Lecture 6

Introduction to UML

Review

- Introduction to Virtualization
- Introduction to UML

Today's Outline

- Use case diagram
- Class diagram
- Deployment diagram
- Sequence diagram
- Collaboration/Communication diagram
- Activity diagram
- State (machine) diagram

Reference: UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition, by Martin Fowler (2003)

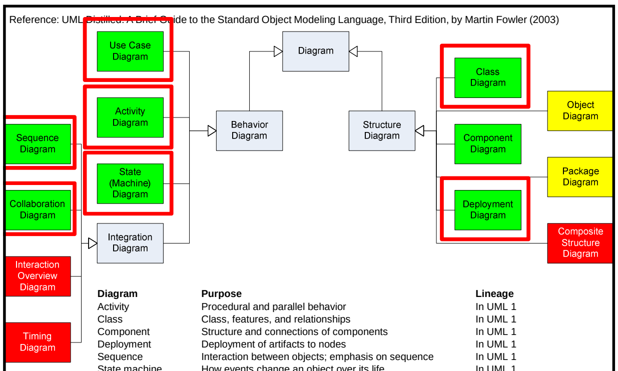
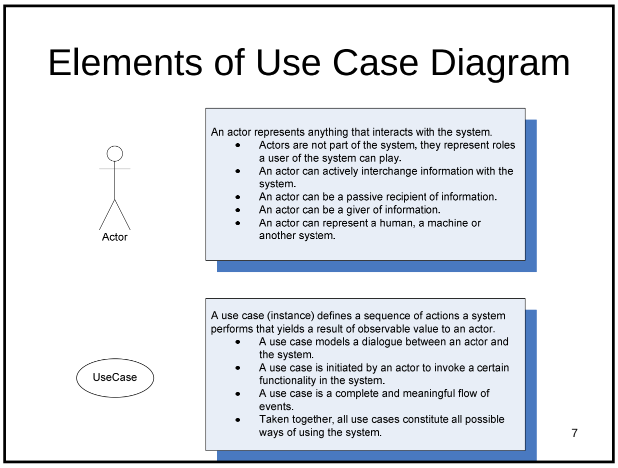


Diagram	**Purpose**	**Lineage**
Activity	Procedural and parallel behavior	In UML 1
Class	Class, features, and relationships	In UML 1
Component	Structure and connections of components	In UML 1
Deployment	Deployment of artifacts to nodes	In UML 1
Sequence	Interaction between objects; emphasis on sequence	In UML 1
State machine	How events change an object over its life	In UML 1
Use case	How users interact with a system	In UML 1
Communication	Interaction between objects; emphasis on links	UML 1 collaboration diagram
Object	Example configurations of instances	Unofficially in UML 1
Package	Compile-time hierarchic structure	Unofficially in UML 1
Composite structure	Runtime decomposition of a class	New to UML 2
Interaction overview	Mix of sequence and activity diagram	New to UML 2
Timing	Interaction between objects; emphasis on timing	New to UML 2

Use Case Diagram

Use Case Diagram

- Describes the requirements of the system, especially functional requirements.
- Use case diagram contains
  - Actors and their descriptions
  - Relationships between different entities
    - <<Include>>: implying that the behavior of the included use case is inserted into the behavior of the including use case
    - <<Extend>>: the behavior of the extension use case may be inserted in the extended use case under some conditions
    - {constraint}: indicate constraints



Elements of Use Case Diagram

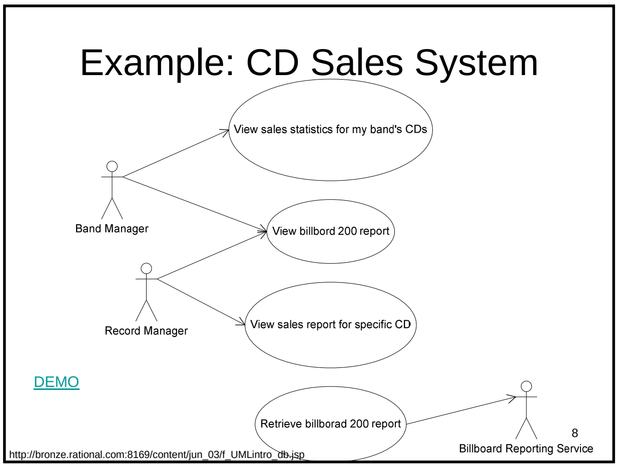
An actor represents anything that interacts with the system.

