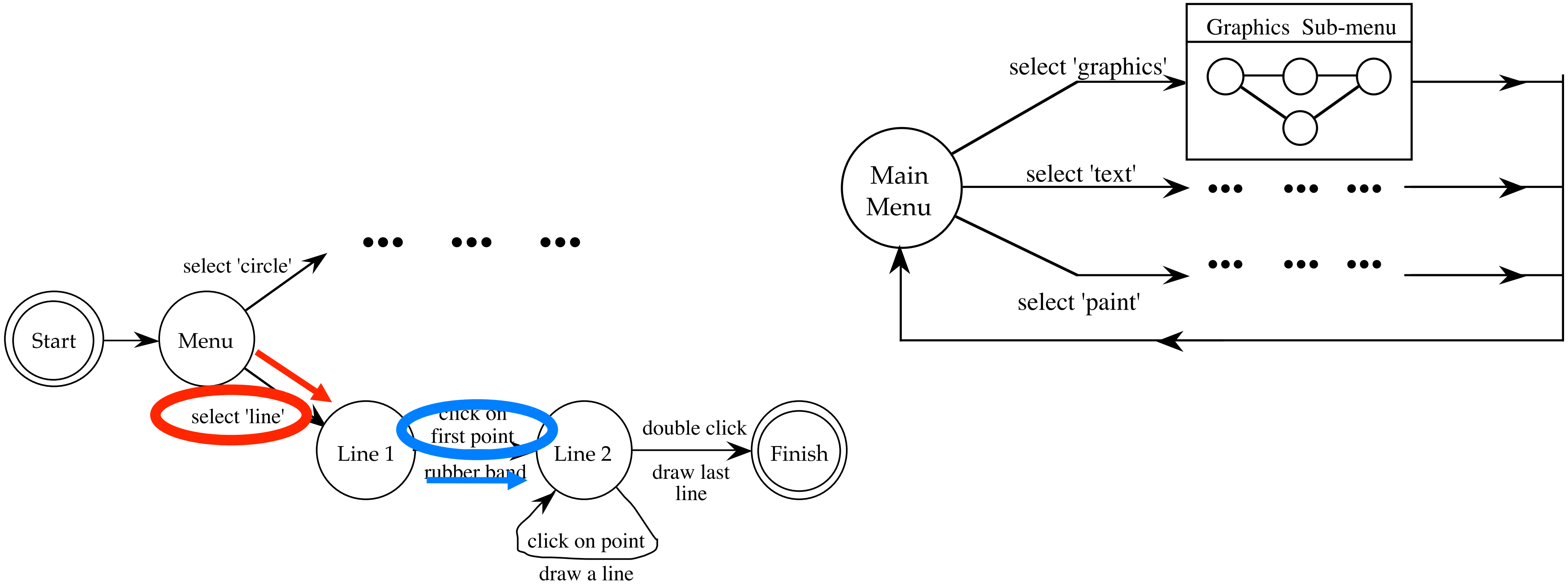
- Reversibility:
  - To reverse “select line” from graphics menu





