Human Computer Interaction

Dialog Notations and Design

Lecture 13

Imran Siddiqi
imran.siddiqi@gmail.com

Dialog

- Conversation between two or more parties

Dialog

- Conversation between two or more parties

Dialog – User Interface Design

- More specific meaning
- **Structure** of the conversation between user and computer
- Levels of computer language
  - Lexical – Lowest level: Shape of icons, actual keys pressed
  - Syntactic – Order of inputs and outputs
  - Semantic – Meaning of conversation, Effect on internal application/data
- In user Interfaces
  - Dialog generally refers to the syntactic level of human–computer ‘conversation’

Structured Human Dialog

- 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
Woman: With this ring I thee wed
Minister: I now pronounce you man and wife

Lessons from Dialog

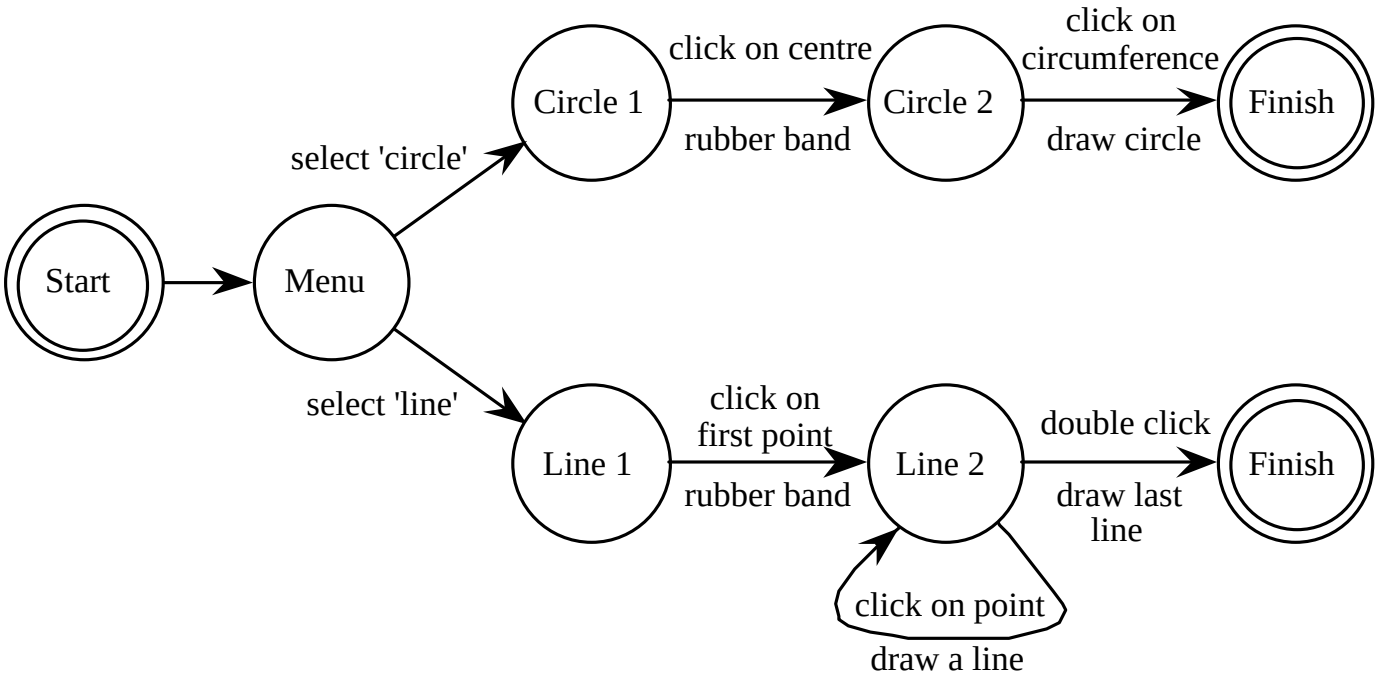
- Wedding service
  - Sort of script for three parties
  - Participants must say certain things in a specifies order
  - Some contributions are predetermined – “I do”
  - Others variable – “do you *man’s name* …”
- If you say these words are you married?
  - Only if in the right place, with marriage licence
  - Syntax not semantics – marriage may be acted out on television

Graphical Dialog Design Notations

State-transition nets (STN)
Petri nets, State charts
Flow charts, JSD diagrams

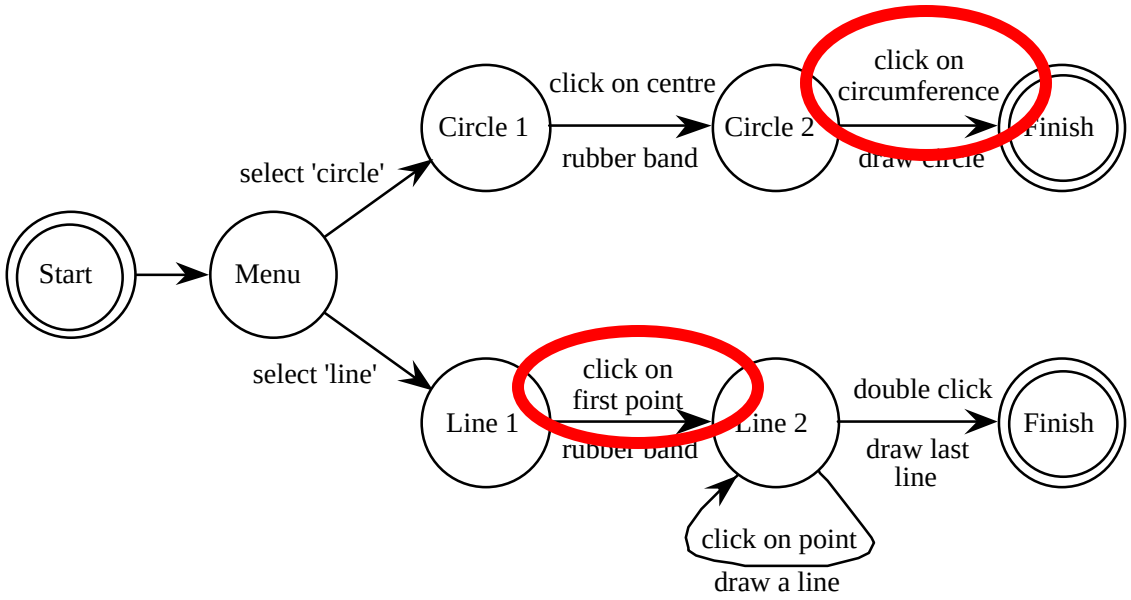
State Transition Networks (STN)

- Circles – states
- Arcs - actions/events



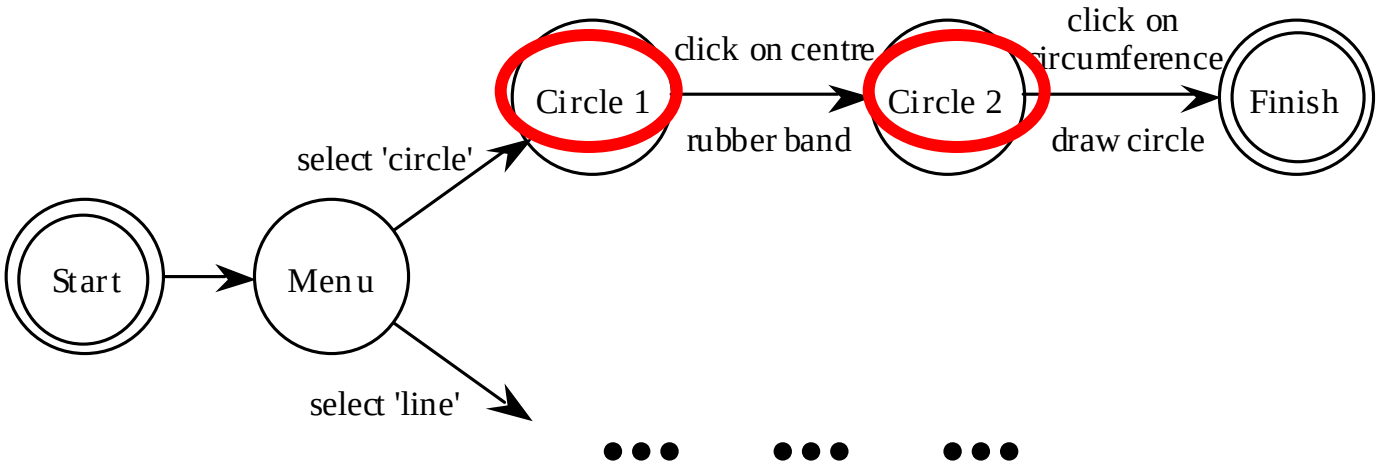
State Transition Networks – Actions/Events

