面向对象程序的分析与设计
Object-Oriented Analysis and Design

Lecture 3

Prof. S. Xu

Contents

- Use Case
- Use case Model

Customize UP, The UP Development Case

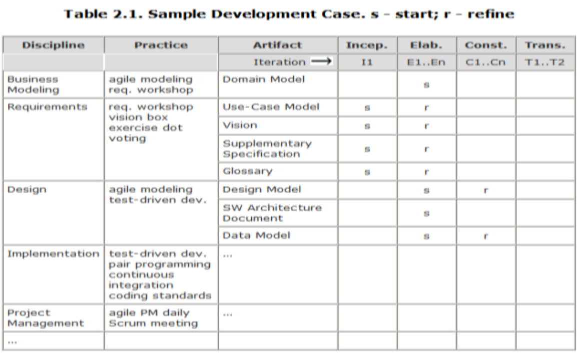
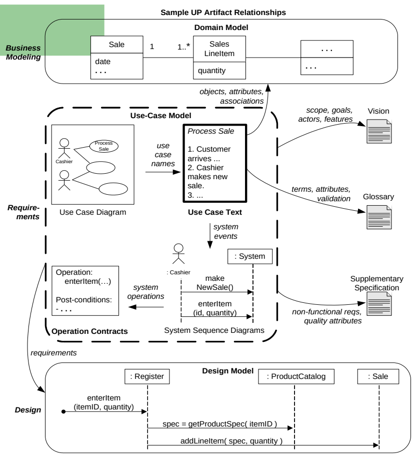


Table 2.1. Sample Development Case. s - start; r - refine

Discipline	Practice	Artifact	Incep.	Elab.	Const.	Trans.
		Iteration ⟶	I1	E1..En	C1..Cn	T1..T2
Business Modeling	agile modeling req. workshop	Domain Model		s		
Requirements	req. workshop vision box exercise dot voting	Use-Case Model	s	r		
		Vision	s	r		
		Supplementary Specification	s	r		
		Glossary	s	r		
Design	agile modeling test-driven dev.	Design Model		s	r	
		SW Architecture Document		s		
		Data Model		s	r	
Implementation	test-driven dev. pair programming continuous integration coding standards	...				
Project Management	agile PM daily Scrum meeting	...				
...						



Sample UP Artifact Relationships

Use Case

Example

- Next Generation POS system

Use Cases

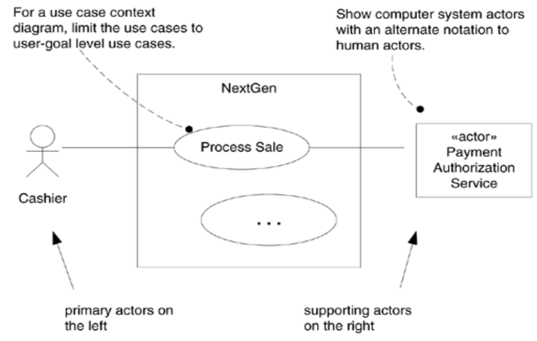
- Informally, use cases are ***text stories of some actor using a system to meet goals***.
- The essence of use cases is discovering and recording functional requirements by writing stories of using a system to fulfill user goals;

Process Sale: A customer arrives at a checkout with items to purchase. The cashier uses the POS system to record each purchased item. The system presents a running total and line-item details. The customer enters payment information, which the system validates and records. The system updates inventory. The customer receives a receipt from the system and then leaves with the items.

7

Use Case Diagrams

Figure 6.4. Notation suggestions.


For a use case context diagram, limit the use cases to user-goal level use cases.
Show computer system actors with an alternate notation to human actors.
NextGen
Process Sale
. . .
Cashier
«actor»
Payment Authorization Service
primary actors on the left
supporting actors on the right

Actor

- An ***actor*** is something with behavior, such as a person (identified by role), computer system, or organization, who uses the system
  - for example, a cashier.

9

Three kinds of actors

- Primary actor
  - has user goals fulfilled through using services of the SuD (system under discussion).
  - For example, the cashier.
- Supporting actor
  - provides a service (for example, information) to the SuD.
  - The automated payment authorization service is an example.
- Offstage actor
  - has an interest in the behavior of the use case, but is not primary or supporting;
  - for example, a government tax agency.

