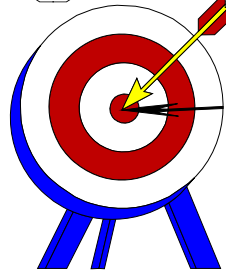




*Object-Oriented Systems  
Development:  
Using the Unified Modeling  
Language*

**Chapter 6:  
Object-Oriented Analysis  
Process: Identifying Use  
Cases**



## *Goals*

- **The use-case approach to object-oriented analysis and the object-oriented analysis process.**
- **Identifying actors.**
- **Identifying use cases.**
- **Documentation.**

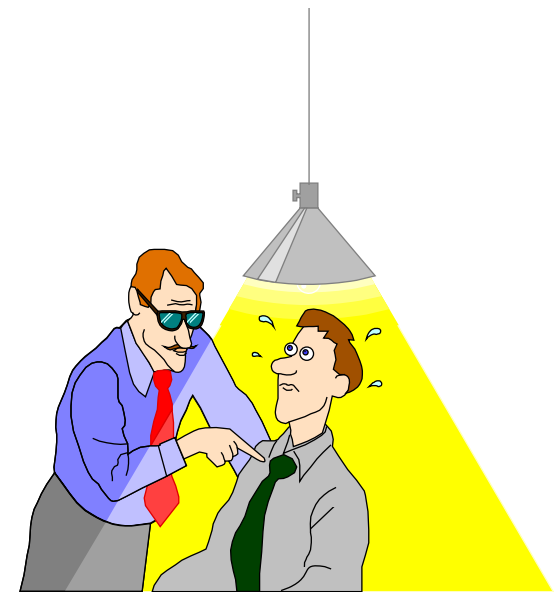
*... just think of all the Christmas presents that are never removed from their boxes before being returned.*

**Weinberg and Gause**



## *What Is Analysis?*

- **Analysis is the process of transforming a problem definition from a fuzzy set of facts and myths into a coherent statement of a system's requirements.**





## *Analysis*

- **The main objective of the analysis is to capture:**
  - a complete, unambiguous, and consistent picture of the requirements of the system and
  - what the system must do to satisfy the users' requirements and needs.

## *Where Should We Start?*

- **1. Examination of existing system documentation.**
- **2. Interviews.**
- **3. Questionnaire.**
- **4. Observation.**



# *Requirements Difficulties*

**Three most common sources of requirements difficulties are:**

- 1. Incomplete requirements.**
- 2. Fuzzy descriptions (such as fast response).**
- 3. Unneeded features.**



# *The Object-Oriented Analysis (OOA) Process*

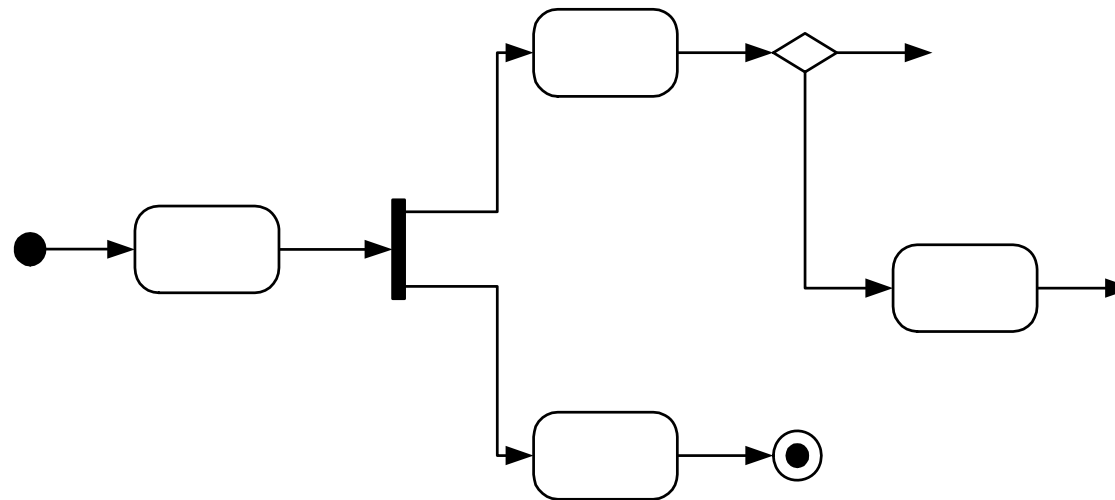
- **The process consists of the following steps:**
- **1. Identify the actors:**
  - Who is using the system?
  - Or, in the case of a new system, who will be using the system?





