




Essentials of Visual Modeling w/ UML 2.0 - Instructor Notes

Instructor Notes:

	
IBM Software Group	
Essentials of Visual Modeling with UML 2.0 Module 4: Use-Case Modeling	
	
	

Essentials of Visual Modeling w/ UML 2.0 - Instructor Notes

Instructor Notes:

Introduce the objectives for the module.

Objectives

- ♦ Describe system behavior and show how to capture it in a model.
- ♦ Demonstrate how to read and interpret:
 - A use-case diagram
 - An activity diagram

2

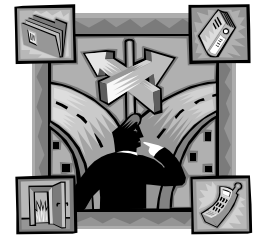


Essentials of Visual Modeling w/ UML 2.0 - Instructor Notes

Instructor Notes:

Where Are We?

- ☆ ♦ Concepts in use-case modeling
 - ♦ Use-case diagrams
 - ♦ Activity diagrams



3



Essentials of Visual Modeling w/ UML 2.0 - Instructor Notes

Instructor Notes:

- ☆ Explain what is meant by "system behavior." How do you capture system behavior?

What Is System Behavior?

- ◆ System behavior is how a system acts and reacts.
 - It comprises the actions and activities of a system.
- ◆ System behavior is captured in use cases.
 - Use cases describe the interactions between the system and (parts of) its environment.

4



No system exists in isolation. Every system interacts with people or automated systems for some purpose. These interactions result in some sort of predictable result. This predictable result is system behavior.

Use cases are the mechanism for capturing the desired behavior for the system that is under development, but do not specify how the behavior is to be implemented.

The UML specifies a model for communicating system behavior, the use-case model.

Essentials of Visual Modeling w/ UML 2.0 - Instructor Notes

Instructor Notes:

☆ Introduce the students to use-case models.
Remember, this is probably the first time that they have seen a use-case model.

Actors and use cases are discussed on the next several slides.

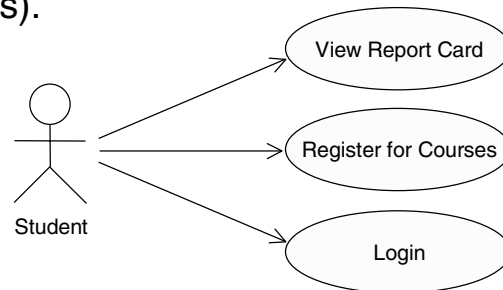
Use this slide to set the context for that discussion.

You might describe a use-case model as a menu. People can place themselves in a role (actor) and see the options available to them on this system.

A use-case model does NOT imply the order that use cases execute.

What Is a Use-Case Model?

- ♦ A model that describes a system's functional requirements in terms of use cases.
- ♦ A model of the system's intended functions (use cases) and its environment (actors).



5



A **use-case model** describes a system's functional requirements in terms of use cases. The use-case model is a model of the system's intended functions and its environment and serves as a contract between the customer and the developers. Because it is a very powerful planning instrument, the use-case model is generally used in all phases of the development cycle.

The customer approves the use-case model. When you have that approval, you know the system is what the customer wants. You can also use the model to discuss the system with the customer during development.

Participants use it to better understand the system.

Designers use it as a basis for their work and to get a system overview.

Testers use it to plan testing activities (use case and integration testing) as early as possible.

Those developing the next version of the system use it to understand how the existing version works.

Documentation writers review the use cases as a basis for writing the system's user guides.

The architect checks the use-case model to identify architecturally significant functionality.

The manager uses it to plan and follow up on the use-case modeling and also the subsequent design.

Essentials of Visual Modeling w/ UML 2.0 - Instructor Notes

Instructor Notes:

Yes, you are selling the concept of a use-case model here. Many of your students may not immediately recognize the need for the use-case model because it is so simple. This slide is intended to bring out some of the problems that this model resolves. Feel free to add your own experiences here.