10

Scenario

- A ***scenario*** is a specific sequence of actions and interactions between actors and the system; it is also called a *use case instance*.

11

Use Cases

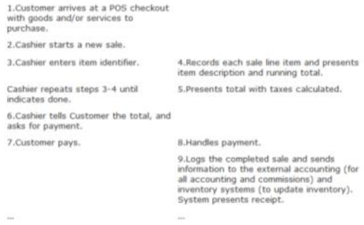
- one format of use case, two column or one column?

Use Case UC1: Process Sale

Primary Actor: ...

... as before ...

Main Success Scenario:



Actor Action (or Intention)	System Responsi
1.Customer arrives at a POS checkout with goods and/or services to purchase.	
2.Cashier starts a new sale.	
3.Cashier enters item identifier.	4.Records each sale line item and presents item description and running total.
Cashier repeats steps 3-4 until indicates done.	5.Presents total with taxes calculated.
6.Cashier tells Customer the total, and asks for payment.	
7.Customer pays.	8.Handles payment.
	9.Logs the completed sale and sends information to the external accounting (for all accounting and commissions) and inventory systems (to update inventory). System presents receipt.
...	...

12

Use Cases

- A use case is a collection of related success and failure scenarios that describe an actor using a system to support a goal.

More example:

Handle Returns

- Main Success Scenario:
  - A customer arrives at a checkout with items to return. The cashier uses the POS system to record each returned item …
- Alternate Scenarios:
  - If the customer paid by credit, and the reimbursement transaction to their credit account is rejected, inform the customer and pay them with cash.
  - If the item identifier is not found in the system, notify the Cashier and suggest manual entry of the identifier code (perhaps it is corrupted).
  - If the system detects failure to communicate with the external accounting system, …

Exercise

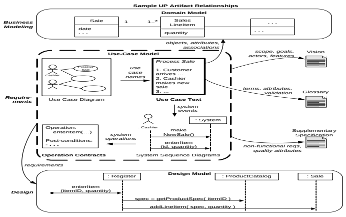
- Elevator system
  - Provide some valid use cases

Future iPhone

Use Case Model

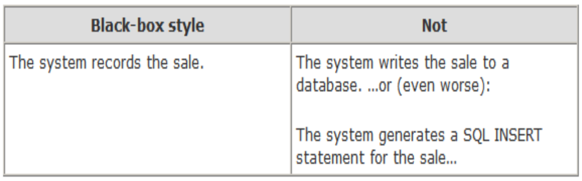
Use Case Model

- Use-case model is a collection of all written use cases.
  - It defines the system’s functionality.
- The Use-Case Model may include a UML use case diagram to show the names of use cases and actors, and their relationships.



Use cases - Guidelines

- Guideline 1
  - Write black-box use cases (why?)
    - Do not describe the internal workings of the system, its components, or design.
  - Rather, the system is described as having ***responsibilities***



Black-box style	Not
The system records the sale.	The system writes the sale to a database. …or (even worse): The system generates a SQL INSERT statement for the sale…

Use cases - Guidelines

- Guideline 2
  - **Take an Actor and Actor-Goal Perspective**
  - Write requirements focusing on the users or actors of a system, asking about their goals.
- Guideline 3

During early requirements work,

"keep the user interface out, focus on user intent."

