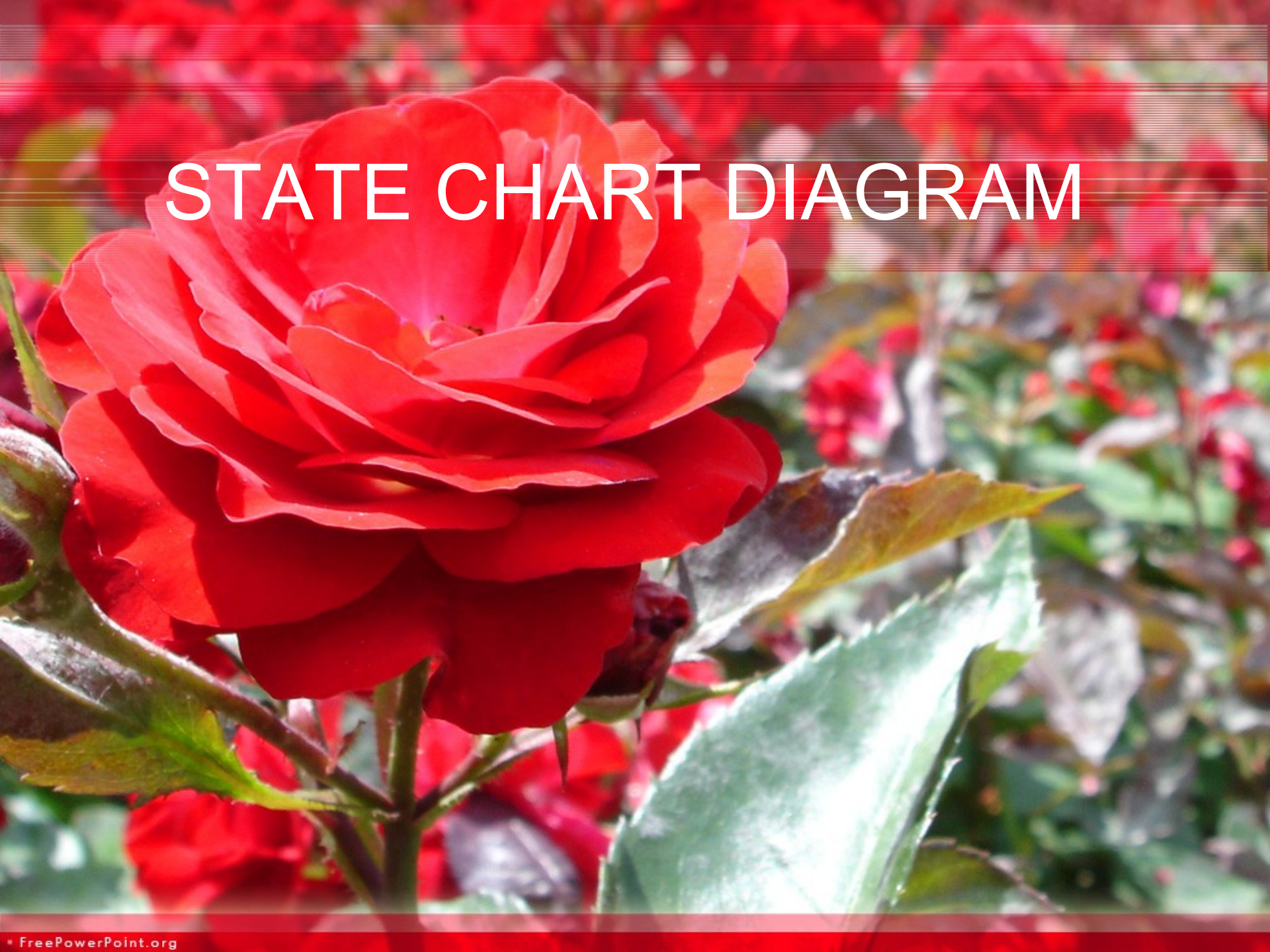


TITLE

- State diagrams (also called State Chart diagrams) are used to help the developer better understand any complex/unusual functionalities or business flows of specialized areas of the system.
- In short, State diagrams depict the dynamic behavior of the entire system, or a sub-system, or even a single object in a system. This is done with the help of ***Behavioral elements***.

Statechart

Statechart diagrams describe the behavior of dynamic model elements

 Systems and entities within a system, such as objects, can be viewed as moving from state to state…events trigger some activity that change the state of the system

 They are closely related to activity diagrams

 Activity diagrams describe flow between areas of work

 Statechart diagrams describe the changes between the states of instances

- examples: a telephon is hung up, dialling, engaged in a call, or disconnected

 Statechart diagrams can be used to model workflow (as activity, sequence or collaborative diagrams)….but statechart diagram models behavior from the perspective of a single entity (such as a class), whereas other diagrams can model the behavior of many entities in a single diagram

