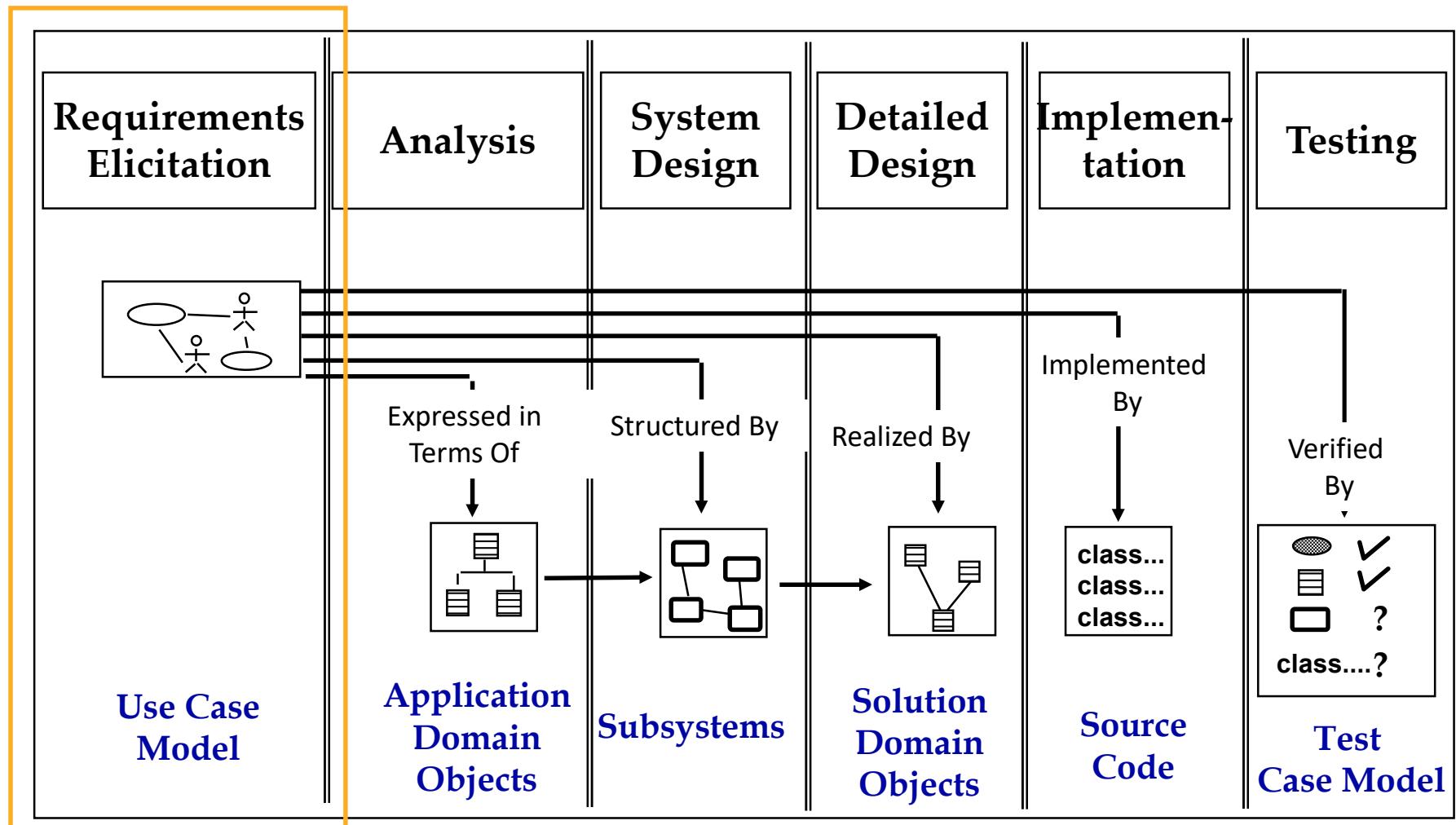


Software Lifecycle Activities



Models

- A model is an abstraction of a system
 - A system that no longer exists
 - An existing system
 - A future system to be built.