## *The OOA Process (Con't)*

- **2. Develop a simple business process model using UML activity diagram.**





## *The OOA Process (Con't)*

- **3. Develop the use case:**
  - **What the users are doing with the system?**
  - **Or, in the case of a new system, what users will be doing with the system?**

**Use cases provide us with comprehensive documentation of the system under study.**



## *The OOA Process (Con't)*

- **4. Prepare interaction diagrams:**
  - Determine the sequence.
  - Develop collaboration diagrams.

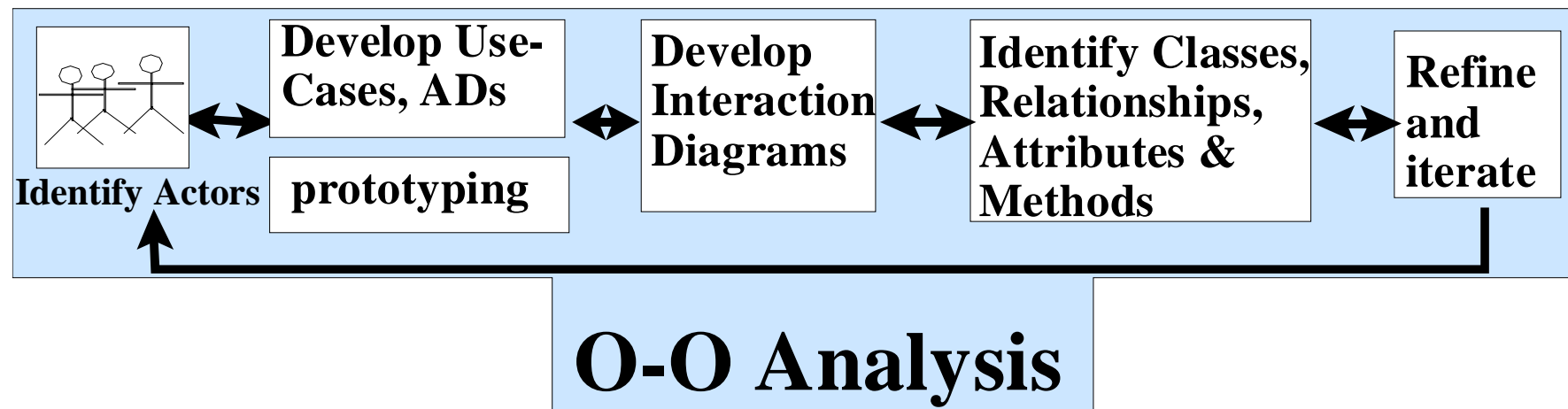


## *The OOA Process (Con't)*

- **5. Classification – develop a static UML class diagram:**
  - Identify classes.
  - Identify relationships.
  - Identify attributes.
  - Identify methods.

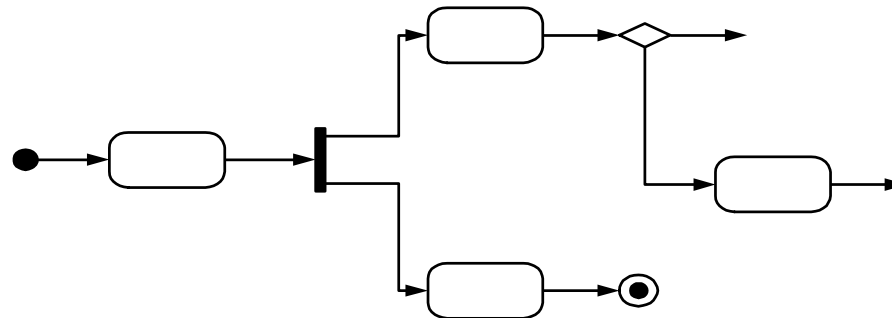
## *The OOA Process (Con't)*

- **6. Iterate and refine: If needed, repeat the preceding steps.**



# Developing Business Processes

- **Developing an activity diagram of the business processes can provide us with an overall view of the system.**