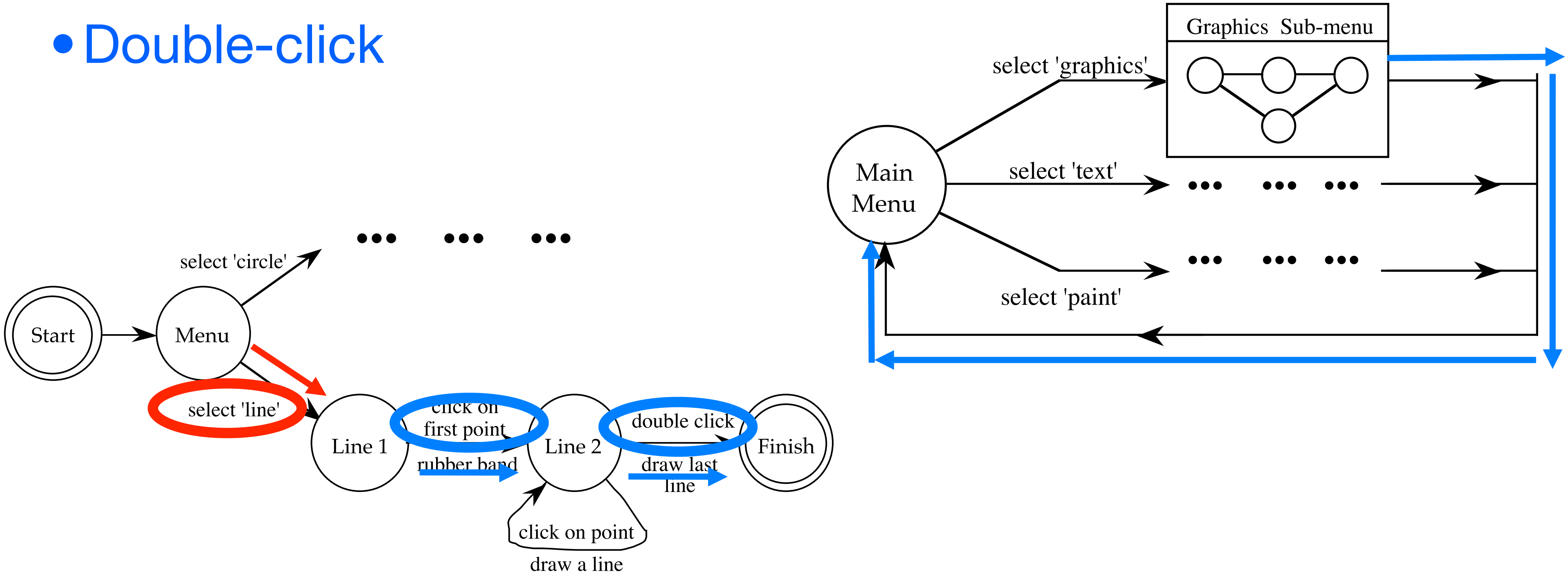
Checking Properties

- Reversibility:
  - To reverse “select line” from graphics menu
  - Click



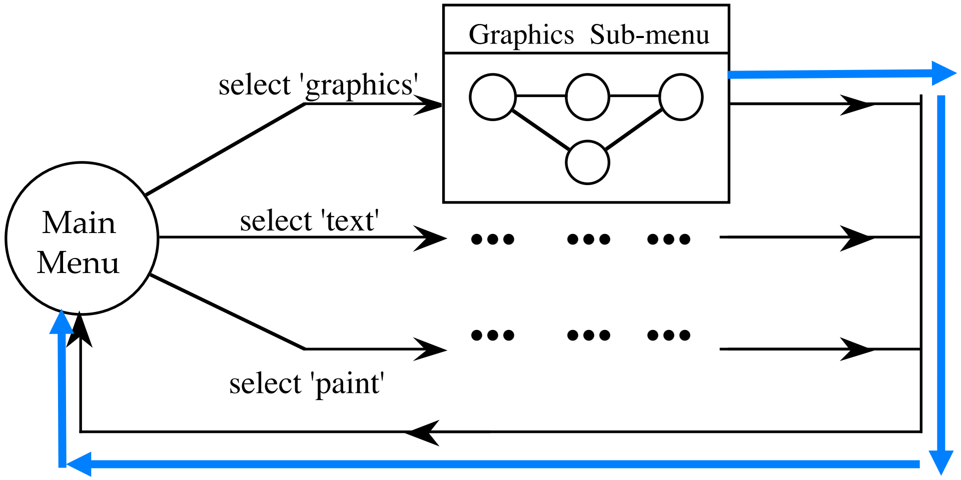
Checking Properties

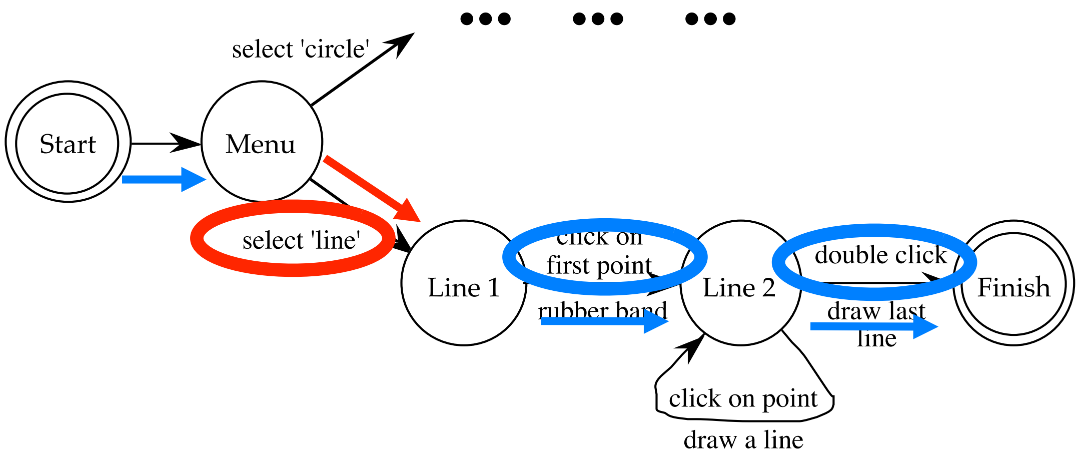
- Reversibility:
  - To reverse “select line” from graphics menu
  - Click
  - Double-click



Checking Properties

- Reversibility:
  - To reverse “select line” from graphics menu
  - Click
  - Double-click
  - Select Graphics





3 actions!

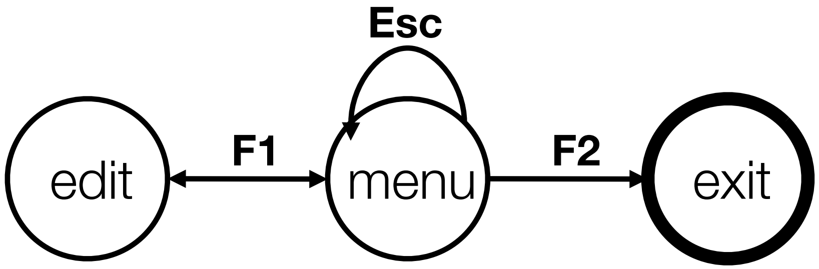
Note: this is not an undo.