- Arc labels a bit cramped because:
  - The actions/events require most detail



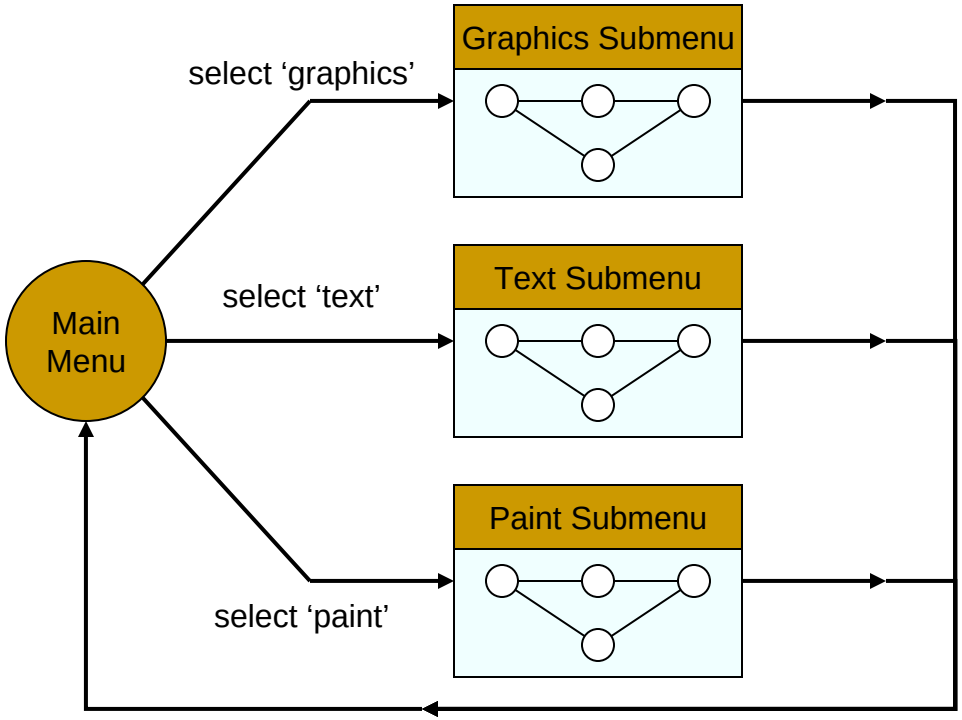
State Transition Networks - States

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



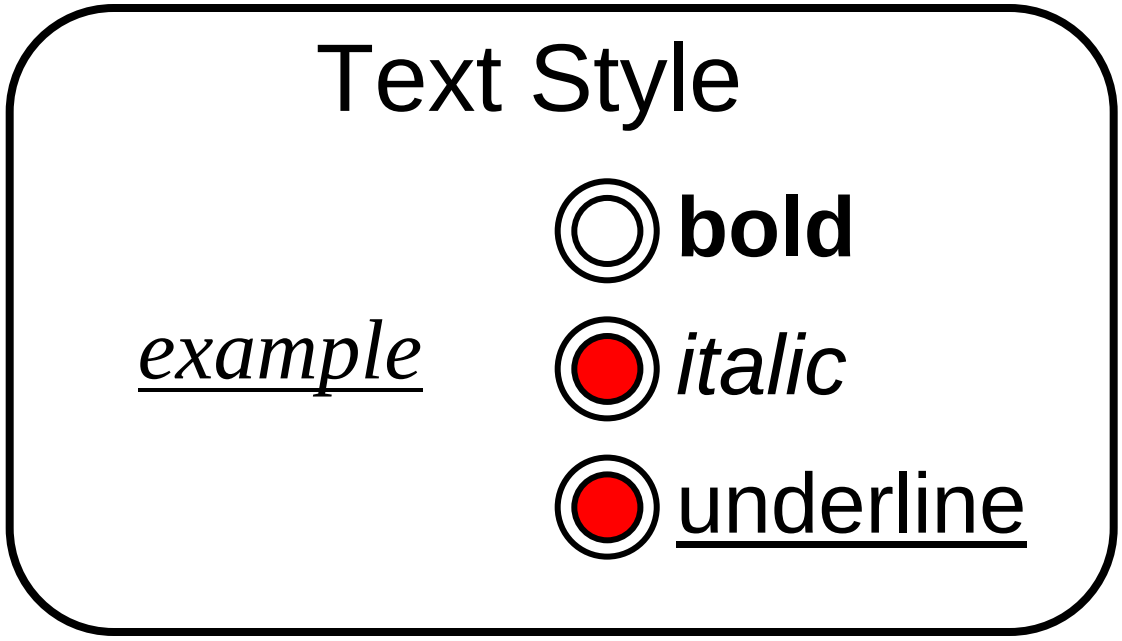
Hierarchical STNs

- Managing complex dialogues
- Named sub-dialogues



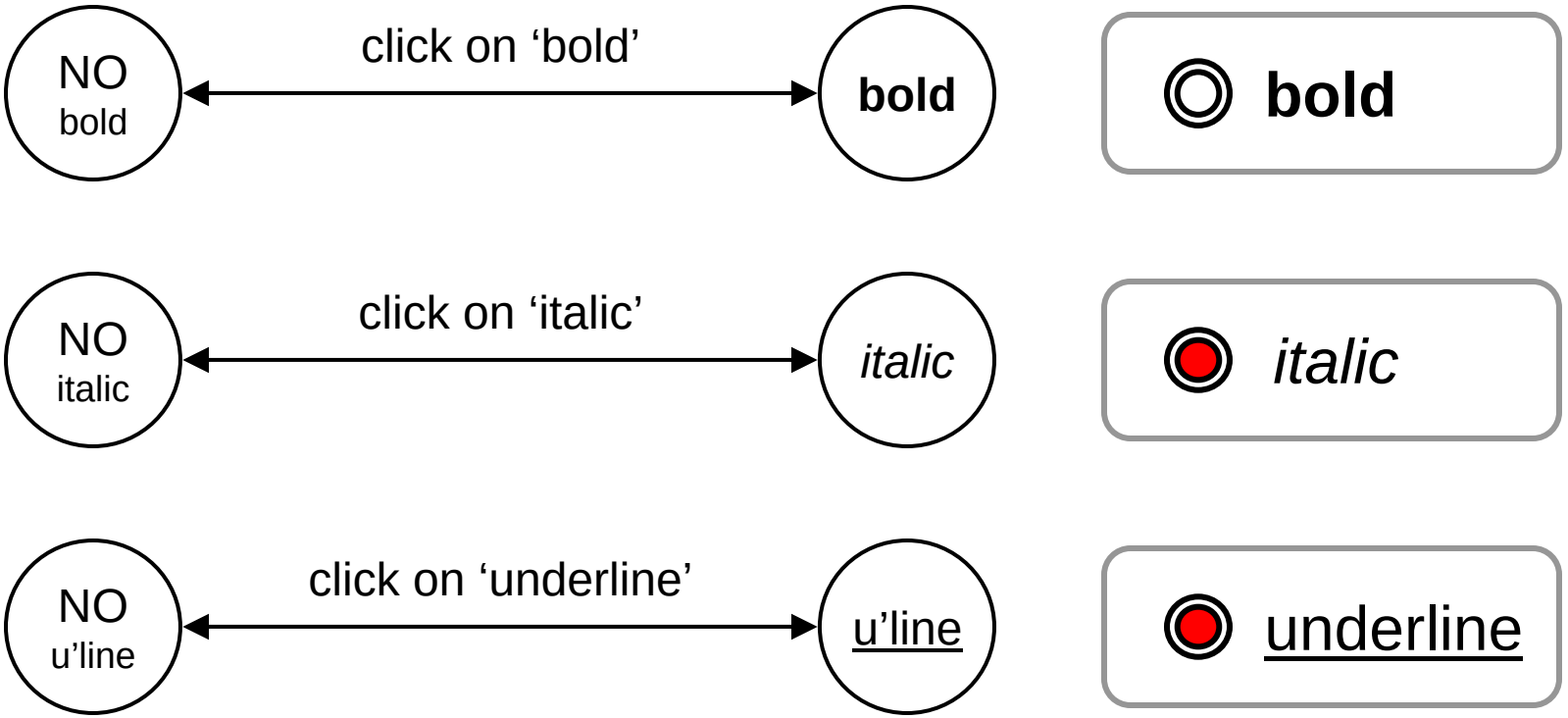
Concurrent Dialogs

- Simple Dialog Box



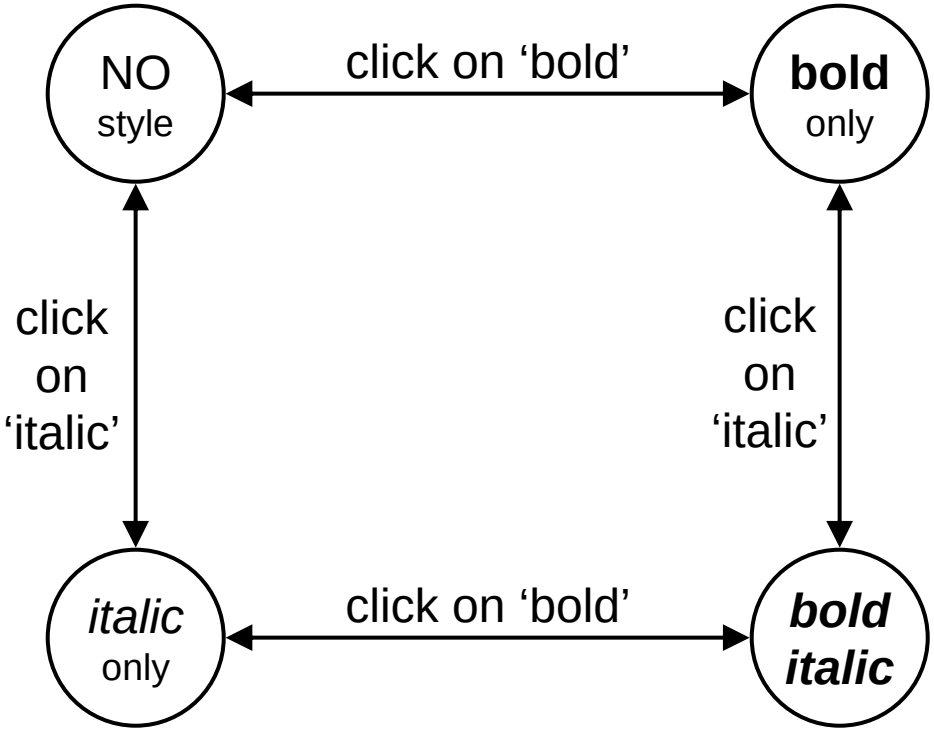
Concurrent Dialogs

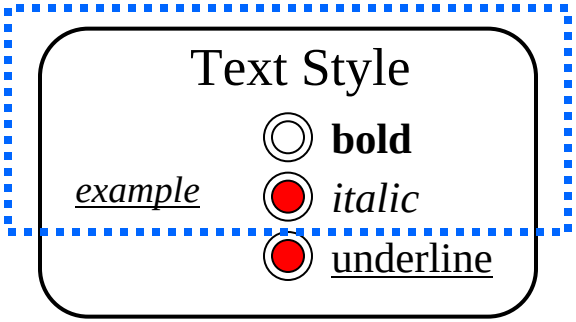
- Three toggles - individual STNs



Concurrent Dialogs

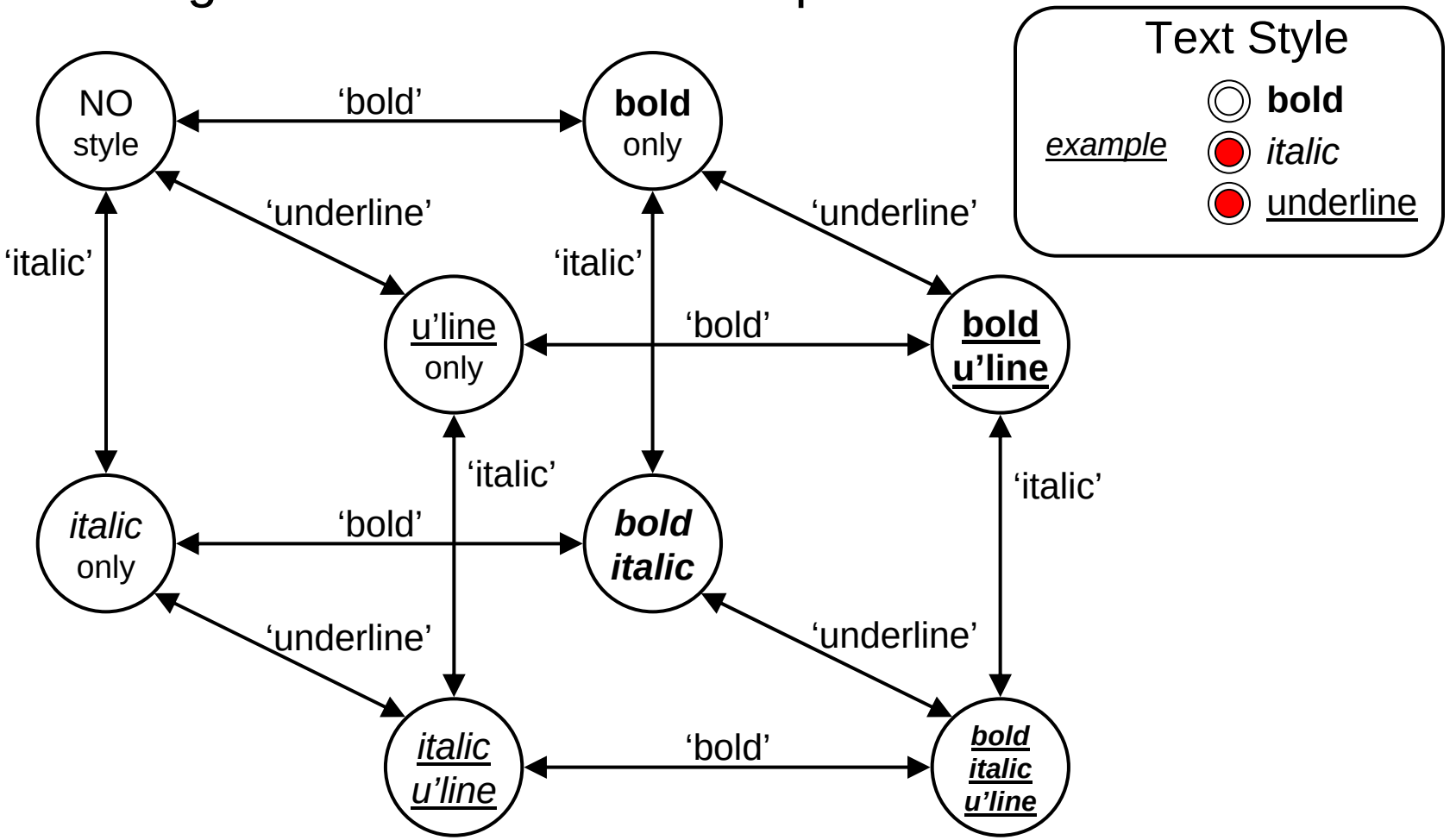
- Bold and Italic Combined





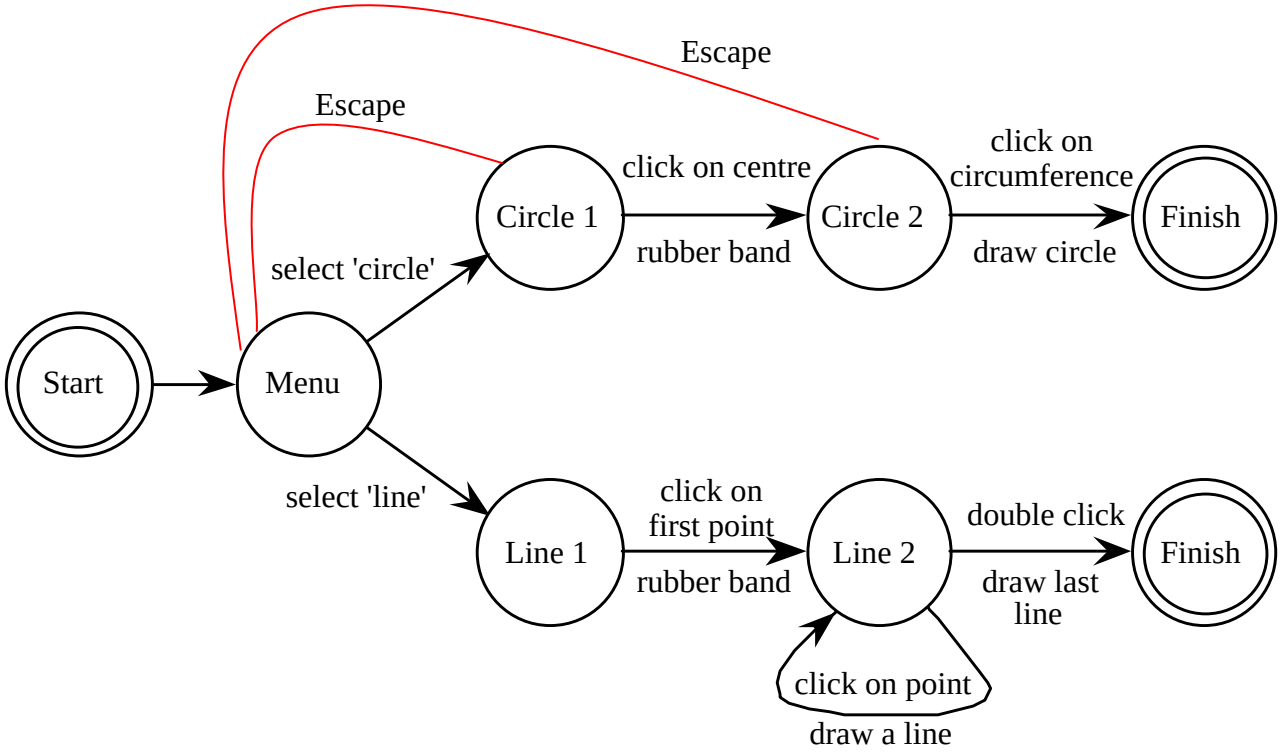
Concurrent Dialogs

- All together - combinatorial explosion



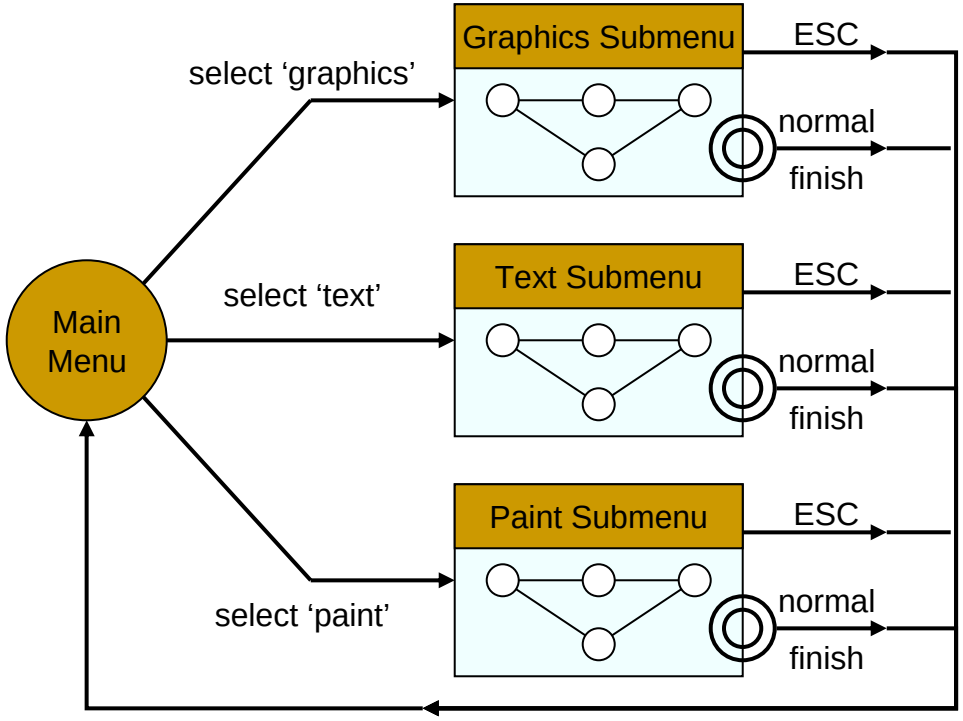
Escapes

- Escape/cancel keys
  - End up with spaghetti of identical behaviors



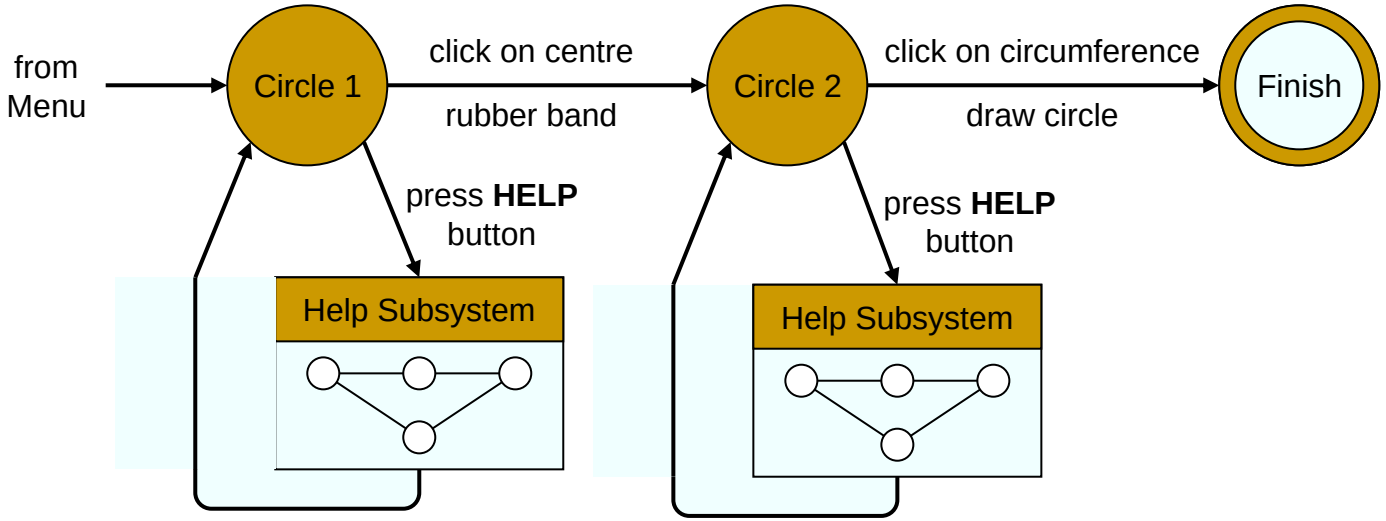
Escapes

- Proposition
  - 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

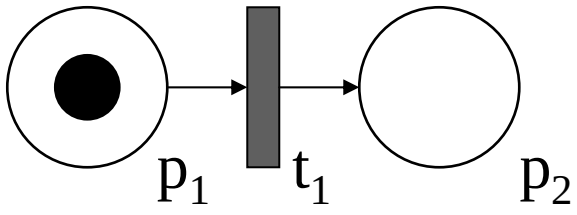


Petri Nets

- One of the oldest notations in computing!
- Flow graph:
  - Places – a bit like STN states
  - Transitions – a bit like STN arcs
  - Counters/Tokens – sit on places (current state)

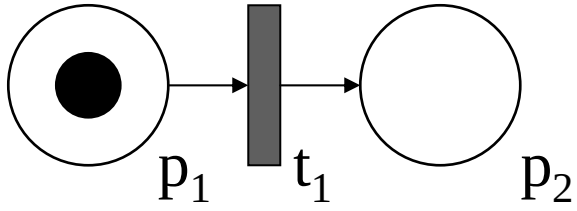
Petri Nets - Basics

- Petri net consist two types of nodes: *places* and *transitions*. An arc exists only from a place to a transition or from a transition to a place.
