Overview

Chapter 4 discusses Steps 2, 3, and 4 of the process for object-oriented analysis. It introduces the Unified Modeling Language (UML), a standard notation for describing object-oriented systems.

The use case model and system sequence diagrams presented here begin to describe the system’s required response to events in its environment.
Use case diagrams show the use cases within the scope of the system and the actors in the environment with which each use case is associated.

There is at least one use case narrative for each use case. A use case narrative is a structured narrative showing what the system must do to respond to a specific event.

We write an expanded use case narrative for every use case corresponding to an external event; these are the use cases in which there is input to the system.

An expanded use case narrative captures the sequence of messages from an actor to the system as well as the system's response to each message. The narrative also reveals the structure and content of the messages.

Use case narratives for analysis should be essential – they should not mention any possible implementing technology.

A system sequence diagram shows graphically the sequence, structure, and content of each message from an actor to the system. It also shows messages from the system to external systems.
There is one system sequence diagram for each use case corresponding to an external event. If the use case is complex, several system sequence diagrams may be desirable.

The specific data elements contained in each message are usually shown in a system sequence diagram.

Use Case Heuristic

- There should be COMPLETE set of use cases
  - Nothing missing
    - Consider cases, alternatives, options, errors, recoveries
  - But, Don't need to generate lots of similar cases with minor differences.
    - Each case should add new, interesting, information

Step 2 of Object-Oriented Systems Analysis

Identify the use cases and produce a use case diagram for the system.
Use Cases

A use case is the sequence of actions which occur when an actor uses a system to complete a process.

A use case is a model of a requirement.

A use case name is a short phrase beginning with a verb.

Each event corresponds to at least one use case.

Events and Use Cases

<table>
<thead>
<tr>
<th>Event</th>
<th>Use case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department submits class schedule.</td>
<td>Submit Department Class Schedule.</td>
</tr>
<tr>
<td>Time to produce University Class Schedule</td>
<td>Produce University Class Schedule.</td>
</tr>
<tr>
<td>Student registers for classes.</td>
<td>Register for Classes.</td>
</tr>
<tr>
<td>Time to produce Class Roster.</td>
<td>Produce Class Roster.</td>
</tr>
</tbody>
</table>

Identifying Actors

An actor is a person, organization, or system which interacts with a system by sending messages to the system or receiving messages from the system.

Actors play roles with respect to the system.

The actors were identified during event analysis.
Types of Actors

An **initiating actor** initiates a use case by initiating an **external event**. Thus, initiating actors provide system inputs.

A **participating actor** is involved in a use case but does not initiate it. Thus, participating actors receive system outputs.

The Use Case Model

1. A **use case diagram** for the entire system – a **graphic model**

2. One or more **use case narratives** for each use case – descriptions in **text**
   - high-level
   - expanded

In the UML, use cases are part of the **User View Model**

Use Case Diagram

![Use Case Diagram](image-url)
Components of a Use Case Diagram

Actor: A named stick figure

Use case: An oval containing the use case name

Association between an initiating actor and a use case: a line with a stick arrowhead

Association between a participating actor and a use case: a line

The system boundary or a subsystem boundary may be shown as a rectangle.

Show Off Time

• Who here is a good requirements thinker?
  – Willing to answer a few questions
• Let’s make a use case diagram
  – Think like a careful customer for a new system….

Step 3 of Object-Oriented Systems Analysis

Write a use case narrative describing the system’s response to each business event.
High-Level Use Case Narrative

**Use case:** Register for Classes

**Actors:** Student

**Purpose:** Register a student for classes and record the student's schedule.

**Description:** A student requests the sections of classes desired for a term. The system adds the student to each section if space is available. On completion, the student receives a list of the classes in which he or she is enrolled.

---

Expanded Use Case Narrative

**Use case:** Register for Classes

**Actors:** Student

**Purpose:** Register a student for classes and record the student's schedule.

**Overview:** The system adds the student to each section if space is available. On completion, the student receives a list of the classes in which he or she is enrolled.

**Type:** Essential

**Preconditions:** Student is known by the system.

**Postconditions:** Student may register for the class.

**Special Requirements:** System must respond within 10 seconds.

**Flow of Events:**

**Actor Action** | **System Response**
--- | ---
1. The registration process begins when a student requests a class.
2. The student provides the student's name, major, and the desired course.
3. The system checks the availability of the desired course.
4. The student confirms the course selection.
5. The student receives a confirmation of the registration.
6. The system reserves the student's seat.

---

**Alternative Flow of Events**

Use Case Name: Student

Actors: Student

Purpose: Register a student for classes and record the student's schedule.

Overview: The system adds the student to each section if space is available. On completion, the student receives a list of the classes in which he or she is enrolled.

Type: Essential

Preconditions: Student is known by the system.