- Actors are not part of the system, they represent roles a user of the system can play.
- An actor can actively interchange information with the system.
- An actor can be a passive recipient of information.
- An actor can be a giver of information.
- An actor can represent a human, a machine or another system.

A use case (instance) defines a sequence of actions a system performs that yields a result of observable value to an actor.

- A use case models a dialogue between an actor and the system.
- A use case is initiated by an actor to invoke a certain functionality in the system.
- A use case is a complete and meaningful flow of events.
- Taken together, all use cases constitute all possible ways of using the system.

7



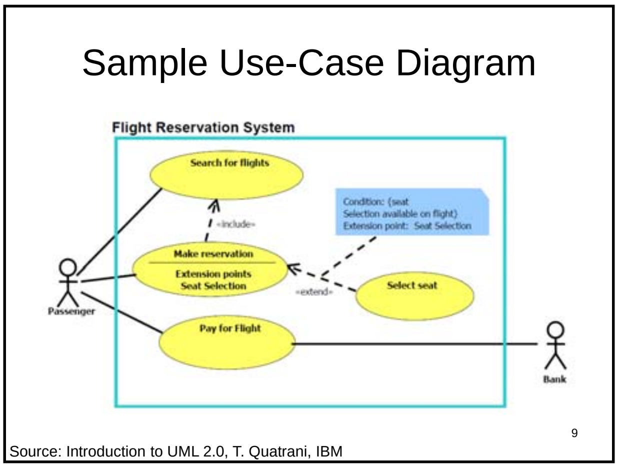
Example: CD Sales System

View sales statistics for my band's CDs
Band Manager
View billbord 200 report
Record Manager
View sales report for specific CD
Retrieve billborad 200 report
Billboard Reporting Service

