

# Unified Modeling Language – Use Cases

## Introduction

The UML object modeling tools consist of nine types of diagrams grouped into two major concept areas—system structure and system behavior. System structure focuses on the objects that compose a system and their relationships, whereas system behavior addresses the dynamic nature of objects as they respond to events, perform actions, and transition to new states.

In MIS 374 we will cover only the most common UML system structure technique: Use Case Diagrams. A use case diagram shows the interaction of users with the system.

A use case diagram shows what the system is, not how it functions. Figure 1 is a generic example of a high level use case diagram.

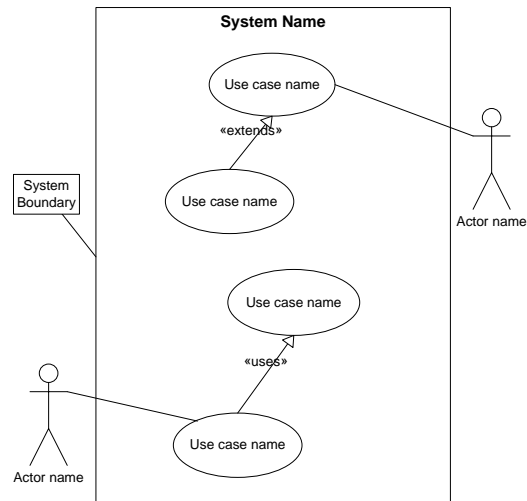
## Purpose

During the Inception Phase and later Construction Iterations, your team may create multiple use case diagrams to describe graphically how multiple users interact with the system. Reviewing these use case diagrams with stakeholders will ensure that your software system does exactly what your client expects.

## Description & Definitions

**The Unified Modeling Language (UML)** is a general-purpose visual language used to specify, visualize, construct, and document the software system. The word “unified” means that it is applicable to all system development methodologies, to all SDLC stages, and to all applications (business and non-business alike), and can be used with all implementation languages and platforms. It is intended to be supported by other visual modeling languages in the form of code generators and report writers.

Figure 1. Key Elements of a Use Case Diagram



## Elements of a Use Case

- The boundary of the system is defined with a rectangle.
- The system name can either be on top of the rectangle or inside.
- Each use case that is included in the system is illustrated with an oval and identified with a verb–object label format like the process bubble labels for DFDs.
- **Actors** are the external entities that interface with the system and are located outside the system boundary. Actors can be organizations, individuals, places, or other systems. Regardless of type, actors are always illustrated as stick figures. The actor’s name appears below the stick figures.
- **Associations** between actors and use cases are illustrated with a straight line. There are two types of associations between use cases and they are explained in the next section.



## Use Case Associations

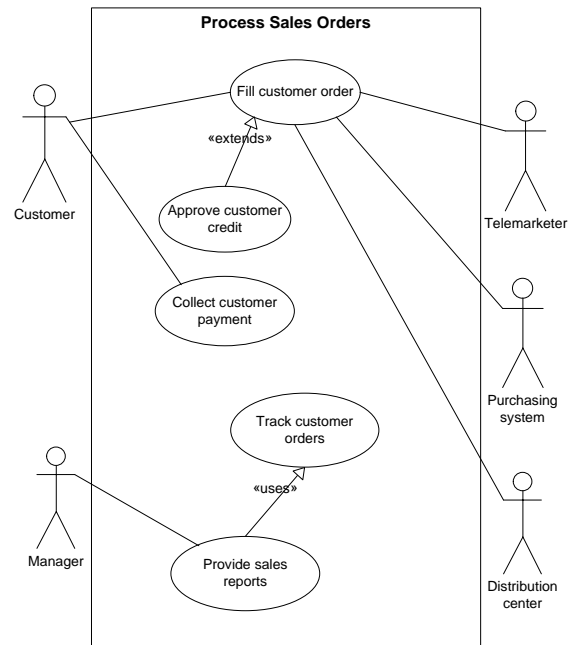
**Extend relationships.** Sometimes, one use case is needed for the other use case to perform its function. This is called a conditional relationship and you show this by choosing the “extends” symbol in Visio as seen in Figures 1 and 2. Make sure that the arrow points to the use case being extended.

