

A decorative L-shaped line in a gold color, consisting of a horizontal segment and a vertical segment meeting at a right angle.

# Chapter 3: Requirements Modeling

---

Requirements Engineering

---

# Objectives

In this chapter, you will learn about:

- Functional requirements
- Modeling requirements
- Overview of basic modeling paradigms

---

# Functional Requirements

Functional Requirements Specification is the second step in Requirements Engineering

- Functional Requirements define the **solution space** for the problem
  - Our goal is to use the customer's needs (see lecture notes 2) and create:
    1. Graphical representations of the required interactions, behaviors and activities
    2. Narrative of the required interactions, behaviors and activities
    3. Algorithms to support your views (You may skip this step for your class project. This step is typically handled by SMEs. It falls mainly in the HOW domain. If you are not an SME, then you will need to brainstorm initially with various Subject Matter Experts)
- The deliverables of the Functional requirements provide a framework for deriving and structuring systems requirements

---

# Functional Requirements Modeling

1. Modeling is a method to schematically describe our “perceived” representations of the customer’s needs, behaviors and interactions for the problem domain

## What are the advantages of Modeling?

- Allows you to restate the requirements in a format different from the original customer’s requests
  - **Force the customer to examine your models in order to validate the model’s accuracy**
- Helps you to better understand the customer’s requirements
  - **Multiple, conflicting outputs to the same input is an indication of inconsistencies in the requirements**
- Allows consideration of multiple interacting aspects and views of the system apriori
- Allows requirements specifications to be visualized in diagrams (“A picture is worth a thousand words”)
- Facilitates efficient communications between the different SDLC organizations

---

# Modeling Requirements

## Modeling Techniques:

- There are several modeling notations to express info about problem's concept, behavior and interactions (properties)
- New (customized) notations are being introduced to adapt to new type of problem domains
- The individual modeling notations, however, are based on less than 10 basic paradigms

We will examine some of the basic paradigms

---

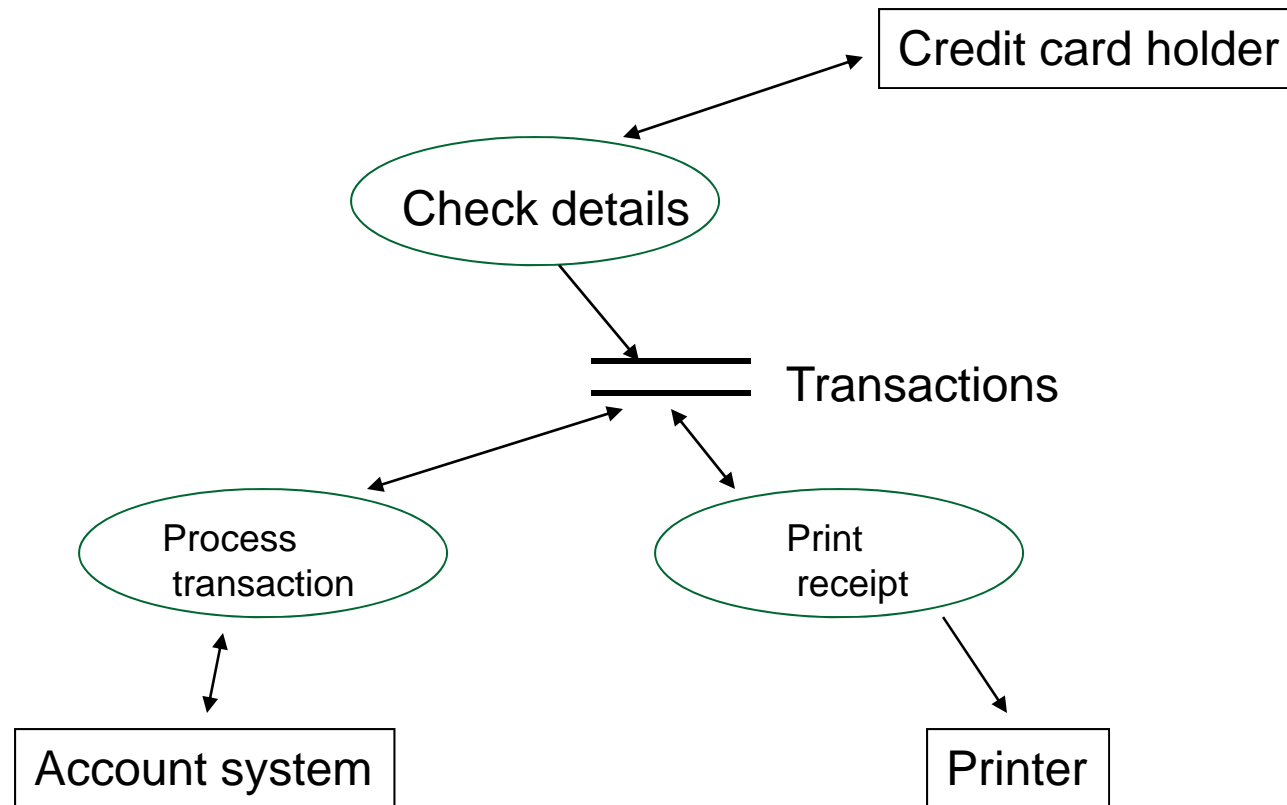
## Modeling Requirements

### Data Flow Diagrams

- **Data Flow Diagrams (DFD)**
  - Provide graphical representation of a fundamental system structure and interfaces
  - Do not show control flows
- **Elements in a Data Flow Diagram:**
  - **Data Flows**
    - **Labeled arrows**: to indicate information exchanged between two transformations
    - Flow type: Continuous, on demand asynchronous etc.
  - **Data Transformations**
    - **Bubbles**: indicate processes/subsystems to depict internal entities
  - **Data Stores**
    - **Horizontal Parallel lines**: indicate Transactions, dB
  - **External Entities**
    - Rectangles: indicates entities that interact with the system
- **DFD must be accompanied by textual descriptions of data Flows, data Transformations and data Stores**
  - Data Dictionary defines data Stores and Flows

## Modeling Requirements

### Data Flow Diagrams

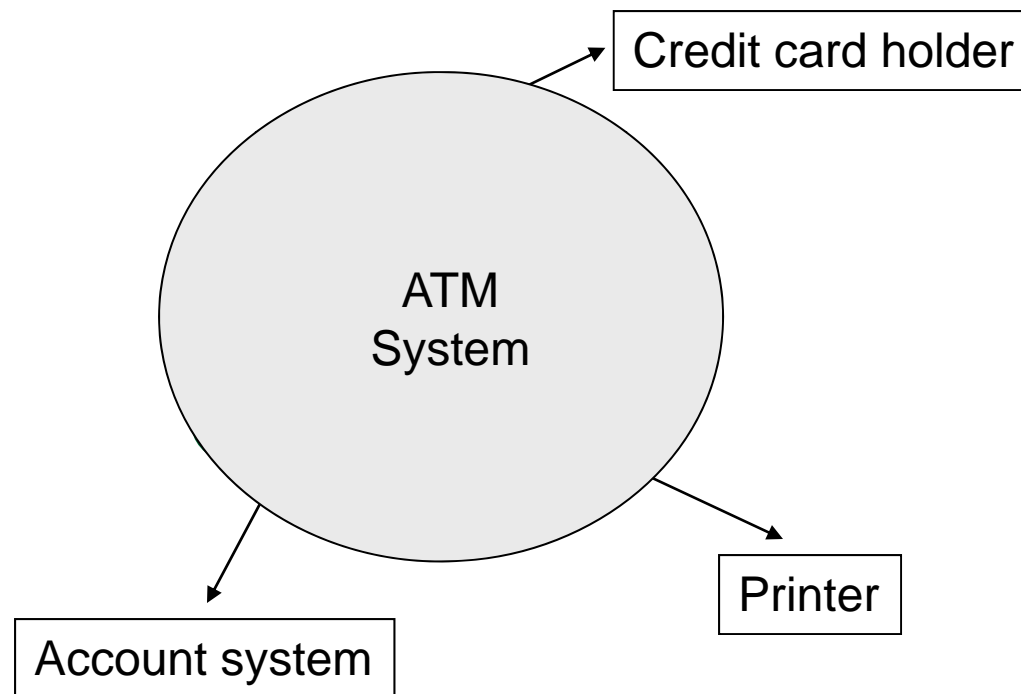


Example Data Flow Diagram for ATM System

## Modeling Requirements

### Developing a Data Flow Diagram

- Start with a Context Diagram
  - Show all external systems that interact with proposed system