- A place may have zero or more *tokens*.
- Graphically, places, transitions, arcs, and tokens are represented respectively by: circles, bars, arrows, and dots.



Petri Nets - Basics

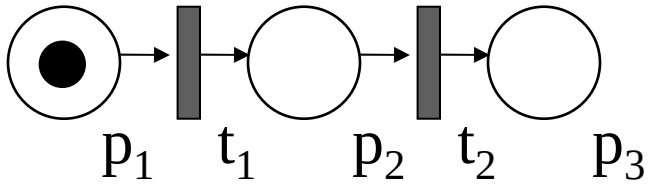
- Below is an example Petri net with two places and one transition.
- Transition node is ready to ***fire*** if and only if there is at least one token at each of its input places



- State transition of form (1, 0) → (0, 1)
- P1:Input place - P2:Output place

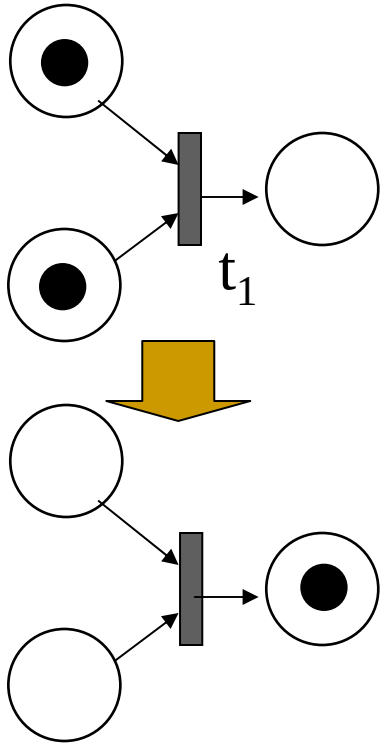
Petri Nets - Properties

- Sequential Execution
  - Transition t2 can fire only after the firing of t1. This impose the precedence of constraints "t2 after t1.“



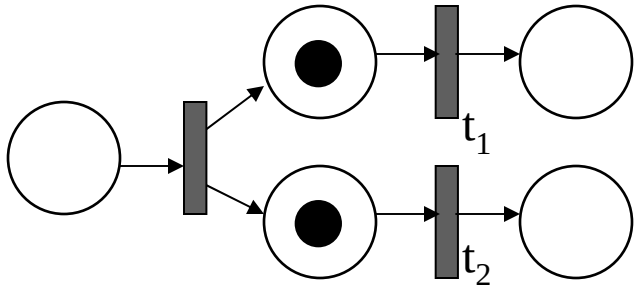
Petri Nets - Properties

- Synchronization
  - Transition t1 will be enabled only when there is at least one token at each of its input places.
- Merging
  - Happens when tokens from several places arrive for service at the same transition.



Petri Nets - Properties

- Concurrency
  - t1 and t2 are concurrent. With this property, Petri net is able to model systems of distributed control with multiple processes executing concurrently in time

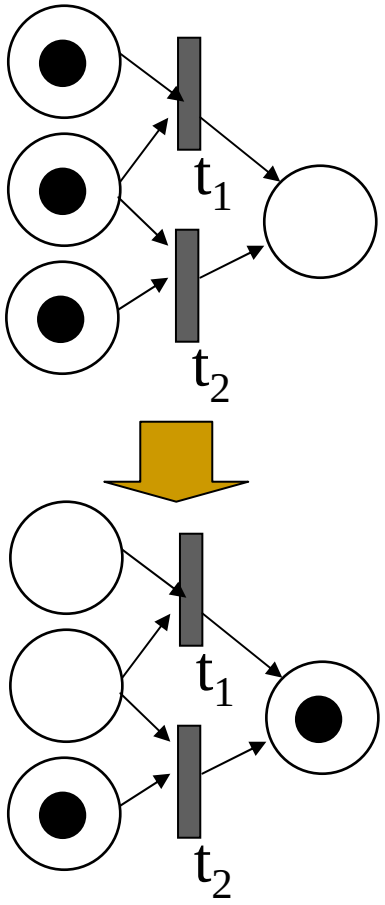


Petri Nets - Properties

- Conflict
  - t1 and t2 are both ready to fire

    but the firing of any leads to the

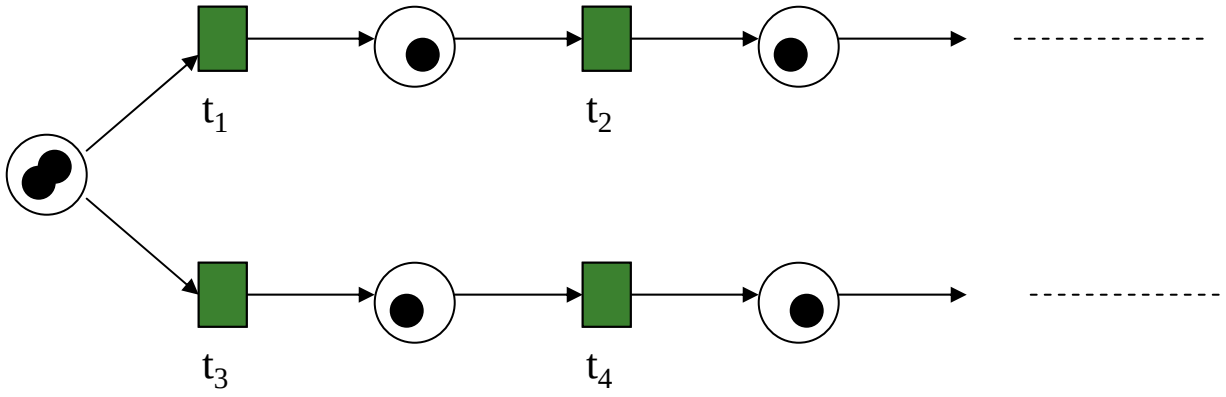
    disabling of the other transitions.



