

CHAPTER 3: Use Cases

USE CASES AND USER GOALS

- **Use case** – an activity the system performs, usually in response to a request by a user
- 2 techniques to identify use cases:
 - **User Goal Technique** - determine what specific goals or objectives must be completed by a user
 - **Event Decomposition Technique** - identify all the business events that will cause the information system to respond, and each event leads to a use case (is the most comprehensive technique)
 - focuses on identifying the events to which a system must respond and then determining how a system must respond (i.e., the system's use cases). When defining the requirements for a system, begin by asking, "What business events occur that will require the system to respond?"
 - it is more comprehensive because it identifies actual business events that will cause the system to respond. By starting with business events, it helps the analyst define each use case at the right level of detail. It not only looks at user-initiated events (like User Goal Technique), but also considers temporal and state events.

The **user goal technique** for identifying use cases includes these steps:

1. Identify all potential users for the new system.
2. Classify potential users in terms of their functional role (e.g., shipping, marketing, sales).
3. Further classify potential users by organizational level (e.g., operational, management, executive).
4. Interview each type of user to find a list of specific goals they will have when using the new system.
5. Create a list of preliminary use cases organized by type of user.
6. Look for duplicates with similar use case names and resolve inconsistencies.
7. Identify where different types of users need the same use cases.
8. Review the completed list with each type of user and then with interested stakeholders.

USE CASES AND EVENT DECOMPOSITION

- **Elementary Business Process (EBPs)** - a task that is performed by one person in one place in response to a business event, adds measurable business value, and leaves the system and its data in a stable and consistent state e.g. Search for item, Fill shopping cart.
- **Event** - something that occurs at a specific time and place, can be precisely identified, and must be remembered by the system. 3 types:
 - **External Event** - an event that occurs outside the system—usually initiated by an external agent or actor. An external agent (or actor) is a person or organizational unit that supplies or receives data from the system e.g. customer
 - **Temporal event** — an event that occurs as a result of reaching a point in time e.g. payroll systems that produce a payslip each month
 - **State event** — an event that occurs when something happens inside the system that triggers the need for processing (aka internal events)
- **System Controls** - checks or safety procedures put in place to protect the integrity of the system
- **Perfect technology assumption** - states that events should be included during analysis only if the system would be required to respond under perfect conditions (i.e. no system breakdowns, capacity problems, user problems). Fixing such items with controls is done later at design phase, not at analysis phase.

Using the Event Decomposition Technique

The event decomposition technique for identifying use cases includes these steps:

1. Consider the external events in the system environment that require a response from the
2. For each external event, identify and name the use case that the system requires.
3. Consider the temporal events that require a response from the system

CHAPTER 3: Use Cases

4. For each temporal event, identify and name the use case that the system requires and then establish the point of time that will trigger the use case.
5. Consider the state events that the system might respond to, particularly if it is a real-time system in which devices or internal state changes trigger use cases.
6. For each state event, identify and name the use case that the system requires and then define the state change.
7. When events and use cases are defined, check to see if they are required by using the perfect technology assumption. Do not include events that involve such system controls as login, logout, change password, and backup or restore the database, as these are put in as system controls.

USE CASES AND CRUD

- **CRUD** - acronym for Create, Read (or Report), Update, and Delete, often introduced with respect to database management.
 - A technique to validate or refine use cases
 - Most useful when used as a cross-check along with the user goal technique. Users will focus on their primary goals, and use cases that update or archive data will often be overlooked.
 - Main purpose is to make sure all possibilities are identified.
 - Sometimes, data entities or domain classes are shared by a set of integrated applications

CRUD technique steps:

1. Identify all the data entities or domain classes involved in the new system.
2. For each type of data (data entity or domain class), verify that a use case has been identified that creates a new instance, updates existing instances, reads or reports values of instances, and deletes (archives) an instance.
3. If a needed use case has been overlooked, add a new use case and then identify the stakeholders.
4. With integrated applications, make sure it is clear which application is responsible for adding and maintaining the data and which system merely uses the data.

USE CASE DESCRIPTION EXAMPLES

Use case	Brief use case description
<i>Create customer account</i>	User/actor enters new customer account data, and the system assigns account number, creates a customer record, and creates an account record.
<i>Look up customer</i>	User/actor enters customer account number, and the system retrieves and displays customer and account data.
<i>Process account adjustment</i>	User/actor enters order number, and the system retrieves customer and order data; actor enters adjustment amount, and the system creates a transaction record for the adjustment.