Postconditions: Student may register for the class.

Special Requirements: System must respond within 10 seconds.

Flow of Events:

**Actor Action** | **System Response**
--- | ---
1. The registration process begins when a student requests a class.
2. The student provides the student's name, major, and the desired course.
3. The system checks the availability of the desired course.
4. The student confirms the course selection.
5. The student receives a confirmation of the registration.
6. The system reserves the student's seat.

---

Alternative Flow of Events:

Use Case: Student

Actors: Student

Purpose: Register a student for classes and record the student's schedule.

Overview: The system adds the student to each section if space is available. On completion, the student receives a list of the classes in which he or she is enrolled.

Type: Essential

Preconditions: Student is known by the system.

Postconditions: Student may register for the class.

Special Requirements: System must respond within 10 seconds.

Flow of Events:

**Actor Action** | **System Response**
--- | ---
1. The registration process begins when a student requests a class.
2. The student provides the student's name, major, and the desired course.
3. The system checks the availability of the desired course.
4. The student confirms the course selection.
5. The student receives a confirmation of the registration.
6. The system reserves the student's seat.

---

Alternative Flow of Events:
Terms in an Expanded Use Case Narrative

**Precondition:** A condition which must be true in order for the use case to begin and produce the desired results.

**Postcondition:** A condition which must be true after the use case has been completed.

---

**Pre and Post Conditions**

**Precise Systems Thinking**

– **Precondition**
  • Assumptions for this part of the system
  • Guaranteed to be met by something else
  • System set up to be sure they are met
  • **No need to check them here**
    – Since they are guaranteed!

– **Postcondition**
  • What is promised to those coming later
  • System state: verifications, updates, changes, deletions…
  • May be something else’s precondition
  – **When used later on**

---

**Terms in an Expanded Use Case Narrative (continued)**

**Special Requirements:** A requirement which is critical to users’ acceptance and use of the system.

**Alternative Flow of Events:** What the system should do in the case of exceptional conditions or errors.
Associations Between Use Cases

The «includes» association always occurs when the use case which includes it occurs.

The occurrence of an «extends» association depends on a true condition in the use case which it extends.

Use Case Scenarios

A use case scenario is a narrative of a single occurrence of a use case. It describes specifics of a real-world enactment of the use case.

Use case scenarios can help discover alternative paths through a use case or test the completeness or correctness of a use case narrative.
Step 4 of Object-Oriented Systems Analysis

Draw a system sequence diagram for each use case scenario.

System Sequence Diagram

A system sequence diagram shows the interaction between an actor and the system for one use case scenario.

It shows:
- The system (as a black box)
- The initiating actor
- Each external system which sends messages to the system
- The messages into and out of the system
- The sequence in which the messages occur

System Sequence Diagram (continued)

Note carefully the UML conventions for the arrows and arrowheads.
Show Off Time

• Who here is a good system thinker?
  – Willing to answer a few questions
• Let’s make an easy, informal sequence diagram
  – Think like a system designer….

System Sequence Diagram

(continued)

A system sequence diagram focuses on the **content and structure** of the system **input**.

It should show whether any messages are **repeated** or are **alternatives**.

A system sequence diagram is **not** the place to show the design of the detailed interaction between the user and the system.

Creating a System Sequence Diagram

• Draw a rectangle representing the **system**. Label the rectangle and draw a lifeline beneath it.

• At the left, draw a stick figure for each **actor**. Label it with the actor’s name and draw a lifeline beneath it.
Creating a System Sequence Diagram (continued)

• For each system input, draw a message arrow from the actor's lifeline to the system's lifeline. Label it with the message name and parameters.

• Confirm that the sequence of messages (from top to bottom) is correct.

Message Formats in System Sequence Diagrams

The UML format for a message consists of a message name followed (in parentheses) by a parameter list.

All names begin with a lower-case letter. There are no spaces in a name. Upper-case letters separate the words within a name.

Names in the parameter list are separated by commas.

Outgoing Messages (System Outputs)

There are two types of outgoing messages:

• A response of the system which completes an event

• A message from the system to an external system requesting action and a reply

Remember that every output must be derivable from the input to the use case combined with stored data.
Review Questions

• See text Review Questions:
  – 4-2, 4-3, 4-4 in particular

• Key Ideas (Deep Knowledge):
  – Use Cases and Sequence Diagrams
    • Know the details of the mechanics of the diagrams
  – Precise Requirements—terminology, steps, relationships, completeness

Learning Objectives

• Identify use cases based on prior event analysis.
• Derive a use case diagram from an event table.
• Draw a use case diagram.
• Discover use case scenarios.

Learning Objectives (continued)

• Write both high-level and expanded essential use case narratives.
• Define system inputs and outputs.
• Draw a system sequence diagram for a use case scenario.