Petri Nets - Properties

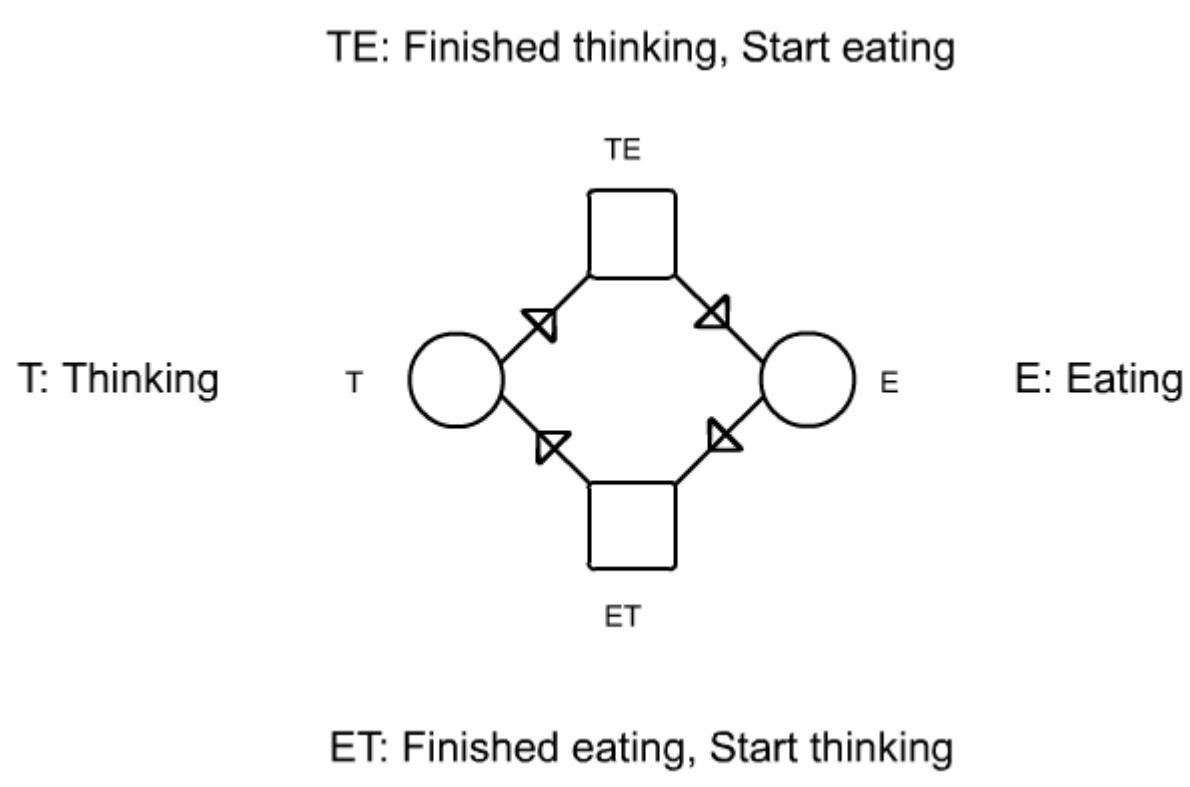
- Conflict (Contd…)
  - The resulting conflict may be resolved in a purely non-deterministic way or in a probabilistic way, by assigning appropriate probabilities to the conflicting transitions

There is a choice of either t1 and t2, or t3 and t4



Petri Nets - Examples

Petri Nets - Philosopher

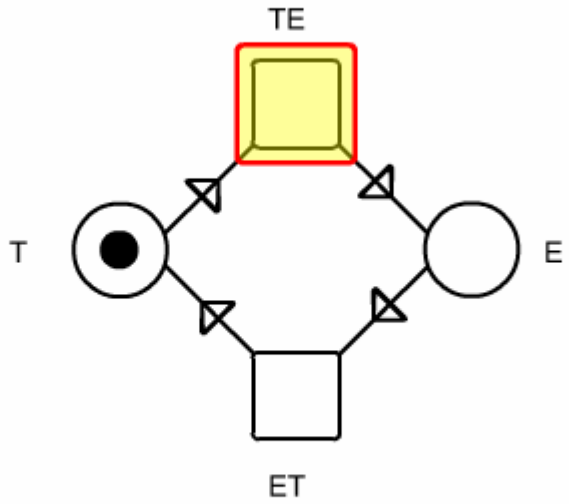


Model

Petri Nets - Philosopher

TE: Finished thinking, Start eating

T: Thinking

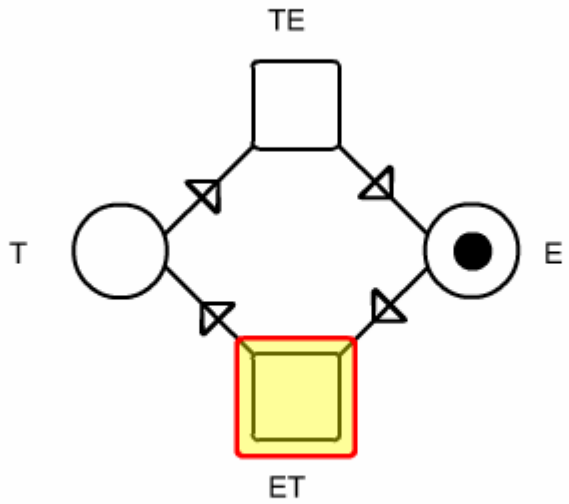


E: Eating

ET: Finished eating, Start thinking

Petri Nets - Philosopher

TE: Finished thinking, Start eating



T: Thinking

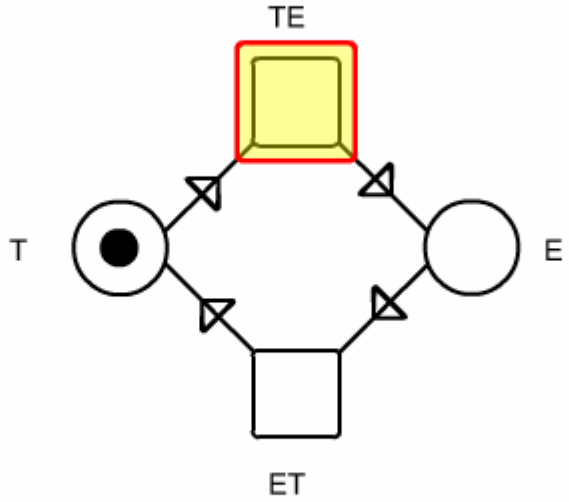
E: Eating

ET: Finished eating, Start thinking

Petri Nets - Philosopher

TE: Finished thinking, Start eating

T: Thinking

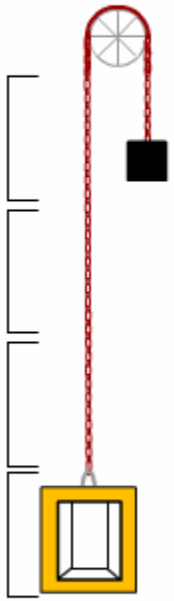


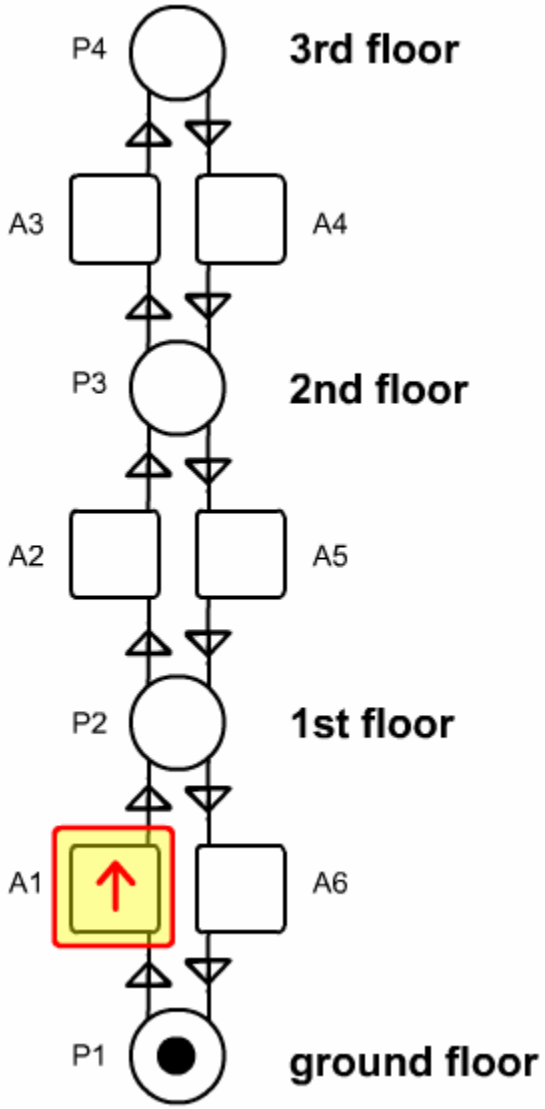
E: Eating

ET: Finished eating, Start thinking

Petri Nets - Elevator

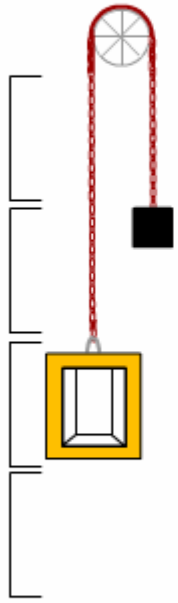
The token represents the elevator.

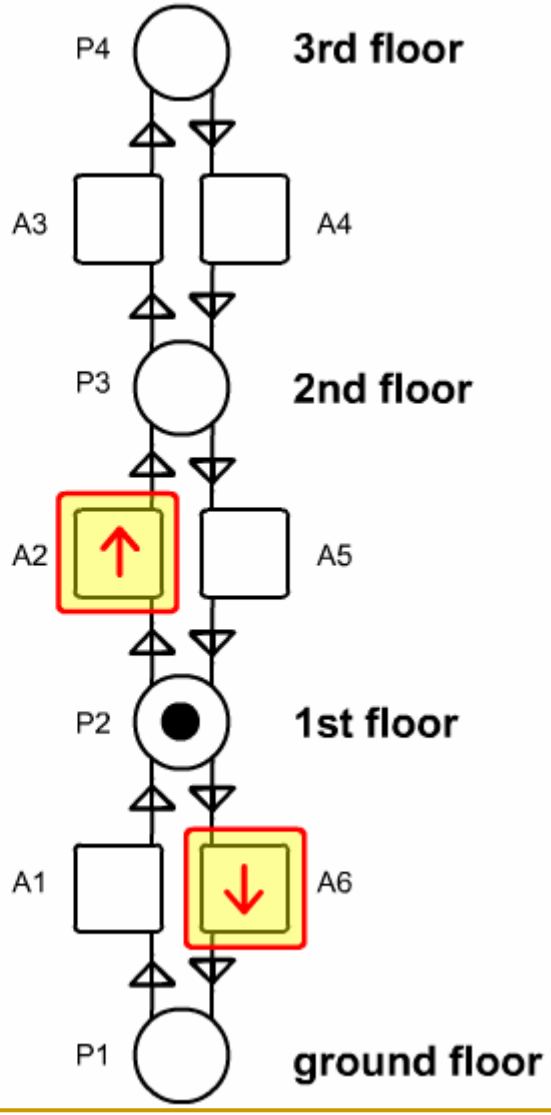




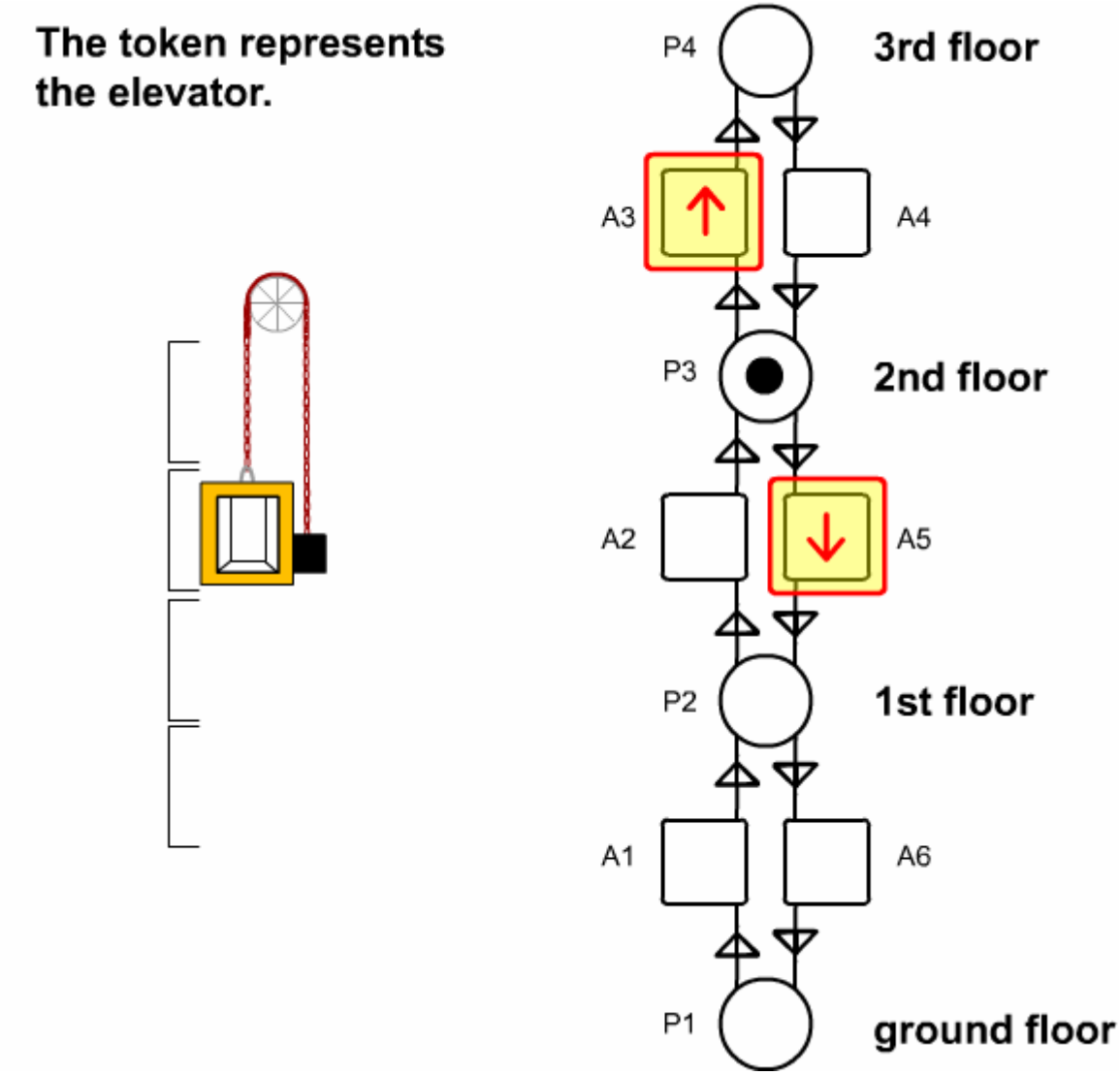
Petri Nets - Elevator

The token represents the elevator.

