CS 619 Introduction to OO Design and Development

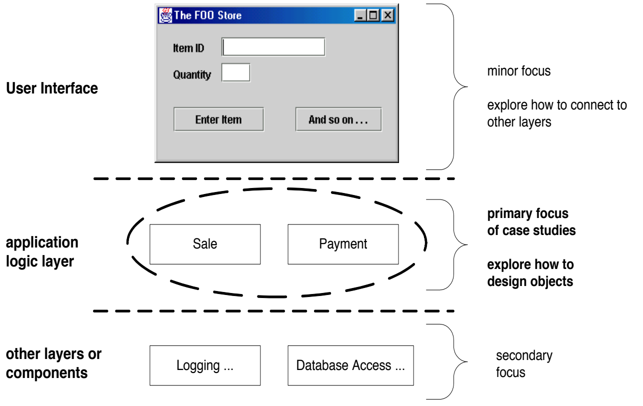
Use Cases

Fall 2013

Define the Problem

- The most critical question:

"Is this the right system to make?"



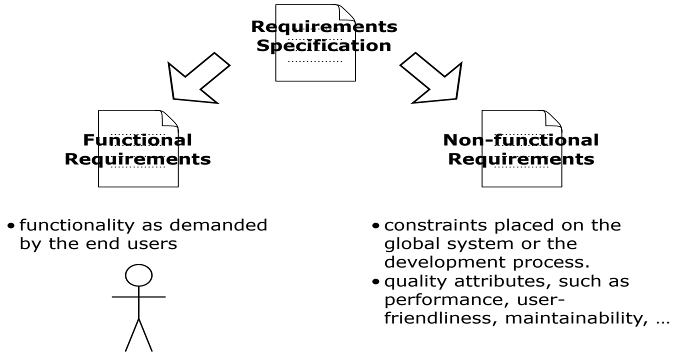
User Interface — minor focus; explore how to connect to other layers

application logic layer — Sale, Payment — primary focus of case studies; explore how to design objects

other layers or components — Logging ..., Database Access ... — secondary focus

Requirements

Requirements are capabilities and conditions to which the system must conform.



Requirements Specification

Functional Requirements

- functionality as demanded by the end users

Non-functional Requirements

- constraints placed on the global system or the development process.
- quality attributes, such as performance, user-friendliness, maintainability, ...

FURPS Requirement Model

In the UP, requirements are categorized according to the **FURPS+ model**

Functional - features, capabilities, security.

Usability **-** human factors, help, documentation.

Reliability - frequency of failure, recoverability, predictability.

Performance - response times, throughput, accuracy, availability, resource usage.

Supportability - adaptability, maintainability, internationalization, configurability.

The "**+**" means

Implementation - resource limitations, languages and tools, hardware..

Interface - constraints imposed by interfacing with external systems.

Operations - system management in its operational setting.

Packaging - for example, a physical box.

Legal - licensing and so forth.

Use Cases

- A **use case** is a text stories of some actor using a system to meet goals.

- An **actor** is something with behavior and have responsibilities.

  To carry out responsibilities, an actor sets goals
  • **Primary actor** (= stakeholder) has unsatisfied goal and needs system assistance
  • **Secondary actor** provides assistance to satisfy the goal

- A **scenario** is a specific sequence of actions and interactions between actors and the system.

  It is one particular story of using a system, or one path through the use case.

Use case in brief format

Rent Videos. A Customer arrives with videos to rent. The Clerk enters their ID, and each video ID. The System outputs information on each. The Clerk requests the rental report. The System outputs it, which is given to the Customer with their videos.

Example:

- Which of these requirements should be represented directly in a use case?

  – Order cost = order item costs * 1.06 tax.
  – Promotions may not run longer than 6 months.
  – Customers only become Preferred after 1 year.
  – A customer has one and only one sales contact.
  – Response time is less than 2 seconds.
  – Uptime requirement is 99.8%.
  – Number of simultaneous users will be 200 max.