Talking about this slide, mention:

Communication with the end users and domain experts

- Provides buy-in at an early stage of system development
- Insures a mutual understanding of the requirements

Identification of system users and what the system should do

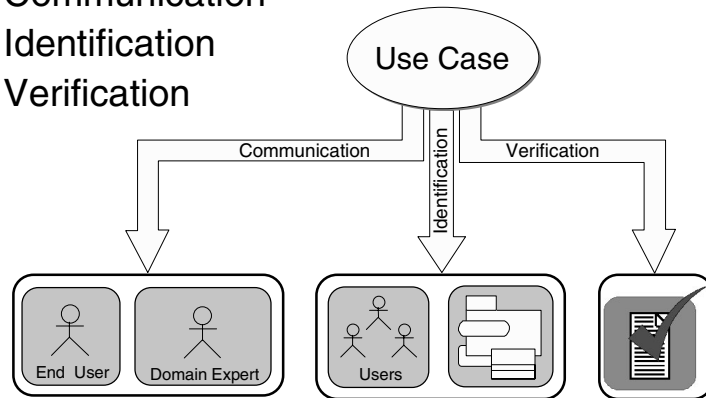
- Considers the requirements for the system interfaces

Verification that all requirements have been captured

- The development team understands the requirements

What Are the Benefits of a Use-Case Model?

- ♦ **Communication**
- ♦ **Identification**
- ♦ **Verification**



6



There are many ways to model a system, each of which may serve a different purpose. However, the most important role of a use-case model is to communicate the system's behavior to the customer or end user. Consequently, the model must be easy to understand.

Communication with the end users and domain experts

- Provide buy-in at an early stage of system development
- Insure a mutual understanding of the requirements

Identification of system users and what the system should do

- The requirements for the system interfaces

Verification that all requirements have been captured

- The development team understands the requirements

Actors are the users and any other system that may interact with the system. Because they represent system users, actors help delimit the system and give a clearer picture of what it is supposed to do. Use cases are developed on the basis of the actor's needs, ensuring that the system turns out to be what the users expected.

Essentials of Visual Modeling w/ UML 2.0 - Instructor Notes

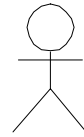
Instructor Notes:

Introduce the two major concepts in use-case modeling.

Do not go into great detail explaining what an actor and use case are in this slide. Detailed actor and use-case slides are coming up soon.

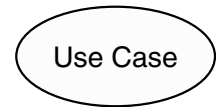
Major Concepts in Use-Case Modeling

- ♦ An actor represents anything that interacts with the system.



Actor

- ♦ A use case describes a sequence of events, performed by the system, that yields an observable result of value to a particular actor.



7

IBM

An **actor** represents a coherent set of roles that one plays when interacting with these use cases. Typically, an actor represents a role that a human, a hardware device, or even another system plays with a system.

A **use case** is a sequence of actions a system performs that yields an observable result of value to a particular actor. A use case describes *what* a system does, but it does not specify *how* it does it.

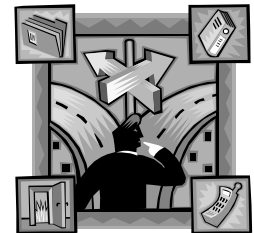
Essentials of Visual Modeling w/ UML 2.0 - Instructor Notes

Instructor Notes:

Transition to the next topic.
The goal of this section is to
simply introduce use-case
diagrams.

Where Are We?

- ◆ Concepts in use-case modeling
- ☆ ◆ Use-case diagrams
- ◆ Activity diagrams



8



Essentials of Visual Modeling w/ UML 2.0 - Instructor Notes

Instructor Notes:

Explain the concept of actors to the students. Remember, this is new to them.

Be sure that the class understands that an actor is NOT JUST A PERSON.

An actor is anything that interacts with the system and is external to the system.

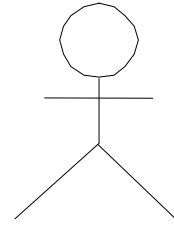
Illustrate this concept by using the example of an ATM machine.