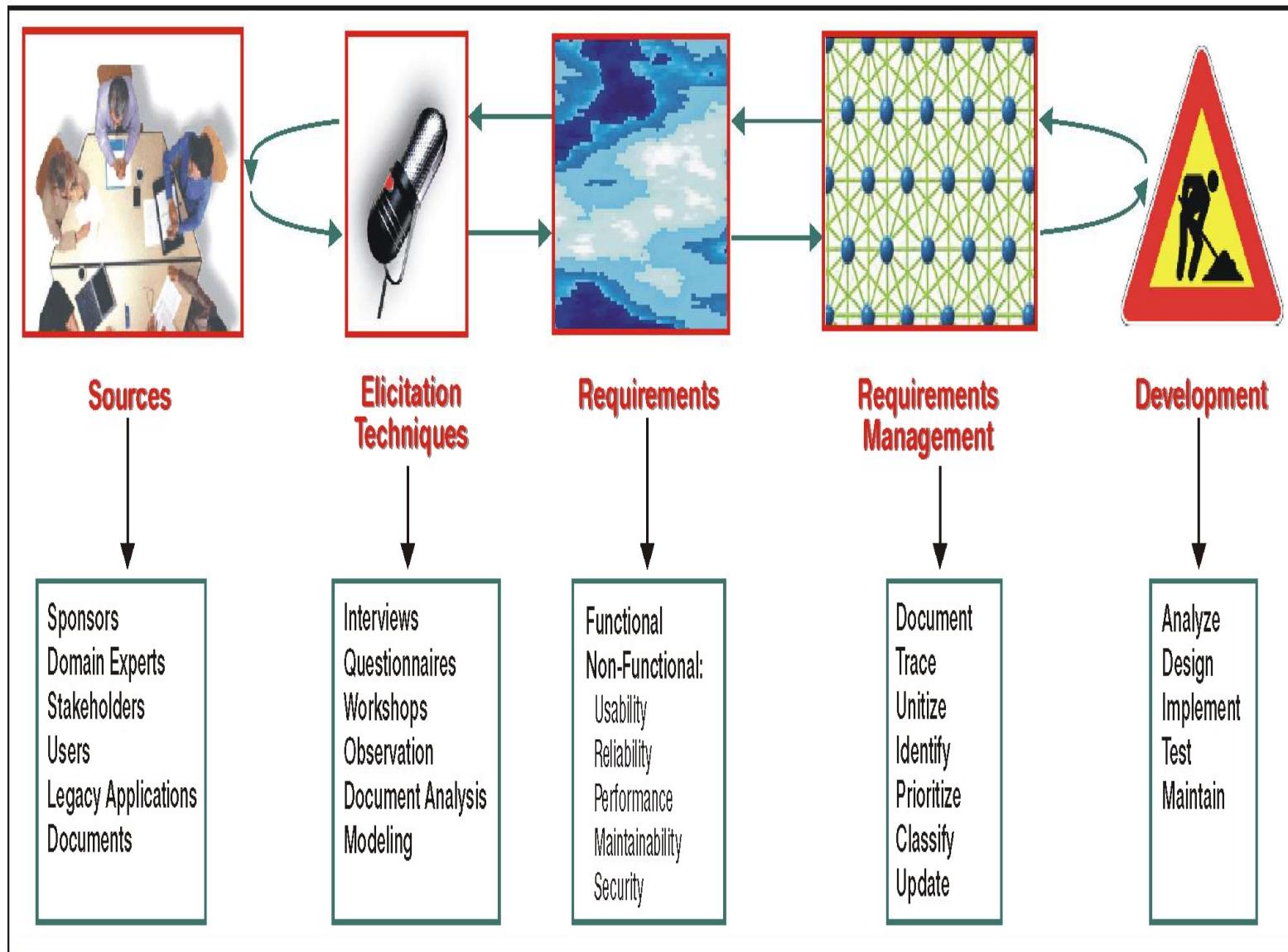
Gathering Requirements

Topics

- Define requirements
- Requirements discovery
- Classifying requirements
- Techniques for eliciting requirements
- Managing requirements
- The case history of Walden Hospital, the main source for examples we will discuss