 Statechart diagrams are used mostly to describe the behavior of classesntent

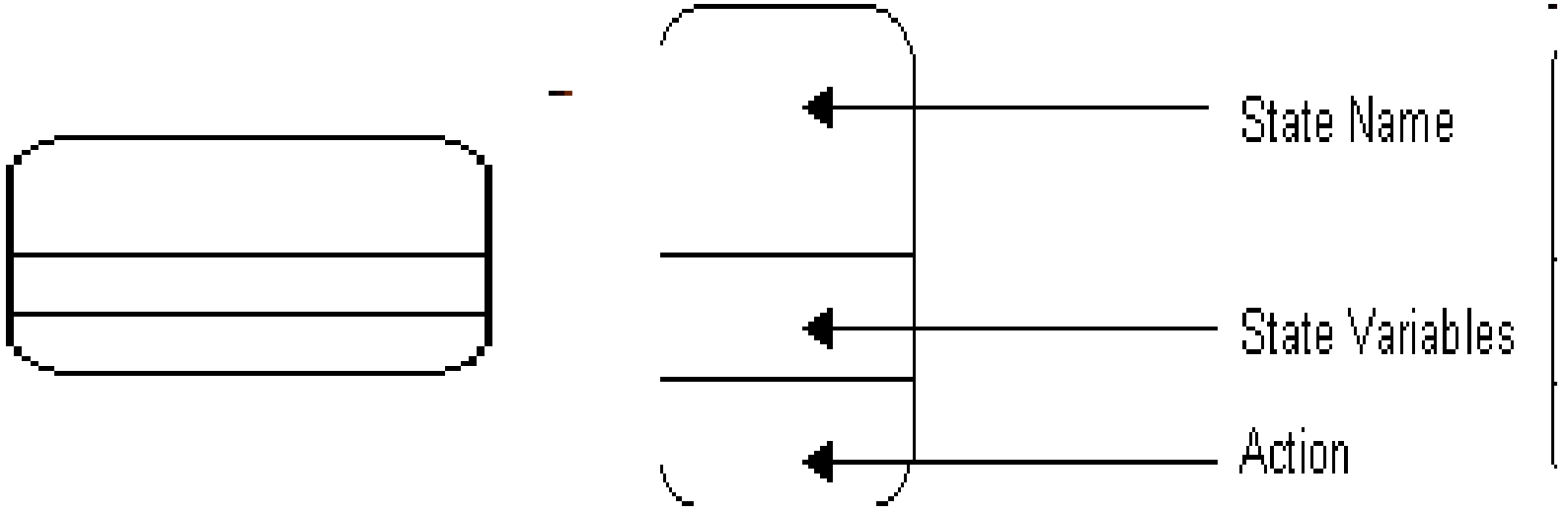
- **Initial State:** This shows the starting point or first activity of the flow. Denoted by a solid circle. This is also called as a "**pseudo state**," where the state has no variables describing it further and no activities.
- Each state diagram should have only one initial state.
- **Final State:**
- Each **final state** is the ending state of the object with reference to the behaviour that the diagram explains. There may be multiple final states for an object.

- **State Machine**

A state machine is a behavior which specifies the sequence of states an object visits during its lifetime in response to events, together with its responses to those events

State

A state is a condition during the life of an object during which it satisfies some condition, performs some activity, or waits for some external event



Event

An event is the specification of a significant occurrence. For a state machine, an event is the occurrence of a stimulus that can trigger a state transition

