Use Cases

Software Design and Analysis

CSCI 2040, Jarek Szlichta

Objectives

- Identify suitable use cases
- Write use cases
- Use the brief, casual and **fully dressed** formats in an essential style
- Use case vs use case diagrams

Use Cases

Use Cases

- **Use cases** are text stories, widely used to discover requirements.
- Use cases are *text stories* of some actor using a system to meet goals.
- They influence many aspects of a project!
- Notice that use cases are not diagrams, they are text.
- Focusing only on UML use case diagrams rather than also use case text is a common mistake for use case novices.

Brief Format Use Case

- **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.

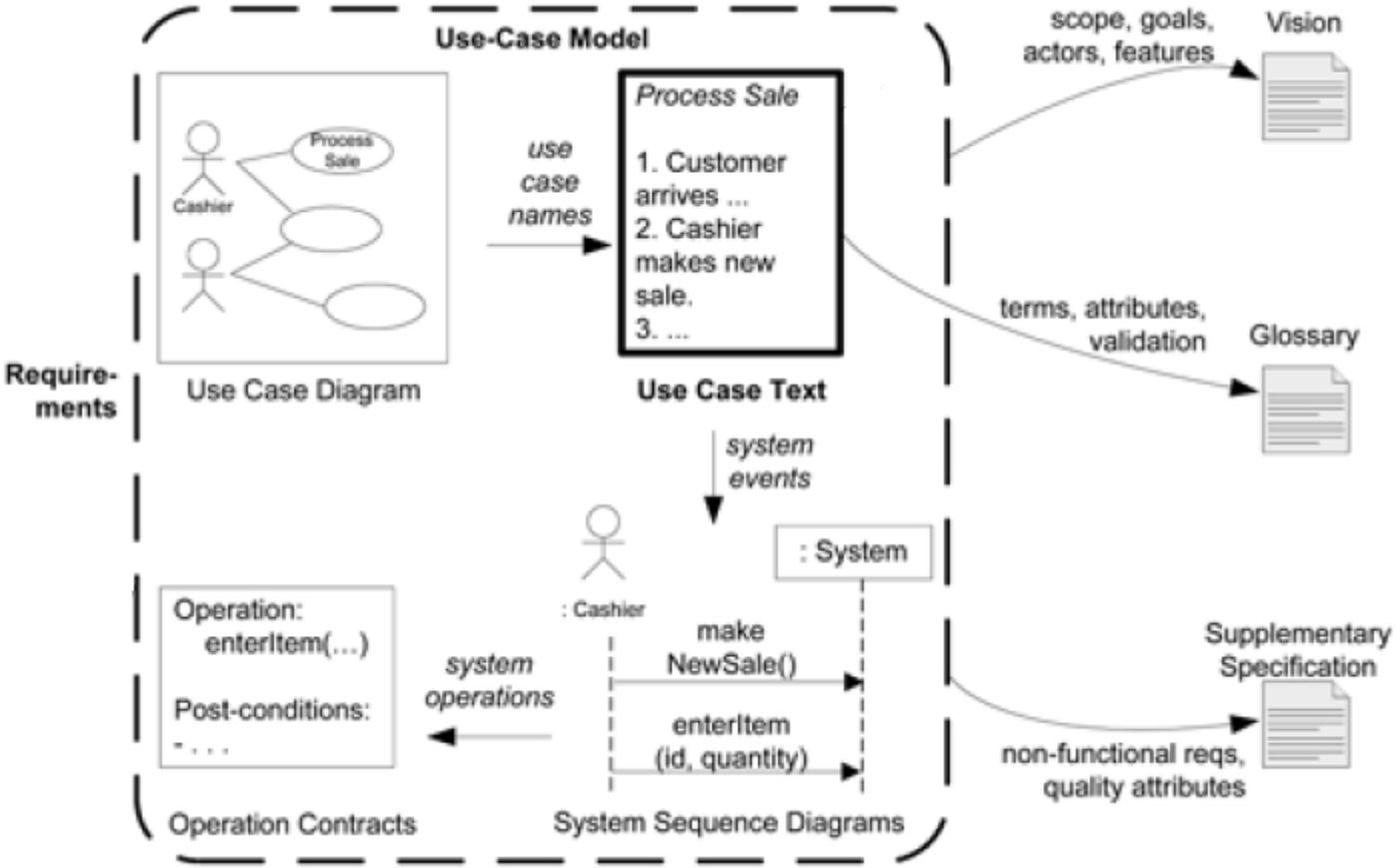
Brief Format Use Case

- Use cases often need to be more detailed and structured than this brief use case example
- But the essence is
  - discovering and
  - recording

  functional requirements by writing stories!