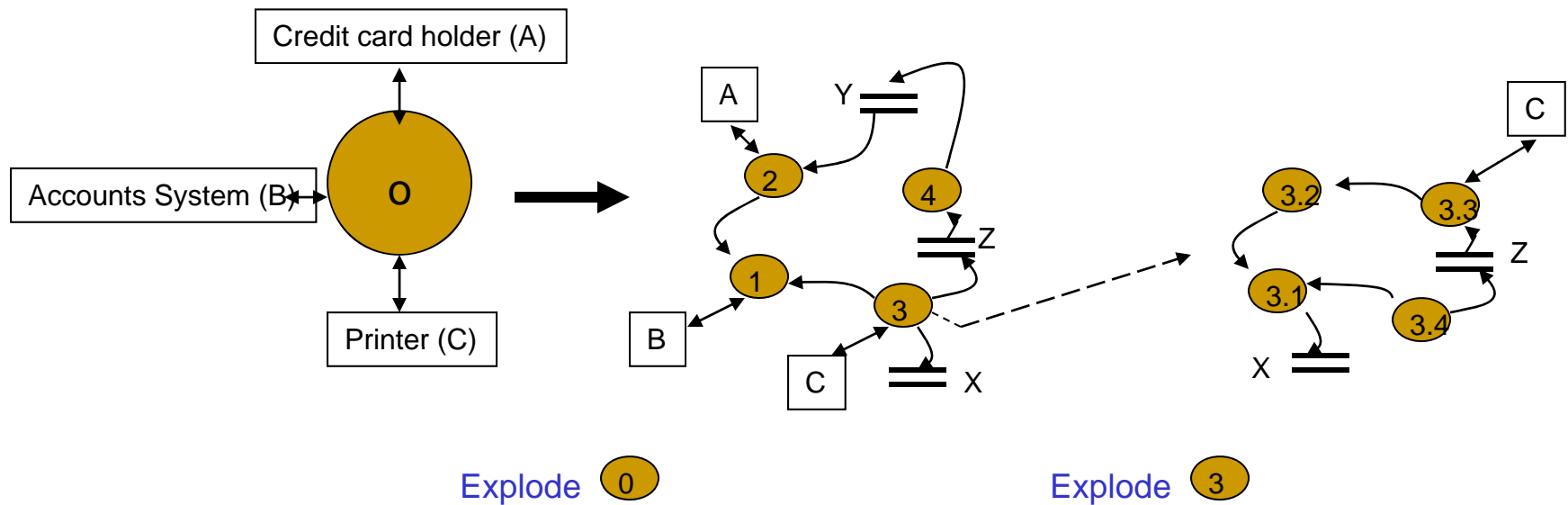
Fundamental Data Flow Diagram for ATM System



## Modeling Requirements

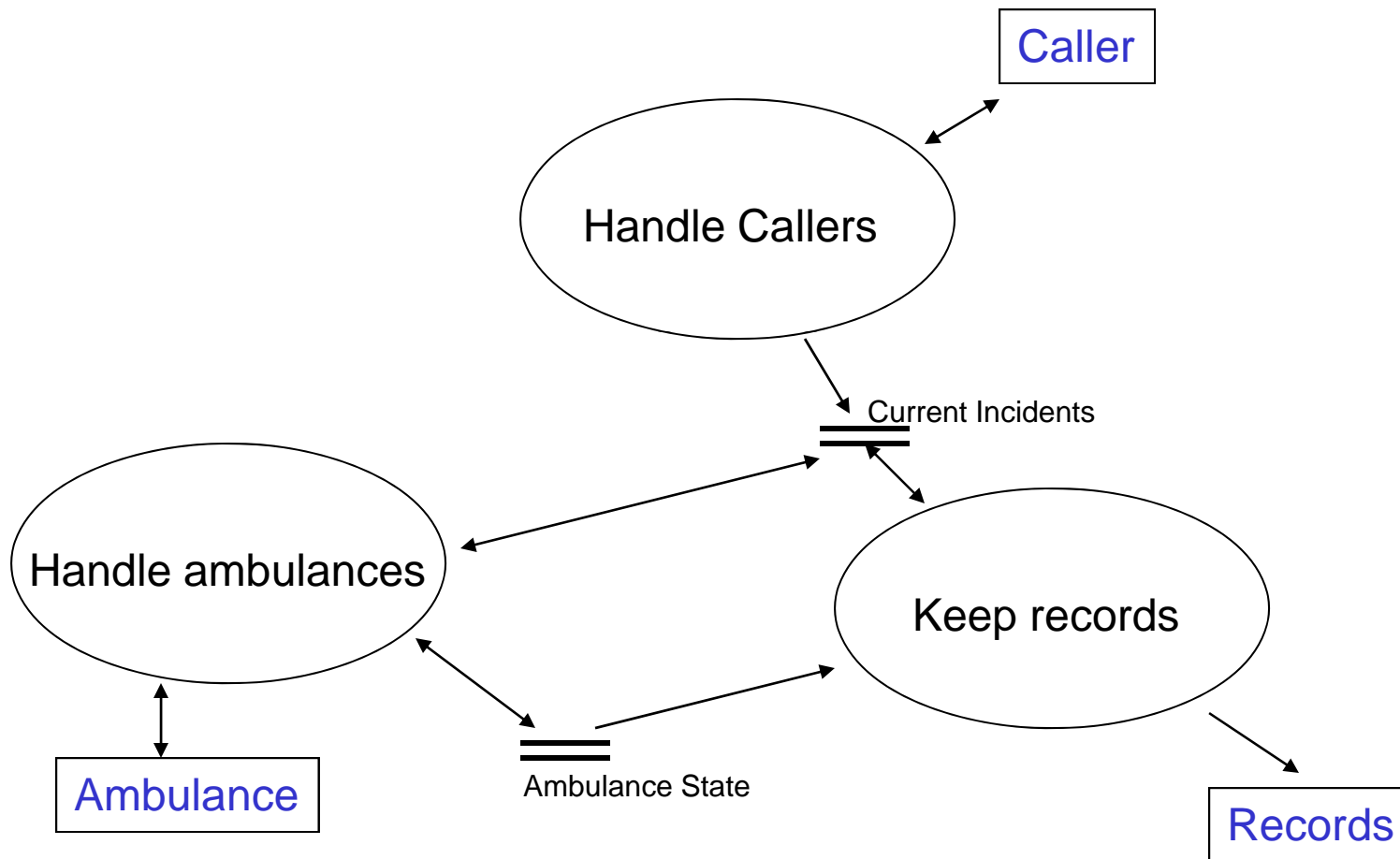
### Developing a Data Flow Diagram

- Next, progressively expand (“explode”) each bubble
  - Detailed processes (bubbles), data stores and/or data flows



## Developing a Data Flow Diagram

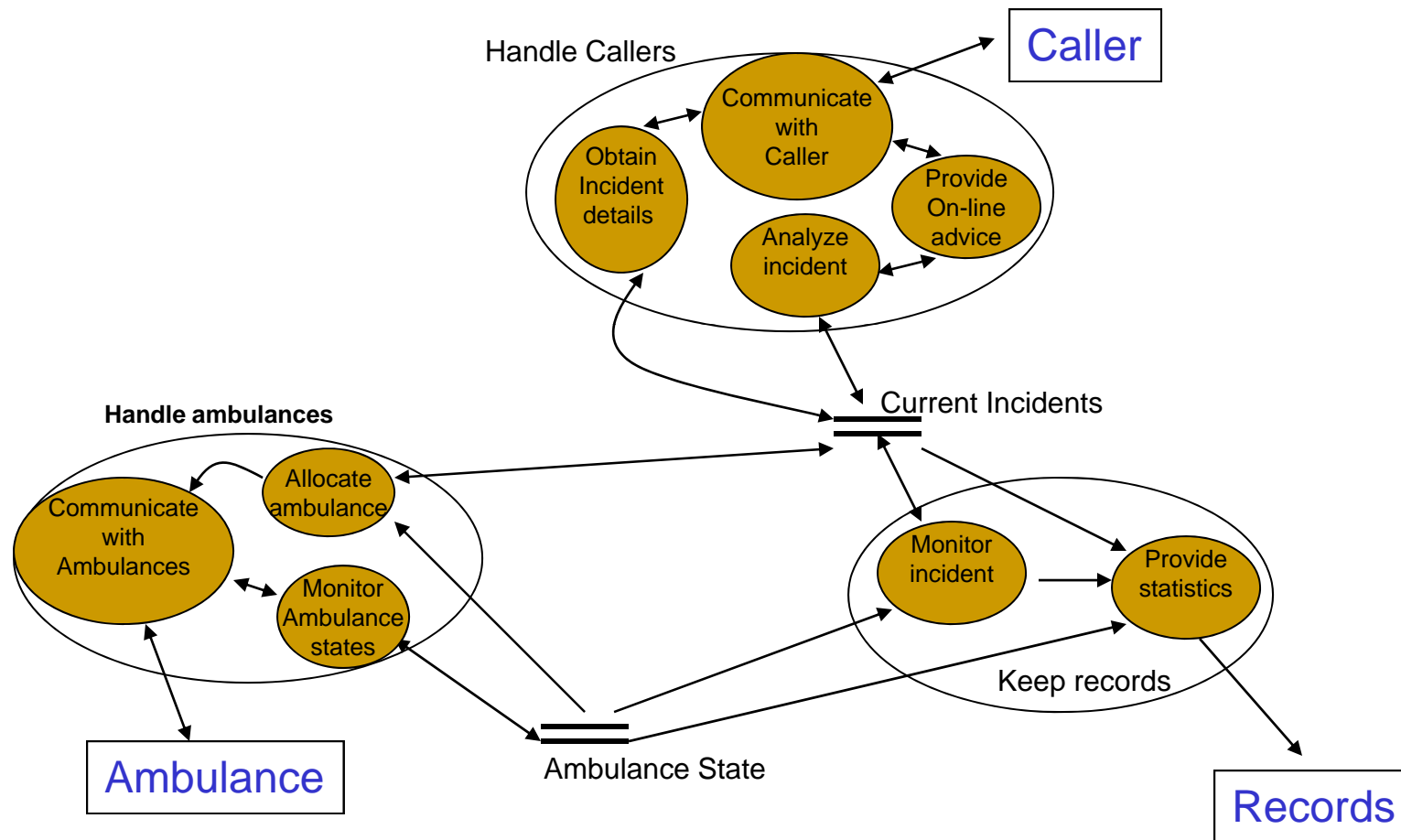
### Model for Ambulance Command & Control (AC&C) System