## *Use Case Model*

- **Use cases are scenarios for understanding system requirements.**
- **The use-case model describes the uses of the system and shows the courses of events that can be performed.**

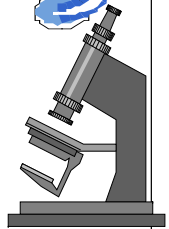


## *Use Case Model (Con't)*

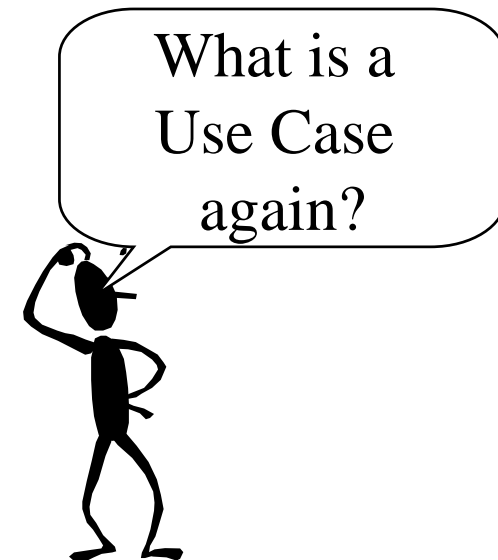
- **Use case defines what happens in the system when a use case is performed.**
- **The use-case model tries to systematically identify uses of the system and therefore the system's responsibilities.**



## *Use Cases Under the Microscope*



- "A *Use Case* is a sequence of *transactions in a system* whose task is to yield results of *measurable value* to an individual *actor* of the system."





## *Use Case Key Concepts*

- *Use case.* Use case is a special flow of events through the system.
- *Actors.* An actor is a user playing a role with respect to the system.
- *In a system.* This simply means that the actors communicate with the system's use case.



## *Use Case Key Concepts (Con't)*

- *A measurable value.* A use case must help the actor to perform a task that has some identifiable value.
- *Transaction.* A transaction is an atomic set of activities that are performed either fully or not at all.



## *Use Associations*

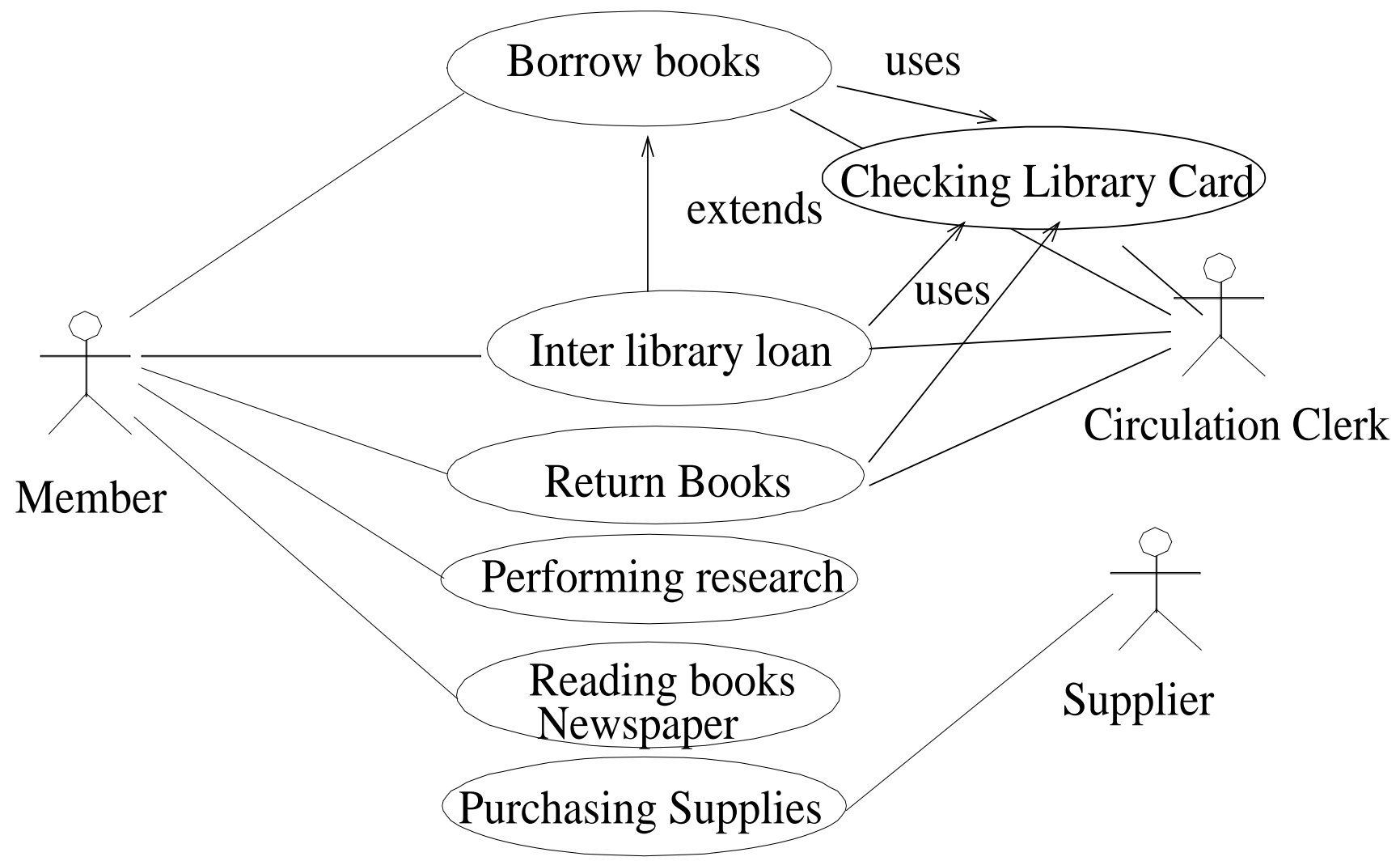
- The *use* association occurs when you are describing your use cases and notice that some of them have common subflows.
- The *use* association allows you to extract the common subflow and make it a use case of its own.