Requirements Gathering

An Ongoing Activity



Requirements Gathering

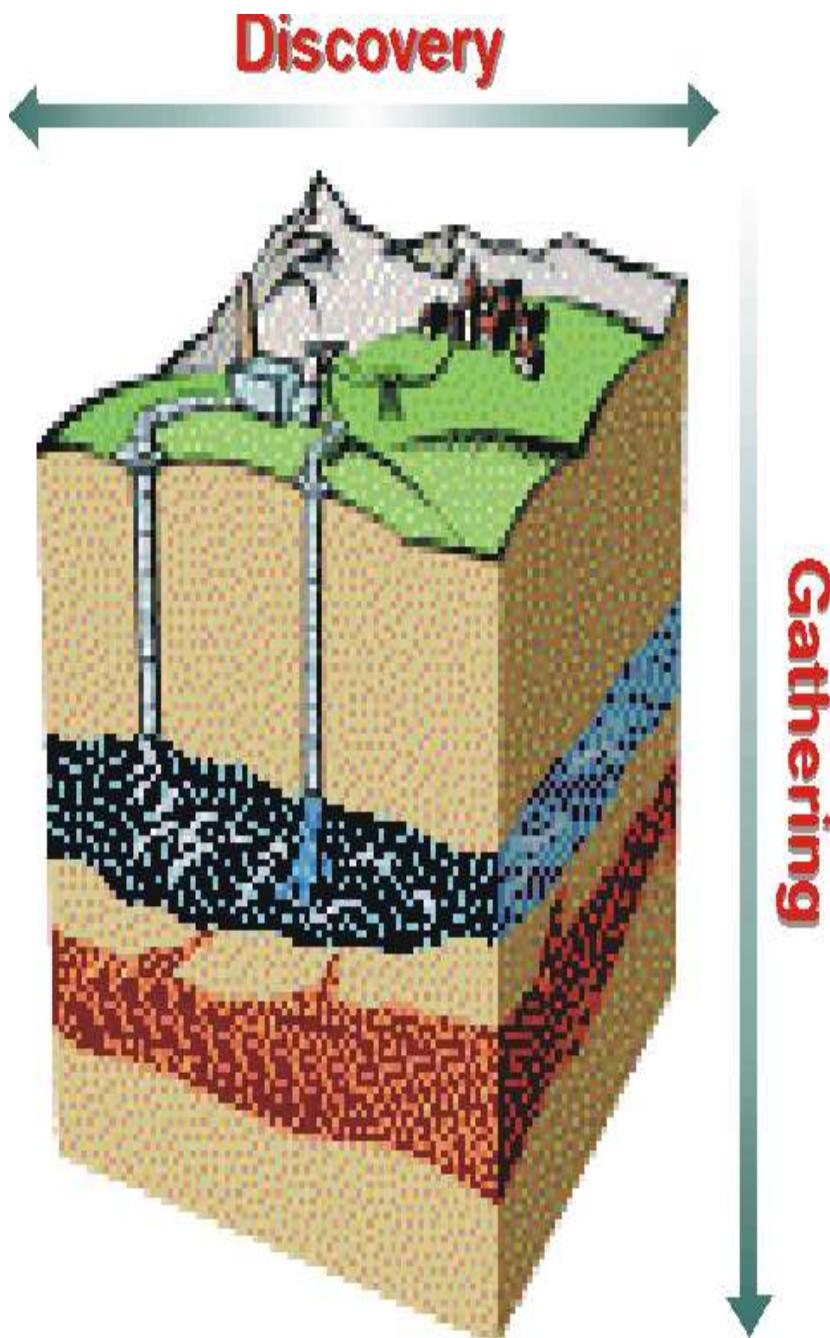
- The task of requirements gathering is to collect and define all features that the information system must have in order to fulfill the objectives that the customer has set.
- The reliability and the correctness of requirements is dependent on their sources, on the techniques that we employ to elicit and verify them, and on their effective management.

Requirements Discovery



Requirements Discovery

- Requirements discovery identifies the scope and the major objectives of the system. Requirements gathering defines what is needed to reach those objectives.



Requirements

Discovery vs. Gathering

Classifying Requirements

- Requirements fall into two broad categories: functional (or behavioral) and non-functional. Since both relate to the same product, they are interrelated and affect each other:
 - Functional Requirements
 - Functional requirements specify the behavior of the system and the constraints on that behavior.
 - Non-Functional Requirements
 - Non-functional requirements specify non-behavioral properties of the system and the constraints on those properties.