Model for Ambulance C&C System

## Developing a Data Flow Diagram

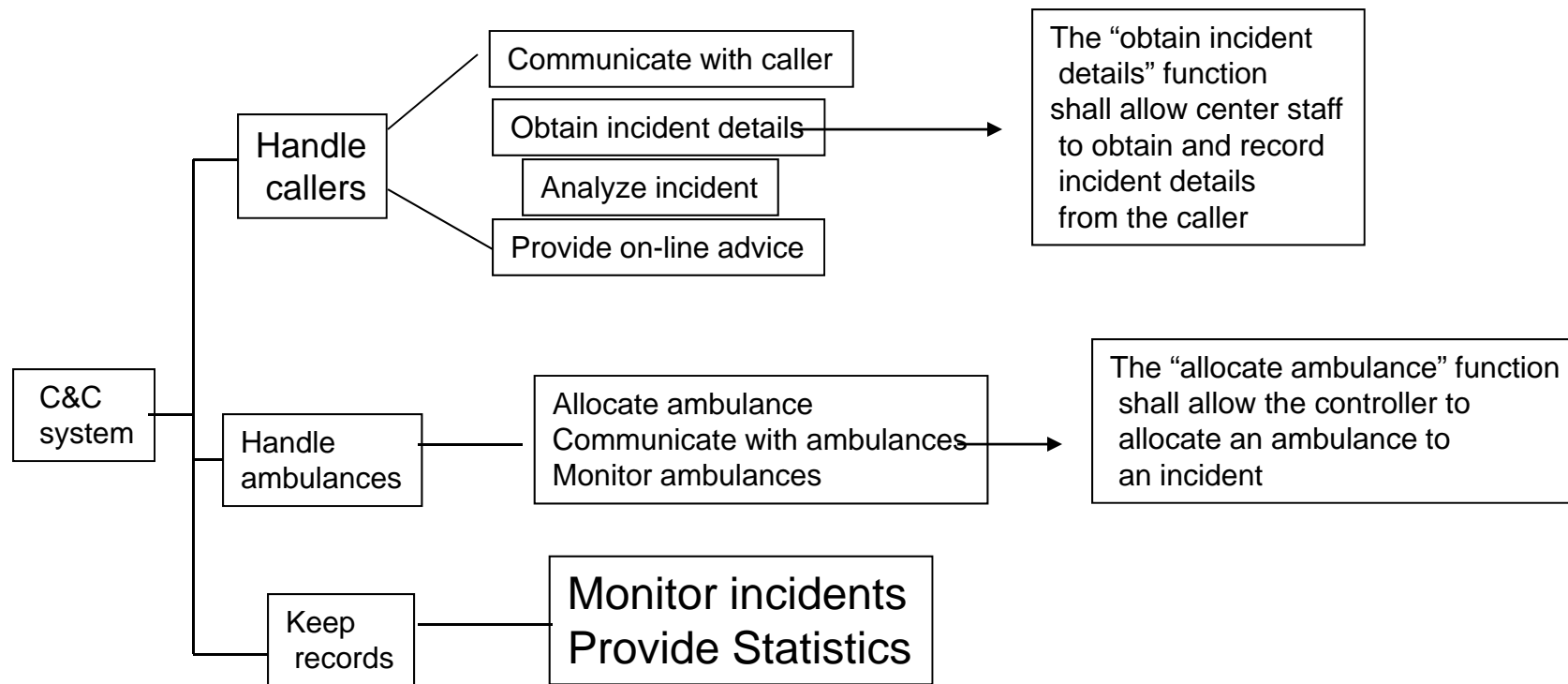
### Model for Ambulance Command & Control (AC&C) System



Detail Model for Ambulance C&C System

## Developing a Data Flow Diagram

### Functional structure of Ambulance Command & Control (AC&C) System



---

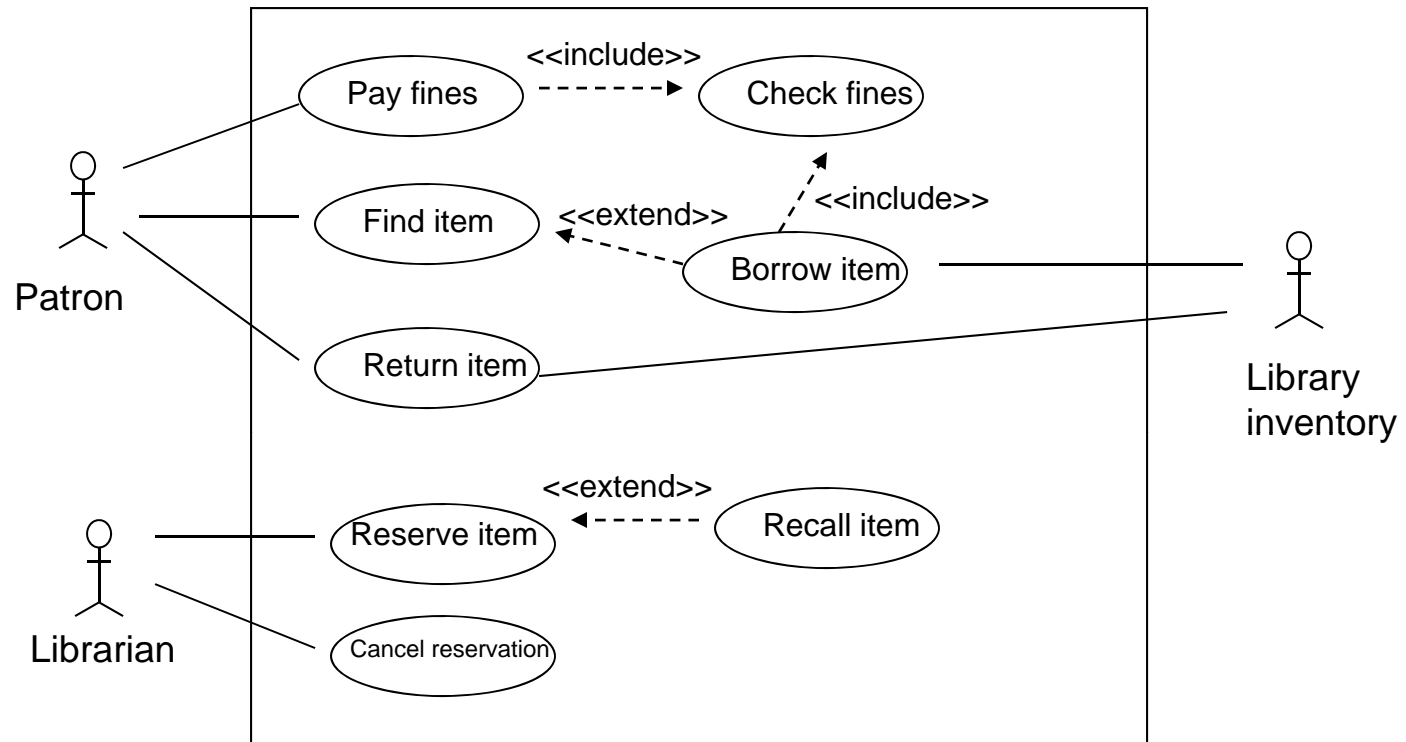
## Modeling Requirements

### Use Cases

- A Use Case:
  - Describes particular functionality of a system
    - Modeling the interactions (dialog) that a user, external system or other entity (**actor**) will have with the system being developed
    - Each use case describes possible scenarios of how the actor interacts with the system
- Use Cases are presented as
  - Drawings of pertinent objects plus
  - Brief textual description of how the function is performed
    - For each scenario the use case identifies:
      - All possible events
      - System's response

The collection of all use cases provides a complete description of the different ways of using the system by all possible entities

## Modeling Requirements Use Cases



Example: Library Use Cases

Can you Identify the actors?  
How about the use cases?