State Properties

- Reachability
  - Can you get anywhere from anywhere else?
  - And how easily?
  - Basic check -- fully connected STN
  - More -- “infinite loops”
- Reversibility
  - Can you get to the previous state?
  - But NOT undo
- Dangerous States
  - Some states you don’t want to get to too easily.

Dangerous States Example

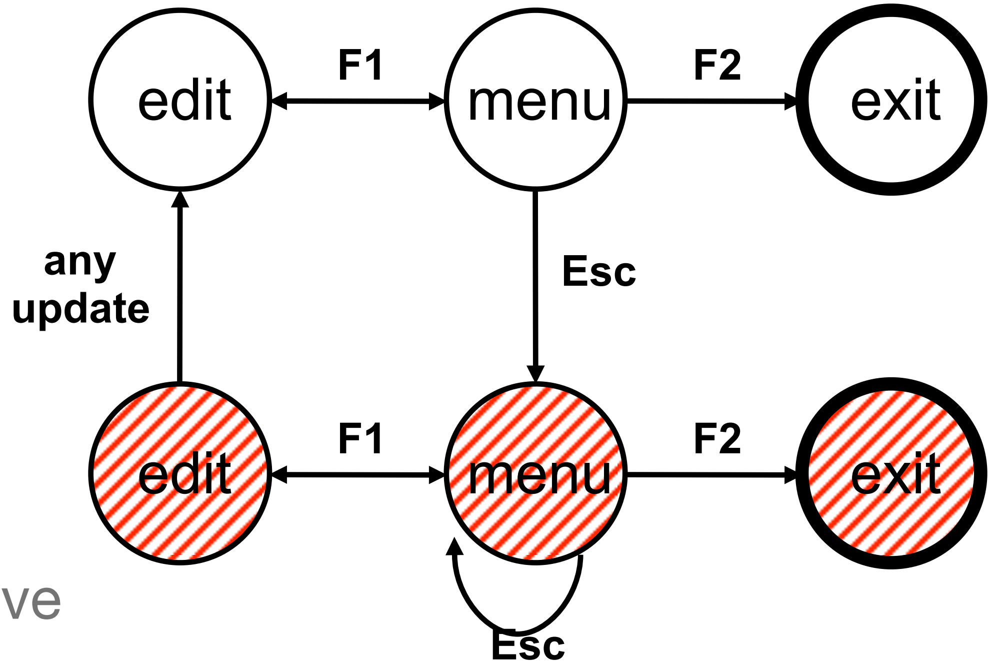
- Word Processor: Two Modes, edit and command (just like vi, emacs)
  - F1: Toggles mode
  - F2: Exit (and save)
  - Esc: No mode changes
  - But — Esc also resets autosave.