DEMO

http://bronze.rational.com:8169/content/jun_03/f_UMLintro_db.jsp

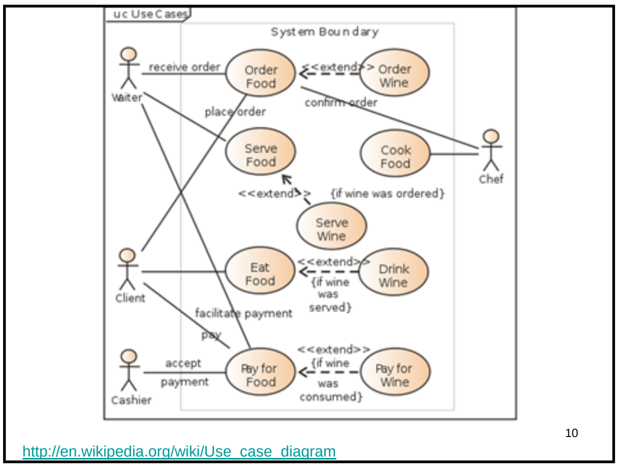
8



Sample Use-Case Diagram

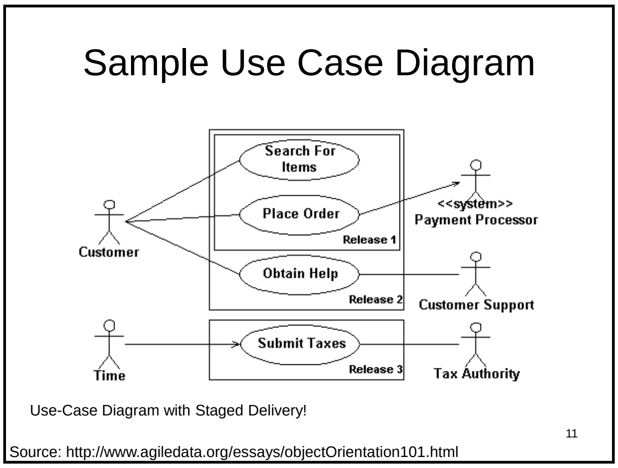
9

Source: Introduction to UML 2.0, T. Quatrani, IBM



10

http://en.wikipedia.org/wiki/Use_case_diagram

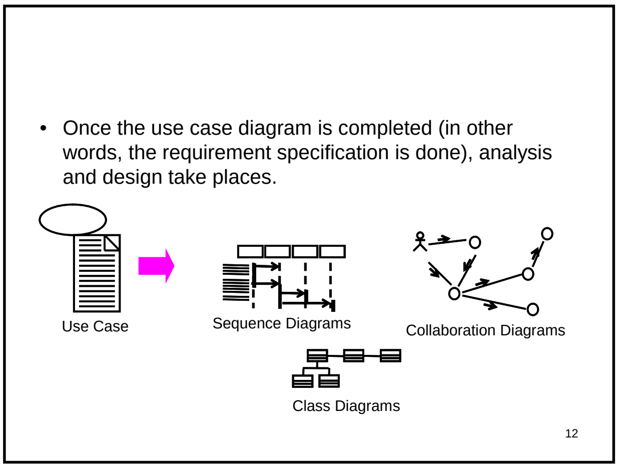


Sample Use Case Diagram

Use-Case Diagram with Staged Delivery!

11

Source: http://www.agiledata.org/essays/objectOrientation101.html



- Once the use case diagram is completed (in other words, the requirement specification is done), analysis and design take places.

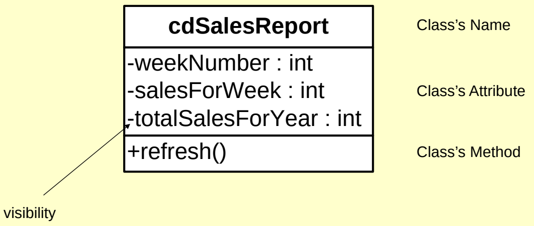
12

Class Diagram

Class Diagram

- Used in requirements, design and implementation:
  - Conceptual, to represent general entities in system
  - Specification, where we specify what each entity (class) will do (but not how)
    - List the methods/actions
  - Implementation
    - Detailed class diagram of actual software (Java or C++)
- List attributes, same as data model
- List methods/operations/functions
  - Activities naturally associated with the data in the entity
- We often don't model everything—too hard to read
  - Focus on key parts of system

A class element

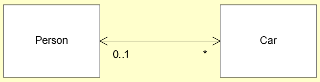


Relationships

Construct	Description	Syntax
Association	A relationship between two or more classifiers that involves connections among their instances.	———
Aggregation	A special form of association that specifies a whole-part relationship between the aggregate (whole) and the component part.	——◇ knows uses refers
Generalization	A taxonomic relationship between a more and a more specific element	Is a kind of ——▶
Dependency	A relationship between two modeling elements, in which a change to one modeling element (the independent element) will affect the other modeling element (the dependent element)	- - - - - ->
Realization	A relationship between a specification and its implementation	implements - - - - -▶
Composition	If a class cannot exist by itself, and instead must be a member of another class, then that class has a Composition relationship with the containing class. A Composition relationship is indicated by a line with a filled diamond.	——◆ has owns

Association

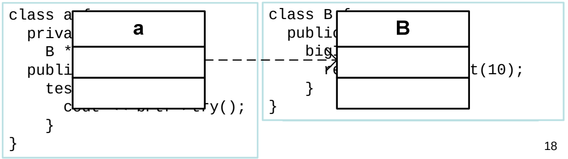
- Associations describe relationships between classes
  - Solid lines are used to describe associations
    - Multiplicity
    - Navigability: Unidirectional vs. bidirectional



•A person may own 0 to many cars
•A car may have 0 or 1 owner

Dependency

- A dependency exists between two elements if changes to the definition of one element (the supplier) may cause changes to the other (the client).
  - one class sends a message to another;
  - one class has another as part of its data;
  - one class mentions another as a parameter to an operation.