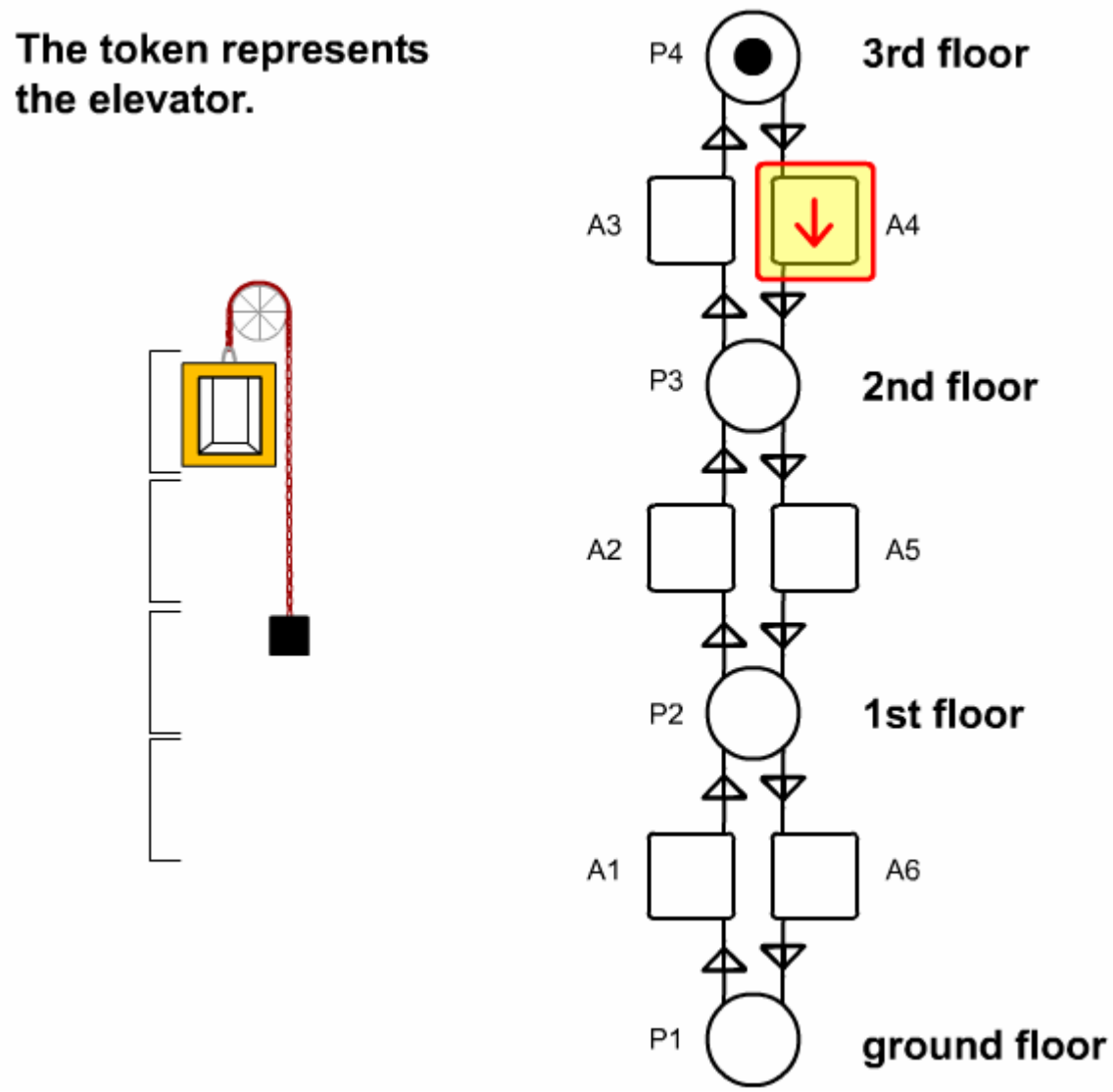




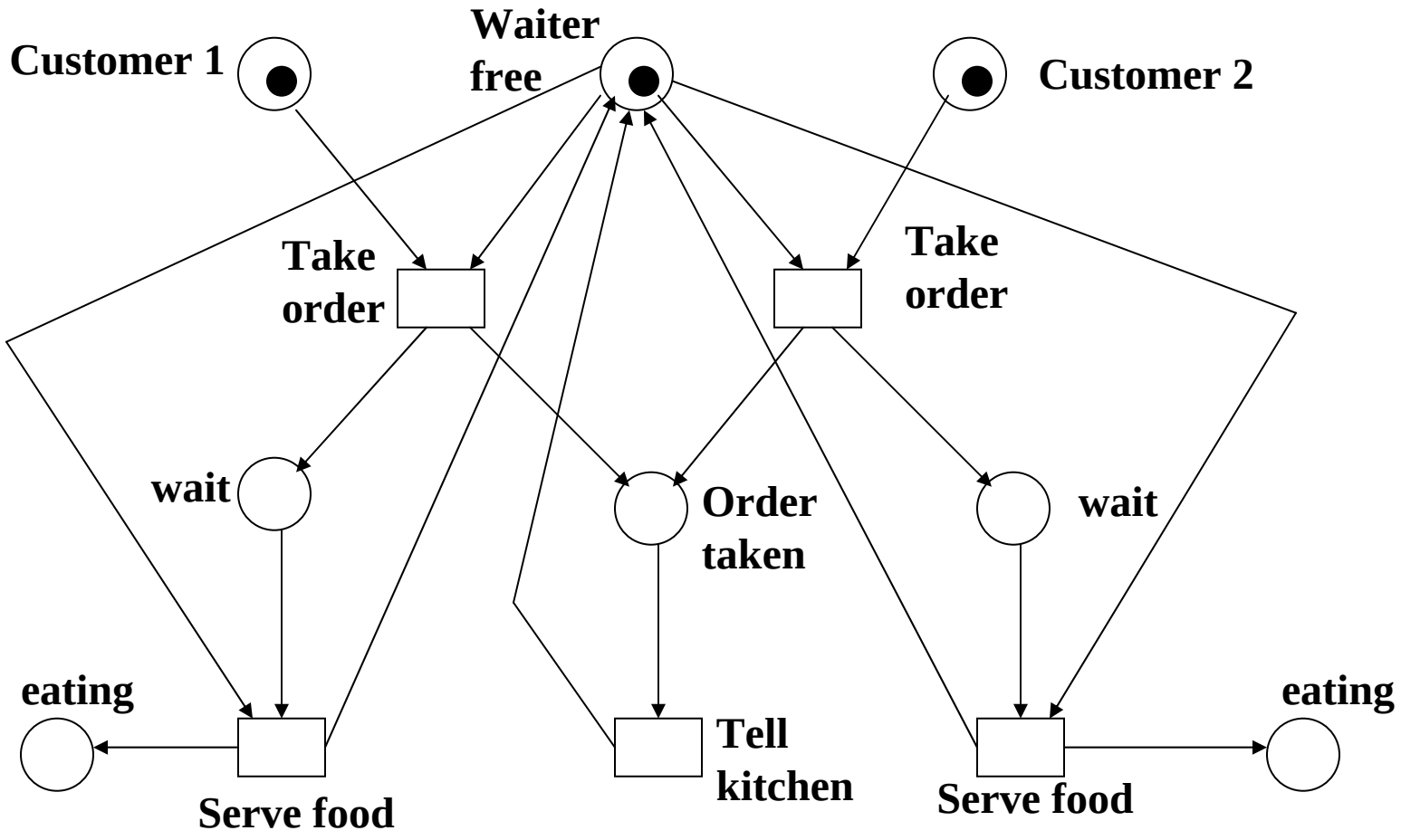
Petri Nets - Elevator



Petri Nets - Elevator



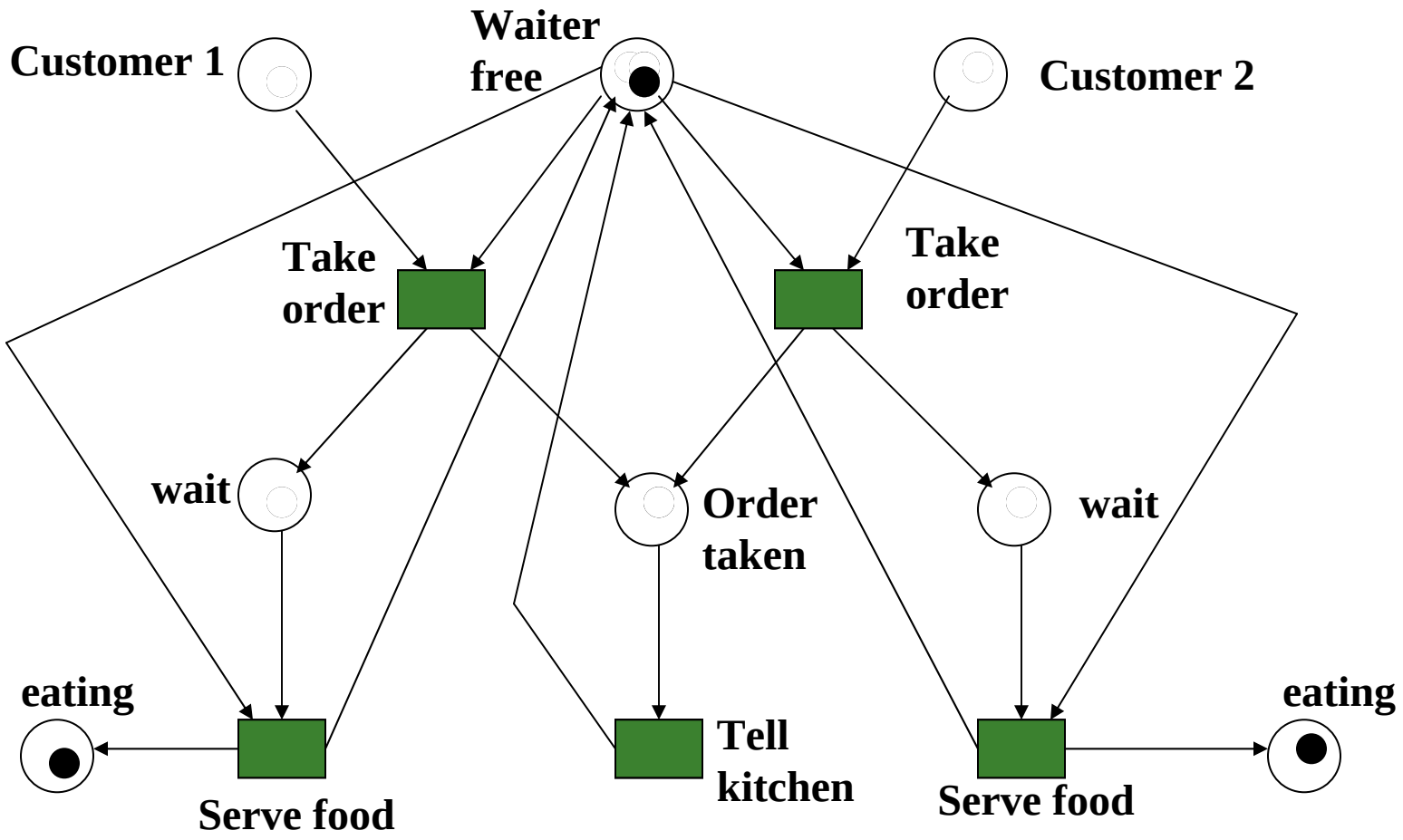
Petri Nets – In a Restaurant



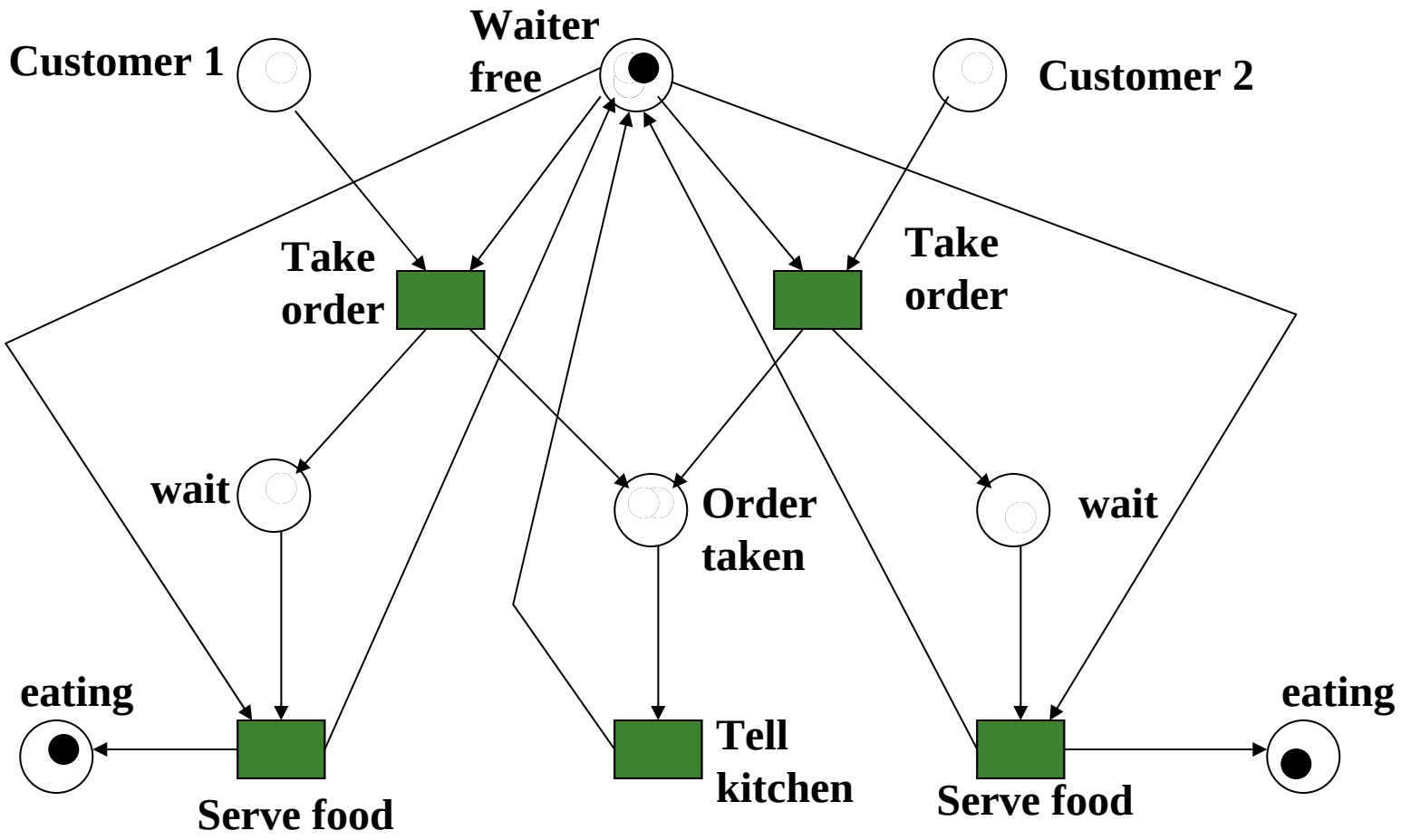
Petri Nets – In a Restaurant

- Scenario 1:
  - Waiter takes order from customer 1; serves customer 1; takes order from customer 2; serves customer 2.
- Scenario 2:
  - Waiter takes order from customer 1; takes order from customer 2; serves customer 2; serves customer 1.

