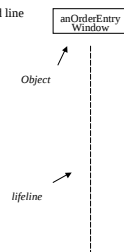


Interaction Diagrams

- Describe how groups of objects collaborate in some behavior
 - often used to show how use cases or parts of use cases behave
- Two kinds of interaction diagrams are
 - sequence diagrams
 - collaboration diagrams
- We will focus on sequence diagrams

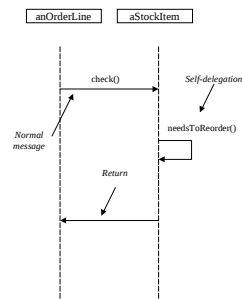
Components

- Object**
 - shown as a box at the top of a vertical dashed line
- lifeline**
 - represents time flow



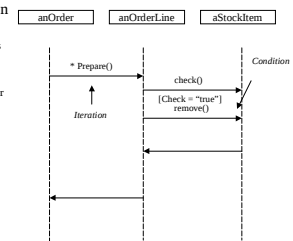
Components

- Messages**
 - communication between objects
- Special message types**
 - self-delegation
 - return

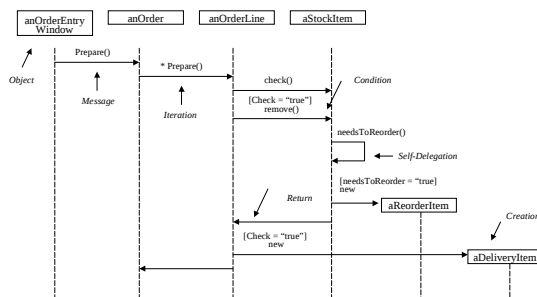


Components

- Two kinds of control information**
 - message conditions
 - message sent only if condition is true
 - iteration marker
 - message sent to multiple receiver objects



Example Sequence Diagram



Tips on Sequence Diagrams

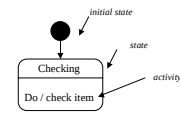
- Show conditional behavior on separate diagrams
- Use to show the behavior of several objects within a use case
 - use a state diagram when you want to focus on one object across many use cases

State Diagrams

- Describe
 - the possible states a single object can get in to
 - how object changes state in response to events

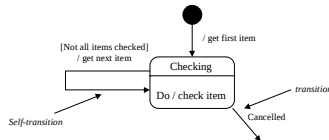
Components

- States
 - represented as a rounded box
 - name
 - activities - process performed by object, long-running, interruptible
 - form1 : do / activityName
 - form2 : eventName / activityName
 - initial state is special, represented as a filled circle

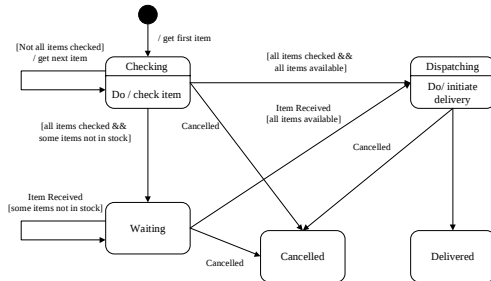


Components

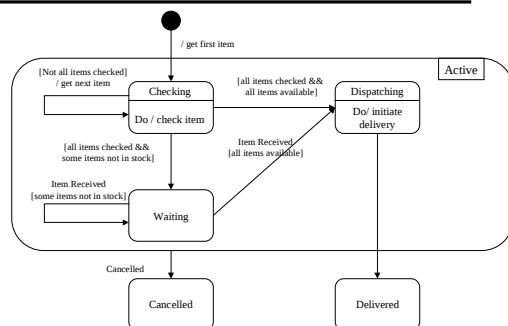
- Transitions
 - allowable state change
 - represented as an arrow from one state to another
 - form : Event (Guard) / Action
 - events can be thought of as signals
 - a guard is boolean condition
 - actions are short-duration, interruptible processes carried out during transitions



Example State Diagram for Order Object

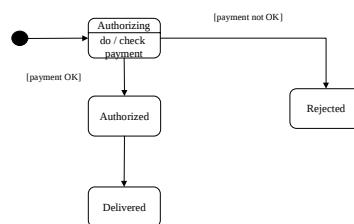


Using Superstates

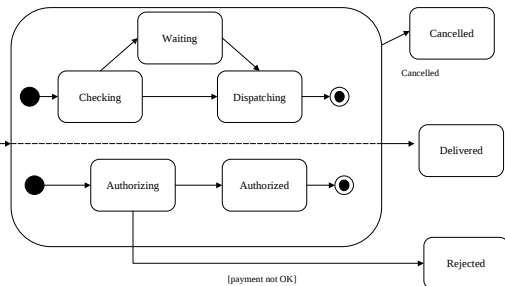


Concurrent State Diagrams

- Payment authorization done concurrently



Concurrent State Diagram



Tips on State Diagrams

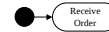
- ♦ Describe behavior of one object, not many objects
 - for many objects use interaction diagrams
- ♦ Don't try to draw them for every object

Activity Diagram

- ♦ Show the flow of a process from activity to activity

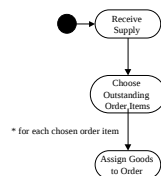
Components

- ♦ Activities
 - task that needs to be done



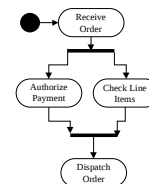
Components

- ♦ Control flow
 - Sequencing
 - » temporal ordering of tasks
 - » can be guarded
 - Decision
 - » if-then-else construct
 - » (not shown here)
 - Multiple triggers
 - » indeterminate looping construct

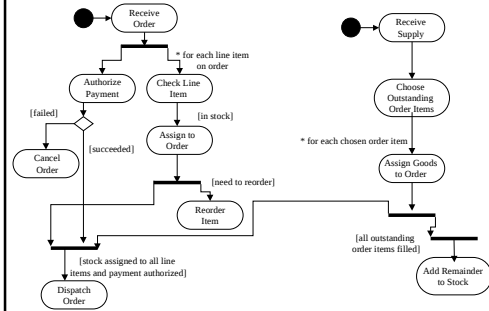


Components

- ♦ Parallelism and synchronization
 - allows parallel threads to be created or destroyed
 - stops progress until all threads reach synchronization point



Example Activity Diagram



Tips on Using Activity Diagrams

- ◆ Good for exploring parallel process flows
 - analyzing a use case
 - understanding workflow across multiple use cases
 - understanding multi-threaded applications
- ◆ Poor at showing connections between objects and actions
 - trying to understand how objects collaborate
 - understanding object lifecycle behavior