Non-Functional Requirements

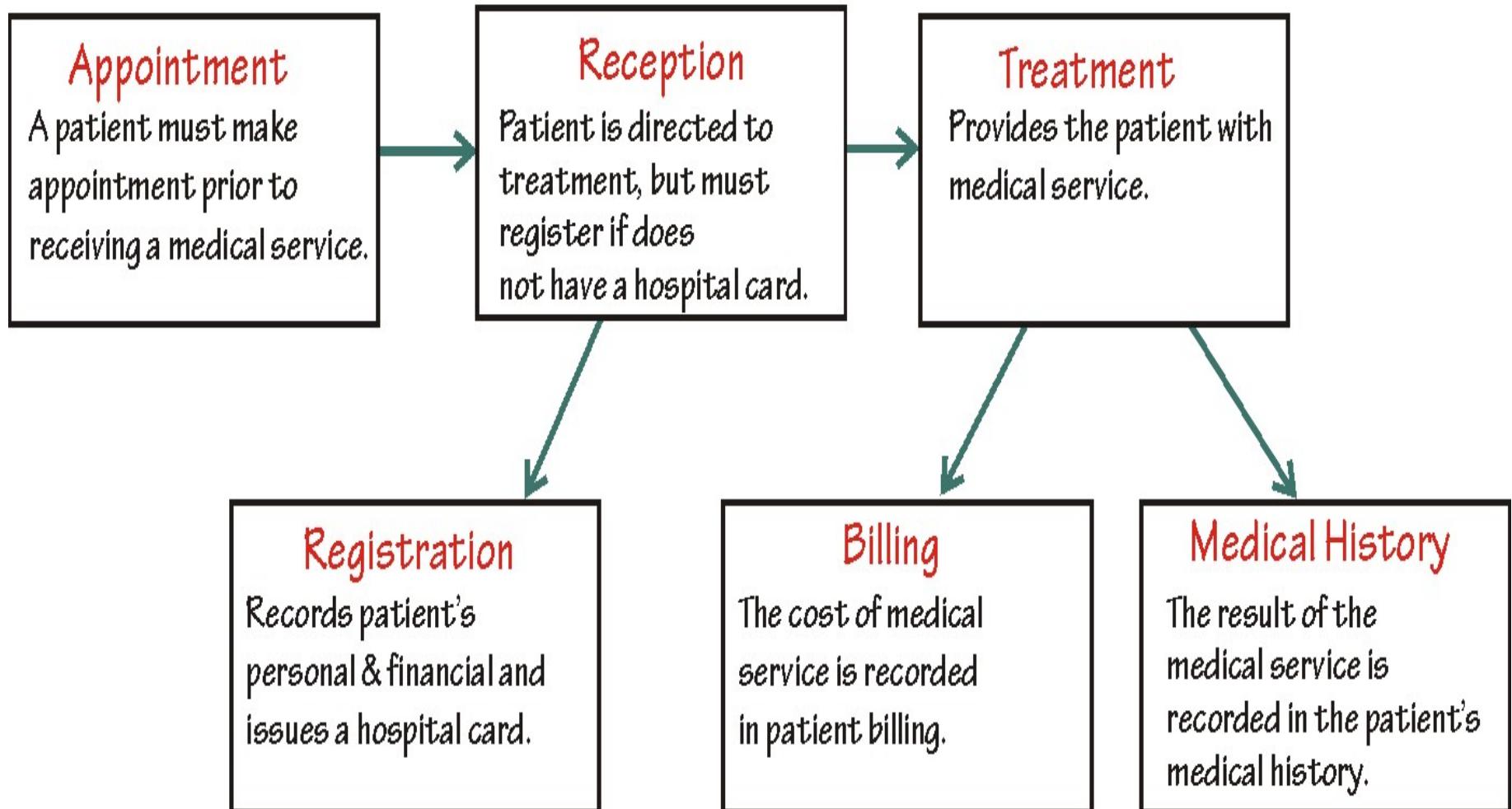
- Usability
- Reliability
- Performance
- Maintainability
- Security

Techniques for Eliciting Requirements

- Interviews
- Questionnaires
- Elicitation Workshops
- Field Trips and Observation
- Modeling
- Mock-Ups