**Uses.** Sometimes a use case needs the assistance of another use case to either perform its function or complete a function. For example, in Figure 2 the use case “Provide sales reports,” can only function if use case “Track customer orders” collects the data it needs. When one use case needs the assistance of another use case to function, select the “uses” symbol in Visio to connect the two use cases.

Using “extends” and “uses” gives you more information regarding the usage requirements of your system.

Figure 2 is a sample use case diagram illustrating a system that processes customer sales orders and consists of five use cases. Notice in the example actors can be organizations (such as customer firms), individuals (telemarketer and manager), other systems (purchasing), and places (distribution center.)

Figure 2. Use Case Example for Processing Sales Orders



**Fill customer order.** A customer places an order by telephone and a telemarketer gathers the required information and enters it in the customer sales order system. The purchasing system determines whether sufficient inventory exists. When it does, the distribution center is notified to ship the merchandise. When the product balance-on-hand reaches the reorder point, the purchasing system is notified to purchase replenishment stock.

**Approve customer credit.** The Approve Customer Credit use case extends the Fill Customer Order use case by providing a credit check.

**Collect customer payments.** The Collect Customer Payments use case maintains the accounts receivable and collects the customer payments.

**Track customer orders.** The Track Customer Orders use case captures key data about each sale.

**Provide sales reports.** The Provide Sales Reports use case uses the customer order data to prepare reports for management.

Notice Figure 2 shows associations need not exist between all use cases.



### High Level Use Cases and Drill Down Use Cases

Begin drawing high level use cases using Visio, eventually, detailed use case diagrams will

enable your team to gain a complete picture of your system requirements, especially if you work with stakeholders in different areas of the client organization to learn the details of each area's system usage.