# Modeling Requirements

## Use Cases

### Actors:

Patron (Human)  
Librarian (Human – but different user)  
Library inventory (system/shelf)

### Base Case:

Pay fines  
Find item  
Return item  
Reserve item  
Borrow item

### Subcase:

#### Check fines

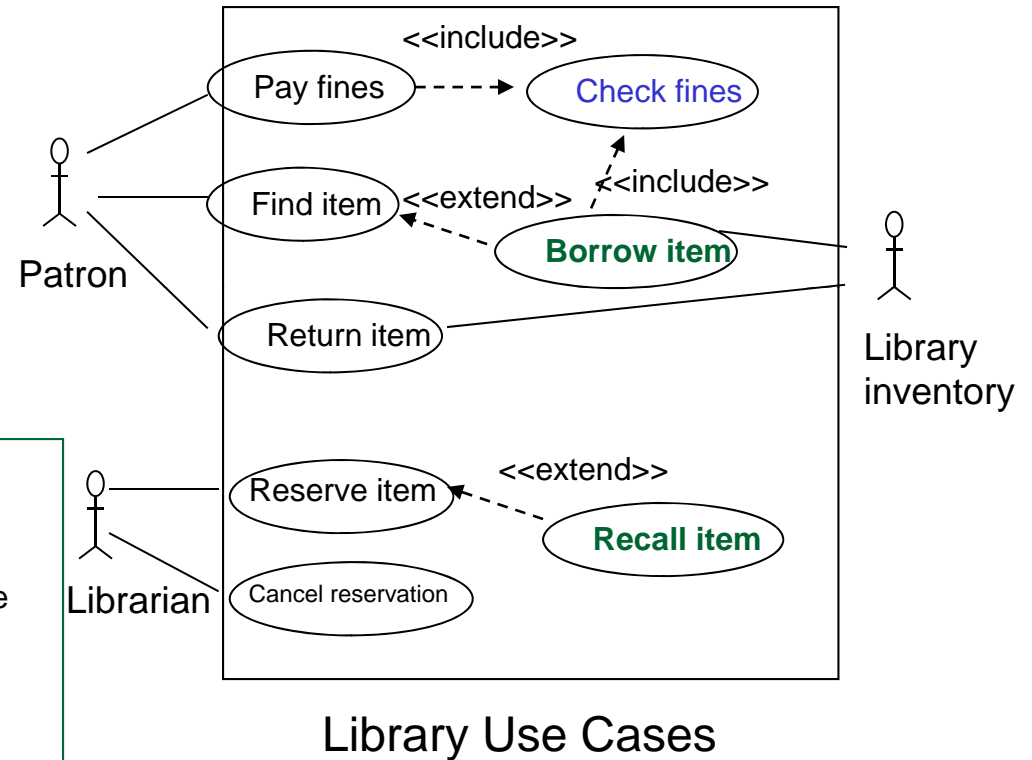
- Common to Base Cases: Pay fines and Borrow item
  - Dashed arrows goes from Base Case to Subcase  
<<include>>

#### Recall item (extension subcase)

- Adds functionality to base case Reserve item
  - Dashed arrowed line points to Base case

#### Borrow item (extension subcase)

- Adds functionality to base case Reserve item
  - Dashed arrowed line points to Base case  
<<extend>>



---

## Use Cases

### Shell Gas Station

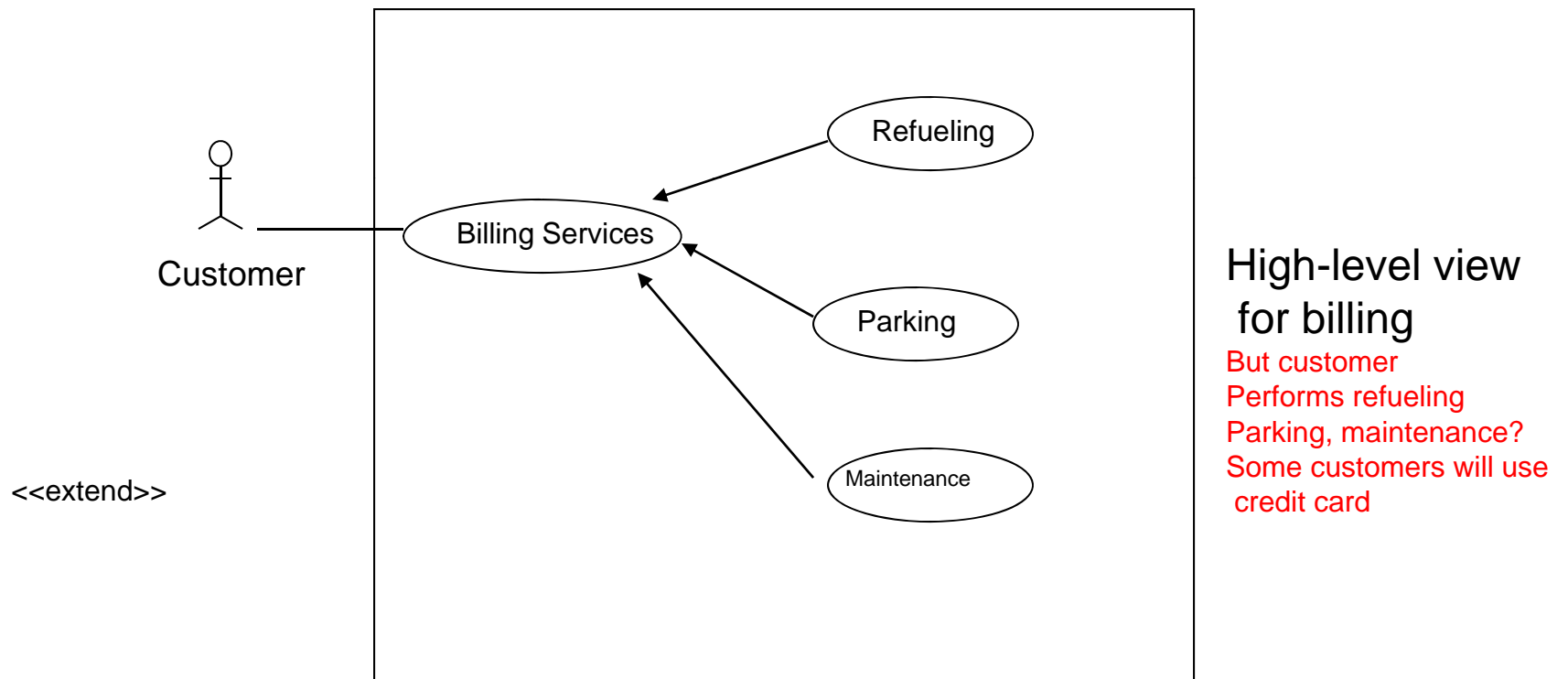
- Lets consider a shell gas station that provides
  - Three types of services: Refueling, Parking and Maintenance
  - Bill to the customer for each service
- Develop the use cases for the system