Ask the class to identify some potential actors. Answers may include: bank customer, maintenance worker, bank teller, bank system, credit card system, and so on.

If you aren't allowed to change it then it is an actor.

What Is an Actor?

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



Actor

9



An **Actor** can be defined as:

Anything that exchanges data with the system and is external to the system.

- To fully understand the system's purpose you must know **who** the system is for. Different user types are represented as actors.
- An actor can be a user, external hardware, or another system. An actor may actively interchange information with the system, be a passive recipient of information, or can represent a human, a machine or another system.
- The difference between an actor and an individual system user is that an actor represents a particular class of user rather than an actual user. Several users can play the same role, which means they can be one and the same actor. In which case, each user constitutes an instance of the actor.
- In some situations, only one person plays the role modeled by an actor. For example, there may be only one individual playing the role of system administrator for a rather small system.
- The same user can also act as several actors. That is, the same person can take on different roles.

Essentials of Visual Modeling w/ UML 2.0 - Instructor Notes

Instructor Notes:

Explain the concept of a use case to the students. This material is also new to them.

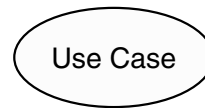
Use cases focus on WHAT the system does, not HOW it does it.

A use case has a set of properties that includes a brief description, flow of events, special requirements, activity diagrams, and so on. These are discussed in more detail later in this module.

Use cases are enclosed in the use-case model artifact. That is, use cases are properties of the use-case model.

What Is a Use Case?

- ◆ Defines a set of use-case instances, where each instance is a sequence of actions a system performs that yields an observable result of value to a particular actor.
 - A use case models a dialogue between one or more actors and the system
 - A use case describes the actions the system takes to deliver something of value to the actor



10



A **Use-Case** can be defined as:

A sequence of **actions** a **system performs** that yields an **observable result of value to a particular actor**.

- **Actions** - An action is a computational or algorithmic procedure. It is invoked either when the actor provides a signal to the system or when the system receives a time event. An action may imply signal transmissions to either the invoking actor or other actors. An action is atomic. That is, it is performed either entirely or not at all.
- **System performs** - The system provides the use case. An actor communicates with a use-case instance of the system.
- **An observable result of value** - You can put a value on a successfully performed use case. A use case should ensure that an actor can perform a task that has an identifiable value. This is important to determine the correct level or granularity for a use case. Correct level refers to achieving use cases that are not too small. In certain circumstances, you can use a use case as a planning unit that includes individuals playing the role of an actor to the system.
- **A particular actor** - The actor is key to finding the correct use case because the actor helps to avoid use cases that are too large. As an example, consider a visual modeling tool. There are two actors to this application: a developer, someone who develops systems using the tool as support and a system administrator, someone who manages the tool. Each of these actors has its own demands on the system and requires its own set of use cases.

Essentials of Visual Modeling w/ UML 2.0 - Instructor Notes

Instructor Notes:

Show the relationship between use cases and actors.

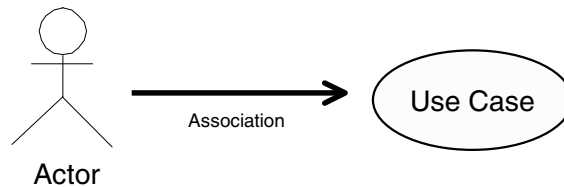
A use case can initiate communication with an actor. Usually, this occurs with a non-human actor.

The <<communicate>> association stereotype was valid for the UML1 example profiles UML Profile for Software Development Process, which is based on the Unified Process for software engineering, and the UML Profile for Business Modeling.

Since an association is the only valid relationship that can exist between actors and use cases, UML 2 has dropped the stereotype.

Use Cases and Actors

- ♦ A use case models a dialog between actors and the system.
- ♦ A use case is initiated by an actor to invoke a certain functionality in the system.



11



It is important to show how actors relate to the use case. Therefore, on finding a use case, establish the actors that interact with it. To do this, you must define an association that helps to clarify the communication between the actor and use case.