Dangerous States (ii)

- Exit with/without save ⇒ dangerous states

- Duplicate states - semantic distinction

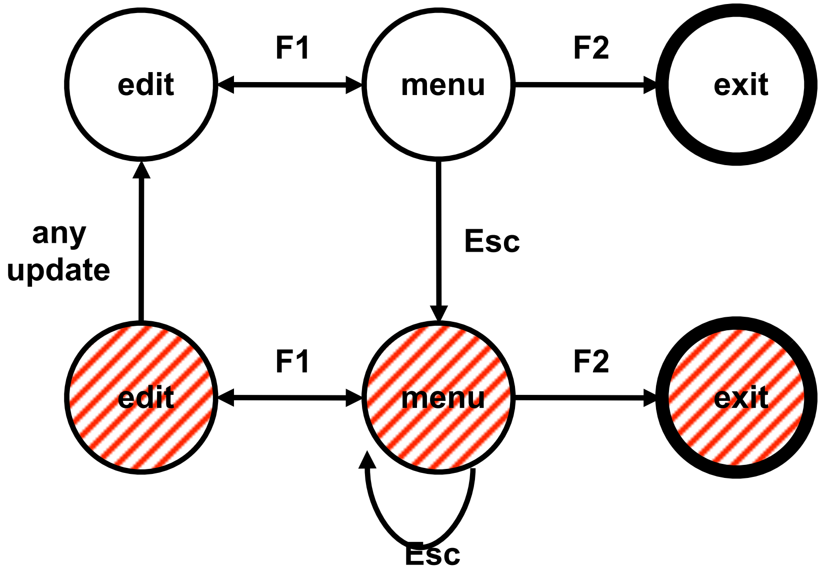


F1-F2 - exit with save

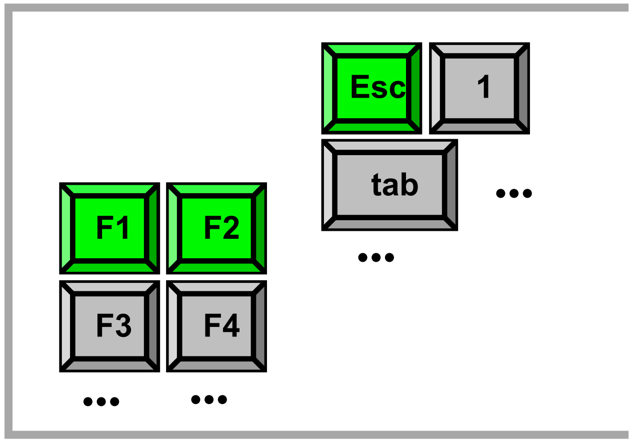
F1-Esc-F2 - exit with no save

Layout Matters

- word processor - dangerous states