## Use Cases

### Shell Gas Station

- Start with the fundamental use case diagram

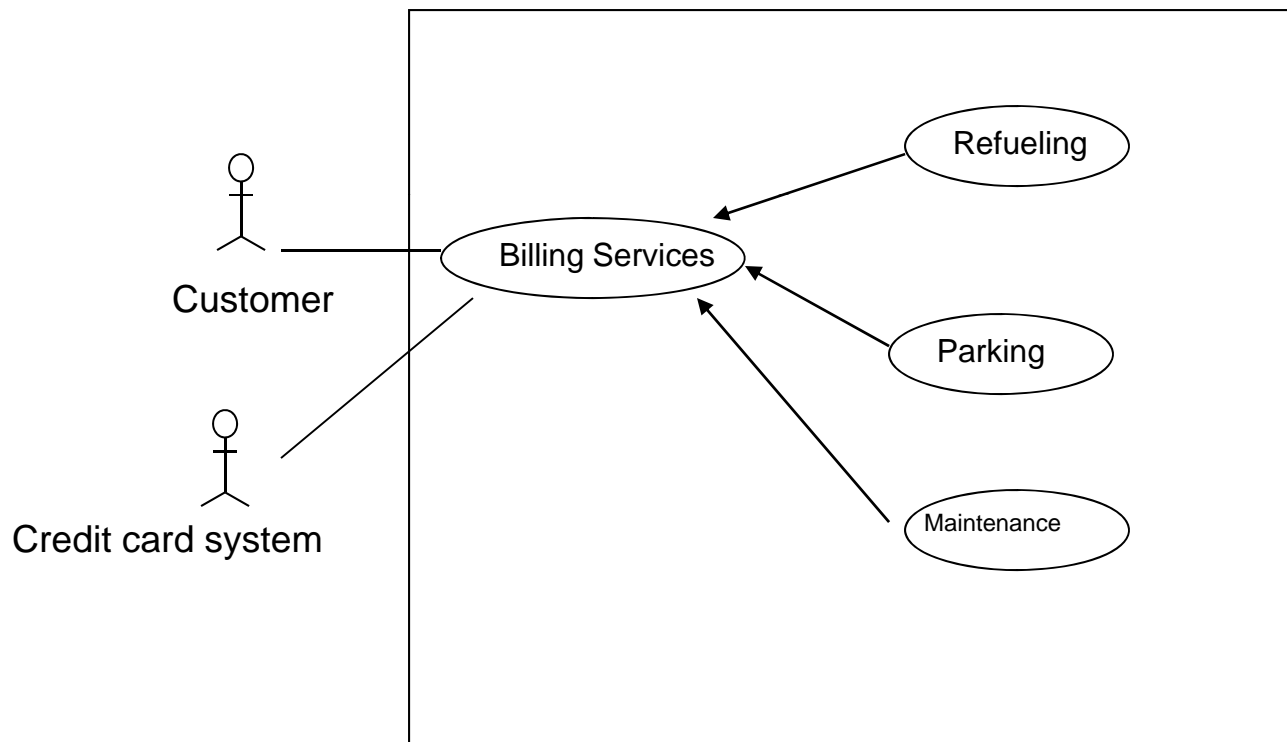


Next, Discover new aspects of the problem by asking questions

## Use Cases

### Shell Gas Station

#### ■ Add credit card payment option



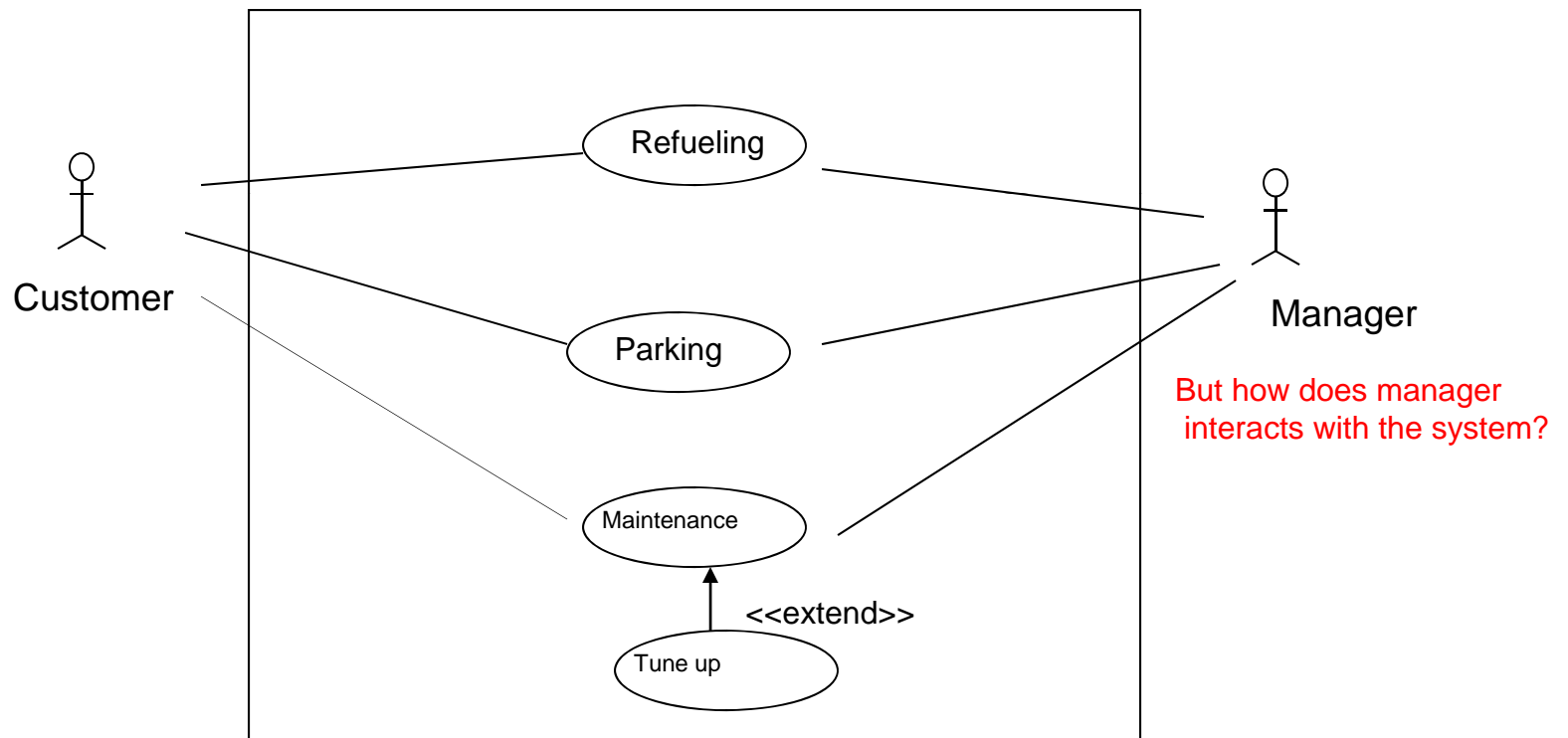
High-level view  
for billing

Some will pay with Visa?

