

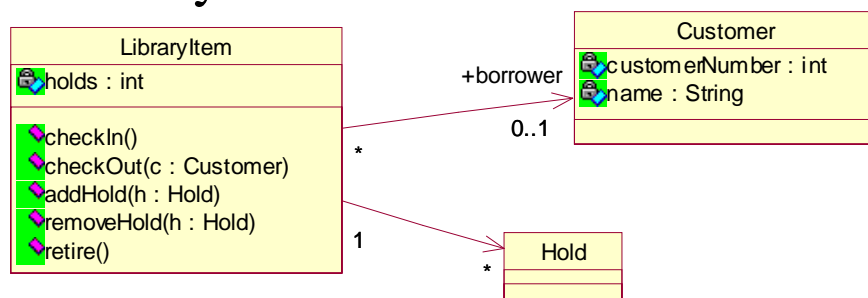
4.4 State Diagrams

- Interaction diagrams show behavior of a set of related objects
- UML State Diagrams show behavior of objects of a single class
 - state changes over lifetime
- Lifetime of an object
 - created with “new”
 - destroyed by garbage collector if not referenced anymore
 - in C++: with “delete”
- State
 - a period when a certain property holds
 - not directly shown in other diagrams or code
- UML version of Finite State Machines (FSMs)
 - also called State Charts, State Machines

© 2004. 497-5-50

State Diagram Example:

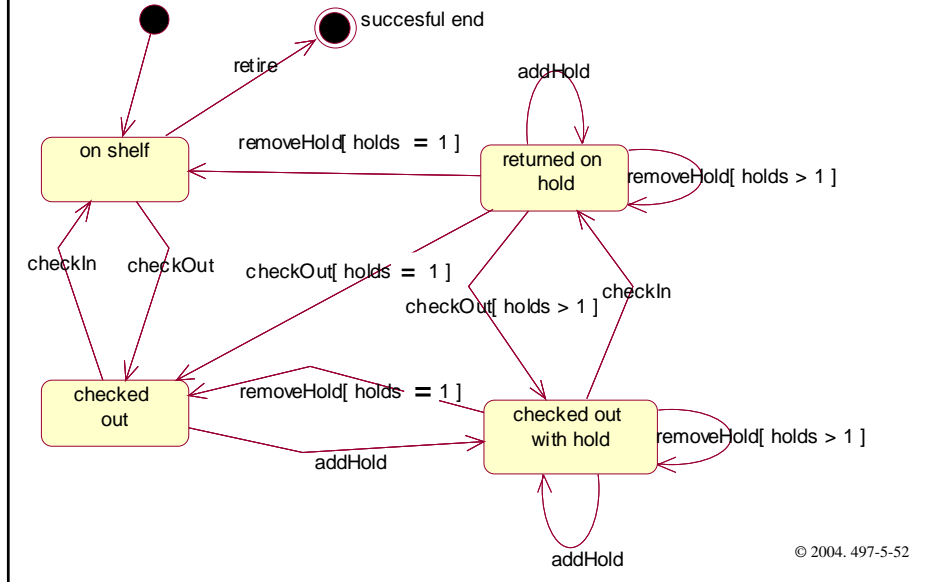
LibraryItem



- A LibraryItem is a book or video that can be borrowed
 - ‘holds’ is the number of holds on the item
- What states do objects of class LibraryItem have?

© 2004. 497-5-51

State Diagram: Class LibraryItem



Elements of State Diagrams



start state



end state

- States
 - rounded rectangles
 - represent properties that exist for a significant period
 - can contain activities
- Transitions
 - represent instantaneous events
 - syntax: event [guard] / action
 - events are generally operation calls
 - guards are conditions that must be true
 - actions are additional things that may happen

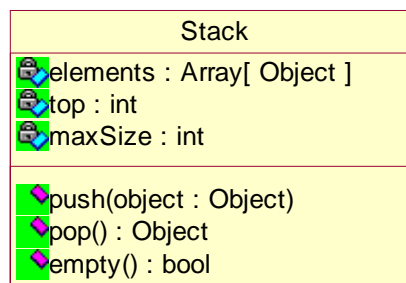
© 2004. 497-5-53

State Diagrams

- Transitions that are not shown are assumed to lead to an error state
 - Example: event “check in” in state “on shelf”
 - make sure that all other possible transitions are shown
- Each diagram has exactly one start state
- There can be several end states
 - for object-oriented systems, the transition to the end state is obvious and can be left out
- Superstates
 - a set of states is a state too
- What are states?
 - sets of values of attributes
 - “checked in”
borrower == null
 - “checked out with holds”
borrower != null *and* holds > 0