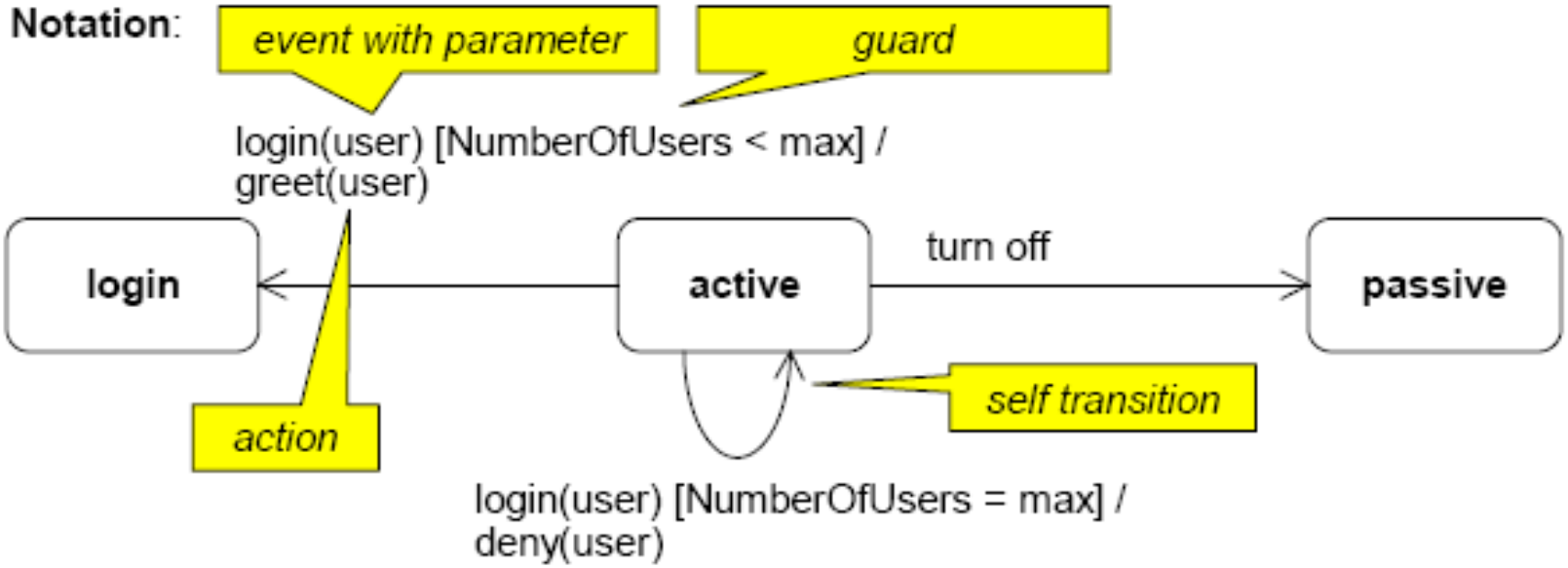
Transition

A transition is a relationship between two states indicating that an object in the first state will, when a specified set of events and conditions are satisfied, perform certain actions and enter the second state.

A transition connects two states and shows the flow of control.

A transition can include a **triggering event**, a **guard** and **actions** to be executed.

Transitions without event and guard are executed **immediately** when an activity is finished respectively all sub states were passed through.



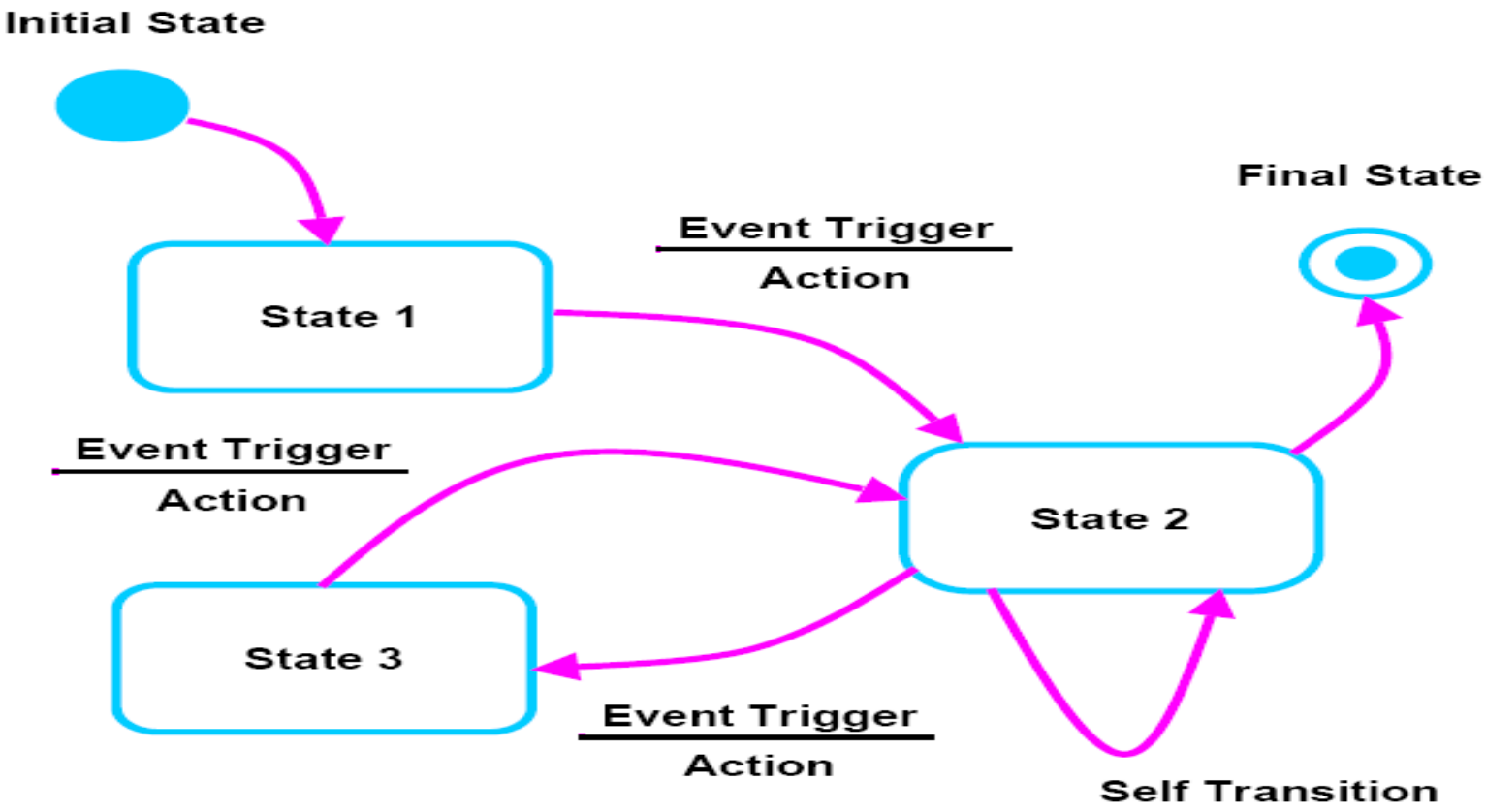
Self-Transition

A self-transition is a transition whose source and target states are the same

Action

. An action is best described as a task that takes place within a state.