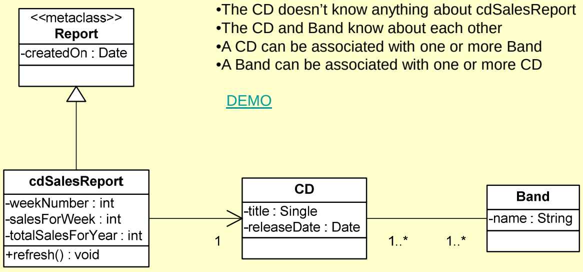
```
class a
  priva
    B *
  publi
    tes
      c              y();
    }
}
```

```
class B
  publi
    big
      r              t(10);
    }
}
```

Example

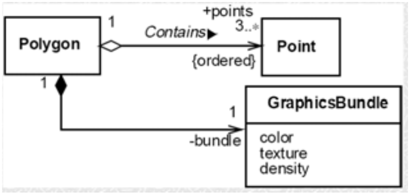
- The cdSalesReport inherits from the Report class
- A cdSalesReport is associated with one CD
- The CD doesn’t know anything about cdSalesReport
- The CD and Band know about each other
- A CD can be associated with one or more Band
- A Band can be associated with one or more CD

DEMO



http://bronze.rational.com:8169/content/jun_03/f_UMLintro_db.jsp

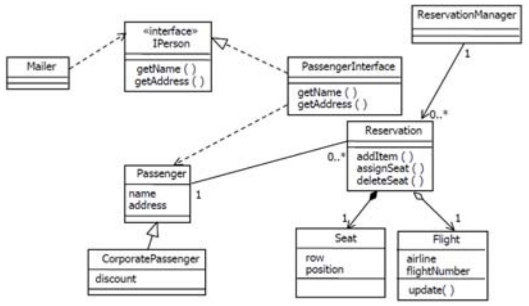
Example



- A polygon contains 3 or more points
- 3 or more points are aggregated in a polygon
- One polygon has one GraphicsBundle, and one GraphicsBundle belongs to exactly one polygon
- Without polygon, graphicsBundle cannot exist by itself.

C. Kobryn (1999), “Object Modeling with UML: Introduction to UML”

Sample Class Diagram



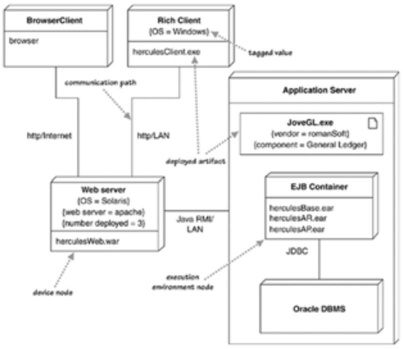
Source: Introduction to UML 2.0, T. Quatrani, IBM

Deployment Diagram

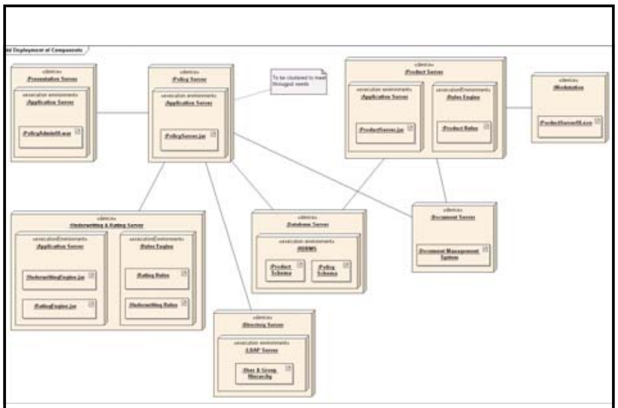
Deployment Diagrams

- Deployment diagrams show the physical relationship between hardware and software in a system.
- The solid lines represent communication links or dependencies.