Actors may be connected to use cases only by an association. An association between an actor and a use case indicates that the actor and the use case instance of the system communicate with one another, each one able to send and receive messages. The arrow head is optional but it's commonly used to denote the initiator.

Essentials of Visual Modeling w/ UML 2.0 - Instructor Notes

Instructor Notes:

In this slide, give students a chance to read a use-case diagram.

Answers to student notes:

1. Student can perform: View Report Card, Register For Courses, and Login. A Professor can: Login, Select Courses to Teach, and Submit Grades. The Course Catalog is involved in: Register for Courses and Select Courses to Teach. The Registrar is involved in: Login, Maintain Professor Information, Maintain Student Information, and Close Registration. The Billing System is involved in: Close Registration.

Disclaimer: Login is a controversial use case. The goal of this course is not to determine when/how/why one should use the Login use case, it is part of the *Mastering Object Oriented Analysis and Design with UML* curriculum so that instructors can have a short use case to demonstrate exercises. It is only here for instructional purposes.

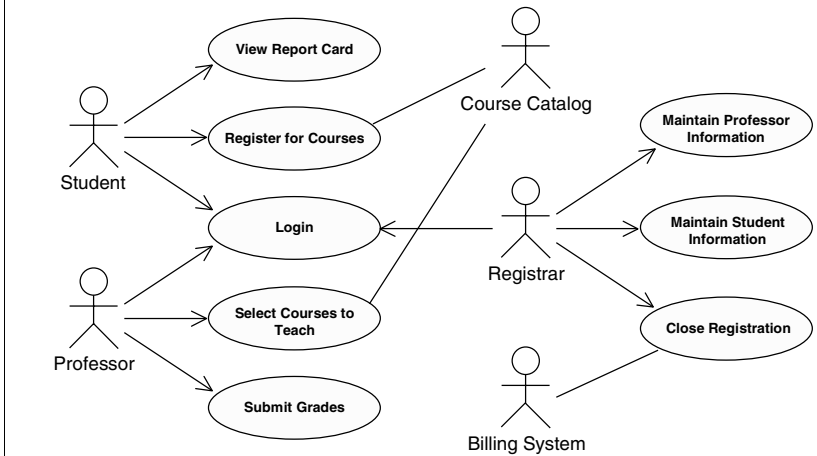
2. Charlie can: View Report Card, Register for Courses, Login, Select Courses to Teach, and Submit Grades.

3. This is a Course Registration System.

4. The Professor initiates the Select Courses to Teach and the Course Catalog is a participant; the Registrar initiates the Close Registration and the Billing System is a participant.

5. Of course, this is a trick question. You can't make that assumption from looking at this model. It isn't intended to show order.

How Would You Read This Diagram?



Answer the following questions:

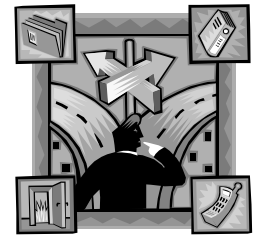
1. Which use cases can a student perform? A professor? The Course Catalog?
2. If Charlie is a student and professor, which use cases can he execute?
3. Describe the functionality of this system.
4. Describe the actor relationships for the Close Registration and Select Courses To Teach use cases.
5. Which use case needs to run first, Register for Courses or View Report Card?

Essentials of Visual Modeling w/ UML 2.0 - Instructor Notes

Instructor Notes:

Where Are We?

- ◆ Concepts in use-case modeling
- ◆ Use-case diagrams
- ☆◆ Activity diagrams



13



Essentials of Visual Modeling w/ UML 2.0 - Instructor Notes

Instructor Notes:

The goal of this section is to introduce the students to the concept of an activity diagram. You are not expected to teach them everything about this diagram at this time.

Activity diagrams can also be used to model the workings of an operation, an object, business modeling, or anything that involves modeling the sequential steps in a computational process.

This course focuses on using activity diagrams to model the flow of events in a use case.