19

Use cases - Guidelines

- Guideline 4
  - **How to find use cases? A Step by step procedure.**

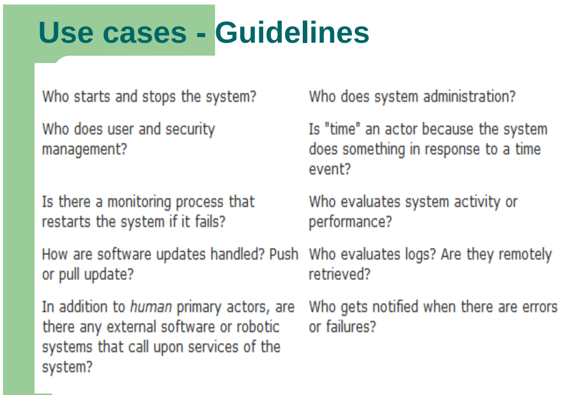
20

Use cases - Guidelines

- Step 1: Choose the System Boundary
  - the POS system itself is the system under design; everything outside of it is outside the system boundary, including the cashier, payment authorization service, and so on.
- **Steps 2 and 3: Find Primary Actors and Goals**
  - in a requirements workshop, people brainstorm and generate a mixture of both. Sometimes, goals reveal the actors, or vice versa.
- In addition, the following questions help identify others that may be missed:

21

Use cases - Guidelines



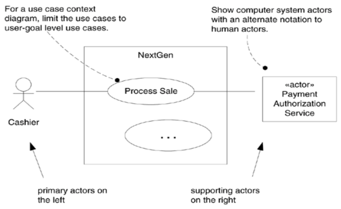
Who starts and stops the system?	Who does system administration?
Who does user and security management?	Is "time" an actor because the system does something in response to a time event?
Is there a monitoring process that restarts the system if it fails?	Who evaluates system activity or performance?
How are software updates handled? Push or pull update?	Who evaluates logs? Are they remotely retrieved?
In addition to *human* primary actors, are there any external software or robotic systems that call upon services of the system?	Who gets notified when there are errors or failures?

22

Use cases - Guidelines

- How to Organize the Actors and Goals?
- Approach:
  - Write an actor-goal list first, review and refine it,
  - Then draw them in a use case diagram, naming the goals as use cases.

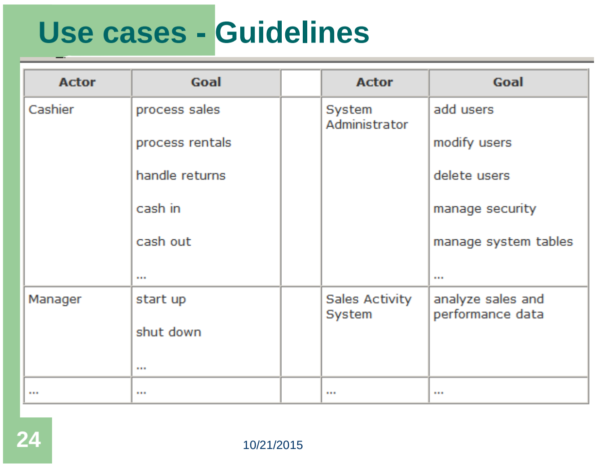
Figure 6.4. Notation suggestions.



For a use case context diagram, limit the use cases to user-goal level use cases.
Show computer system actors with an alternate notation to human actors.
NextGen
Process Sale
...
Cashier
«actor» Payment Authorization Service
primary actors on the left
supporting actors on the right

23

Use cases - Guidelines

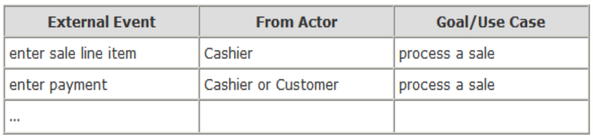


Actor	Goal		Actor	Goal
Cashier	process sales process rentals handle returns cash in cash out ...		System Administrator	add users modify users delete users manage security manage system tables ...
Manager	start up shut down ...		Sales Activity System	analyze sales and performance data
...	...		...	...

24

Use cases - Guidelines

- Use Event Analysis to find Actors and Goals.
- The idea:
  - to identify external events.
  - What are they,
  - where from, and why?
  - Often, a group of events belong to the same use case.
- For example:



External Event	From Actor	Goal/Use Case
enter sale line item	Cashier	process a sale
enter payment	Cashier or Customer	process a sale
...		

Use cases - Guidelines

Step 4: **Define Use Cases.**

- In general, define one use case for each user goal.
  - For example,
    - Goal: process a sale;
    - Use Case: Process Sale.

- Start the name of use cases with a verb.
- A common exception to one use case per goal is to collapse CRUD (create, retrieve, update, delete) separate goals into one CRUD use case, idiomatically called Manage <X>. For example, the goals "edit user," "delete user," and so forth are all satisfied by the Manage Users use case.