## Use Cases

### Shell Gas Station

- Add manager; gas station offers Tune up services

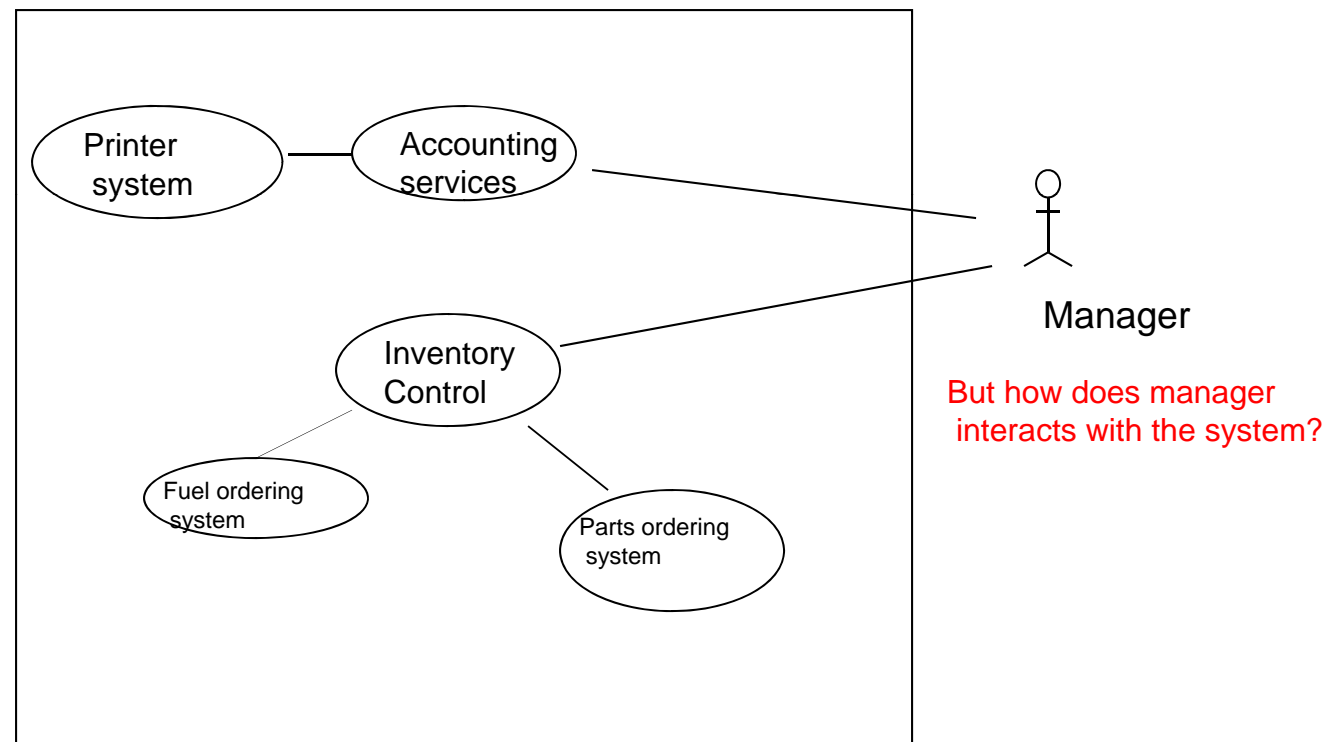


Next, Discover new aspects of the problem by asking questions

## Use Cases

### Shell Gas Station

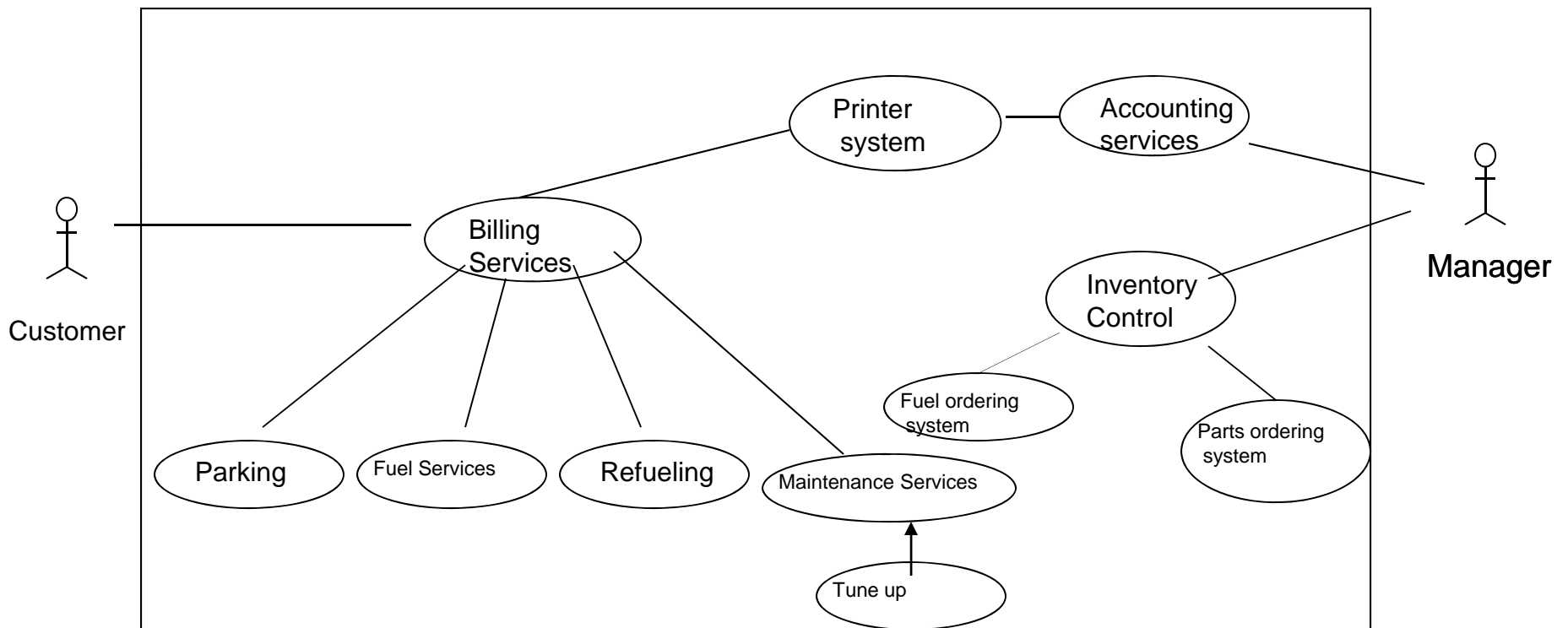
- Managerial role is different from customer
  - Accounting, inventory, orders parts for the services



## Use Cases

### Shell Gas Station

- Collection of Use Cases for the various scenarios

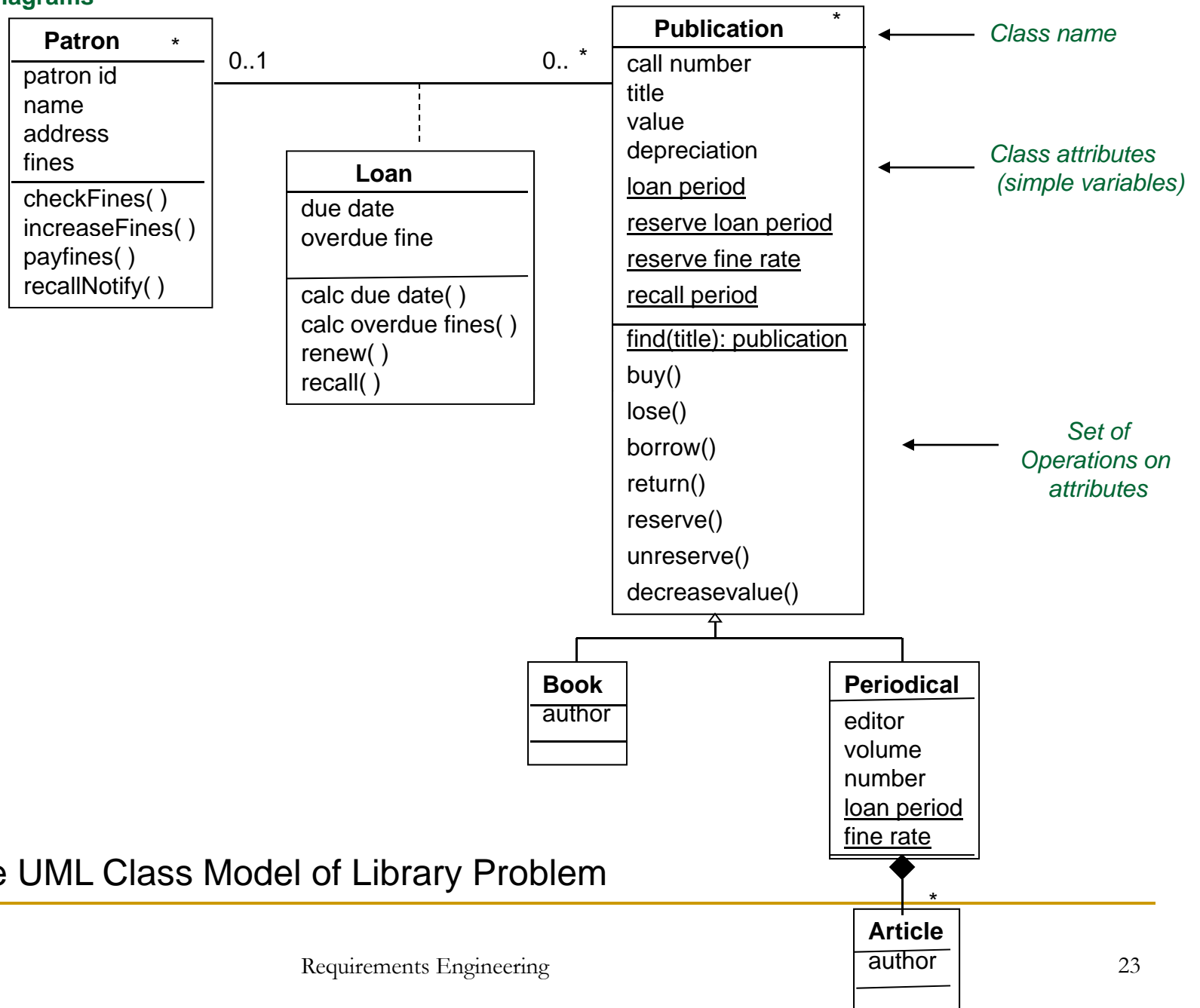


## ■ UML

- ❑ Represents systems in terms of objects and methods
- ❑ Objects (analogous to Entities)
  - Organized in classes that have inheritance
  - Each provides methods that perform actions on the object variables
  - As objects executes they:
    - ❑ Send messages to invoke each others method
    - ❑ Acknowledge actions
    - ❑ Transmit data