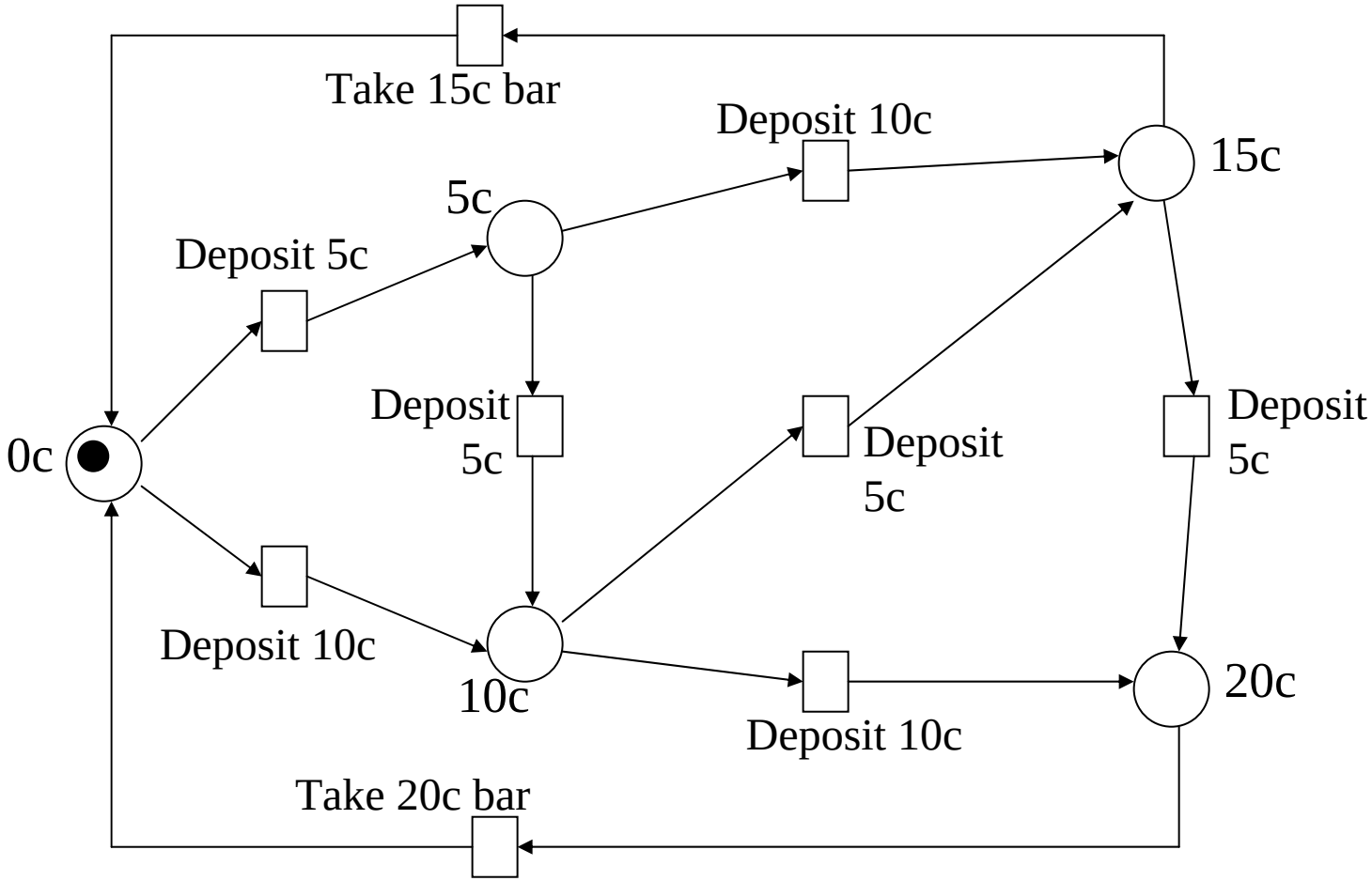
Petri Nets – In a Restaurant (Scenario 1)



Petri Nets – In a Restaurant (Scenario 2)



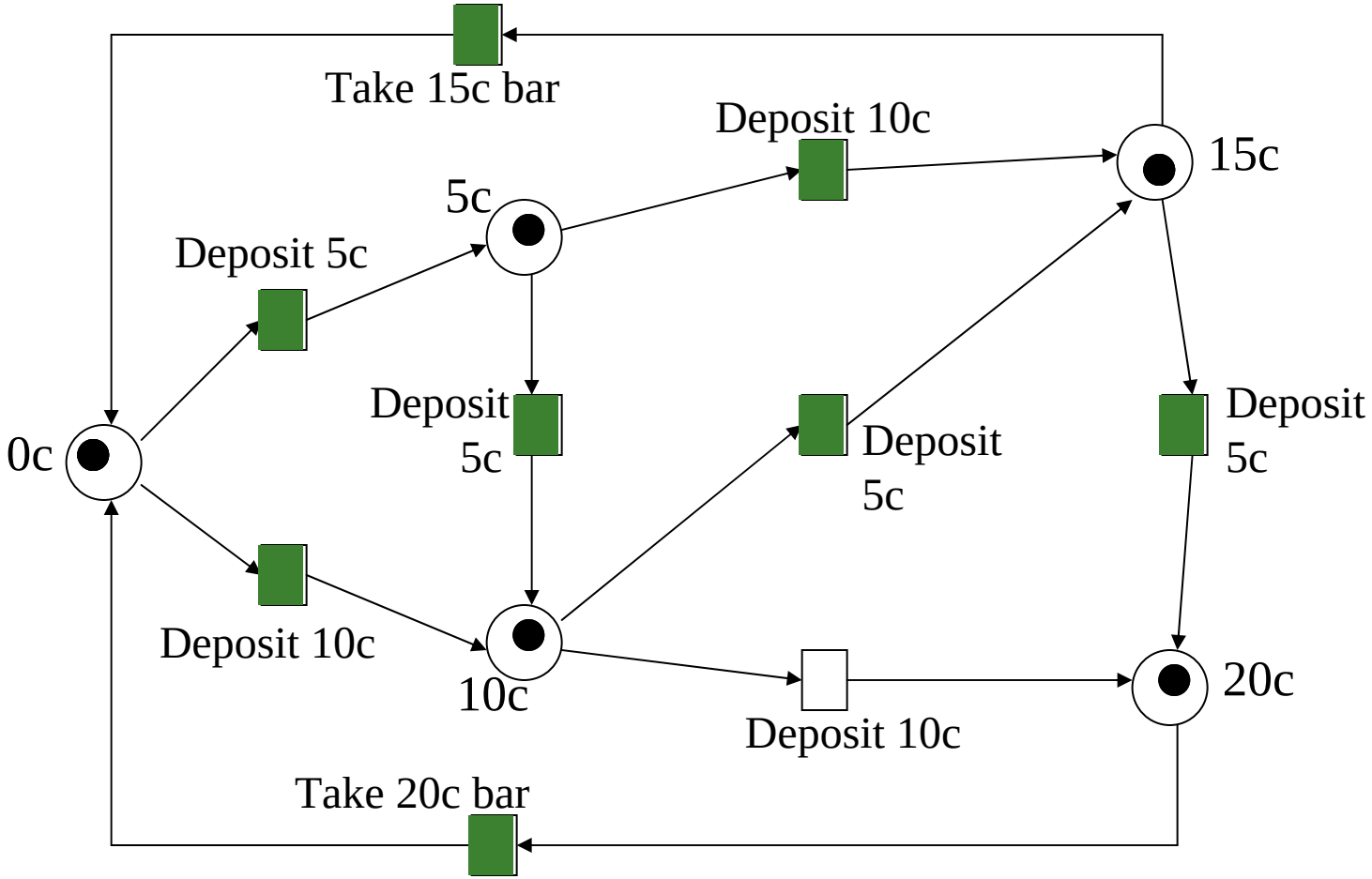
Petri Nets – Vending Machine



Petri Nets – Vending Machine

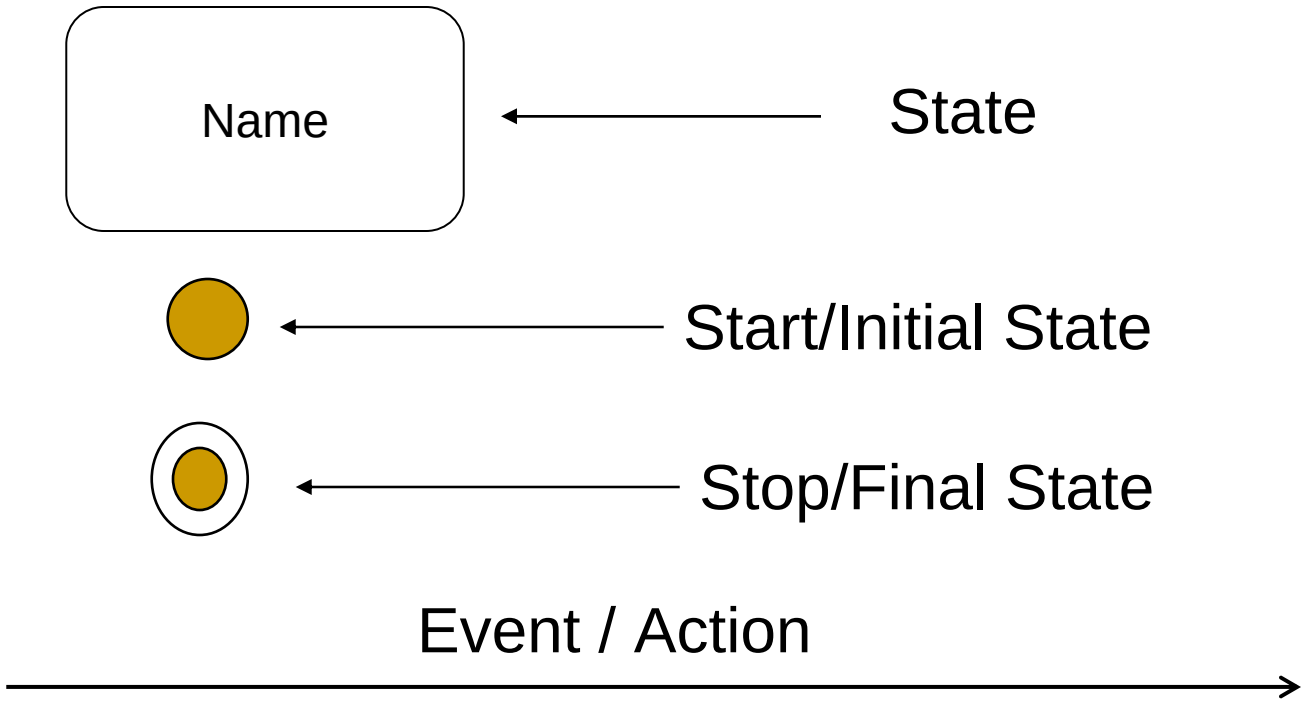
- Scenario 1:
  - Deposit 5c, deposit 5c, deposit 5c, deposit 5c, take 20c snack bar.
- Scenario 2:
  - Deposit 10c, deposit 5c, take 15c snack bar.
- Scenario 3:
  - Deposit 5c, deposit 10c, deposit 5c, take 20c snack bar.