An Owner's View

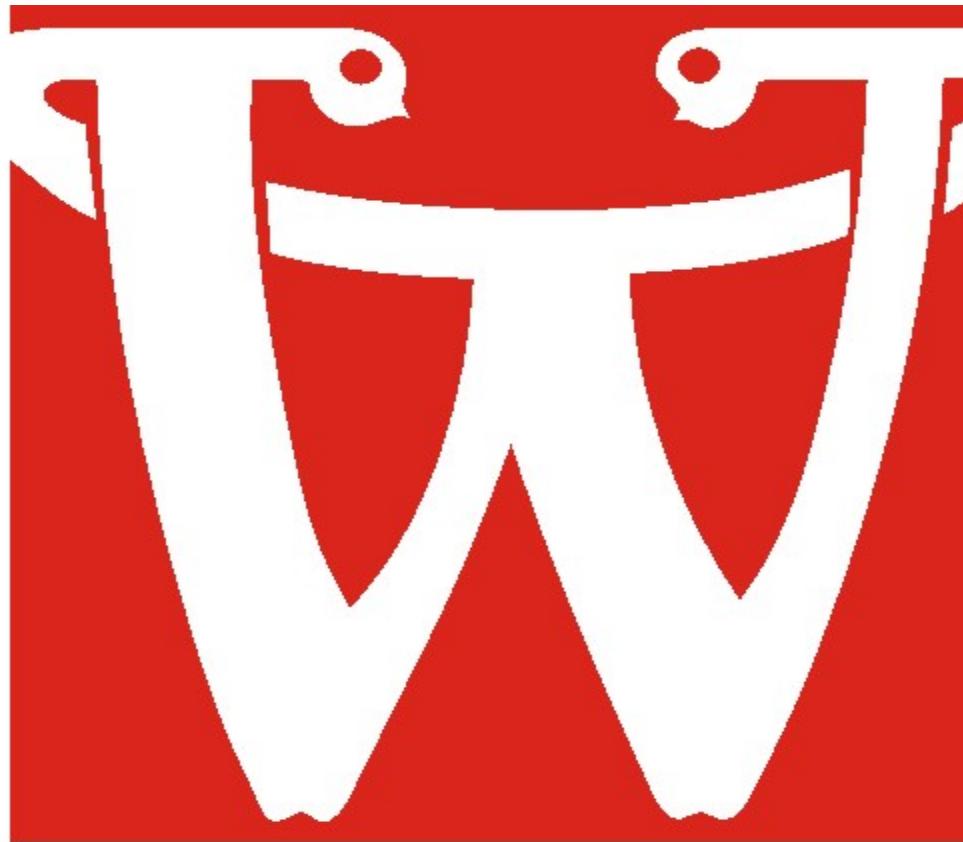
Of Patient Treatment



Managing Requirements

- ① Document and update requirements
- ② Document sources
- ③ Separate requirements into distinct units
- ④ Uniquely identify each requirement
- ⑤ Verify requirements and document verifications
- ⑥ Prioritize requirements
- ⑦ Classify requirements meaningfully

Case History: Walden Medical Center



We will use Walden's case history as our main example.

Initial Requirements

- The consultant concluded that to achieve the goals of the capital-improvement project, an integrated, comprehensive electronic information system is indispensable.

Next: Domain Analysis

- While requirements identify the features of the product, domain analysis places those feature in the context of business domains.
- We will learn about **problem space** and the **solution space**.
- We will also explore the distinctions between requirements, problems, solutions as methods, and solutions as products.

Domain Analysis

Topics

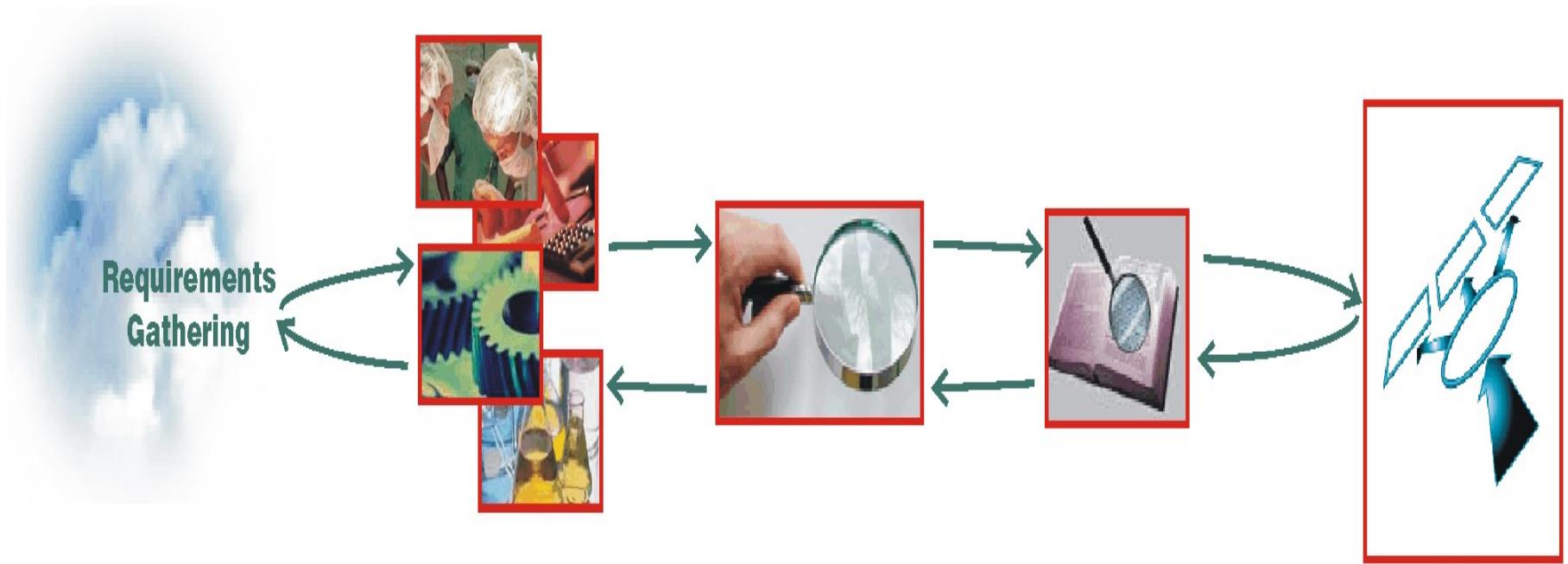
- The three components of problem solving.
- The problem space vs. the solution space.
- Requirements vs. product specifications.
- Domains and their boundaries.
- Identifying domain concepts for analysis and modeling.
- Domain dictionaries and domain catalog.
- Identifying and organizing business rules.