© 2004. 497-5-54

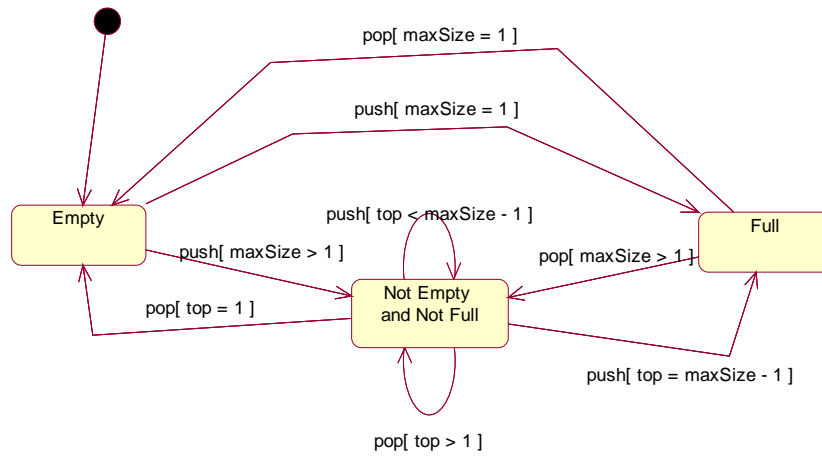
State Diagram: Stack Example



- Stack
 - stores Objects
 - only the Object stored last can be retrieved (popped)
 - can say if it is empty
 - has a maximum size
 - top represents the location of the next element that is added

© 2004. 497-5-55

State Diagram: Stack

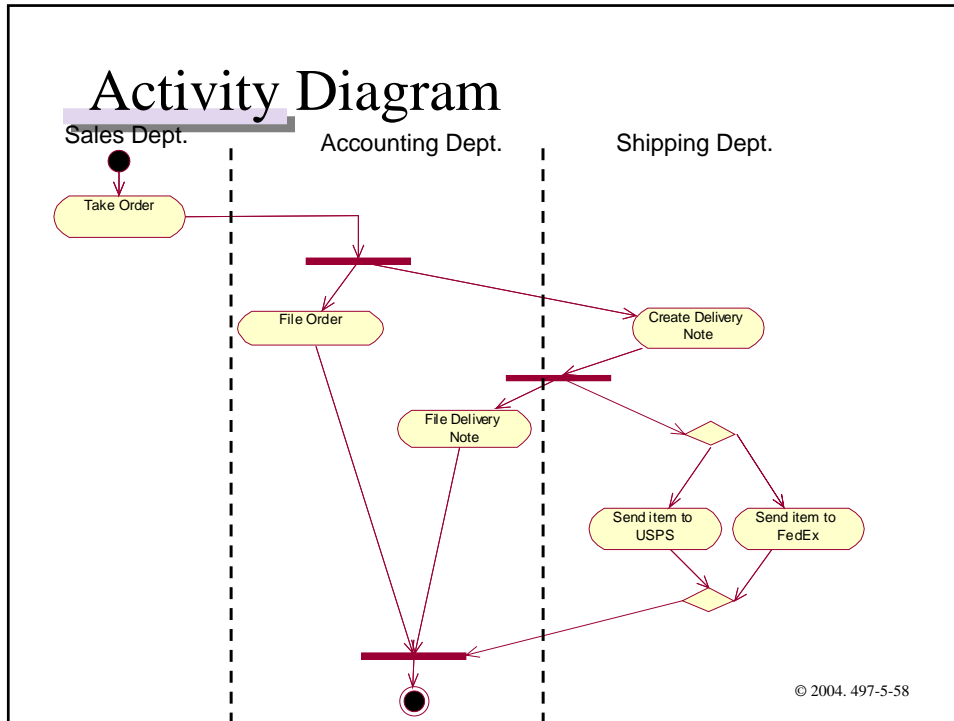


© 2004. 497-5-56

4.5 Activity Diagrams

- Activity diagrams show sequence of activities
- Good for modeling workflows in analysis
- Similar to ANSI flow charts
 - outdated

© 2004. 497-5-57



Elements of Activity Diagrams

- Elements known from state diagrams:
 - activity states
 - transitions
 - start state
 - end state
- New elements
 - fork (horizontal bars)
 - A process is split into two (or more) concurrent processes
 - join (horizontal bars)
 - Several concurrent processes are joined into one
 - branch / merge (diamonds)
 - like if-statements: “then” branch and “else” branch
 - partitions
 - also known as: swimlanes
 - to classify activities