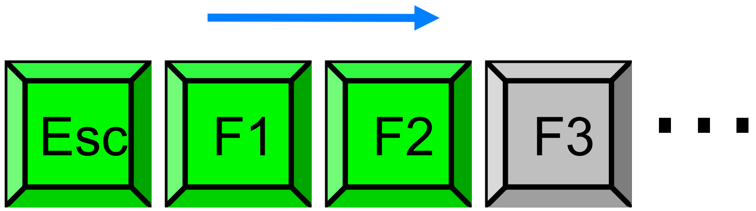


- old keyboard - OK



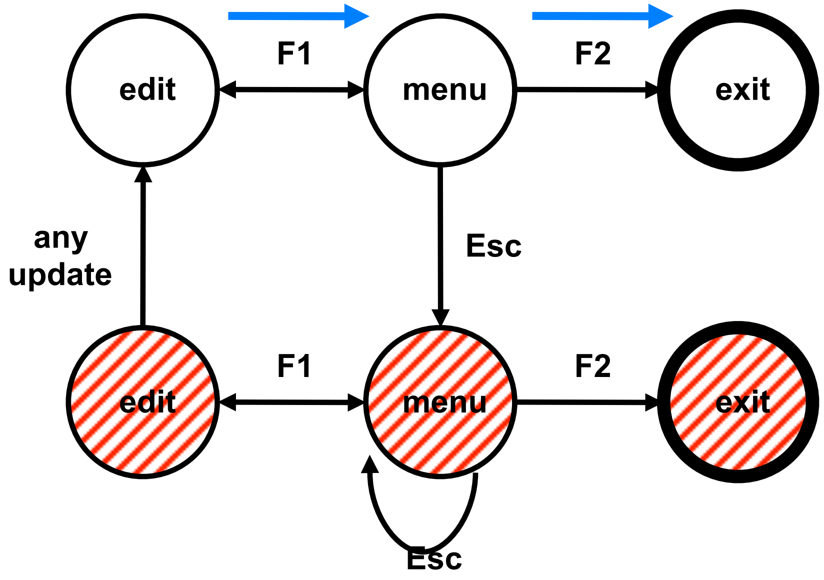
Layout Matters

- new keyboard layout



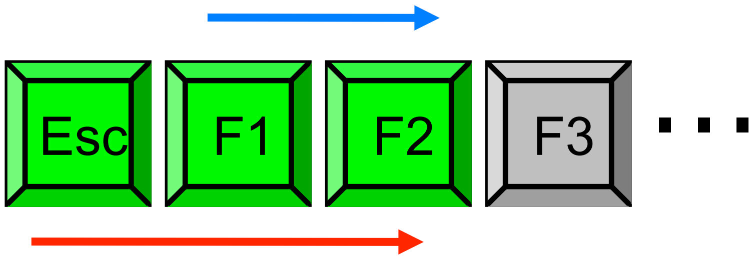
intend F1-F2 (save)

finger catches Esc



Layout Matters

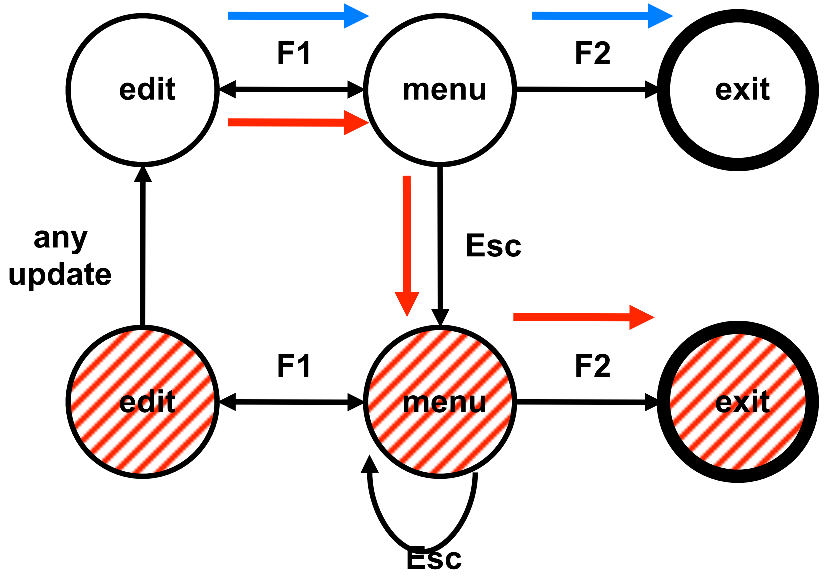
- new keyboard layout



intend F1-F2 (save)