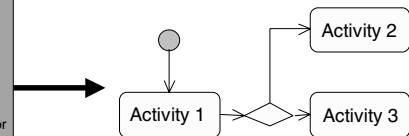
What Is an Activity Diagram?

- ♦ An activity diagram in the use-case model can be used to capture the activities and actions performed in a use case.
- ♦ It is essentially a flow chart, showing flow of control from one activity or action to another.

Flow of Events

This use case starts when the Registrar requests that the system close registration.

1. The system checks to see if registration is in progress. If it is, then a message is displayed to the Registrar and the use case terminates. The Close Registration processing cannot be performed if registration is in progress.
2. For each course offering, the system checks if a professor has signed up to teach the course offering and at least three students have registered. If so, the system commits the course offering for each schedule that contains it.



14



The workflow of a use case describes that which needs to be done by the system to provide the value the served actor is looking for.

It consists of a sequence of activities and actions that together produce something for the actor.

The workflow often consists of a basic flow and one or several alternative flows.

The structure of the workflow can be described graphically with the help of an activity diagram.

Essentials of Visual Modeling w/ UML 2.0 - Instructor Notes

Instructor Notes:

If a high-level activity is viewed as a tree of nested activities, the leaves of the tree are actions.

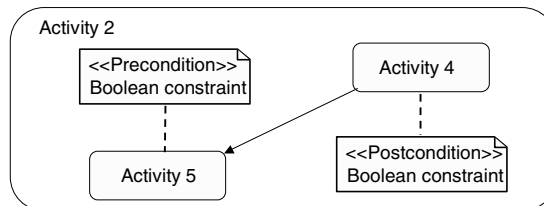
Actions include arithmetic and string functions, manipulations of objects and their values, communications among objects, and similar things.

Kinds of actions include:

- Accept call
- Accept event
- Apply function
- Broadcast event
- Call
- Create
- Destroy
- Raise exception
- Read
- Reply
- Return
- Send
- Time
- Write

What Is an Activity?

- ♦ A specification of behavior expressed as a flow of execution via sequencing of subordinate units.
 - Subordinate units include nested activities and ultimately individual actions.
- ♦ May contain boolean expression constraints when the activity is invoked or exited



An activity is notated as an activity diagram. An activity definition is shown as a large rounded border containing a graph of node symbols and flow arrows representing the decomposition of the activity into its constituents. Activity preconditions and postconditions use the note notation with the keywords <<Precondition>> and <<Postcondition>> respectively.

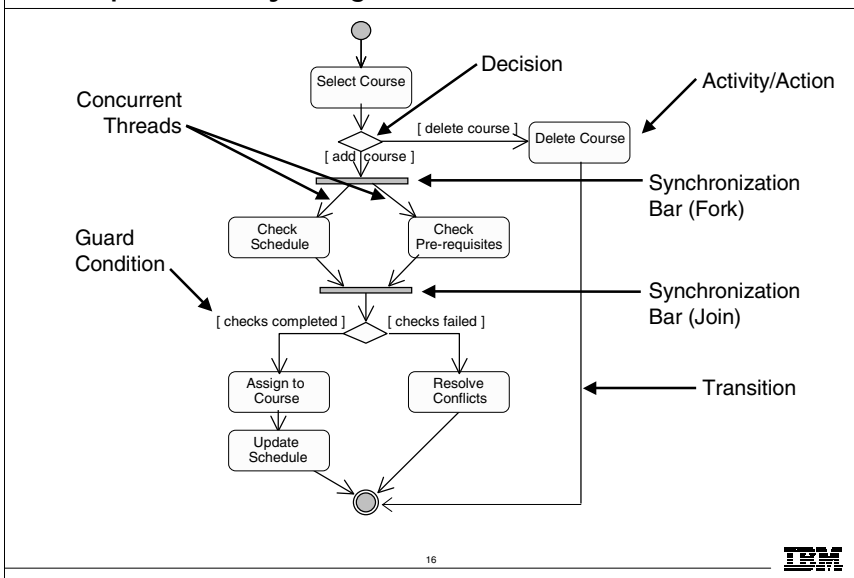
An action is a primitive activity which is the smallest computation that can be expressed. An action is an activity that *does* something to the state of the system or extracts information from it. An action is drawn as a rectangle with rounded corners. Action preconditions and postconditions use the note notation with the keywords <<localPrecondition>> and <<localPostcondition>> respectively.