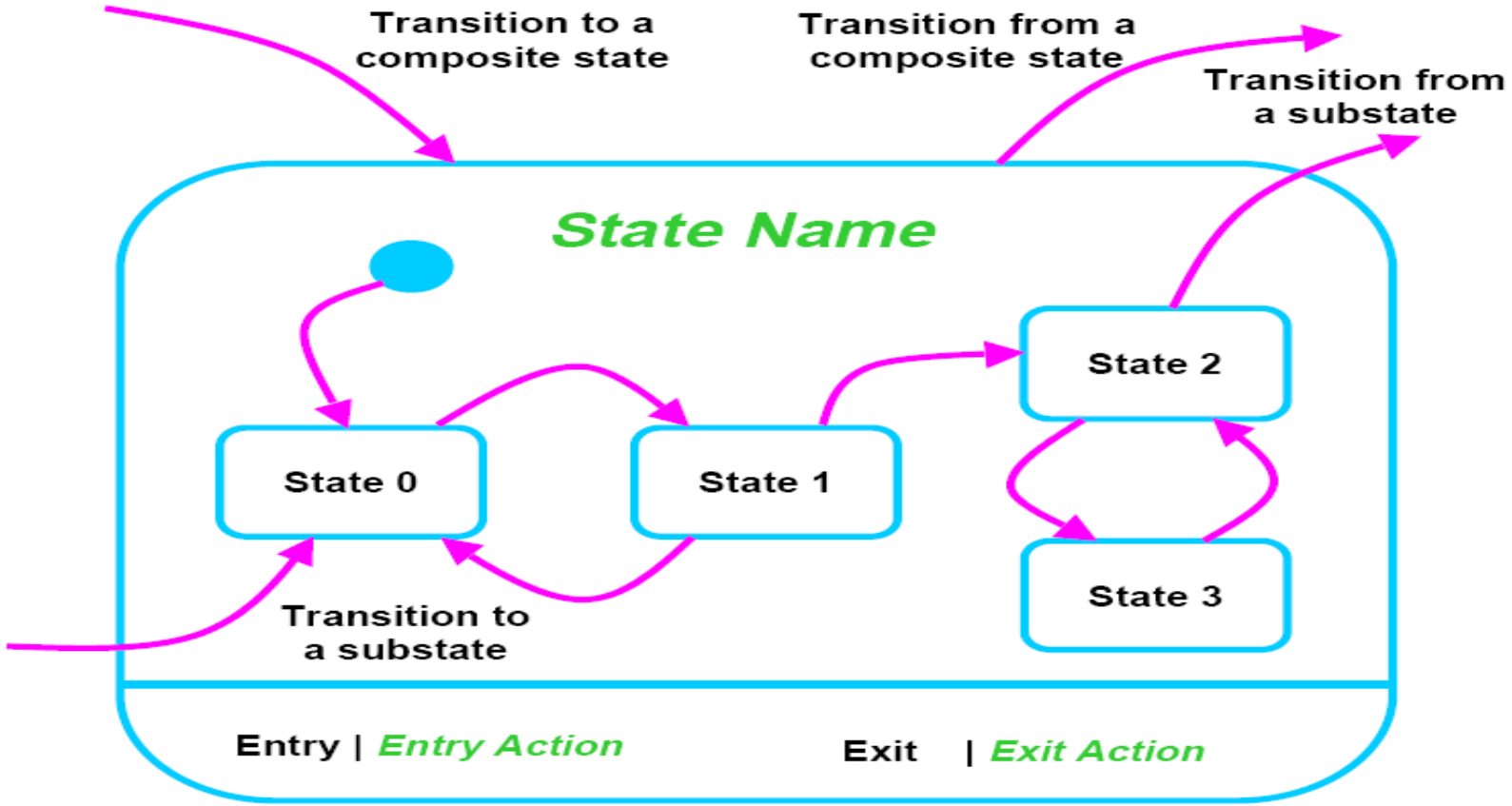
- An action is an executable, atomic (with reference to the state machine) computation.
- Actions may include operations, the creation or destruction of other objects, or the sending of signals to other objects (events).
- There are four possible actions within a state:
- On entry
- On exit
- Do
- On event


Initial State
State 1
Event Trigger
Action
Final State
Event Trigger
Action
State 2
State 3
Event Trigger
Action
Self Transition

