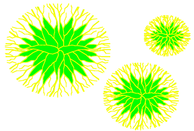
IV. State and Activity Diagrams

State Diagrams
Events and States
Superstates
Activity Diagrams
Petri Nets



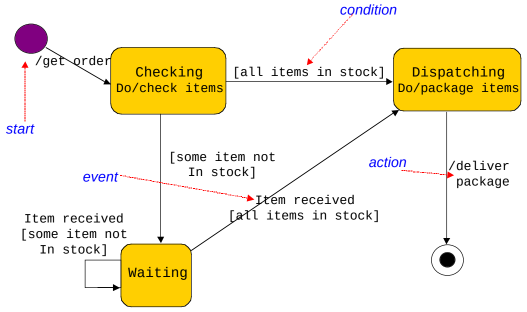
State Diagrams (Statecharts)

- These are state transition diagrams (with some interesting additions) which can be used to describe the operating environment of the system, interactions of the system with that environment, also the lifetime of some object (a person, a student,…) within or without the system.
- Transitions are supposed to represent actions which occur “quickly” and are not interruptible. A transition can have an associated triple

  `Event[Condition]/Action`

  all parts of this triple are optional.
- States are supposed to represent longer-running activities (or other things). What constitutes “quickly” and “longer-running” depends on the application.

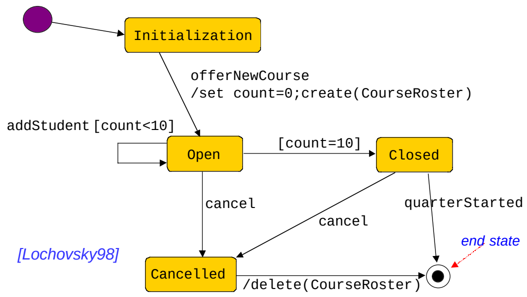
State Diagram for Purchase Order



Events

- An ***event*** is a happening that the system needs to know about.
- For example, completing an assignment, failing an exam, or a system crash are all events.
- An event may trigger an action by an actor or the system being developed.
- In UML, there are four types of events:
  - ✓ ***Change events*** occur when a condition becomes true, e.g., `when(balance < 0);`
  - ✓ ***Signal events*** designate the receipt of an explicit (real-time) signal from one object or actor to another;
  - ✓ ***Call events*** indicate the receipt of a call for an operation by an object or actor;
  - ✓ ***Time events*** mark the passage of a designated period of time, e.g., `after(10 seconds)`

Course Lifetimes

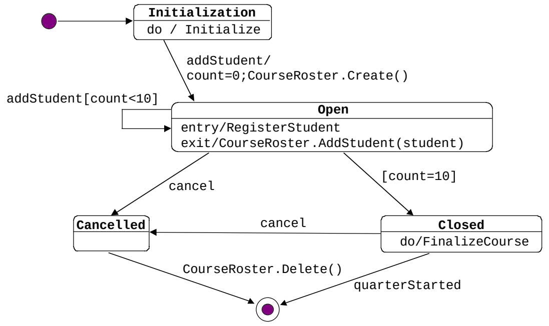


[Lochovsky98]

States

- A state represents a time period during which
  - ✓ A predicate is true, e.g., budget - expenses >0,
  - ✓ An action is being performed, e.g., check inventory for order items, or
  - ✓ Someone waits for an event to happen, e.g., arrival of a missing order item.
- A state can be “on” or “off”.
- When a state is “on”, all its outgoing transitions are eligible to fire. For a transition to fire, its event must occur and its condition must be true. When a transition does fire, its action is carried out.
- States can have associated activities. Special activity constructs include:
  - ✓ do/stateDiagramName(parameterList) -- “calls” another state diagram;
  - ✓ entry/action -- carry out the action when entering the activity;
  - ✓ exit/action -- carry out the action when exiting;

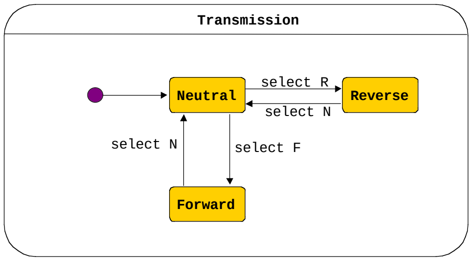
Course Lifetimes, Again



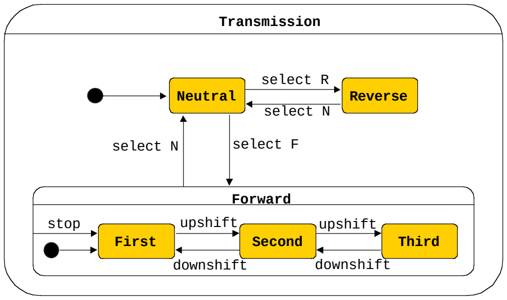
Superstates

- State transition diagrams can be very hard to read once they grow to more than a few dozen states.
- For UML state diagrams, states can be composed into nested states, or ***superstates***. Such compositions make it possible to view a state diagram at different levels of abstraction.
- A superstate consists of one or more states.
- When an ***AND superstate*** is "on", all its states are also "on".
- When an ***OR superstate*** is "on", one of its states is "on".