Figure 3. High-level use case diagram for ASTA skill-set system

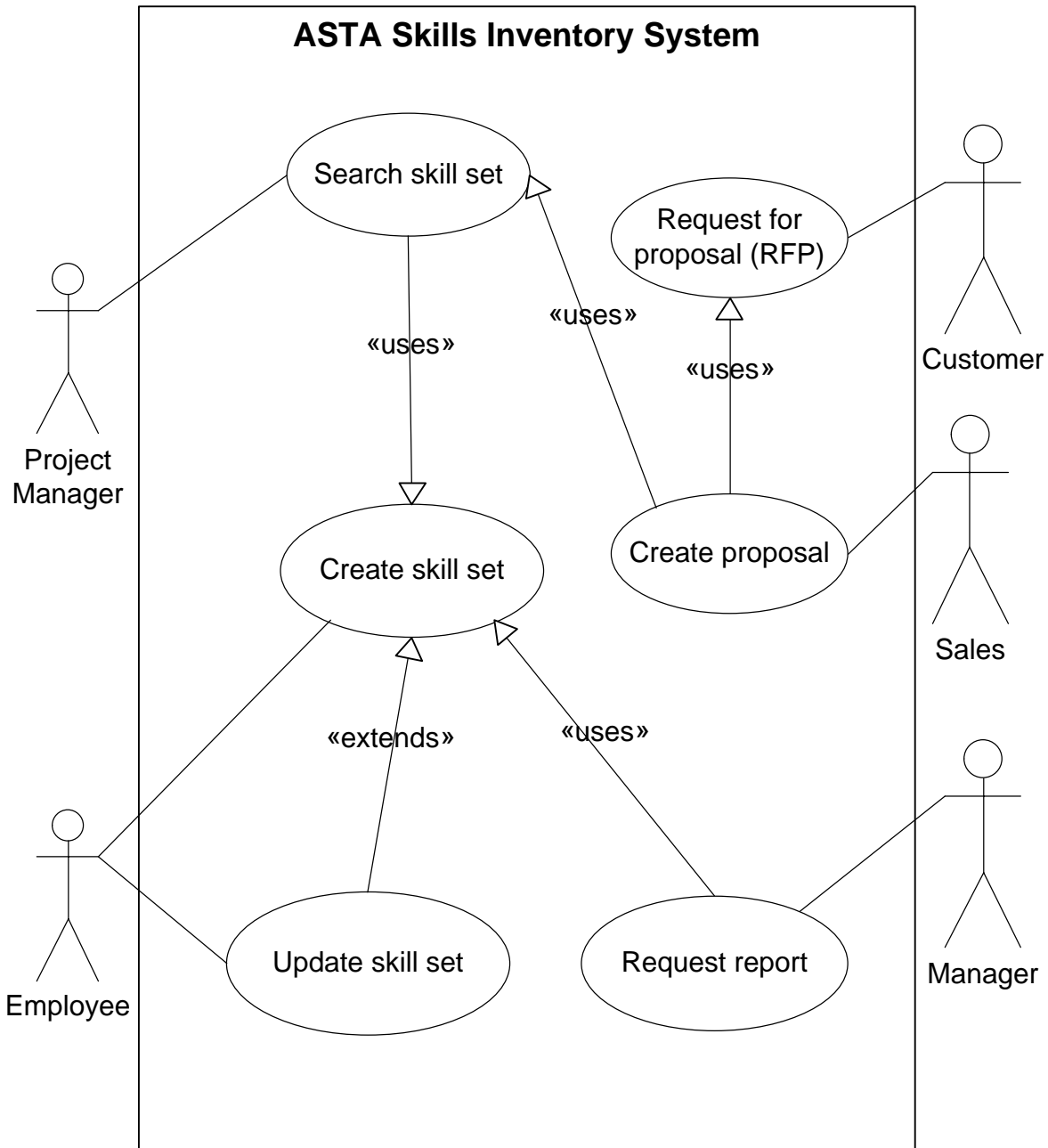
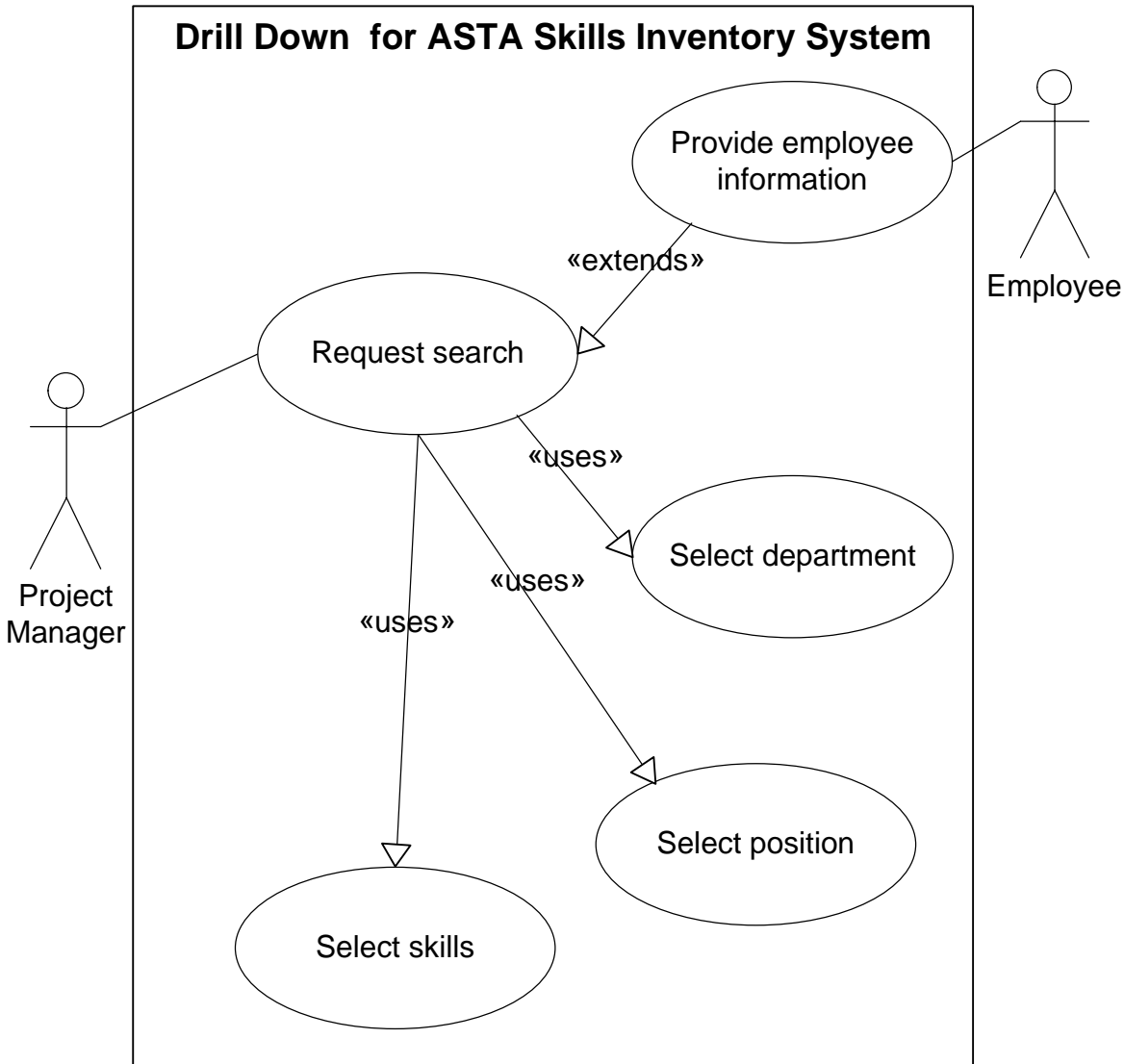