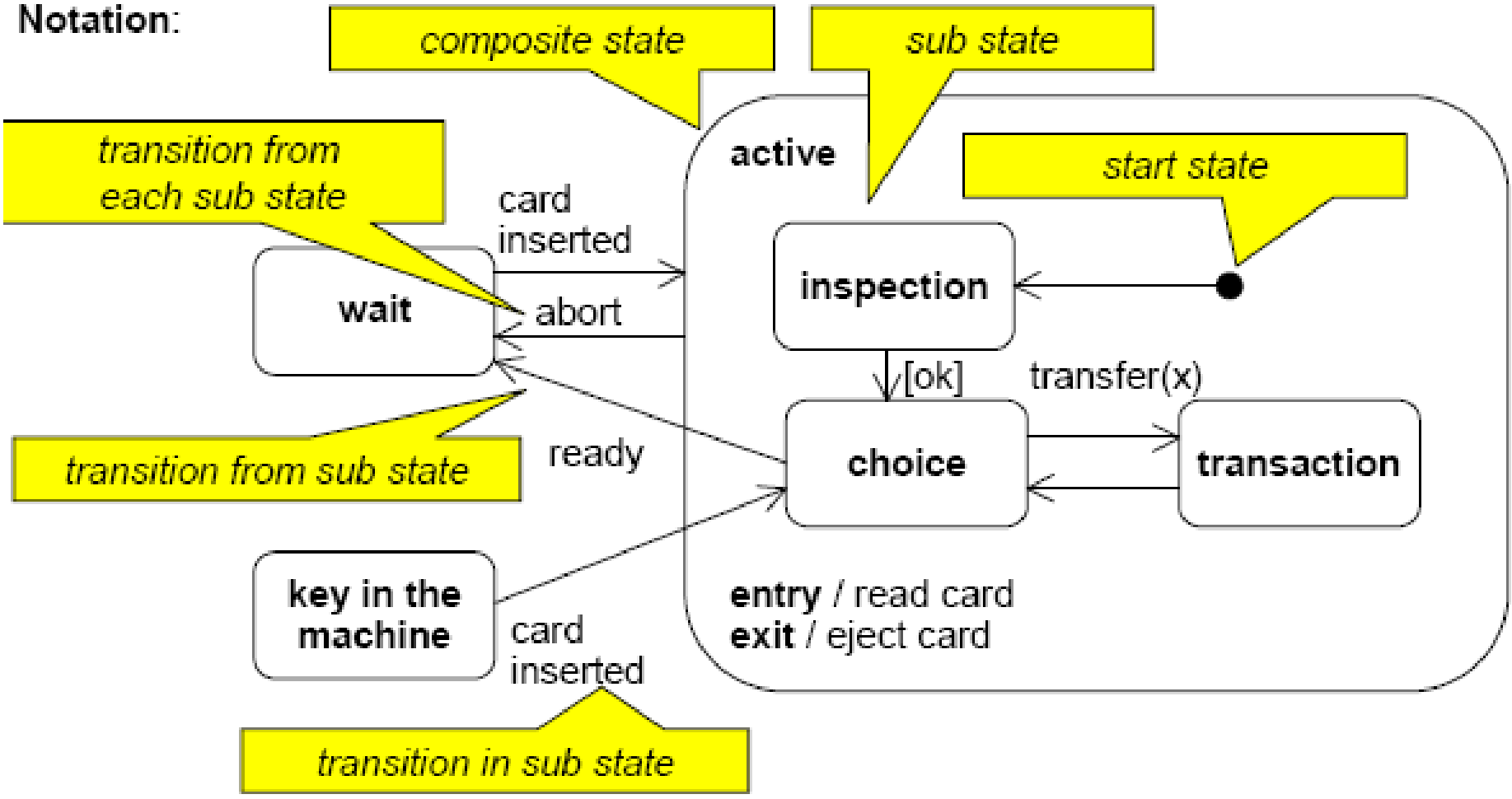
Entry and Exit Actions

- **Substates**
  - A ***substate*** is a state that is nested in another state
  - A state that has substates is called a ***composite state***
  - A state that has no substates is called a ***simple state***
  - Substates may be nested to any level

Sub-States


Transition to a composite state
Transition from a composite state
Transition from a substate
State Name
State 0
State 1
State 2
State 3
Transition to a substate
Entry | Entry Action
Exit | Exit Action