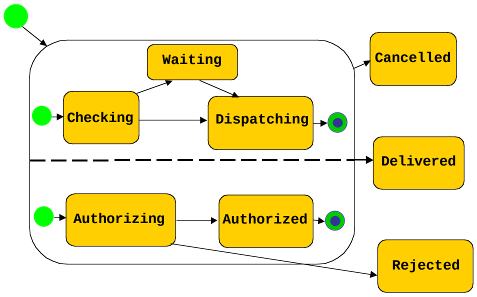
An OR Superstate



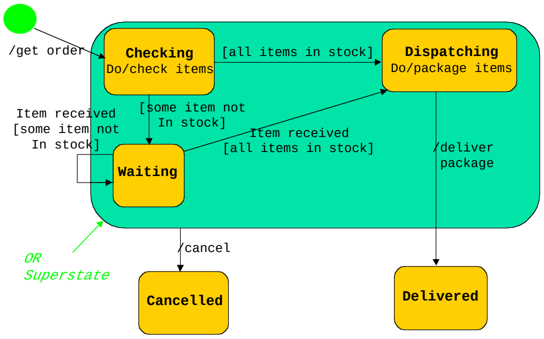
...More on Transmission



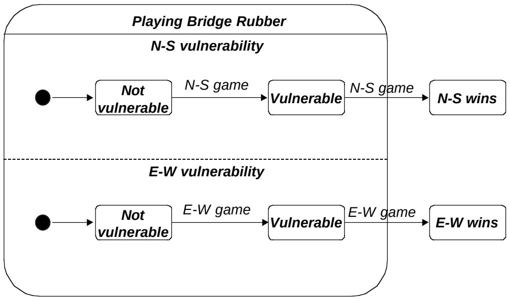
An AND Superstate



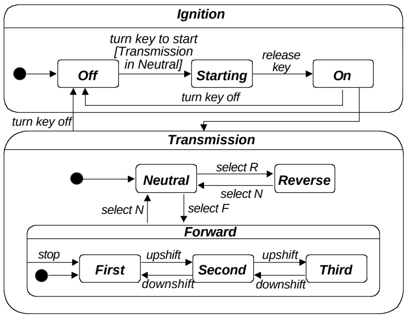
State Diagram for Purchase Order



Bridge Vulnerability Rules



Auto Transmission

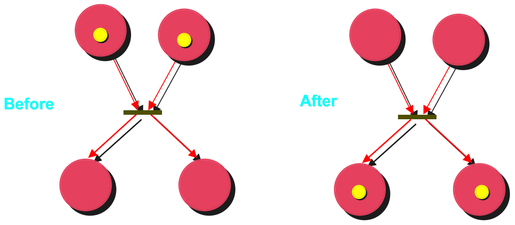


Activity Diagrams

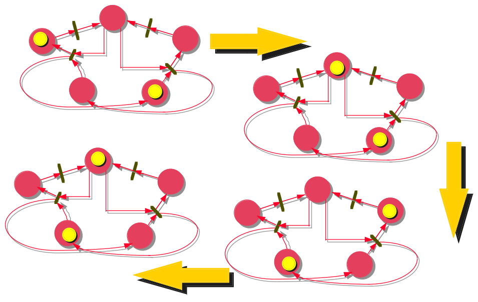
- Activity diagrams describe activities which involve concurrency and synchronization.
- Activity diagrams are a variation of state diagrams that focuses on the flow of actions and events.
- Can be used
  - ✓ To model a human task (a business process, for instance).
  - ✓ To describe a system function that is represented by a use case.
  - ✓ In operation specifications, to describe the logic of an operation.

Firing a Transition

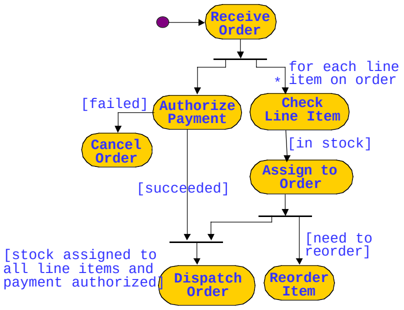
- Activity diagrams use a Perti net-like notation to describe activities which involve concurrency and synchronization.
- Petri nets use transitions which involve several input and output states:



An Example

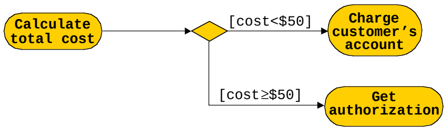


Order Processing



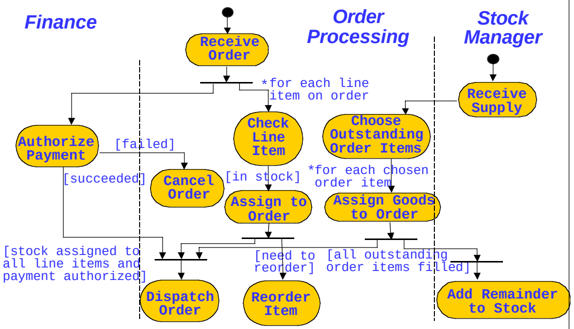
Decision Points

- ***Decision points:***



- ***Dead ends:*** *there may be transitions in an activity diagram with no destination state; this can mean that:*
  - ✓ *Not all processing has been specified,*
  - ✓ *Or, that another activity diagram will take over.*

Swimlanes



When to Use What?

- *State diagrams are good for modeling the lifetime of an object or actor, also for modeling user interfaces and business processes which involve many states.*
- *Activity diagrams are good for modeling business processes and system processes* ***which involve a lot of concurrency****.*
- *Sequence and collaboration diagrams (to be discussed later in the course) are useful for modeling interactions;* ***several of them*** *can be used to model dialogue structure for a user interface, or a business process.*

Additional Readings

- *[Booch99] Booch, G. et al., The Unified Modeling Language User Guide, Chapters 19, 20, 21, 24. Addison-Wesley.*
- *[Fowler00] Fowler, M., UML Distilled: A Brief Guide to the Standard Object Modelling Language, Chapters 8, 9. Addison-Wesley.*