Use Cases in Casual Format

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, …

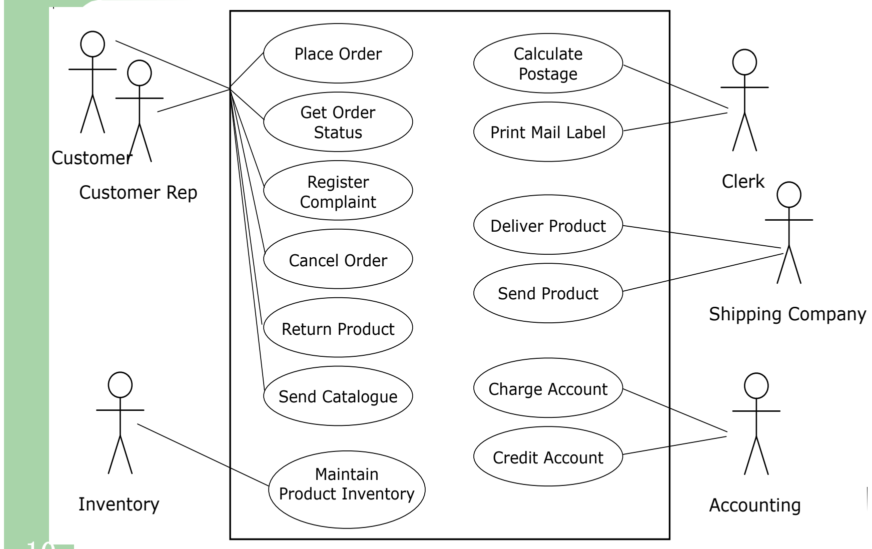
A *use case* is a collection of related success and failure scenarios that describe an actor using a system to support a goal.

Three kinds of actors

- Primary actor
  - has user goals fulfilled through using services of the SuD. E.g., the cashier.
    - Why identify? To find user goals, which drive the use cases.
- Supporting actor
  - provides a service to the SuD. E.g. automated payment authorization service. Often a computer system, but could be an organization or person.
    - Why identify? To clarify external interfaces and protocols.
- Offstage actor
  - has an interest in the behavior of the use case, but is not primary or supporting. E.g. a government tax agency.
    - Why identify? To ensure that all necessary interests are identified and satisfied. Offstage actor interests are sometimes subtle or easy to miss unless these actors are explicitly named.

Primary and *supporting* actors will appear in the action steps of the use case text.

Example: Actorss Use Cases Diagram



Naming Use Cases

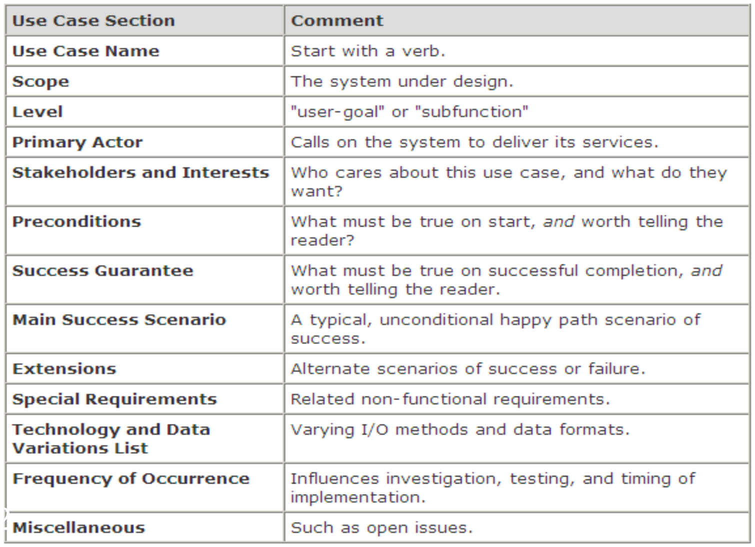
Each use case should have a name that indicates what value (or goal) is achieved by the actor's interaction with the system

- Must be a complete process from the viewpoint of the end user.
- Usually in verb-object form and active voice.
- Use enough detail to make it specific

Examples:

Purchase Concert Ticket
Purchase Concert Tickets
Purchase Ticket
Ticket Purchase
Ticket Order
Pay for Ticket

A Sample Template of Fully dressed use case format



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.

Fully Dressed Format Explained

- **Preconditions and Success Guarantees (Postconditions)**
  - **Preconditions** state what must always be true before a scenario is begun in the use case.
    - Preconditions communicate noteworthy assumptions that the writer thinks readers should be alerted to.
  - Success guarantees (or **postconditions**) state what must be true on successful completion of the use case - either the main success scenario or some alternate path.
    - The guarantee should meet the needs of all stakeholders.

Fully Dressed Format Explained

- **Main Success Scenario and Steps (or Basic Flow)**
  - "happy path" scenario, or the more prosaic "Basic Flow" or "Typical Flow.”
  - It describes a typical success path that satisfies the interests of the stakeholders.
- Guideline
  - defer all conditional handling to the Extensions section.