Petri Nets – Vending Machine

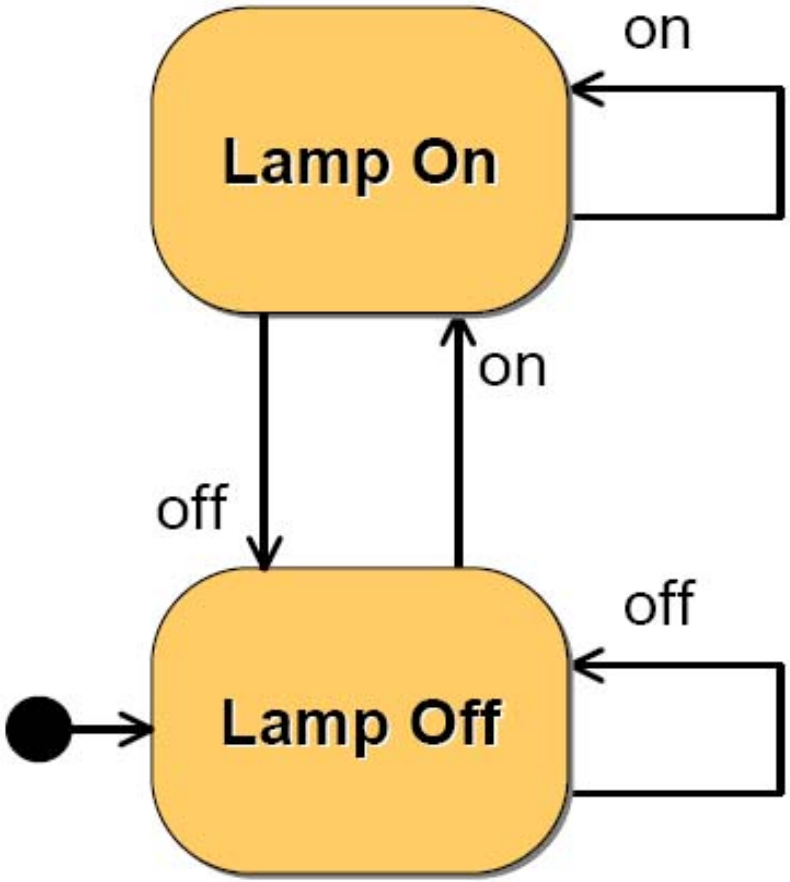


State Charts

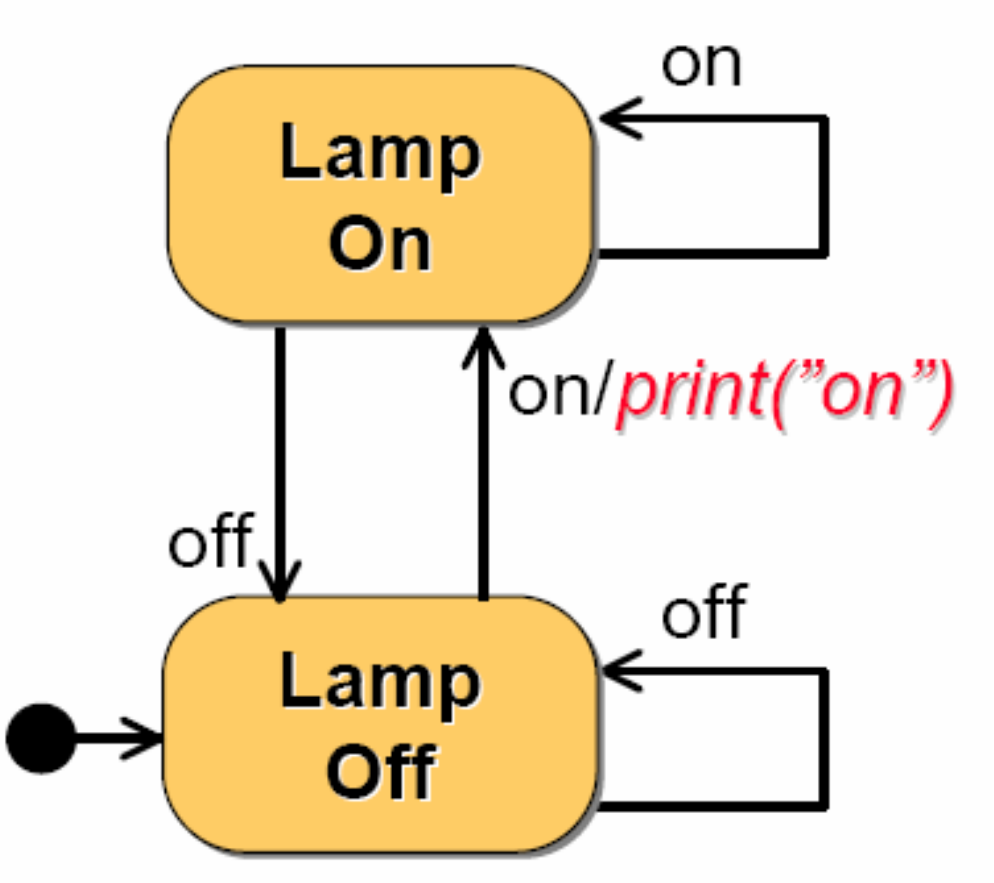
- Used in UML
- Extension to STN



State Charts

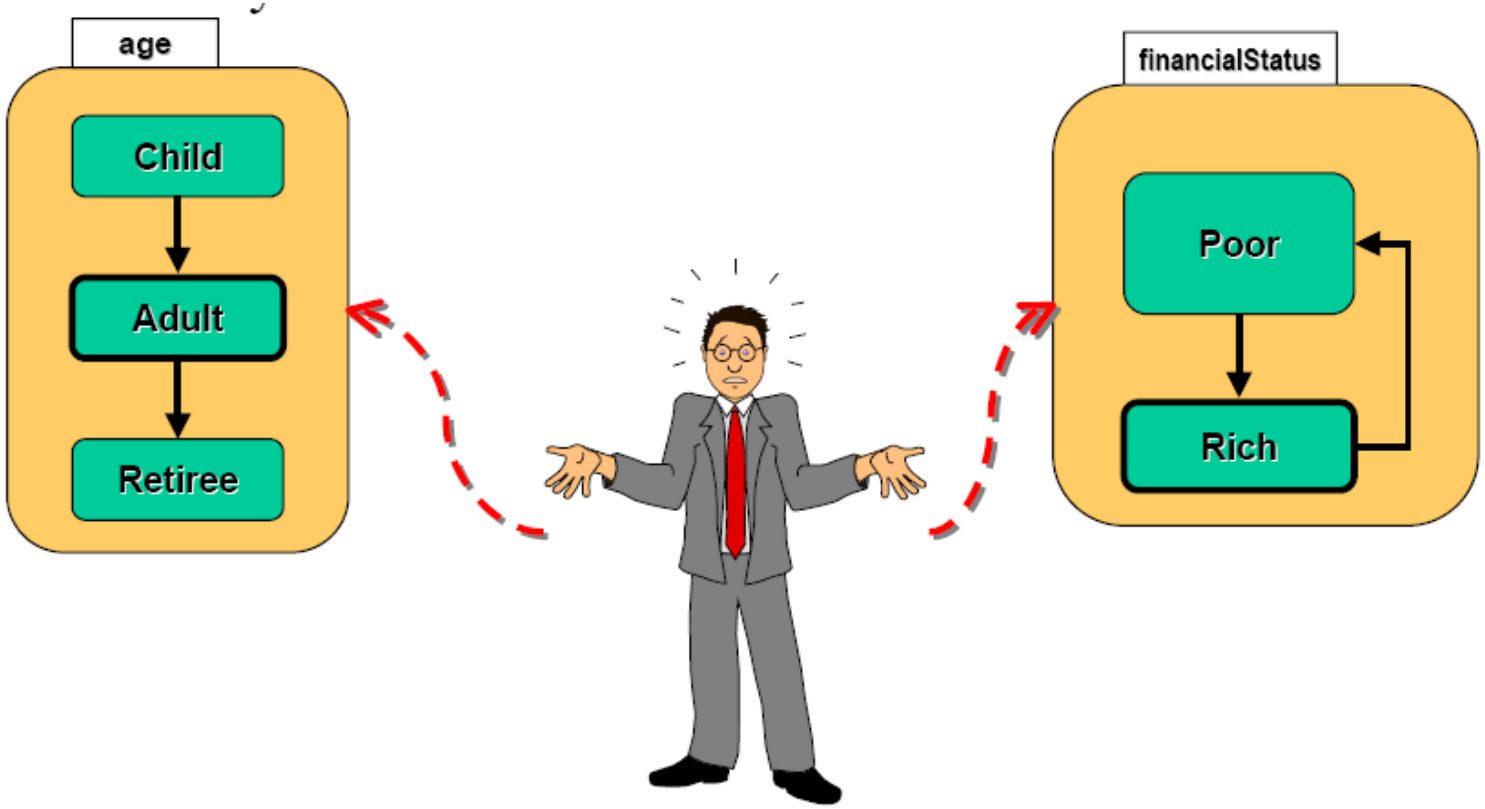


State Charts



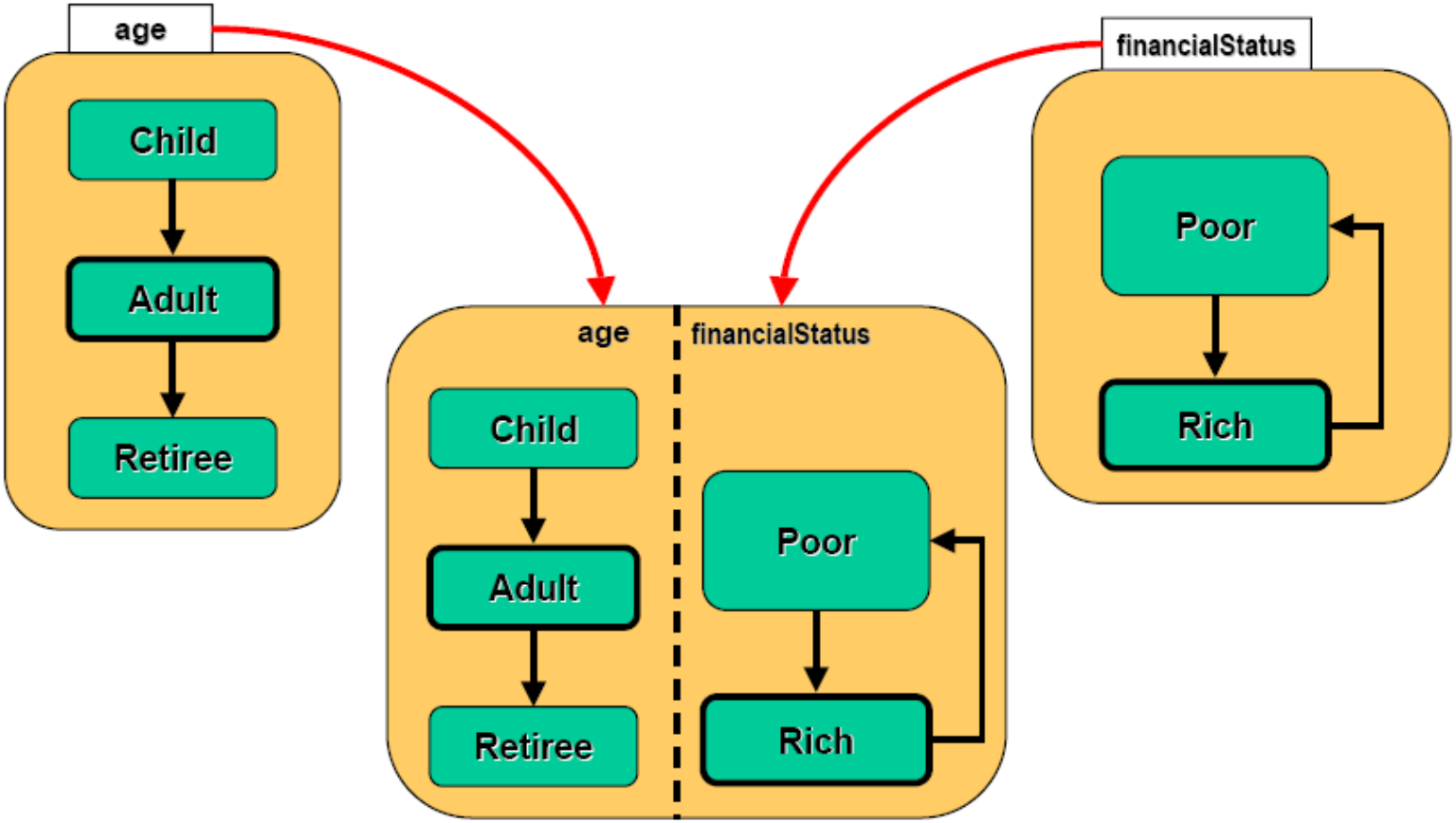
State Charts

- Multiple simultaneous perspectives on the same entity

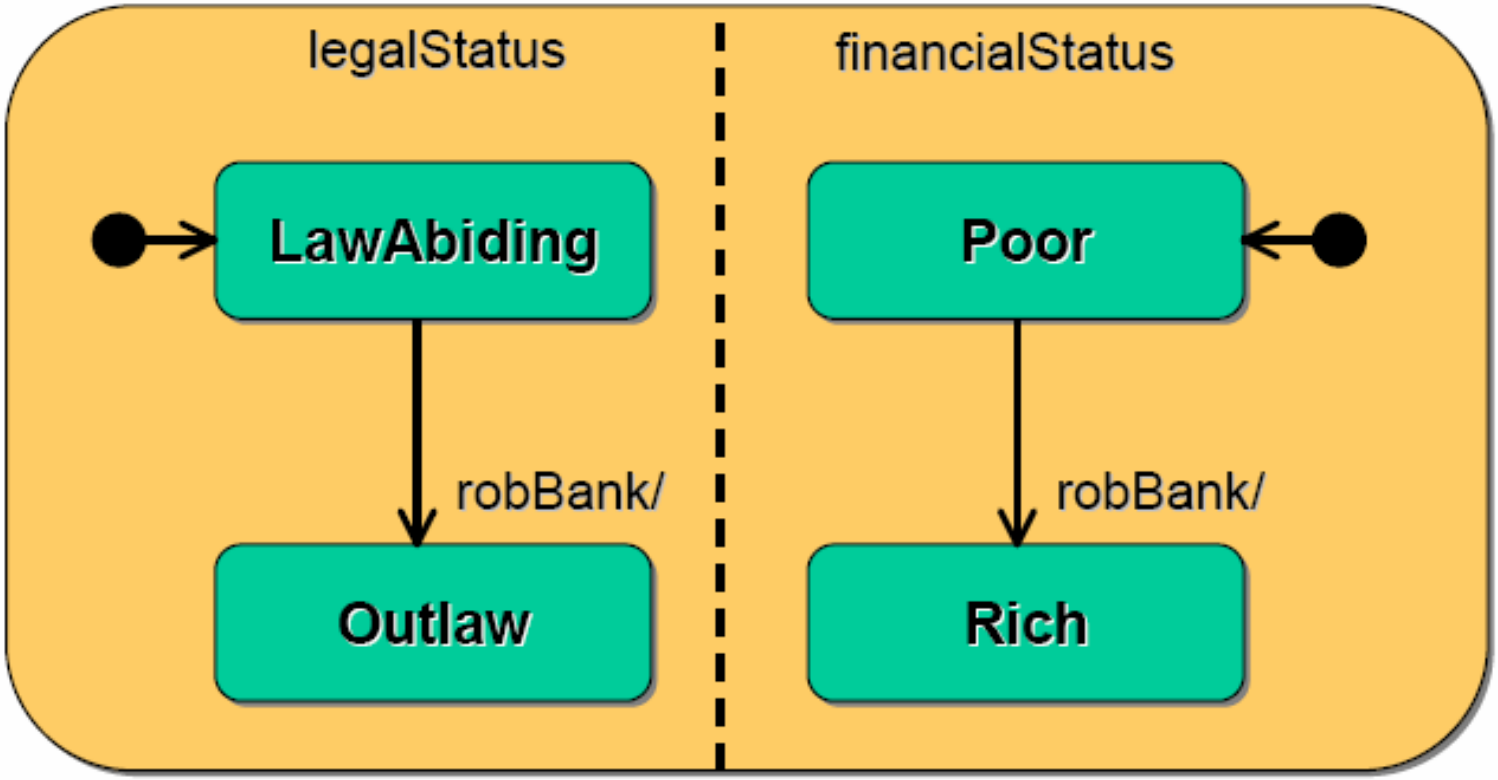


State Charts

- Combine multiple simultaneous descriptions

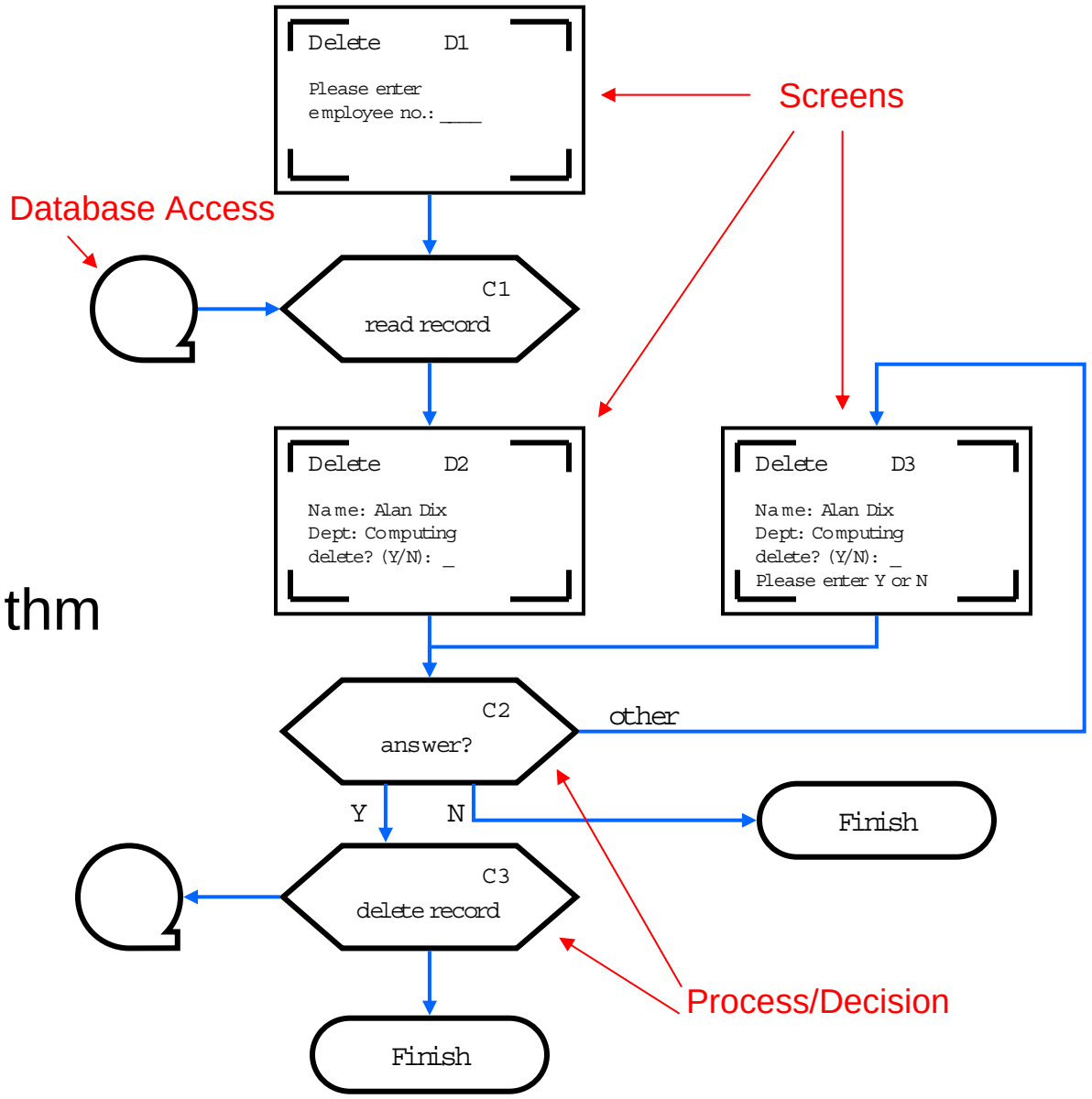


State Charts



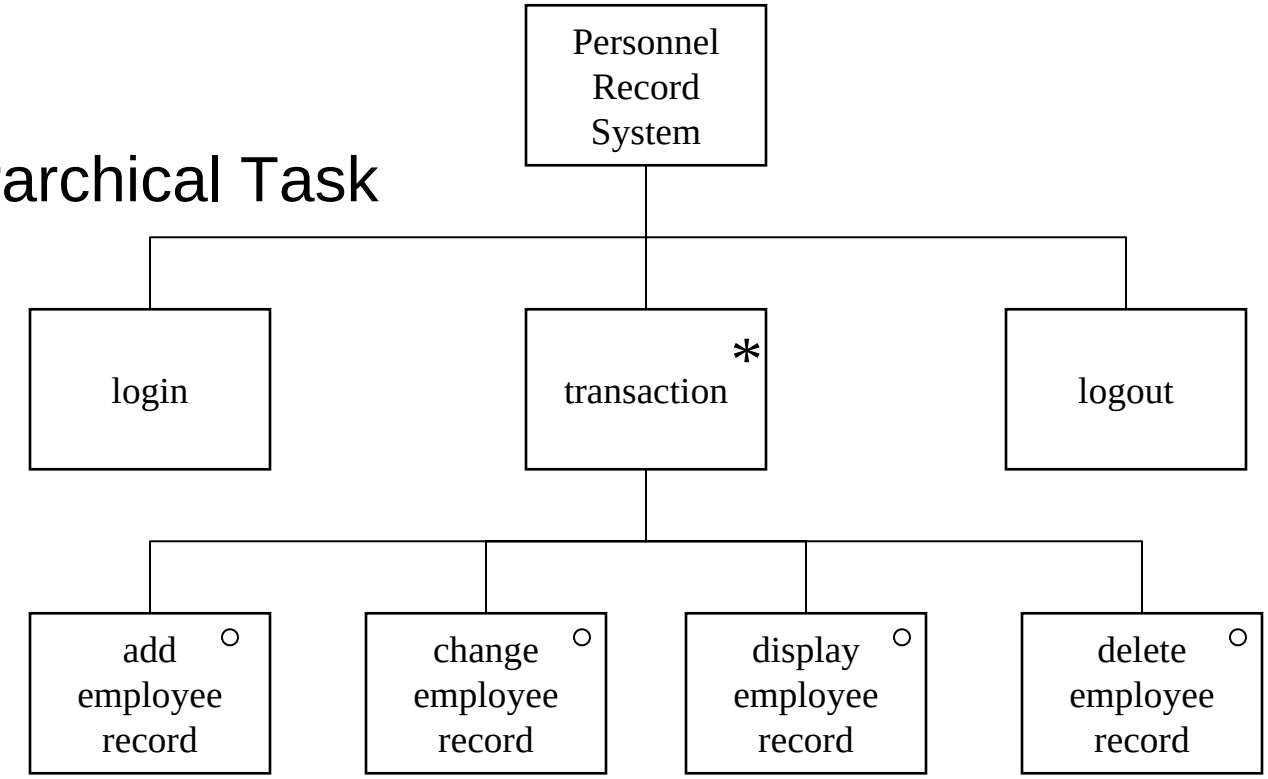
Flow Charts

- Familiar to programmers
- Used for Dialog