## *Extends Associations*

- The *extends* association is used when you have one use case that is similar to another use case but does a bit more or
- Is more specialized; in essence, it is like a subclass.

# Library





## *Types of Use Cases*

- Use cases could be viewed as **concrete** or **abstract**.
- An *abstract use case* is not complete and has no initiation actors but is used by a *concrete use case*, which does interact with actors.

## Identifying the Actors

- The term *actor* represents the role a user plays with respect to the system.
- When dealing with actors, it is important to think about roles rather than people or job titles.







## *Identifying the Actors (Con't)*

- **Candidates for actors can be found through the answers to the following questions:**
  - **Who is using the system? Or,**
  - **Who is affected by the system? Or,**
  - **Which groups need help from the system to perform a task?**



## *Identifying the Actors (Con't)*

- **Who affects the system? Or,**
- **Which user groups are needed by the system to perform its functions? These functions can be both main functions and secondary functions, such as administration.**
- **Which external hardware or other systems (if any) use the system to perform tasks?**



## *Identifying the Actors (Con't)*

- What problems does this application solve (that is, for whom)?**
- And, finally, how do users use the system (use case)? What are they doing with the system?**



## *Guidelines for Finding Use Cases*

- **For each actor, find the tasks and functions that the actor should be able to perform or that the system needs the actor to perform.**
- **Name the use cases.**
- **Describe the use cases briefly by applying terms with which the user is familiar.**

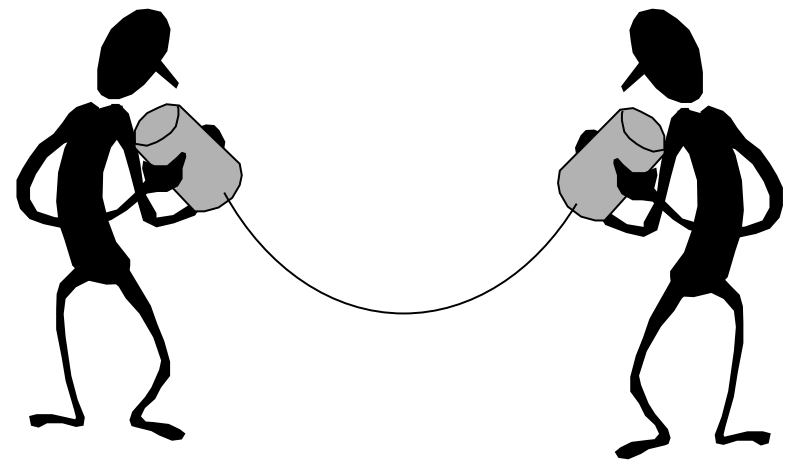


## *Separate Actors From Users*

- **Each use case should have only one main actor.**
- **Isolate users from actors.**
- **Isolate actors from other actors (separate the responsibilities of each actor).**
- **Isolate use cases that have different initiating actors and slightly different behavior.**