**Class Diagram** is an ER that relates classes (entities) in a UML model

## UML Class Diagrams



Example UML Class Model of Library Problem

---

# UML Class Diagrams

## Notations

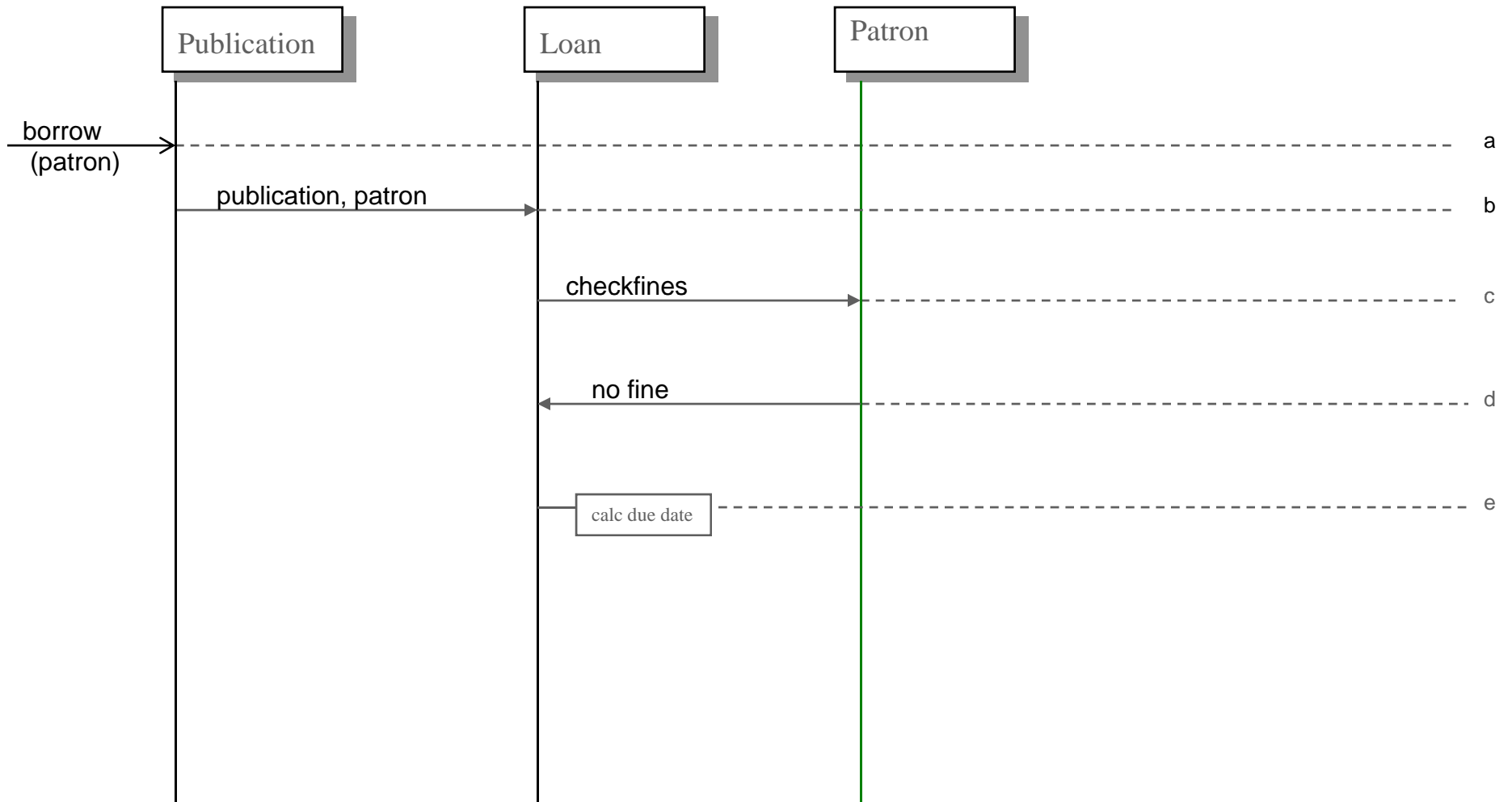
- **Class-Scope Attribute (Underlined attributes)**
  - Attributes associated with the class rather than instances of the class
    - Loan period, fine rate apply to all publications
    - Librarian can modify loan duration for all books, all periodicals but not for individual books or individual periodicals
- **Class-Scope Operations (Underlined operations)**
  - Operation performed by abstract class on a new instance or a whole collection of instances
- **Association (Line between two entities)**
  - Indicates relationship between the classes' entities
    - Patron *borrow*s a Publication
- **Aggregate Association (line with clear diamond end)**
  - **Class** at the diamond end owns (includes instances) of class at the other end of the line
  - **Composition Association** (line with black diamond)
    - Instances of compound class are physically constructed from classes of component classes
    - *Periodicals is composed of Articles*
- **Generalization Association (Line with triangle on one end)**
  - **Publication** is a parent class of subclasses: **Book** and **Periodical**
  - **A subclass inherits all the parents class's attributes, operations and associations**
    - It is not necessary to draw a line between Patron and Book, because Book inherits this association from Publication
- **Multiplicities**
  - **Constraints on number of links between entities**
    - Number, range, unlimited numbers (\*)



- **MSC** provides graphical representations of
  - Sequences of events exchanged between real-world entities
- Elements in **MSC**:
  - **Participating Entity**
    - Depicted as a vertical line
  - **Message**
    - Depicted as an arrow from the sending entity to receiving entity
  - **Actions**
    - Invoked operations or changes to variable values
    - Depicted as labeled rectangles

## Message Sequence Chart (MSC)

### Library Loan Request (Item available)



---

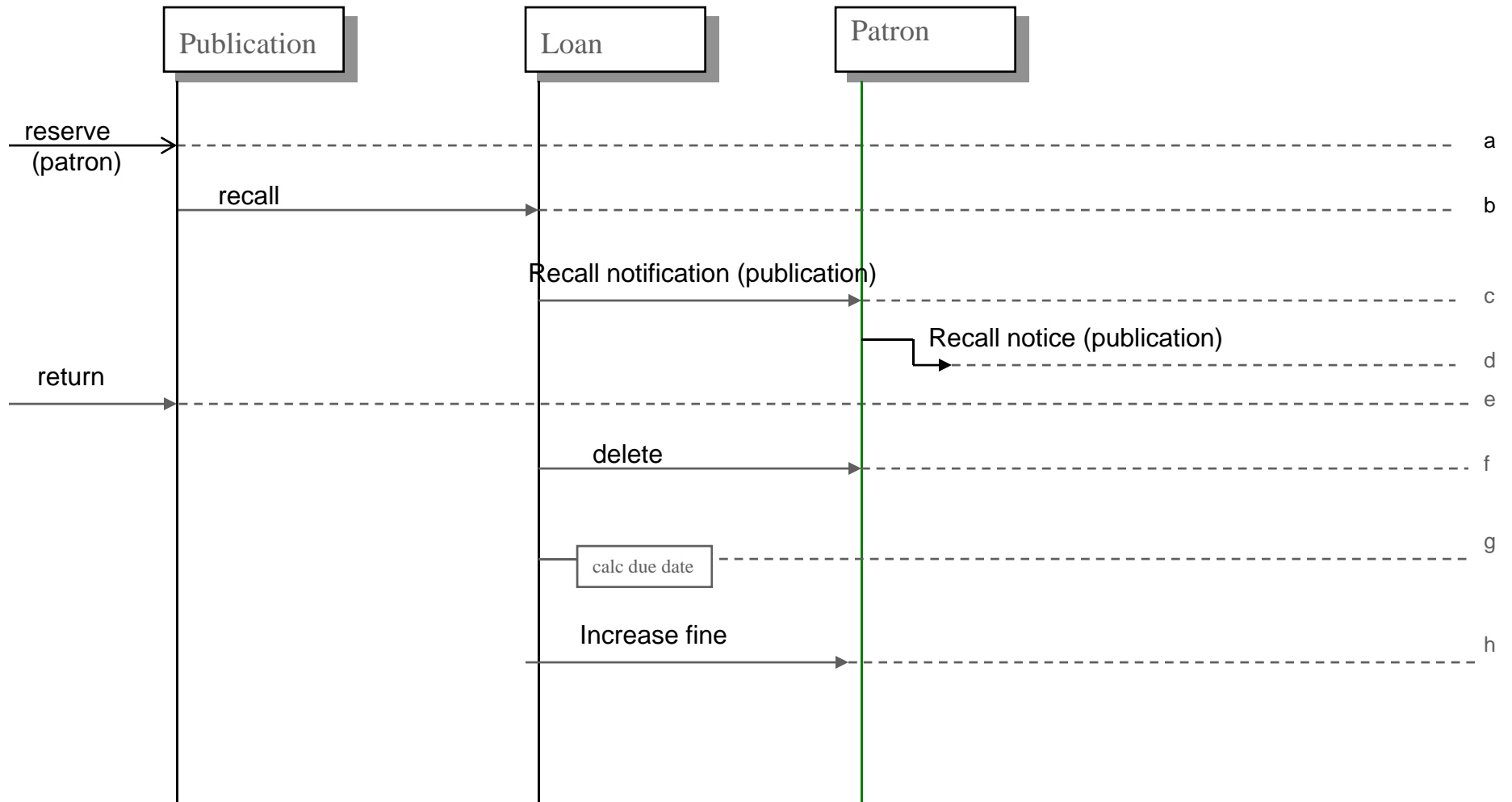
## MSC

### Operational Details

- a. Library loan request is sent to the Publication entity to be borrowed
- b. Publication creates a loan entity to manage the loan specific data
- c. Check if patron has a fine
- d. Patron does not have fines
- e. Calculate due date to return borrowed item