CSMS Customer account subsystem	
Use cases	Users/actors
Create/update customer account	Customer, customer service representative, store sales representative
Process account adjustment	Management
Send message	Customer
Browse messages	Customer

CHAPTER 3: Use Cases

USE CASES, ACTORS, AND NOTATION

Actor - the user in Unified Modelling Language (UML)

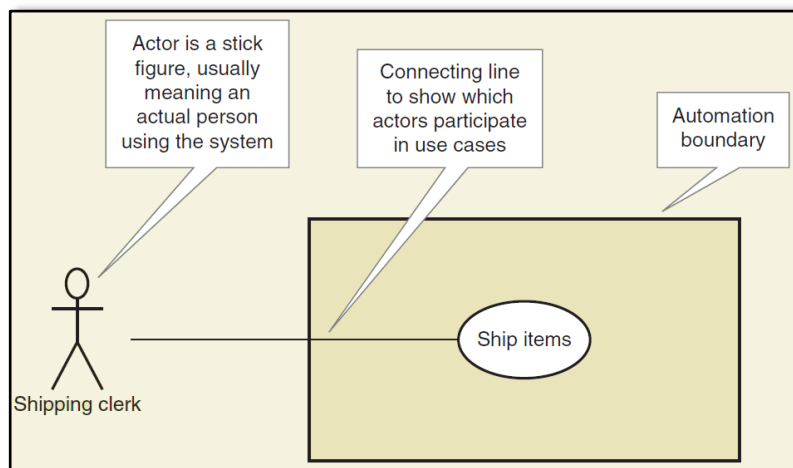
- Is always outside the automation boundary of the system
- May be part of the manual portion of the system
- Can also be another system or device that receives services from the system
- Is represented by a stick figure and assigned a name that characterizes the role the actor is playing

Use case – An activity the system performs represented by an oval with the name of the use case inside

- The connecting line between the actor and the use case indicates that the actor is involved with that use case

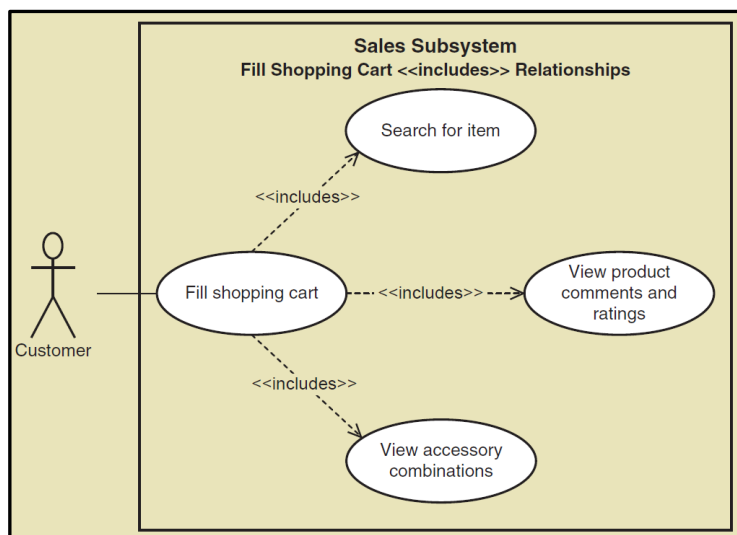
Automation boundary - defines the border between the computerized portion of the application and the people operating the application

- Shown as a rectangle containing the use case
- The actor’s communication with the use case crosses the automation boundary



Simple Use Case Diagram

- During the development of a use case diagram, it frequently becomes apparent that one use case may use - or "includes" - the services of another use case
- The relationship between these use cases is denoted by the dashed connecting line with the arrow that points to the use case that is included
- This relationship is referred to in UML as the <<includes>> relationship or the <<uses>> relationship



CHAPTER 3: Use Cases

Developing a Use Case Diagram

Steps:

1. Identify all the stakeholders and users who would benefit by having a use case diagram.
2. Determine what each stakeholder or user needs to review in a use case diagram. Use a case diagram is generally produced for each subsystem, for each type of user, for use cases with the <<includes>> relationship, and for use cases that are of interest to specific stakeholders.
3. For each potential communication need, select the use cases and actors to show and draw the use case diagram.
4. Name each use case diagram and state how and when the diagram should be used to review use cases with stakeholders and users