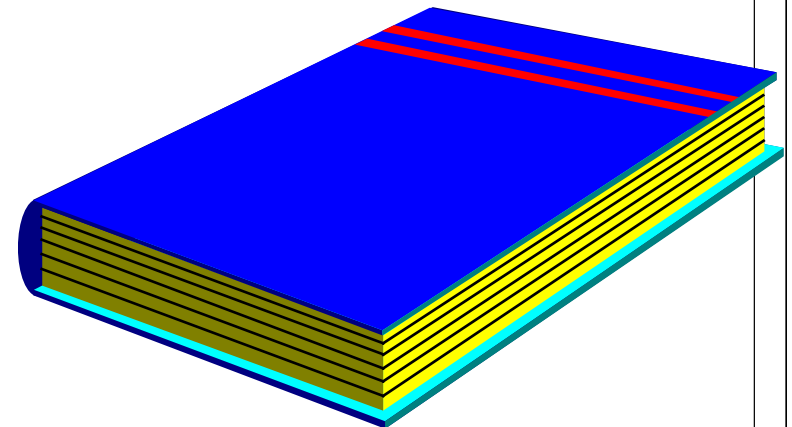
# *Documentation*

- **An effective document can serve as a communication vehicle among the project's team members, or it can serve as initial understanding of the requirements.**



# *Effective Documentation:* *Common Cover*

- **All *documents* should share a common cover sheet that identifies the document, the current version, and the individual responsible for the content.**





## *80-20 Rule*

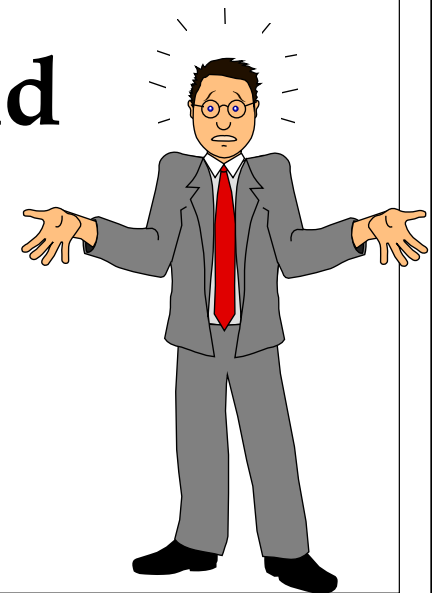
- **80 percent of the work can be done with 20 percent of the documentation.**
- **The trick is to make sure that the 20 percent is easily accessible and the rest (80 percent) is available to those (few) who need to know.**

80%-20%



## *Familiar Vocabulary*

- Use a vocabulary that your readers understand and are comfortable with.
- The main objective here is to communicate with readers and not impress them with buzz words.



# *Make the Document as Short as Possible*

- **Eliminate all repetition;**
- **Present summaries, reviews, organization chapters in less than three pages;**
- **Make chapter headings task oriented so that the table of contents also could serve as an index.**





## *Organize the Document*

- **Use the rules of good organization (such as the organization's standards, college handbooks, Strunk and White's *Elements of Style*, or the University of Chicago *Manual of Style*) within each section.**

## *Summary*

- **The main objective of the analysis is to capture a complete, unambiguous, and consistent picture of the requirements of the system.**
- **Construct several models and views of the system to describe what the system does rather than how.**





## *Summary (Con't)*

- **Capturing use cases is one of the first things to do in coming up with requirements.**
- **Every use case is a potential requirement.**



## *Summary (Con't)*

- **The key in developing effective documentation is to eliminate all repetition; present summaries, reviews, organization chapters in less than three pages.**
- **Use the 80-20 rule: 80 percent of the work can be done with 20 percent of the documentation.**