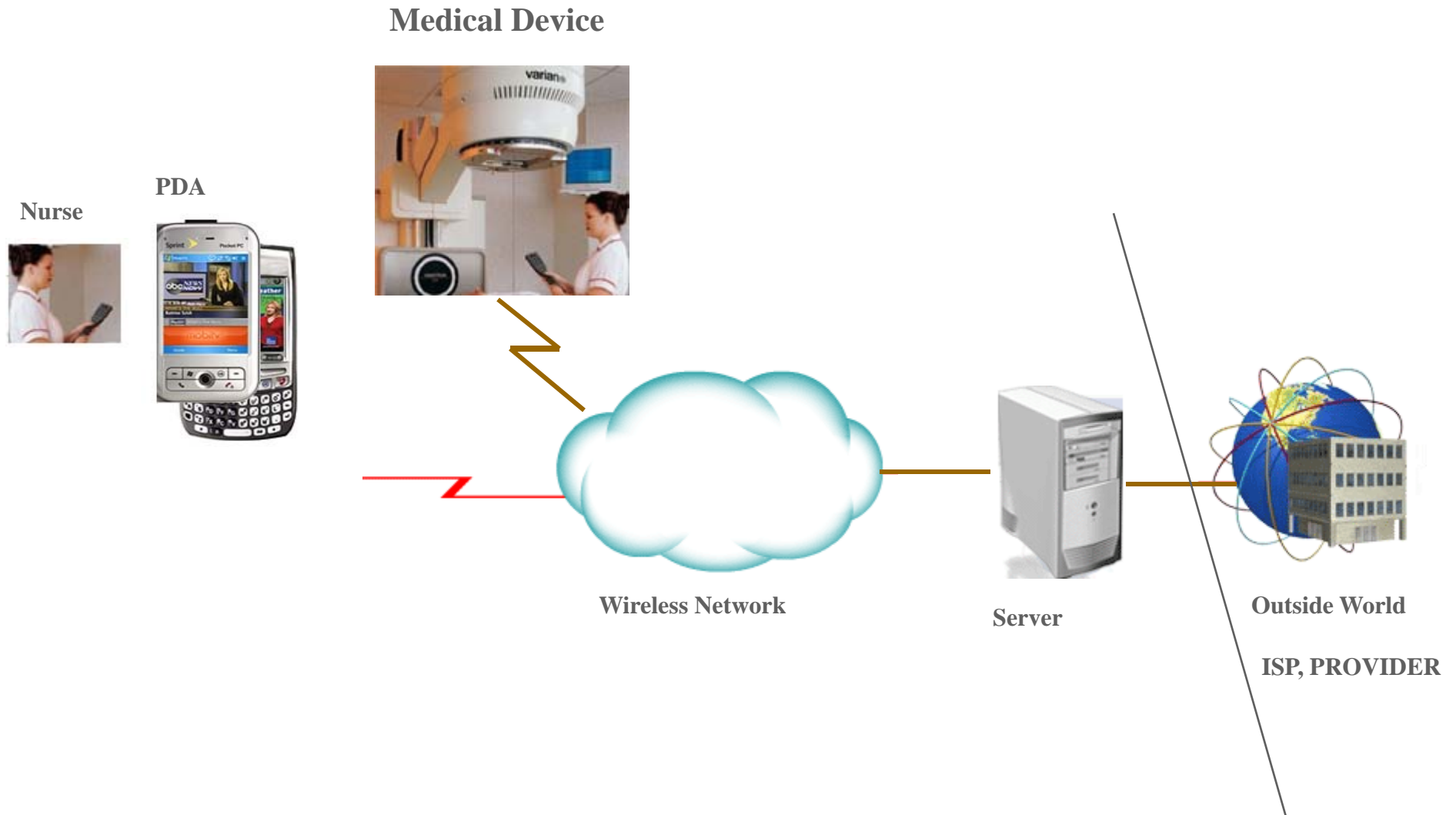
## Message Sequence Chart (MSC)

### Library Loan Request (item checked out ~ reserve)



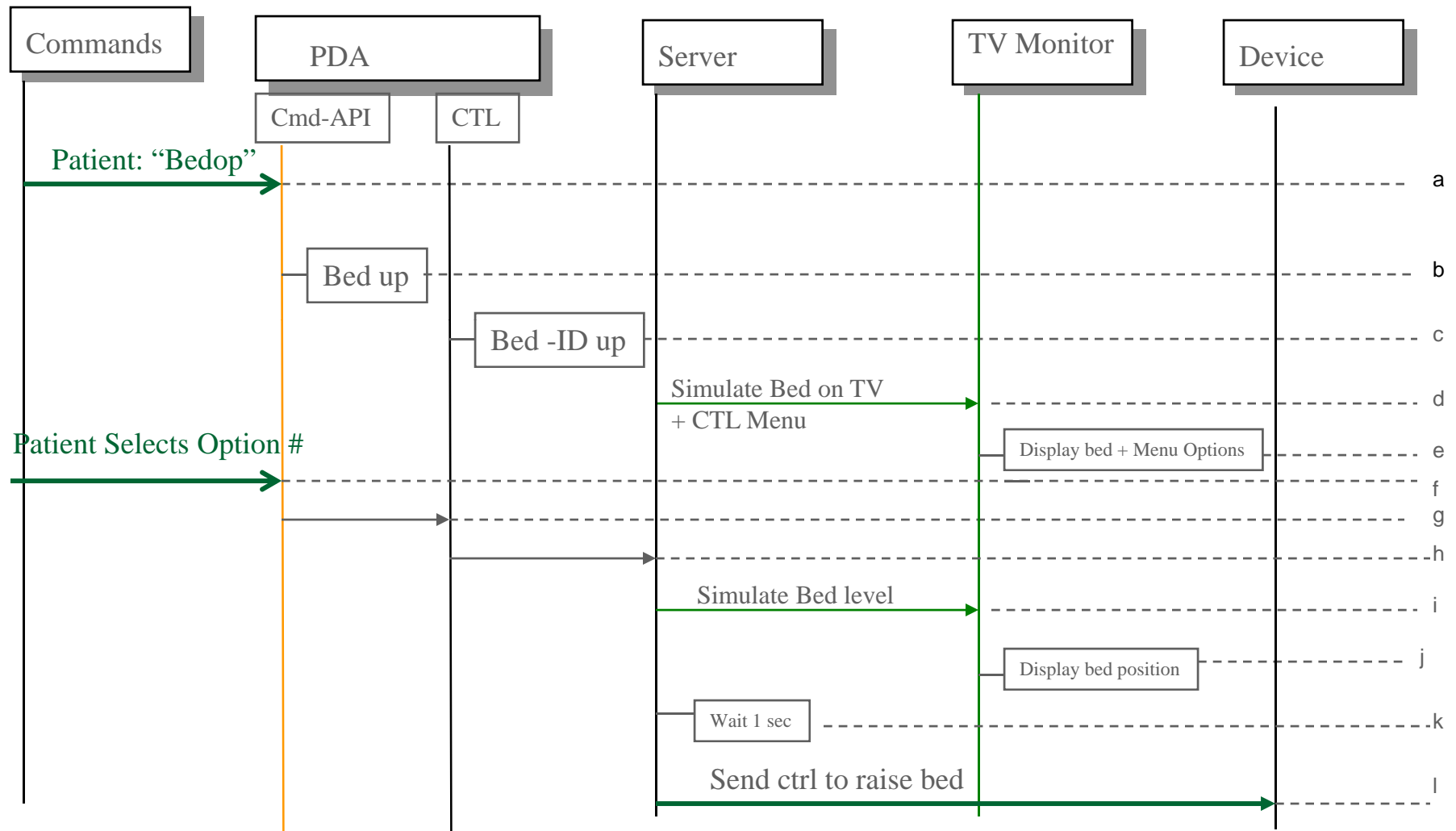
## Modeling Requirements

### Use Case Scenario & Message Sequence Chart



# Patient uses speech command to raise bed

## Use Case Scenario & MSC



## Glossary of Terms

“This section provides a glossary of terms used in this document.  
It is not meant to be a glossary of all relevant terms used in the product.”

Acronym	Definition
CCD	Control Command Definition
GUI	Graphical User Interface
JVM	Java Virtual Machine
MS	Microsoft
NE	Network Element
NEI	Network Element Interface
OA&M	Operations Administration and Maintenance
QA	Quality Assurance
SSN	Social Security Number

Example

Refers to customer (business) & technical jargons

# Case Study

Case Study materials will be presented in class