Use Case Model

Use Case = Use Case Text



Use Cases and Use Case Model

- **Use cases are text documents, not diagrams.**
- **Use-case modeling is primarily an act of writing text, not drawing diagrams.**
- The Use-Case Model may optionally include a UML use case diagram
  - to show the names of use cases and actors, and their relationships.
- This gives a nice **context diagram** of a system and its environment.
- It also provides a quick way to list the use cases by name.

Motivation – Why Use Cases?

- Use cases are a good way to help keep it simple.
- They make it possible for domain experts to participate in writing use cases.
- *They emphasize the user goals and perspective*
  - Who is using the system? What are their typical scenarios of use, and what are their goals?
- Scale both up and down in terms of sophistication and formality.

Actors and Scenarios in Use Cases

- An **actor** is something with behavior, such as a person, computer system, or organization;
  - for example, a cashier.
- A **scenario** is a specific sequence of actions and interactions between actors and the system.
  - It is one particular story of using a system
  - for example, the scenario of successfully purchasing items with cash, or the scenario of failing to purchase items because of a credit payment denial

Use Case

- A **use case** is a collection of related success and failure scenarios that describe an actor using a system to support a goal.
- For example, next slide is a *casual format* use case with alternate scenarios.

Casual Format Use Case – 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*:
  - …

Casual Format Use Case – Handle Returns

- *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).

What are Three Kinds of Actors?

- **Primary actor** has user goals fulfilled through using services.
  - For example, the cashier
  - Why identify? To find user goals, which drive the use cases.
- **Supporting actor** provides a service
  - The automated payment authorization service is an example
  - Why identify? To clarify interfaces and protocols
- **Offstage actor** has an interest in the behavior of the use case, but is not primary or supporting;
  - for example, a government tax agency.
  - Why identify? To ensure that *all* necessary interests are identified and satisfied.

Three Common Use Case Formats

- **Brief** Short one-paragraph summary, usually of the main success scenario.
  - The prior *Process Sale* example was brief.
- **Casual** Multiple informal paragraphs that cover various scenarios.
  - The prior *Handle Returns* example was casual.
- **Fully dressed** All steps and variations are written in detail, and there are supporting sections, such as preconditions and success guarantees etc.

Use Case – Fully Dressed

Fully Dressed Style – Template

Use Case Section	Comment
Use Case Name	Start with a verb.
Scope	The system under design.
Level	"user-goal" or "subfunction"
Primary Actor	Calls on the system to deliver its services.
Stakeholders and Interests	Who cares about this use case, and what do they want?
Preconditions	What must be true on start, *and* worth telling the reader?
Success Guarantee	What must be true on successful completion, *and* worth telling the reader.
Main Success Scenario	A typical, unconditional happy path scenario of success.
Extensions	Alternate scenarios of success or failure.
Special Requirements	Related non-functional requirements.
Technology and Data Variations List	Varying I/O methods and data formats.
Frequency of Occurrence	Influences investigation, testing, and timing of implementation.
Miscellaneous	Such as open issues.

Use Case UC1: Process Sale

- **Scope**:
  - NextGen POS application
- **Level**:
  - User goal
- **Primary Actor**:
  - Cashier

Use Case UC1: Process Sale

- **Stakeholders and Interests**:
  - Cashier: Wants accurate, fast entry, and no payment errors, as cash drawer shortages are deducted from his/her salary.
  - Salesperson: Wants sales commissions updated.
  - Customer: Wants purchase and fast service with minimal effort. Wants easily visible display of entered items and prices. Wants proof of purchase to support returns.
  - Manager: Wants to be able to quickly perform override operations, and easily debug Cashier problems.
  - …

Use Case UC1: Process Sale

- **Stakeholders and Interests**:
  - Company: Wants to accurately record transactions and satisfy customer interests.
  - Government Tax Agencies: Want to collect tax from every sale. May be multiple agencies, such as national, state, and country.
  - Payment Authorization Service: Wants to receive digital authorization requests in the correct format and protocol.