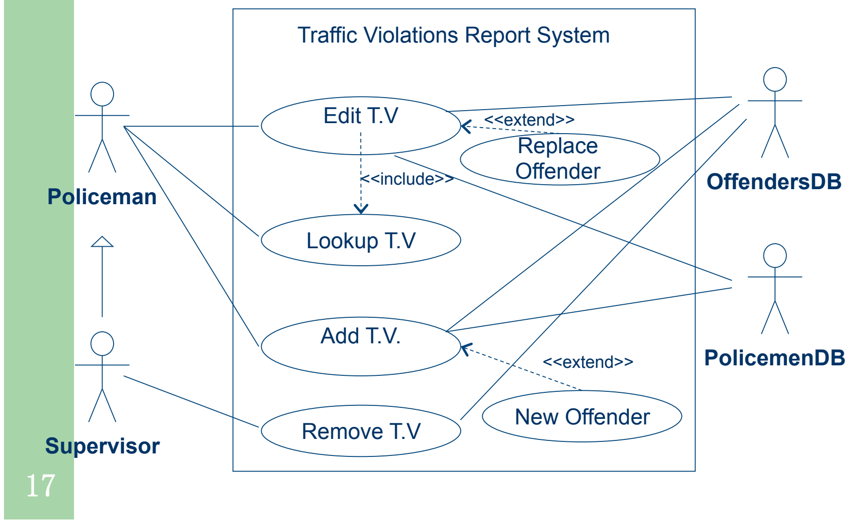
Fully Dressed Format Explained

- **Extensions (or Alternate Flows)**
  - Normally comprise the majority of the text.
  - They indicate all the other scenarios or branches, both success and failure.
  - Extension scenarios are branches from the main success scenario, and so can be notated with respect to its steps 1…N.
  - An extension has two parts: the condition and the handling.

Fully Dressed Format Explained

- Performing Another Use Case Scenario
  - 3a. Invalid item ID:
    1. System signals error and rejects entry.
    2. Cashier responds to the error:
       - 2a. …
       - 2c. Cashier performs ***Find Product Help*** to obtain true item ID and price.

Example: TVRS Use Case



TVRS – Remove TV

Use Case Name: Remove Traffic Violation

Scope: TVRS

Level: user-goal

Primary Actors: Supervisor

Stakeholders and Interests:

Supervisor: Remove an existing Traffic Violation

…

Preconditions:

Normal Course of “Lookup Traffic Violation” UC is completed, and the details of an existing Traffic Violation are displayed

Postconditions:

Removed Traffic Violation is no longer stored in the TVRS.

Traffic Violation is removed from the offender’s record in the OffendersDB

"Lookup Traffic Violation" form is displayed

TVRS - Remove TV

Main Success Scenario:

1. Supervisor calls for deletion of the chosen Traffic Violation
2. TVRS prompts Supervisor for confirmation
3. Supervisor confirms
4. TVRS requests OffendersDB to delete the Traffic Violation from the offender’s record
5. OffendersDB approves that the Traffic Violation has been deleted
6. TVRS allows Supervisor to look up a new Traffic Violation as described in the “Lookup Traffic Violation” UC

TVRS - Remove TV

– Extensions:

- 3a: Supervisor cancels:

  3a1: TVRS Continues to item 6 without removing the Traffic Violation

- 5a: Traffic Violation is not removed from the OffendersDB

  5a1: TVRS displays an error message describing the failure

  5a2: TVRS continues to item 6 without clearing chosen Traffic Violation details, and without deleting the Traffic Violation

Identifying Use Case Tests

Boss Test: Especially for the “architecturally significant” use cases, your boss should think this is essential to the business.

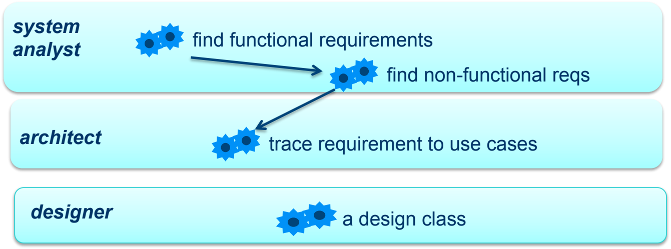
EBP Test: An Elementary Business Process is defined as

> A task performed by one person in one place at one time, in response to a business event, that adds measurable value to the business and leaves business data in a consistent state.

Size Test: Use cases that you can express in less than a page are often not significant. Fully-dressed use cases take 3-10 pages to explain.

Exercise:

- Are the following possible use cases?
  - negotiate a supplier contract
  - handle returns
  - log in
  - move pieces on a game board



system analyst

find functional requirements

find non-functional reqs

architect

trace requirement to use cases

designer

a design class

- You don’t have always have 1-1 mapping between functional requirements and use-cases
- But make sure that you have cover all functional requirement.

Benefits of Use Cases

- Captures functional requirements from user’s perspective. Gives a clear and consistent description of what the system should do
- A basis for performing system tests
- Provides the ability to trace functional requirements into actual classes and operations in the system
- Serves as a unit of estimation
- The smallest unit of delivery
  - Each increment that is planned and delivered is described in terms of the Use Cases that will be delivered in that increment