© 2004. 497-5-59

Use of Activity Diagrams

- Analyzing workflows
 - well-suited for parallel processes, workflows
 - similar use as dataflow diagrams
 - difference: focus on activities, not data
- Modeling control flow in methods (design phase)
 - algorithms
 - activities are mapped to statements
- Similar to:
 - flowcharts / control-flow diagrams
 - Petri nets
- Disadvantage: not very object-oriented
 - do not show classes, objects, operations
 - interaction diagrams are usually a better choice

© 2004. 497-5-60

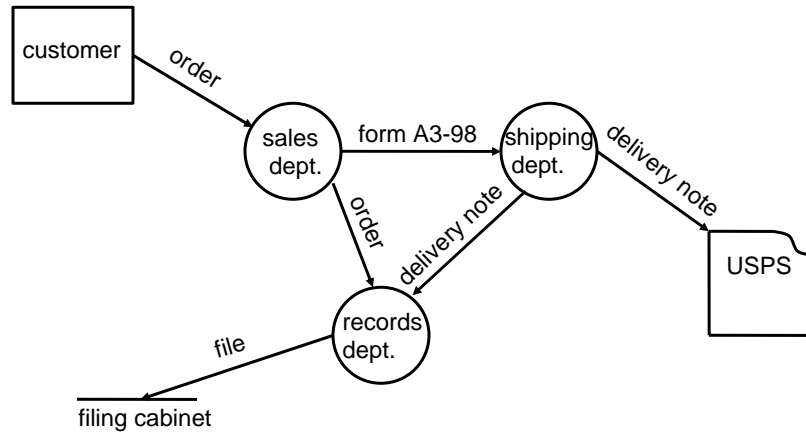
4.7 Specification Languages of Olden Days

- A couple of older specification languages
- Not part of the UML
- Still used though

- Dataflow Diagrams
- Entity-Relationship Diagram

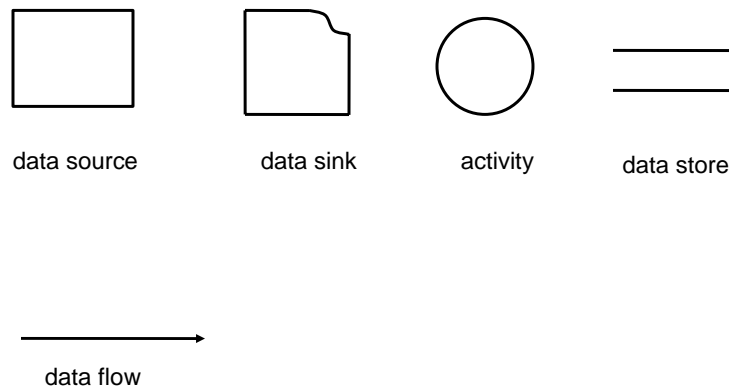
© 2004. 497-5-61

Dataflow Diagram



© 2004. 497-5-62

Elements of Dataflow Diagrams



© 2004. 497-5-63

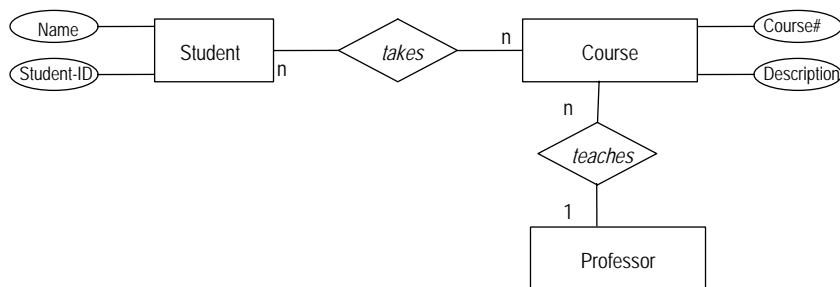
Dataflow Diagrams

- Focus on data
 - as opposed to control flow diagrams (like Activity diagrams)
 - no notion of time or sequence
- Good for modeling work in real-life organizations
 - activities are inherently parallel
 - data: forms or papers
 - activities: people or organizations
- No direct equivalent in UML

© 2004. 497-5-64

ER Diagram

- Notation:
 - Entities (with entity attributes)
 - Relations (with relation attributes)
 - Cardinality of relationships (1:1, 1:m, n:m)



© 2004. 497-5-65

ER Diagrams

- Entity-Relationship Diagrams
- Have been replaced by UML class diagrams
 - entities = classes
 - relationships = associations
- Were often used with dataflow diagrams
- Are still used for modeling of relational databases

© 2004, 497-5-66