Domain Analysis

- Domain analysis identifies business concepts that will be refined into the building blocks of an analysis model for the information system.

Domain Analysis

Understanding the Context



Requirements Discovery

Defines the scope of requirements gathering.

Domain Definition

Domain is an area of related activities that operates on a set of shared rules and concepts.

Domain Analysis

Defines the context of requirements by discovering domain concepts & rules.

Domain Dictionary

Defines and classifies domain concepts & rules for analysis and conceptual modeling.

Conceptual Modeling

A conceptual realization of the system by analyzing & modeling domain concepts.

Problems, Solutions, and Methods

- Solving problems involves not a pair but a trio of components: problem, solution as method or process, and solution as answer. Each one can be understood only in relation to the whole.

Problems, Solutions, and Requirements

- To build a software system — a product — we must understand the problem. We must also understand **what** is required to solve the problem *before* we can decide **how** to solve it.

Problems, Solutions, and Methods

- Three components of problem solving: the problem that we want to solve, the answer to the problem, and the method to arrive at the answer.

Solution as Two Concepts

Method vs. Answer



Context, Method, and Solution

- Merely knowing the features of the solution is inadequate for building it correctly.
 - We must understand the context to:
 - find the proper method(s), and
 - design a solution that takes the context into account.

Requirements as The Answer

'What' Needs 'Why', 'Where' & 'How'



Problem Space

- Problem space is the context from which the problem arises and in which the solution must operate.

Solution Space

- Solution space defines the territory in which concrete decisions about the information system — as opposed to its features — are made.

Requirements Versus Product Specifications

- Requirements specify the *desired* features of the product or service.
- Product specifications define the product that must realize those features.

Products Are Solutions to Real or Perceived Problems

Product	Requirements
Hammer	Drive in a nail and pull out a nail.
Watch	Tell time and attach to wrist.
Telephone	Enable people to talk to others across vast distances in real time.
Movie	Entertain with sound, music and moving pictures.
Plane	To fly people from location to location.
Rocket	To carry people into space.

Products and Requirements

- Sometimes more than one product or service can satisfy a requirement.
- The choice depends on many factors.

Alternate Solutions

For the Same Requirement

Problem

Shellfish spoils fast



Various Solutions

Can be used separately or combined;
Can be bought, built, or customized



1



2



3



4



5



Requirement

Get the catch to the consumers
before it goes bad

Domain Definition

- A business domain is an area of related activities that operate on a set of shared rules and concepts:
 - Business domains are organized domains.
 - Business domains are goal-oriented.
 - Business domains can change fast.