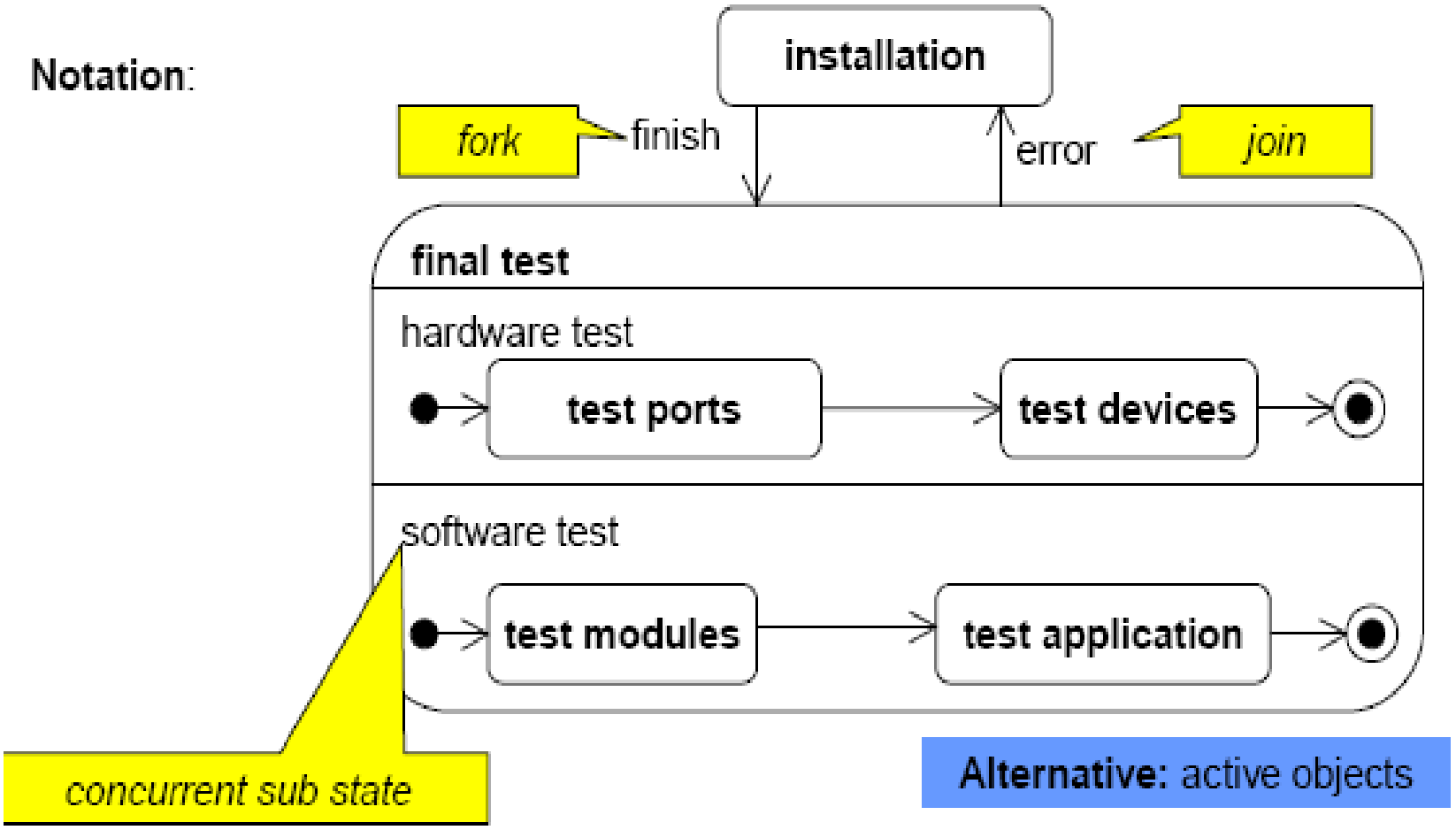
Composite States



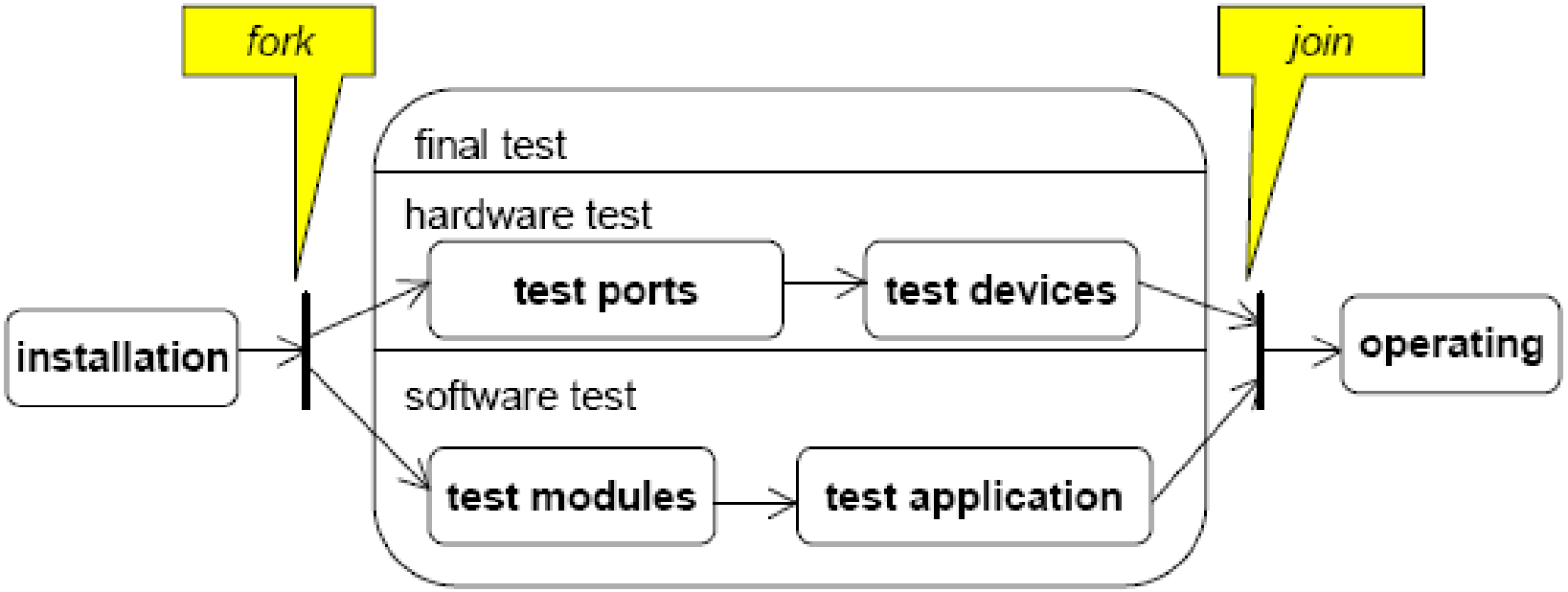
Concurrent Sub States