Use cases - Guidelines

- Which of these is a valid use case?
  - Negotiate a Supplier Contract
  - Handle Returns
  - Log In
  - Move Piece on Game Board

Use cases - Guidelines

- There are several rules of thumb, including:
  - The Boss Test
  - The Size Test

Use cases - Guidelines

The Boss Test

- Your boss asks, "What have you been doing all day?" You reply: "Logging in!" Is your boss happy ?☺
- If not, the use case fails the Boss Test, which implies
  - It may be a use case at some low goal level, but not the desirable level for requirements analysis.
- User authentication may fail the boss test, but may be important and difficult.

Use cases - Guidelines

The Size Test

- A use case is very seldom a single action or step; rather, a use case typically contains many steps.
- A common mistake in use case modeling is to define just a single step within a series of related steps as a use case by itself.

Use cases - Guidelines

Example: Applying the Tests

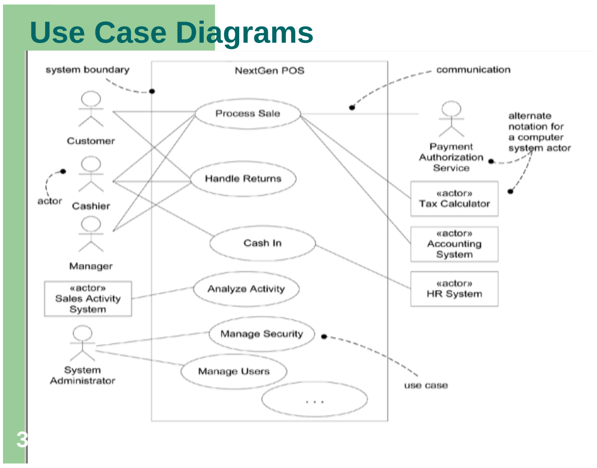
- **Negotiate a Supplier Contract**
  - Much broader and longer than an EBP. Could be modeled as a business use case, rather than a system use case.
- **Handle Returns**
  - OK with the boss. Size is good.
- **Log In**
  - Boss not happy if this is all you do all day!
- **Move Piece on Game Board**
  - Single step - fails the size test.

Future Manufacturing

Use Case Diagrams

- The UML provides ***use case diagram*** notation to *illustrate the names of use cases and actors*, and the relationships between them.

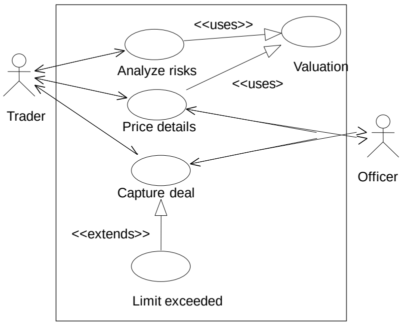
Use Case Diagrams



Exercise

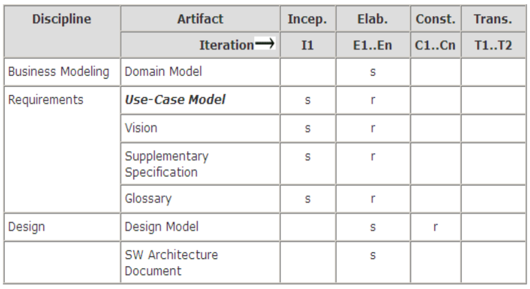
- How to draw a use case diagram for a stock trading system?

A use case diagram



Use Cases into Perspective

Table 6.2. Sample UP artifacts and timing. s - start; r - refine



Discipline	Artifact	Incep.	Elab.	Const.	Trans.
	Iteration⟶	I1	E1..En	C1..Cn	T1..T2
Business Modeling	Domain Model		s		
Requirements	*Use-Case Model*	s	r		
	Vision	s	r		
	Supplementary Specification	s	r		
	Glossary	s	r		
Design	Design Model		s	r	
	SW Architecture Document		s		

Use cases

- The Use-Case Model is started in inception, with perhaps only 10% of the architecturally significant use cases written in any detail.
- The majority are incrementally written over the iterations of the elaboration phase.