Use Case UC1: Process Sale

- **Preconditions:**
  - Cashier is identified and authenticated.
- **Success Guarantee (or Postconditions):**
  - Sale is saved.
  - Tax is correctly calculated.
  - Accounting and Inventory are updated.
  - Commissions recorded.
  - Receipt is generated.
  - …

Main Success Scenario (Basic Flow)

1. Customer arrives at POS checkout with goods and/or services to purchase.
2. Cashier starts a new sale.
3. Cashier enters item identifier.
4. System records sale line item and presents item description, price, and running total. Price calculated from a set of price rules.

Cashier repeats steps 3-4 until indicates done.

5. System presents total with taxes calculated.

Main Success Scenario (Basic Flow)

6. Cashier tells Customer the total, and asks for payment.
7. Customer pays and System handles payment.
8. System logs completed sale and sends sale and payment information to the external Accounting system (for accounting and commissions) and Inventory system (to update inventory).
9. System presents receipt.
10. Customer leaves with receipt and goods (if any).

Optional: Two Column Variation

Primary Actor: ...

... as before ...

Main Success Scenario:

Actor Action (or Intention)	System Responsibility
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.

Extensions (or Alternative Flows)

- …
- *9a. There are product rebates…*
  - 1. System presents the rebate forms and rebate receipts for each item with a rebate.
- …

See Details in Chapter 6 (Use Cases), Page 51 Applying UML and Patterns, Craig Larman!

Use Case UC1: Process Sale

- **Special Requirements:**
  - Touch screen UI on a large flat panel monitor. Text must be visible from 1 meter.
  - Credit authorization response within 30 seconds 90% of the time.
  - Language internationalization on the text displayed.
  - …
- **Technology and Data Variations List**
  - 3a. Item identifier entered by bar code laser scanner (if bar code is present) or keyboard.
  - 7a. Credit account information entered by card reader or keyboard.

Use Case UC1: Process Sale

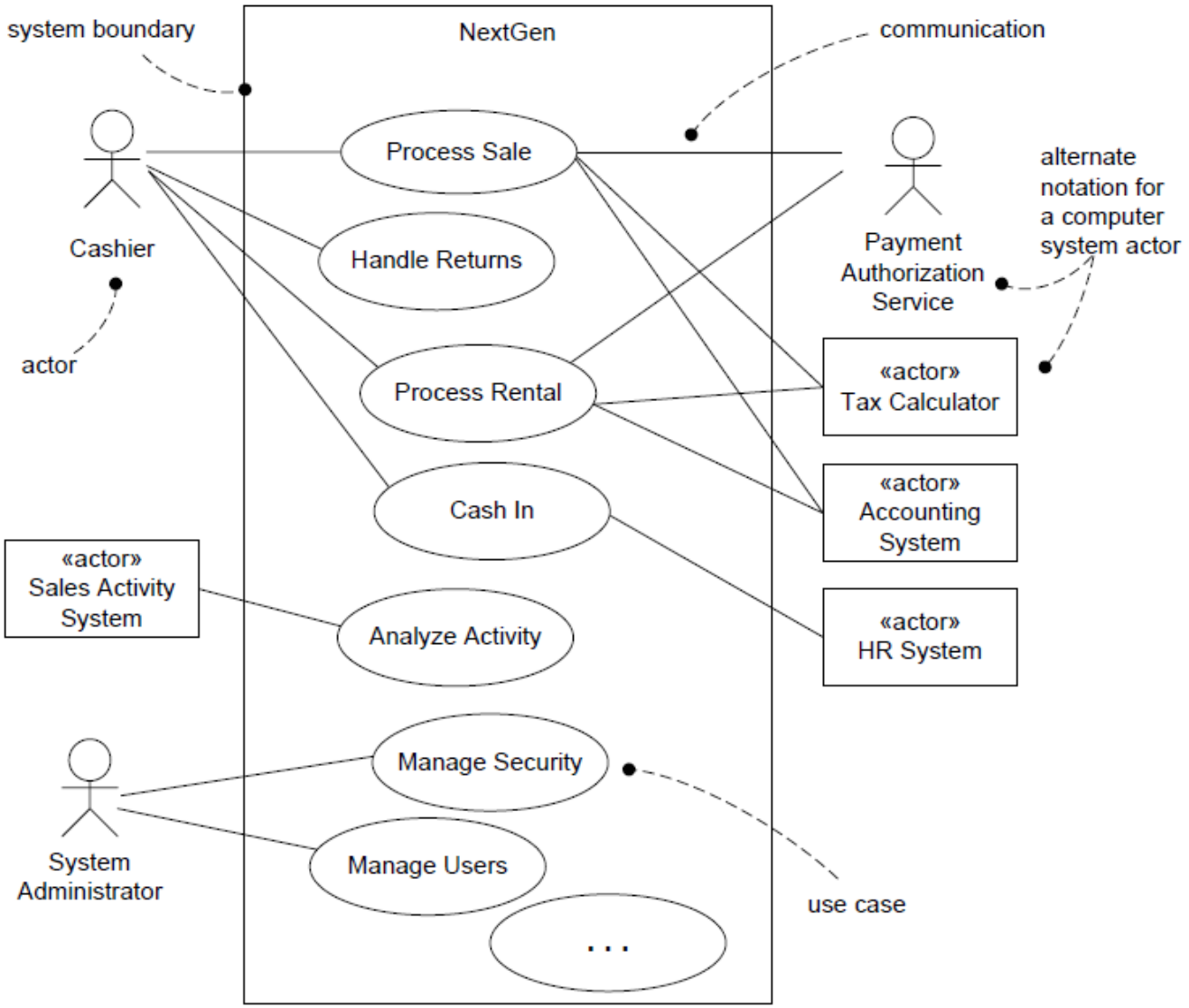
- **Frequency of Occurrence**
  - Could be nearly continuous.
- **Open Issues**
  - What are the tax law variations?
  - Must a cashier take their cash drawer when they log out?
  - …