Figure 4. Drill-down use case diagram for the search skill set use case shown in Figure 3



### Benefits

Paramount to producing a software system that your client will love using is knowing all of the system’s usage requirements. To gain this knowledge, create multiple iterations of use case diagrams thereby ensuring that you have given the proper attention to the usage requirements before investing time and resources to build and test the code.

### Tips for completeness

- Identify as many actors as possible.
- Then ask: “How do they interact with the system?”
- Connect actors with all appropriate use cases.

### FAQs

**Q1:** What’s the difference between Use Cases and Use Case Diagrams? **Answer:** A diagram illustrates a system or subsystem with two or



more user interactions (use cases) with the system. Figure 4 above is a Use Case diagram with five use cases.

For a complete specification, all of the use cases within a use case diagram are further specified with a use case script. (We do not require scripts for MIS 374 projects).

**Q2:** How do Use Case diagrams differ from data flow diagrams (DFDs)? **Answer:** Use case diagrams work well to show the high level interaction of “web-based” systems with user categories.

Data flow diagrams show more details for processing data and for manual processes. These can be very important for “back end” systems like inventory or other supply chain activities. DFDs are also helpful for detailing system update responsibilities for non-expert computer users, like the example provided in the Process Modeling reading for Central Texas Pediatric Orthopedic project on the Resources page.

## High Quality Delivery Tips

- Encapsulate the system boundary with a rectangle.
- Include a title that states what system is being modeled.
- System titles may be inside the rectangular boundary of the system or on top of the rectangle.
- Identify each use case with an oval with a verb-object label.
- Identify actors (individuals, organizations, and systems) that interact with each use case with stick figures with names below
- Place all actors outside the rectangular system boundary.
- Indicate associations between actors and use cases with a straight line.
- Indicate associations between use cases with an arrow.

## Examples on Resources Page:

- ASTA Use Cases (Visio file)
- Latinitas Use Case Diagrams (slides)
- Latinitas Use Case Specification