Essentials of Visual Modeling w/ UML 2.0 - Instructor Notes

Instructor Notes:

Walk the students through the activity diagram and explain each component (decision, fork, join, and so on).

Example: Activity Diagram



An activity diagram may include the following elements:

- **Activity/Action** represents the performance of a step within the workflow.
- **Transitions** show the activity/action that follows.
- **Decisions** evaluate conditions defined by guard conditions. These guard conditions determine which of the alternative transitions will be made and, thus, which activities are performed. You may also use the decision icon to show where the threads merge again. Decisions and guard conditions allow you to show alternative threads in the workflow of a use case.
- **Synchronization bars** show parallel sub-flows. They allow you to show concurrent threads in the workflow of a use case.

Essentials of Visual Modeling w/ UML 2.0 - Instructor Notes

Instructor Notes:

A. **System behavior** is how a system acts and reacts. It is an outwardly visible and testable activity of a system.

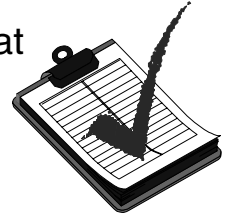
B. A **use-case model** describes a system's functional requirements in terms of use cases. It is used to communicate with the end users and the domain experts. A **benefit** includes buy-in at an early stage of system development.

C. An **actor** is anything that exchanges data with the system and is external to the system. A **use case** is a sequence of actions a system performs that yields an observable result of value to a particular actor.

D. An **activity diagram** in the use-case model can be used to capture the activities in a use case. It is essentially a flow chart, showing flow of control from activity to activity.

Review

- ♦ What is system behavior?
- ♦ What is a use-case model? What are its benefits?
- ♦ What is an actor? A use case?
- ♦ What is an activity diagram?



17

IBM

Essentials of Visual Modeling w/ UML 2.0 - Instructor Notes

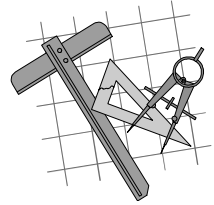
Instructor Notes:

The goal of the exercise is to have the students draw a UML diagram. Remember, all of this is new to them and they may struggle with drawing the diagrams the right way.

The “givens” are provided in the student notes for this slide. The students are provided with a textual description of a use case and must then build a diagram that accurately reflects the textual description.

Exercise

- ♦ **Given:**
 - Use cases, actors and associations
- ♦ **Draw:**
 - A use-case diagram
- ♦ **Given:**
 - Action states and activity edges
- ♦ **Draw:**
 - An activity diagram



18



1. Draw a use-case diagram using the following elements:

- There are four actors on the diagram: Prospective Buyer, E-Mail System, Loan System, and Credit Reporting System.
- There are four use cases on the diagram: Find Realtor, Maintain Personal Planner, Search for a Home, and Apply for a Loan.
- Document the following association;
 - Prospective Buyer to Find Realtor
 - Prospective Buyer to Maintain Personal Planner
 - Prospective Buyer to Search For A Home
 - Prospective Buyer to Apply For A Loan
 - Maintain Personal Planner to E-Mail System
 - Search for a Home to E-Mail System
 - Apply for a Loan to Loan System
 - Apply for a Loan to Credit Reporting System

Study the diagram that you have drawn. What does it say? What doesn't it say? Describe the functionality that a Prospective Buyer expects from the system?

2. Draw an activity diagram that reflects the following four action states:

Choose Profile, Find Buyer Profile, Log on, and Create New Profile. Starting with Choose Profile, go to Find Buyer Profile; then go from the Find Buyer Profile to Create New Profile, providing a profile does NOT exist. If a profile does exist, you can go to Log on.