



UML Tutorial Part 1: UML Use Cases



Presented by Igor Ivković
iivkovic@swen.uwaterloo.ca

Why UML Use Cases?

- **Use Case**
 - A set of scenarios related by a common actor and a goal
 - A description of sequences of actions performed by a given system to produce a result for an actor
- Use cases specify the expected behaviour [**what**], and not the exact method of making it happen [**how**]
- Use cases are created based on identified functional requirements but are not mapped one-to-one to requirements
- Use cases once specified can be denoted using a clear and precise visual modeling language such as UML

Agenda

- Terms and Concepts
 - Identifying Actors, Use Cases, and Scenarios
 - UML Use Cases Examples
 - Summary and References

3

Use Case Names

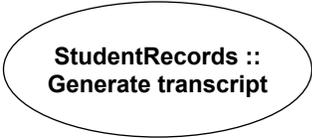
- **Use Case Names**

- Simple Name



Register student

- Path Name
Package name :: Use case name



**StudentRecords ::
Generate transcript**

4

Actors

- **Actors**

- Represent roles that humans, hardware devices, or external systems play while interacting with a given system
- They are not part of the system and are situated outside of the system boundary
- Actors may be both at input and output ends of a use case



5

Scenarios

- **Scenarios**

- Specify behaviour of use case by description, not modeling
 - Examples include informal structured text, formal structured text with conditions, and pseudocode
- Typically specifies:
 - How and when the use case starts and ends
 - Interaction with the actors and the exchange of objects
 - Flow of events: main / typical (success), alternative (success), and exceptional (failure) flows

6

Scenarios Example

- Scenarios Example
 - In a human resources system, for the “Hire Employee” use case, the following scenarios may apply:
 - Typical success scenario: Hire a person from outside of the company, for example, from another company
 - Alternative success scenario: Hire a person from within the same company, for instance, from another division
 - Exceptional failure scenario: No qualified person could be hired

7

Conditions and Quality Requirements

- **Entry and Exit Conditions**
 - Entry conditions describe the environment under which the use case is invoked
 - Exit conditions reflect the impact of the use case on the environment through its execution
- **Quality Requirements**
 - Describe quality attributes in terms of a specific functionality
 - For example, requires system response in < 30 seconds

8

Relationships

- **Relationships**

- Organize use cases by grouping them in packages as in the organization of classes
- *Generalization*: The child use case exhibits a more specific variation in behavior than as specified for its parent
- *Include*: Common behaviour of more than one use case is referenced as a separate instance to avoid repetition
- *Extend*: Implicit integration of the behaviour of another use case by declaring the extension points / events in the base

9

Agenda

- ✓ Terms and Concepts
- Identifying Actors, Use Cases, and Scenarios
 - o UML Use Cases Examples
 - o Summary and References

10

Identifying Actors

- Define system boundary to identify actors correctly
- Identify users and systems that depend on the system's primary and secondary functionalities
- Identify hardware and software platforms with which the system interacts
- Select entities that play distinctly different roles in the system
- Identify as actors external entities with common goals and direct interaction with the system
- Denote actors as nouns

11

Identifying Use Cases

- Business / Domain Use Cases:
 - Interactions between users and the business (or domain)
- System Use Cases:
 - Interactions between users and the system
 - One business use cases contains a set of system use cases
- To name the use cases, give it a verb name to show the action that must be performed
 - Describe a transaction completely
 - No description of user interface whatsoever

12

Capture Use Cases

- Capture use cases during requirements elaboration
- Use cases are not mapped one-to-one to requirements
 - Each requirement must be covered by at least one use case
 - However, use cases may contain many requirements
- Use scenarios to model assumptions and define system scope
- List exceptions separately

13

Identifying Scenarios

- Extract the functionality that is available to each actor
- Establish specific instances and not general descriptions
- Denote situations in the current and future systems
- Identify:
 - Tasks to be performed by the user and the system
 - Flow of information to the user and to the system
 - Events that are conveyed to the user and to the system
 - For the events flow, name steps in active voice

14

Agenda

- ✓ Terms and Concepts
- ✓ Identifying Actors, Use Cases, and Scenarios
- UML Use Cases Examples
- o Summary and References

15

Courseware System Description

▪ Informal Description:

- For this case study, the task is of constructing the design elements for a system that can be used to manage courses and classes for an organization that specializes in providing training. The name of the system is Courseware System. The organization offers courses in a variety of areas such as learning management techniques and understanding different software languages and technologies. Each course is made up of a set of topics. Tutors in the organization are assigned courses to teach according to the area that they specialize in and their availability. The organization publishes and maintains a calendar of the different courses and the assigns tutors every year. There is a group of course administrators in the organization who manage the courses including course content, assigning courses to tutors, and defining the course schedule. The training organization aims to use the Courseware System to get a better control and visibility to the course management and to also streamline the process of generating and managing schedules for different courses.