- Not internal algorithm

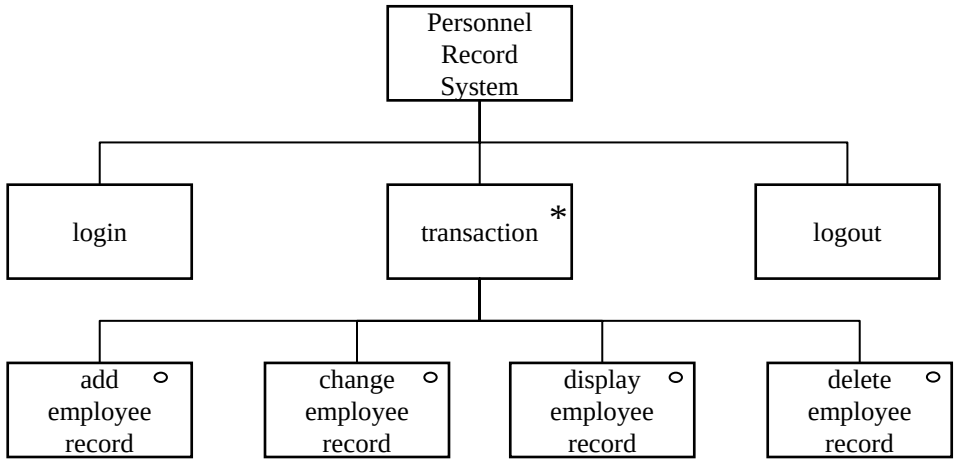


JSD Diagrams

- JSD – Jackson Structured Design
- Similar to Hierarchical Task Analysis



JSD Diagrams



- Sequence – Left to Right
- Can not do any transactions before login
- ‘o’ – Optional elements
  - *Transaction may be anyone of the four operations*
- ‘*’ – Iterations: Any number of repetitions
  - *Login – Any number of transactions - Logout*

References

- Chapter 16 - Human Computer Interaction by Dix et al.
- Interactive Tutorials on Petri Nets, Wil van der Aalst, et al. TU Eindhoven, Netherlands
- Modeling and Simulation, P. Fishwick