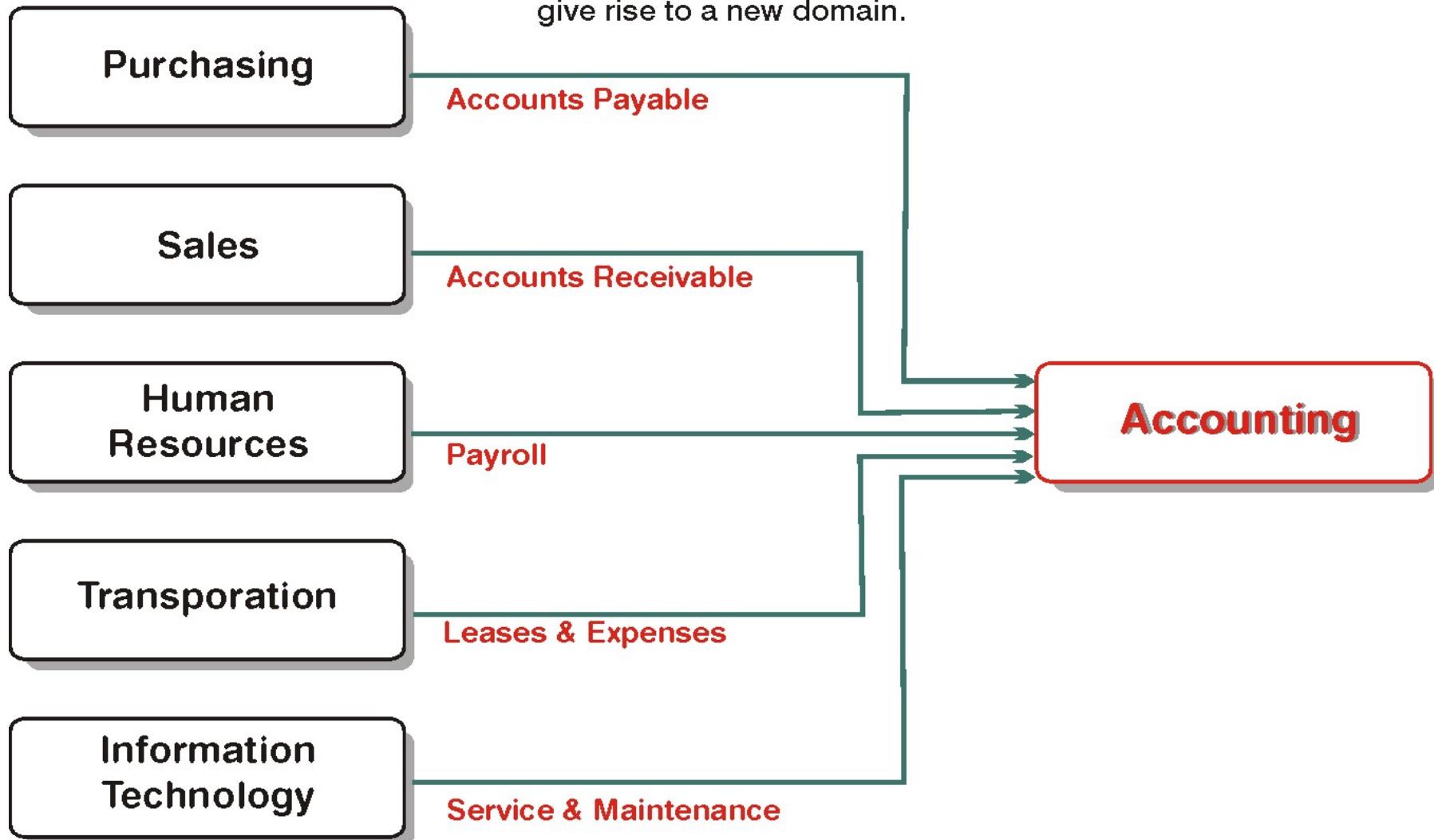
The Domain Scope

- Domain scope defines the boundaries that separate shared activities, rules and concepts within a domain from those on the outside.

Derivative Domains

The Case of Accounting

Common activities, concepts and rules across several domains may give rise to a new domain.



Domains and Subsystems

- Domain definition provides a framework for the conceptual subsystems within the information system.

Walden Medical Center:

Domain Definitions

(Partial Listing)

Domain	Scope Outline
Patient Management:	All activities that directly come into contact with patients fall within this domain, including: <ul style="list-style-type: none">▪ Referrals▪ Scheduling▪ Registration, Admissions▪ Treatments▪ Patient Billing Issues
Drug Inventory & Purchasing:	<ul style="list-style-type: none">▪ Pharmaceutical Inventory▪ Drug Supply Chain
...	...
Medical & Lab Technology:	<ul style="list-style-type: none">▪ Medical Equipment Purchasing▪ Medical Equipment Inventory▪ Medical Equipment Maintenance
House Services:	<ul style="list-style-type: none">▪ Laundry▪ Cleaning▪ Food Preparation & Diets
...	...

The Scope of Walden's Patient Management

- Referral
- Appointment
- Registration
- Medical Service
- Hospitalization
- Cost & Record Keeping
- Discharge
- Patient Billing

Domain Analysis

- Domain analysis is analyzing the context of requirements. It has a two-fold task:
 - define the context in which the information system will operate.
 - discover and define concepts that the product must incorporate or take into account in order to meet its objectives.