http://www.developer.com/design/article.php/10925_2109801_4

16

Courseware Overview

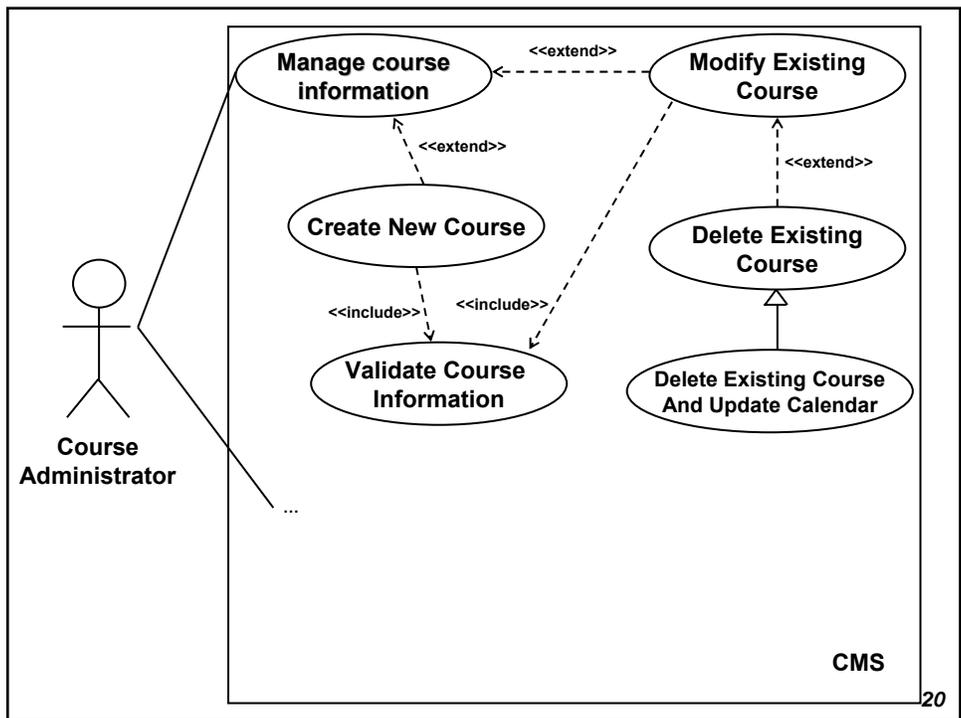
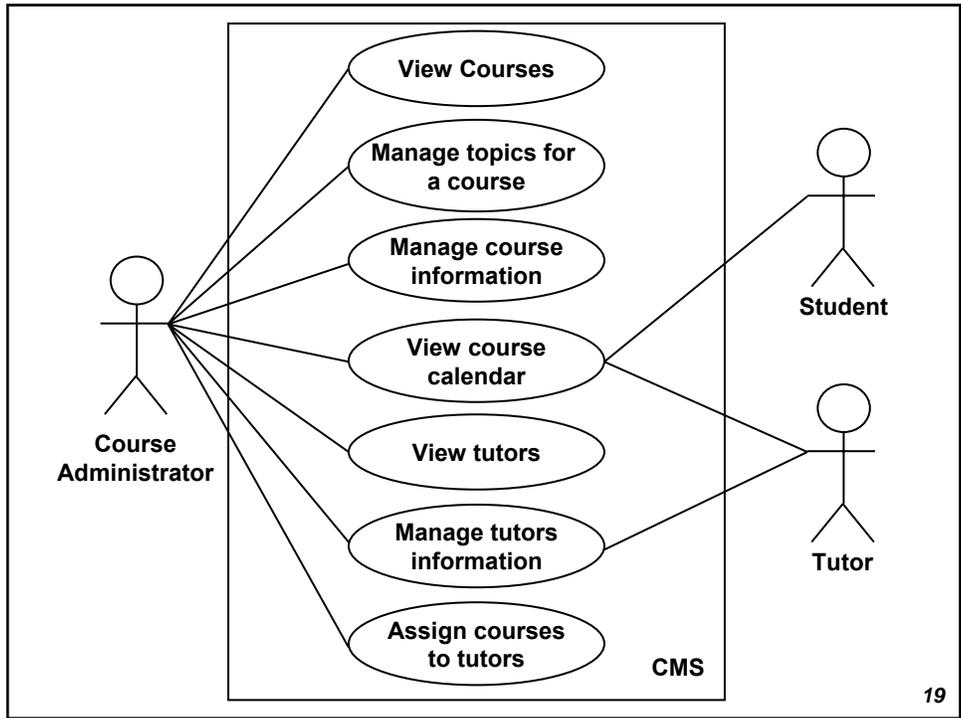
- The following terms and entities are specific to the system:
 - Courses and Topics that make up courses
 - Tutors that teach courses
 - Course Administrators who manage the assignment of courses to tutors
 - Calendars and Course Schedules that are generated as a result of the work performed by the course administrators
 - Students who refer to Calendars and Course Schedules to decide which courses they wish to take up for study

17

Courseware Actors and Use Cases

- Actors: Tutor, Student, Course Administrator (main actor)
- Use Cases (primary business: secondary user)
 - Manage courses: View courses, Manage topics for a course, and Manage course information
 - Manage tutors: View course calendar, View tutors, Manage tutor information, and Assign courses to tutors

18



Courseware Scenarios Example /1

- **Use Case: Manage Course Information (UC_ID1)**
 - Participating Actors: Course Administrator
 - Entry Conditions: Course Administrator is logged into CourseWare
 - Exit Conditions: Course Administrator has received an acknowledgement from the system that the selected transaction is complete, or if not complete, a message explaining the failure
 - Quality Requirements: (Performance) Course Administrator receives a response from the system in less than 3 seconds
 - Related Requirements: Create, Modify, and Delete Course
 - ...