Example



Reference: UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition, by Martin Fowler (2003)


Deployment of Components
«device» Presentation Server
«execution environment» Application Server
PolicyAdminUI.war
«device» Policy Server
«execution environment» Application Server
PolicyServer.jar
To be clustered to meet throughput needs
«device» Product Server
«execution environment» Application Server
ProductServer.jar
«executionEnvironment» Rules Engine
Product Rules
«device» Workstation
ProductServerUI.exe
«device» Underwriting & Rating Server
«executionEnvironment» Application Server
UnderwritingEngine.jar
RatingEngine.jar
«executionEnvironment» Rules Engine
Rating Rules
Underwriting Rules
«device» Database Server
«execution environment» RDBMS
Product Schema
Policy Schema
«device» Document Server
Document Management System
«device» Directory Server
«execution environment» LDAP Server
User & Group Hierarchy

25

http://upload.wikimedia.org/wikipedia/commons/b/b9/Deployment_Diagram.PNG

Sequence Diagram

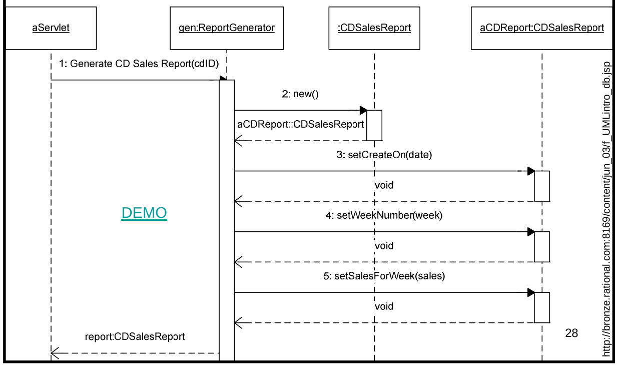
26

Sequence Diagram

- Shows a detailed or partial flow for a specific use case
- Shows the calls between different objects in their sequence
- 2-D:
  - Vertical: time/order
  - Horizontal: object instances

27

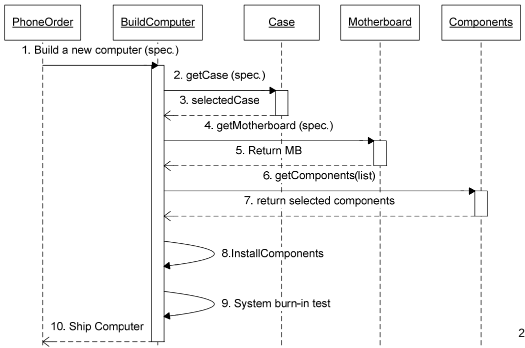
Example



http://bronze.rational.com:8169/content/jun_03/f_UMLintro_db.jsp

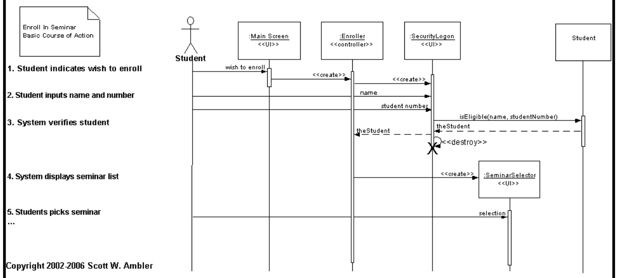
28

Example



29

Example



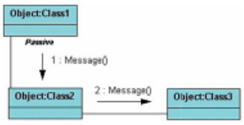
30

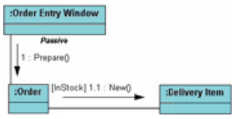
http://www.agiledata.org/essays/objectOrientation101.html