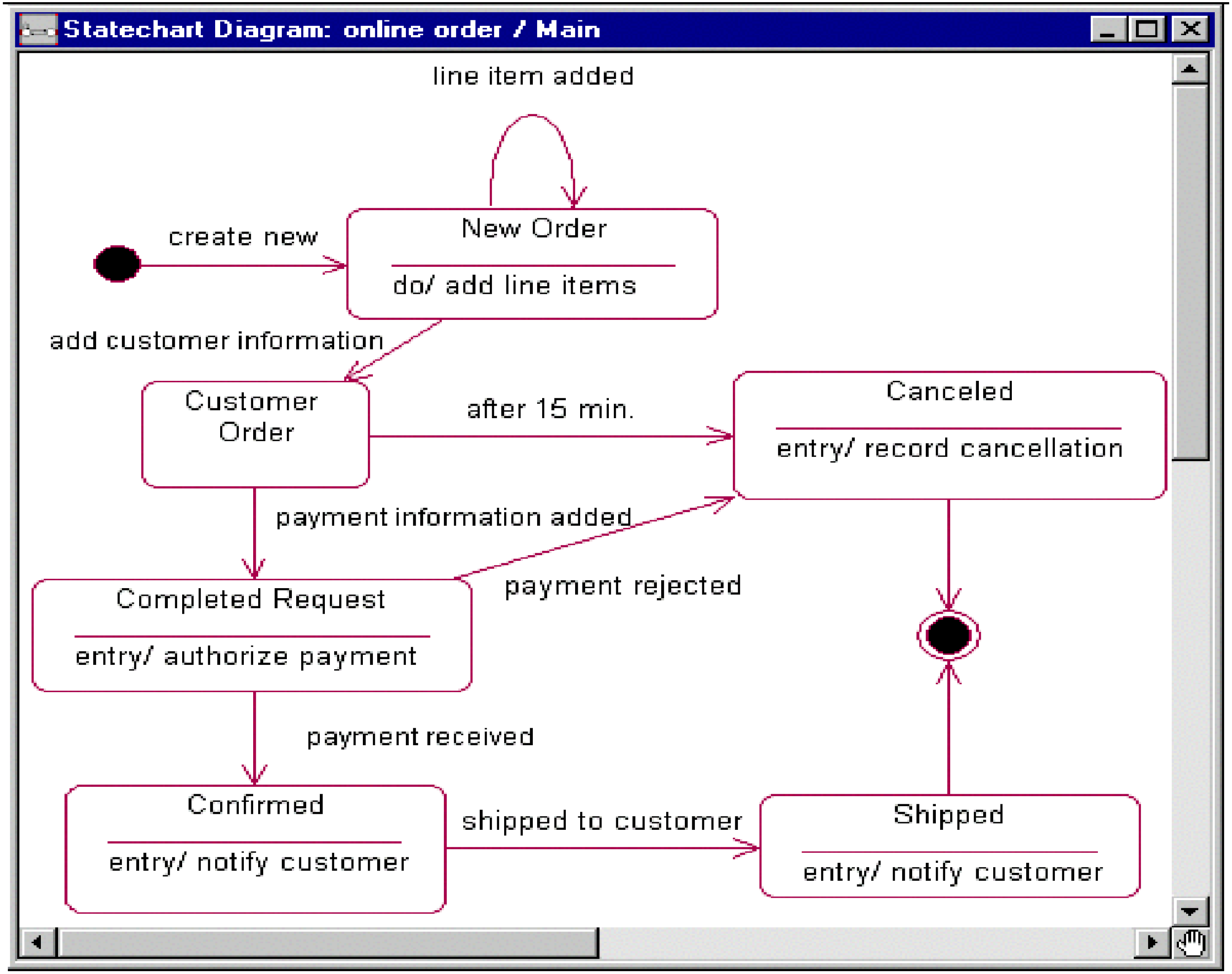
In a state several sequences of sub states described by own state machines can be performed concurrently.