Use Case Diagrams

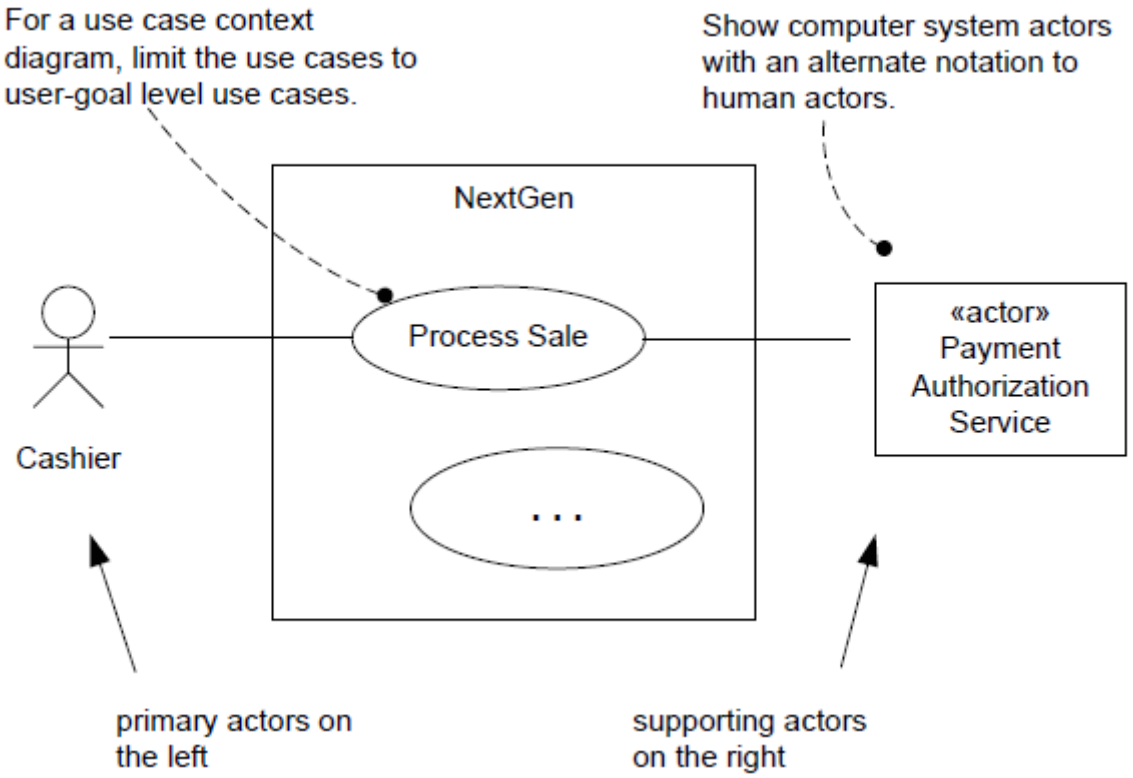
Applying UML: Use Case Diagrams

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

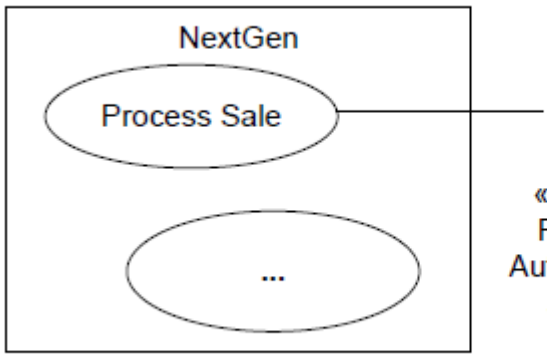
UML Use Case Diagram

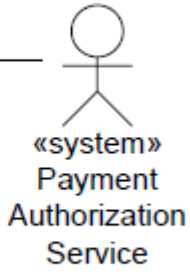


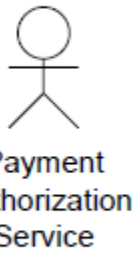
Suggestions

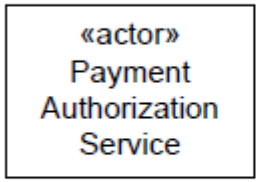


Alternate Notations







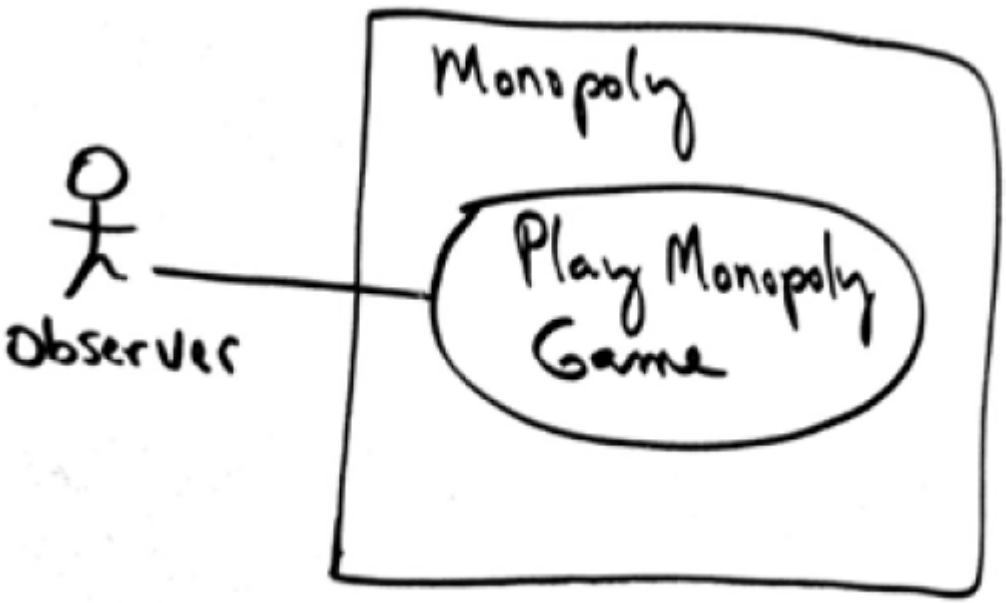


Some UML alternatives to illustrate external actors that are other computer systems.

The class box style can be used for any actor, computer or human. Using it for computer actors provides visual distinction.

Monopoly System Simulation

Use Case Diagram for Monopoly System



Use Case: Play Monopoly Game

- **Scope**:
  - Monopoly application
- **Level**:
  - User goal
- **Primary Actor**:
  - Observer
- **Stakeholders and Interests**:
  - Observer wants to easily observe the output of the game simulation.

Use Case: Play Monopoly Game

- **Main Success Scenario**:
  1. Observer requests new game initialization, enters number of players.
  2. Observer starts play.
  3. System displays game trace for next player move.

  Repeat step 3 until a winner or Observer cancels.

Use Case: Play Monopoly Game

- **Extensions**:
  - ***a.** At any time, System fails: (To support recovery, System logs after each completed move)
    - **1.** Observer restarts System.
      - System detects prior failure, reconstructs state, and prompts to continue.
      - Observer chooses to continue (from last completed player turn).

Use Case: Play Monopoly Game

- **Special Requirements**:

  Provide both

  - graphical and
  - text

  trace modes

Quiz

- Are use cases functional or non-functional requirements?
- What is the difference between a use case and use case diagram? What use case model consists of?
- What is the notation suggestion for system actors in use case diagrams?
- What are three kinds of actors?
- What are three common types of use case formats?

Actions

- Review Slides.
- Read Chapter 6 (Use Cases)
  - *Applying UML and Patterns*, Craig Larman