Communication Diagram

Communication Diagram

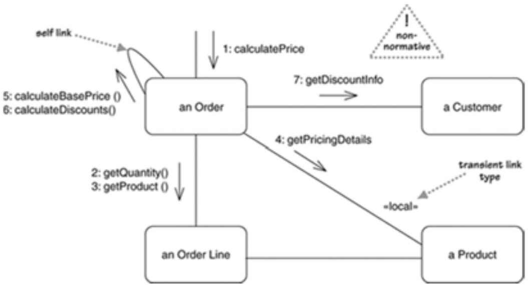
- Collaboration diagram in UML 1.x
- Show the relationship between objects and the order of messages passed between them.
- Collaboration diagram has almost the same information as sequence diagram, but different view/perspective.





Source: http://pigseye.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/interaction.htm

Communication Diagram

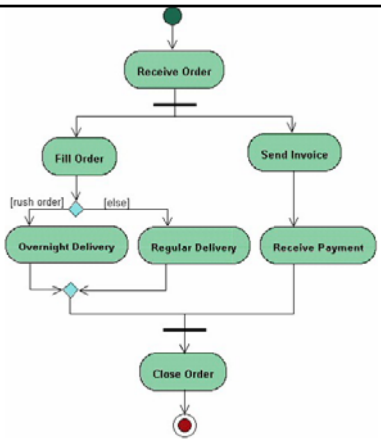


Reference: UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition, by Martin Fowler (2003)

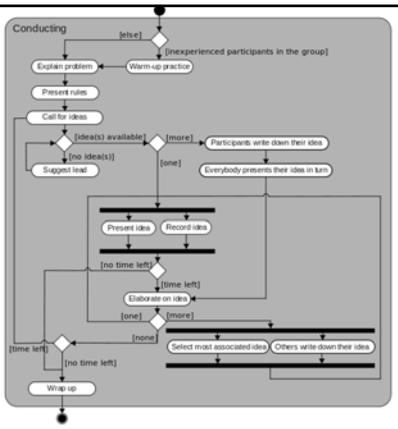
Activity Diagram

Activity Diagrams

- Activity diagrams are a technique to describe procedural logic, business process, and work flow.
- Activity diagram is very much like flowcharts previously introduced.
- The diagrams describe the state of activities by showing the sequence of activities performed.
- Show activities that are conditional or parallel.



http://pigseye.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/activity.htm



http://en.wikipedia.org/wiki/Activity_diagram

State Diagram

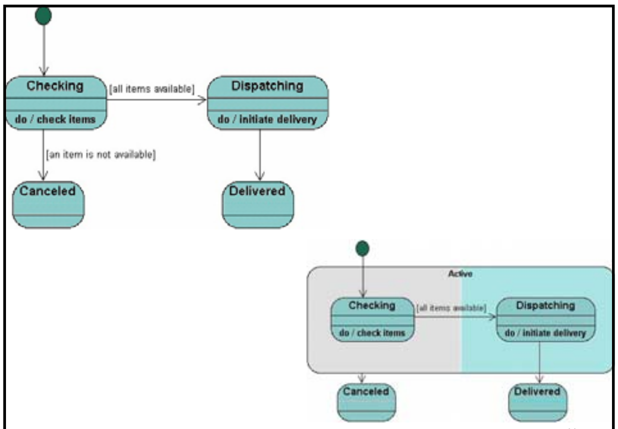
State Diagram

- Rename to state-machine diagram in UML 2.x
- Describe all of the **possible states of an object** as events occur. Each diagram usually represents objects of a single class and track the different states of its objects through the system.
- Use state diagrams to demonstrate the state-dependent behavior of an object through many use cases of the system.
  - Only use state diagrams for classes where it is necessary to understand the behavior of the object through the entire system.
  - Not all classes will require a state diagram and state diagrams are not useful for describing the collaboration of all objects in a use case.
- One object's state diagram may refer to other objects' state. Often combined with collaboration diagram…