finger catches Esc

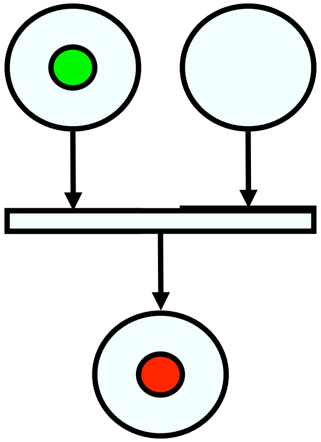
F1-Esc-F2 - disaster!



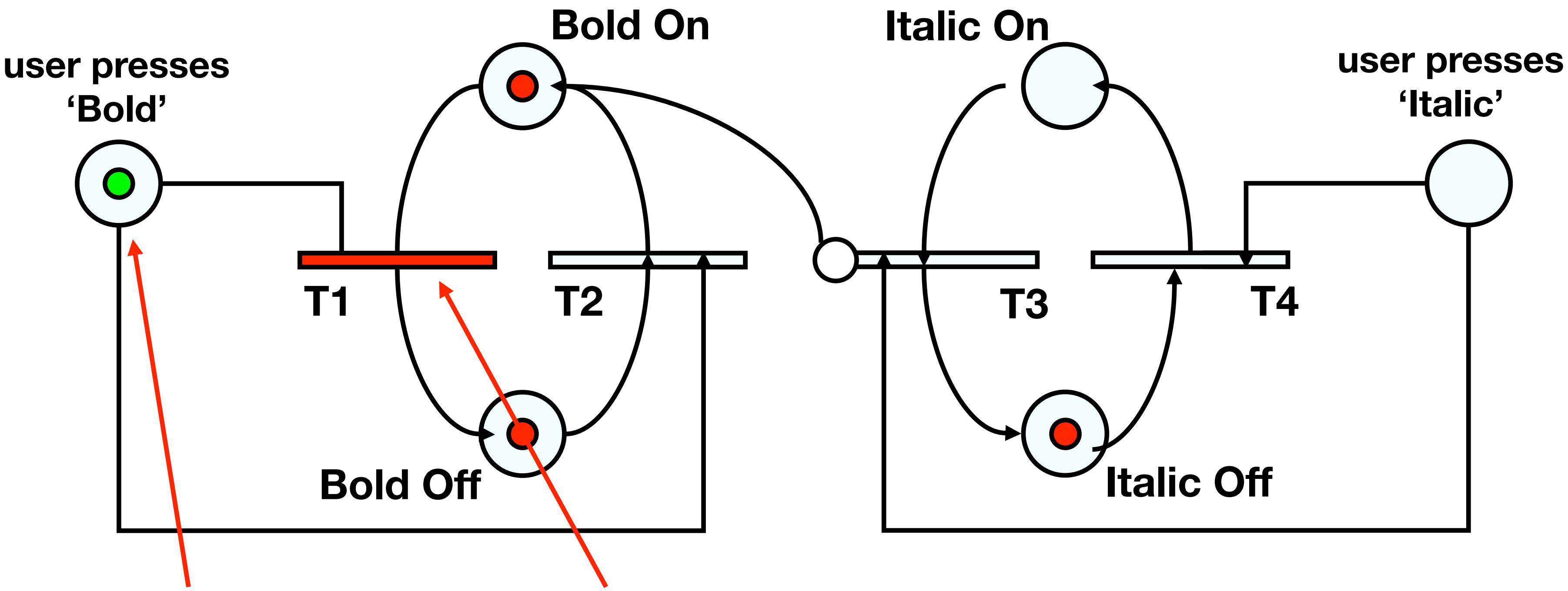
Other kinds of notations

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 are allowed for concurrent dialogue states.
- Used for UI specification