Use cases

- **How to Write Use Cases in Inception?**
  - Not all use cases are written in their fully dressed format during the inception phase.
  - The earlier part of the workshop day is spent identifying goals and stakeholders, and speculating what is in and out of scope of the project.
  - A use case context diagram is started.
  - After a few hours, perhaps 20 use cases are identified by name, including Process Sale, Handle Returns, and so on.
  - After this, 10% to 20% of the use cases that represent core complex functions, require building the core architecture, or that are especially risky in some dimension are rewritten in a fully dressed format.
  - Perhaps this means two use cases: Process Sale and Handle Returns.

Use cases

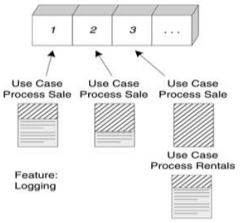
- **How to Write Use Cases in Elaboration?**
  - The "majority" of requirements identified and clarified.
  - However, not all use cases are investigated in each workshop. They are prioritized; early workshops focus on a subset of the most important use cases.
  - Each subsequent short workshop is a time to adapt and refine the vision of the core requirements, which will be unstable in early iterations, and stabilizing in later ones.
  - By the end of elaboration, "80-90%" of the use cases are written in detail.

Use cases

- **How to Write Use Cases in Construction?**
  - The construction phase is composed of timeboxed iterations (for example, 20 iterations of two weeks each) that focus on completing the system.
  - The risky and core unstable issues have settled down in elaboration.
  - There may still be some minor use case writing and perhaps requirements workshops, but much less so than in elaboration

Use cases

Figure 8.1. Use case implementation may be spread across iterations.



A use case or feature is often too complex to complete in one short iteration.

Therefore, different parts or scenarios must be allocated to different iterations.

It is common to work on varying scenarios of the same use case over several iterations and gradually extend the system to ultimately handle all the functionality required . On the other hand, short, simple use cases may be completed within one iteration.

Review

- Which of the following is true?

A: An object is an instance of a class
B: A class is an abstract definition for a set of objects
C: An object can be in more than one class
D: An object has a life span.
E: All of the above

43

Review

- What is TRUE about Use Cases? (Pick 3)(Multiple Choices)

A. Use cases provide the basis of communication between sponsors and developers in planning phase

B. Use cases description provides a good source to identify domain concepts

C. A fully-dressed use case should include both “whats” and “hows” so that they are ready for “realization”

D. A use case is an interaction between a user and a system.

- UML is concerned with object-oriented analysis and design. (Pick 1)

A. True

B. False

44

Review

- Use cases CANNOT be used for : (Single select)

A. modeling the nonbehavioral aspects such as data requirements, performance and security.
B. capturing behavioral aspect of a system.
C. capturing functional aspect of a system.

45

Review

- Whereas [redacted] are concrete examples illustrating a single case, use cases are abstractions describing all possible cases
- Object-oriented analysis is concerned with modeling the [redacted] domain. Object-oriented design is concerned with modeling the solution domain. Both modeling activities use the same representations (i.e., classes and objects)
- Use cases describe the behavior of the system as seen from [redacted]oint of view.
- The client, also called the customer, is responsible for the formulation of scenarios and the requirements. The client is expected to be able to interact with developers. [redacted] [redacted] the person who will be using the delivered system.

46

Review

- Whereas scenarios are concrete examples illustrating a single case, use cases are abstractions describing all possible cases
- Object-oriented analysis is concerned with modeling the application domain. Object-oriented design is concerned with modeling the solution domain. Both modeling activities use the same representations (i.e., classes and objects)
- Use cases describe the behavior of the system as seen from an actor's point of view.
- The client, also called the customer, is responsible for the formulation of scenarios and the requirements. The client is expected to be able to interact with developers. The end user is the person who will be using the delivered system.

47

Review

- Both use case model and design model will start in elaboration phase. **[True/False]**
- Use case modeling in the inception phase will usually describe around 80 percent of the user requirements. **[True/False]**
- We continue writing use cases in the construction phase **[True/False]**

48