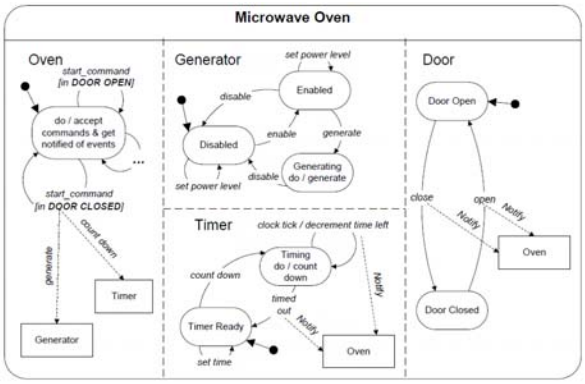
http://pigseye.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/state.htm

Conventions

- Notations:
  - State: rounded rectangles
  - Arrows: events causing state transition. Source of these events are not specified.
  - Filled circle: initial state
  - Hollow circle enclosing filled circle: final state
- Terms:
  - Activities: operations that take time to complete.
  - Actions: operations of very short duration.
  - Events: causing transitions or changes in state. Transitions can tigger actions.

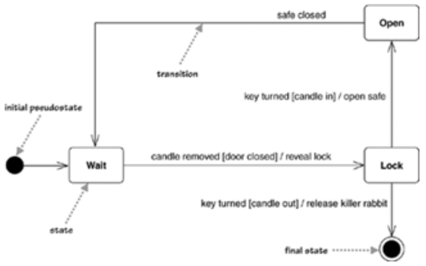


http://pigseye.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/state.htm



http://www.gui.net/portfolio/csci4448/courseNotes/10_UMLStateDiagrams.pdf

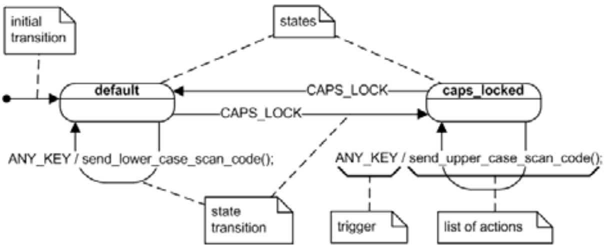
- To keep valuables in a safe that's hard to find. So to reveal the lock to the safe, one have to remove a strategic candle from its holder, but this will reveal the lock only while the door is closed. Once the lock is seen, insert the key to open the safe. For extra safety, the safe can be opened only if the candle is replaced first. If a thief neglects this precaution, a nasty monster will be unleashed to devour him.



safe closed
Open
transition
key turned [candle in] / open safe
initial pseudostate
Wait
candle removed [door closed] / reveal lock
Lock
key turned [candle out] / release killer rabbit
state
final state

Reference: UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition, by Martin Fowler (2003)

Example State Diagram



UML state diagram representing the computer keyboard

http://en.wikipedia.org/wiki/UML_state_machine

Summary

- Brief Introduction to UML Diagrams
  - Use case diagram
  - Class diagram
  - Deployment diagram
  - Sequence diagram
  - Communication/Collaboration diagram
  - Activity digram
  - State (machine) diagram

References for UML

http://www.agilemodeling.com/essays/umlDiagrams.htm

http://www-01.ibm.com/software/rational/uml/

http://atlas.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/index.htm

http://www.sparxsystems.com.au/uml-tutorial.html

UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition, by Martin Fowler (2003), ISBN 0-321-19368-7

Online UML Examples

- Google: uml examples