Notation:



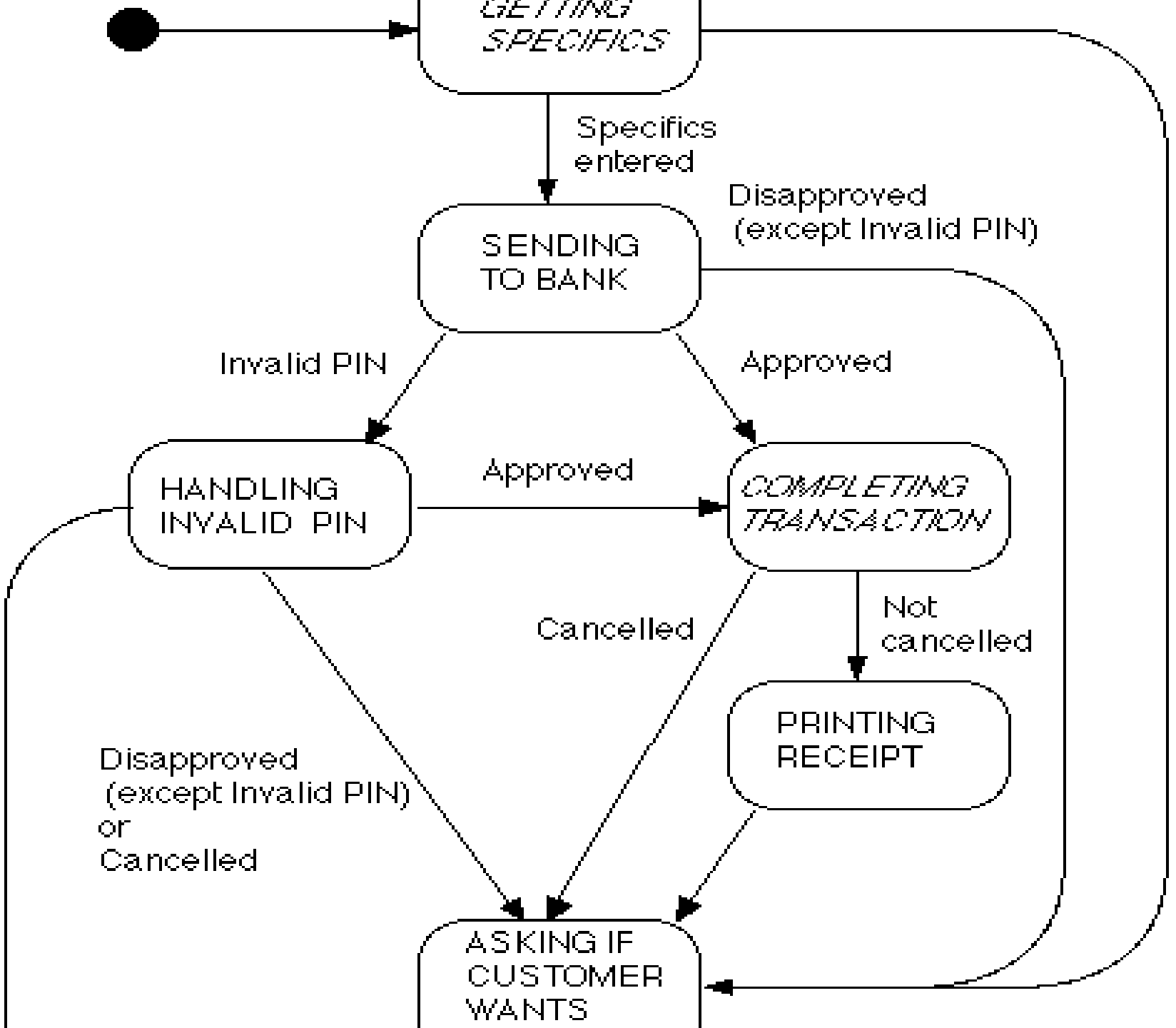
Concurrent Sub States: Alternative




Statechart Diagram: online order / Main
line item added
create new
New Order
do/ add line items
add customer information
Customer Order
after 15 min.
Canceled
entry/ record cancellation
payment information added
payment rejected
Completed Request
entry/ authorize payment
payment received
Confirmed
entry/ notify customer
shipped to customer
Shipped
entry/ notify customer

Main purposes of using State chart diagrams:

- To model dynamic aspect of a system.
- To model life time of a reactive system.
- To describe different states of an object during its life time.
- Define a state machine to model states of an object


GETTING SPECIFICS
Specifics entered
SENDING TO BANK
Disapproved (except Invalid PIN)
Invalid PIN
Approved
HANDLING INVALID PIN
Approved
COMPLETING TRANSACTION
Cancelled
Not cancelled
PRINTING RECEIPT
Disapproved (except Invalid PIN) or Cancelled
ASKING IF CUSTOMER WANTS

State Transition Diagram for One Session