21

Courseware Scenarios Example /2

- **Use Case: Manage Course Information (UC_ID1)**
 - Typical flow of events:
 1. Course Administrator selects Create New Course
 - a) System invokes Create New Course use case
 2. Course Administrator selects Modify Existing Course
 - a) System invokes Modify Existing Course use case

22

Courseware Scenarios Example /3

- **Use Case: Create New Course (UC_ID2)**
 - Participating Actors: Course Administrator
 - Extends: Manage Course Information (UC_ID1)
 - Entry Conditions: Course Administrator has selected Create New Course option
 - Exit Conditions: Course Administrator has received an acknowledgement from the system that a course has been created, or if not, a message explaining the failure
 - Quality Requirements: (Performance) Course Administrator receives a response from the system in less than 3 seconds
 - Related Requirements: Create Course
 - ...

23

Courseware Scenarios Example /4

- **Use Case: Manage Course Information (UC_ID2)**
 - Typical flow of events:
 1. Course Administrator enters New Course Information
 - a) System invokes Validate Course Information use case
 - b) For a valid response, system creates a new course entry and sends an acknowledgment back to the actor
 - Exceptions:
 1. Course Administrator enters New Course Information
 - b) Invalid response received, so system reports failure with a message indicating invalid course information

24

Courseware Scenarios Example /5

- **Use Case: Modify Existing Course (UC_ID3)**
 - Participating Actors: Course Administrator
 - Extends: Manage Course Information (UC_ID1)
 - Entry Conditions: Course Administrator has selected Modify Existing Course option
 - Exit Conditions: Course Administrator has received an acknowledgement from the system that a course has been modified, or if not, a message explaining the failure
 - Quality Requirements: (Performance) Course Administrator receives a response from the system in less than 3 seconds
 - Related Requirements: Modify and Delete Course
 - ...

25

Courseware Scenarios Example /6

- **Use Case: Modify Existing Course (UC_ID3)**
 - Typical flow of events:
 1. Course Administrator selects Find Existing Course option
 - a) System searches for a selected course and returns existing course information
 2. Course Administrator enters new course information
 - a) System invokes Validate Course Information use case
 - b) For a valid response, system updates the existing course entry and sends an acknowledgment back to the actor
 - ...

26

Courseware Scenarios Example /7

- **Use Case: Modify Existing Course (UC_ID3)**
 - Alternatives:
 2. Course Administrator selects Delete Existing Course option
 - a) System invokes Delete Existing Course use case
 - Exceptions:
 1. Course Administrator selects Find Existing Course option
 - a) System searches for a selected course and returns failure stating that the course could not be found
 2. Course Administrator enters New Course Information
 - b) Invalid response received, so system reports failure with a message indicating invalid course information

27

Courseware Scenarios Example /8

- **Use Case: Delete Existing Course (UC_ID4)**
 - Participating Actors: Course Administrator
 - Extends: Modify Existing Course (UC_ID3)
 - Entry Conditions: Course Administrator has selected Delete Existing Course option
 - Exit Conditions: Course Administrator has received an acknowledgement from the system that a course has been deleted, or if not, a message explaining the failure
 - Quality Requirements: (Performance) Course Administrator receives a response from the system in less than 3 seconds
 - Related Requirements: Delete Course
 - ...

28

Courseware Scenarios Example /9

- **Use Case: Delete Existing Course (UC_ID4)**
 - Typical flow of events:
 - a) System deletes a selected course and sends an acknowledgment back to the user
 - Exceptions:
 - a) System cannot delete a selected course so it returns failure stating that the course could not be deleted

29

Agenda

- ✓ Terms and Concepts
- ✓ Identifying Actors, Use Cases, and Scenarios
- ✓ UML Use Cases Examples
- Summary and References

30

Tutorial Summary

- In this tutorial, we have introduced UML use cases
- We have revisited key elements of use cases including actors, scenarios, conditions, quality requirements, and relationships
- Through examples, we have demonstrated how to write fully decorated use cases
- We have also shown how to create complete UML use case diagrams including different types of use case associations

31

References

- G. Booch, J. Rumbaugh and I. Jacobson. *The UML User Guide*, Addison-Wesley, 1999.
- M. Fowler and K. Scott. *UML Distilled*, Addison-Wesley, 2000.
- B. Bruegge and A. H. Dutoit, *Object-Oriented Software Engineering*, Prentice Hall, 2004.

32