Petri Nets

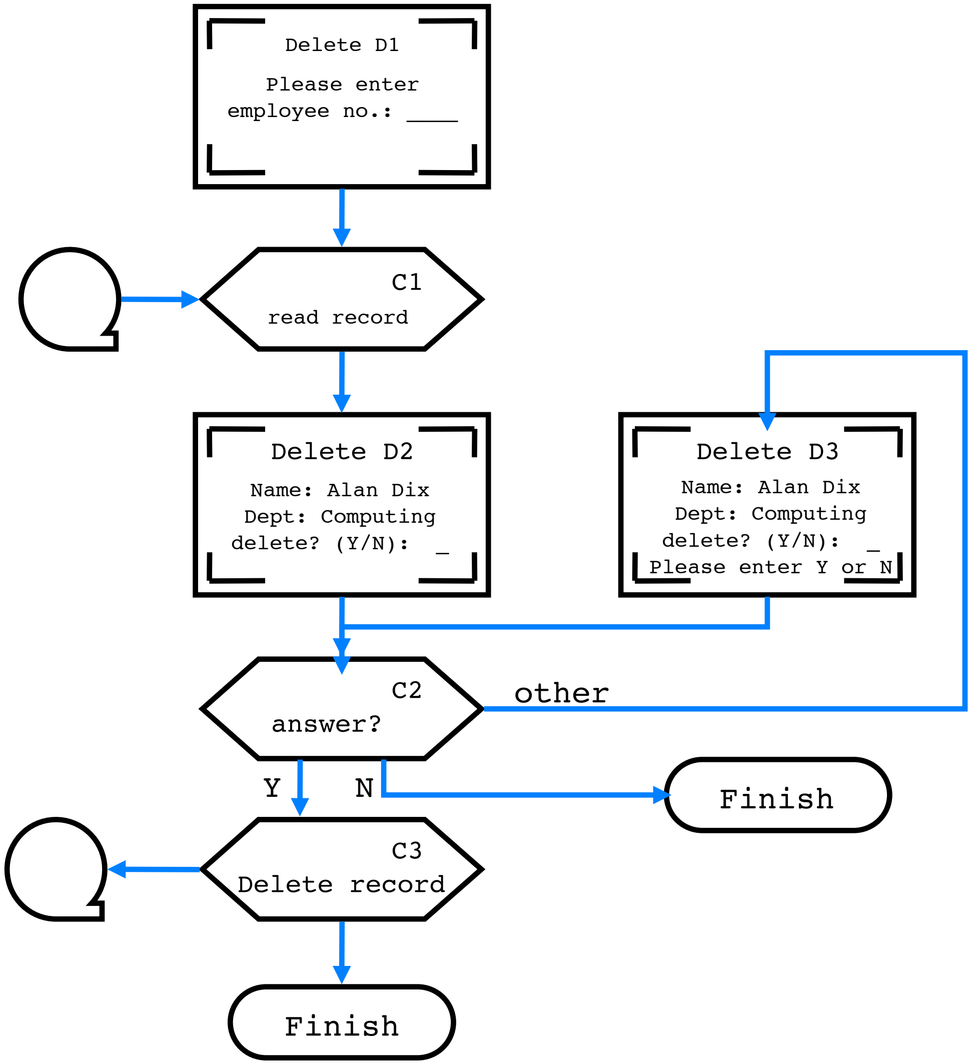


user actions represented as a new counter

transition 'fires' when all input places have counters

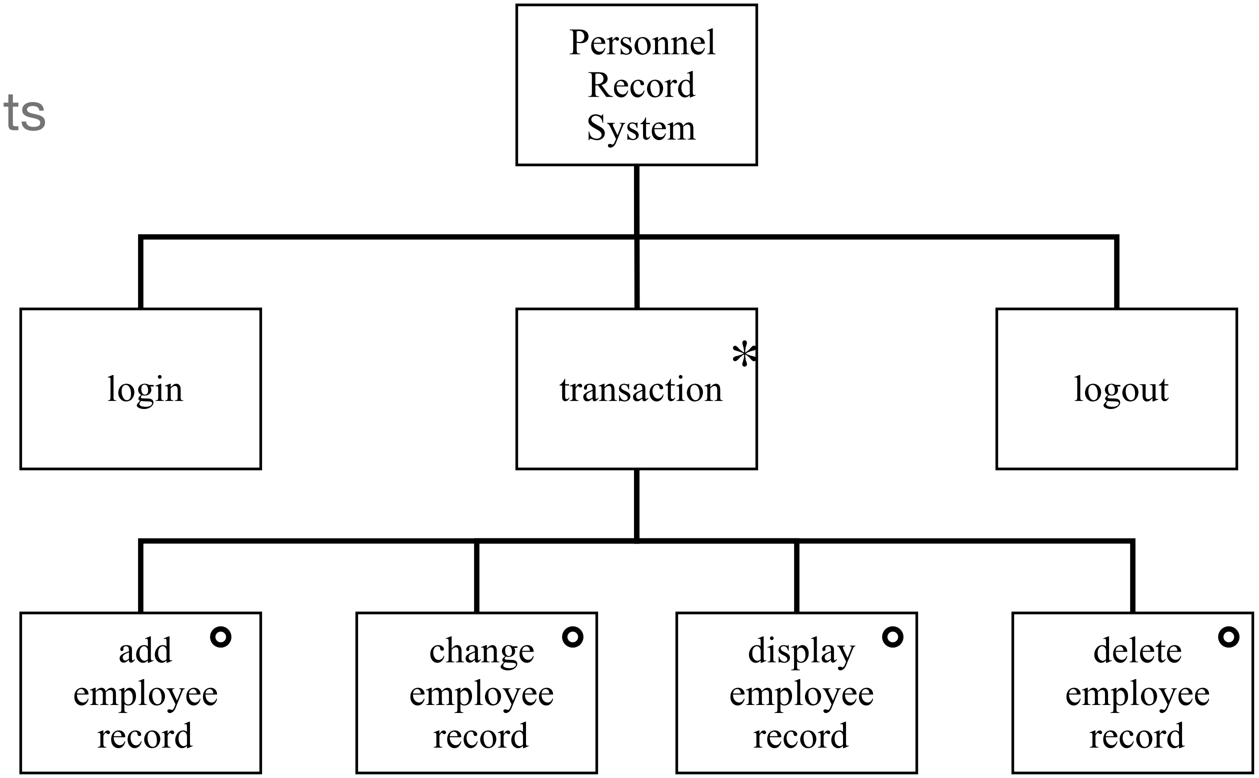
Flowcharts

- Familiar to programmers
- Boxes — Process/event (not state)



Jackson Structured Design (JSD) Diagrams

- Hierarchical Task Analysis + Dialog Design
- For tree-structured dialogs
  - o -- optional elements
  - * -- iteration
- Less expressive
- Greater clarity



Textual — Grammars

- Regular Expressions

  ```
  sel-line click click* dble-click
  ```

  - Same computational cost as STNs
  - Mainly deals with sequential ordering of tokens.
  - Uses operators to capture patterns:
    - +: one or more
    - ?: zero or one
    - *: zero or more
  - Examples:
    - The UNIX copy command: cp filename+ directory

Grammars

- BNF
  - symbol ::= expression

```
expr ::= empty
              | atom expr
              | ‘(‘ expr ‘)’ sentence
```

  - More powerful than regular expressions or STNs
  - Still cannot handle concurrent dialogs

Dialog Notations: Summary

- Diagrammatic
  - STN, JSD, Flowcharts, etc
- Textual
  - BNF, regular expressions, etc
- Some notations essentially equivalent, some more expressive
- Issues
  - Event-based vs. State-based
  - Power vs. Clarity
  - Model vs. Notation
  - Sequential vs. Concurrent