Finding Domain Concepts

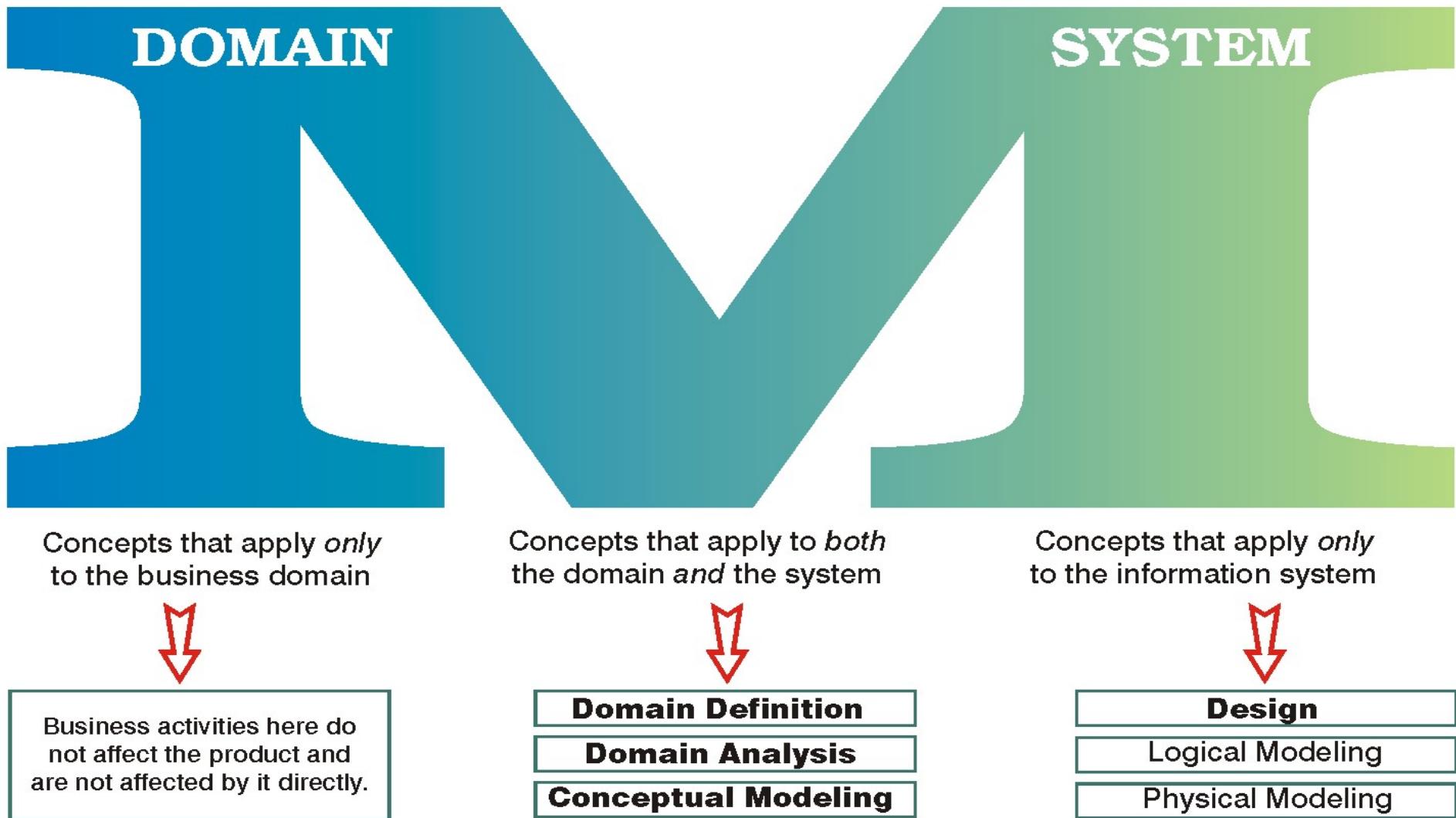
- Domain concepts are objects, processes, people and rules that constitute the goals, the behavior and the structure of a domain.

Finding Domain Concepts

- ① Discover the essence of the requirements.
- ② Discover problems that the requirements are supposed to solve.
- ③ Discover the components of the problem.
- ④ Discover related domain concepts.

'M' for Model

The Passage from The Problem Domain to the Solution Domain



Domain Dictionary

- Domain dictionary organizes and brands domain concepts. It is the link between stakeholders who must verify the concepts and the analysts who would use them as the foundation for building a conceptual model of the system.

Domain Dictionary

- The dictionary organizes domain concepts (the **problem space**) pertinent to the system (the **solution space**).
- To populate the dictionary, we turn to all the products of information gathering: requirements, interviews, manuals.

Domain Dictionary

- The most promising concepts are:
 - Subjects
 - “The noun, noun phrase, or pronoun in a sentence or clause that denotes the **doer of the action**.”
 - Objects
 - “**a.** A noun or substantive that receives or is affected by the action of a verb within a sentence. **b.** A noun or substantive following and governed by a preposition.”
 - These nouns are candidates for becoming objects in an **object-oriented** sense.
 - Verbs
 - They can indicate processes, but they can also hide nouns, or grammatical objects: “ordering a book” is a variant of “placing *an order* for book.” The object “order” is hidden in the verb “order.”

Domain Dictionary

An Example

- **Medical Service.** Depending on the nature of the medical service, doctors, nurses and lab technicians provide the patient with appropriate service (s) for which the appointment has been made.

The Dictionary Template

- Name
- Type
- Description
- Source
- Notes

The Dictionary Template

- **Type**

- Process
 - A series of actions, changes, or functions bringing about a result.
 - A series of operations performed in the making or treatment of a product.
- Function
 - The purpose or the result of one action or a set of actions.
- Role
 - A grouping of any entity.
- Object
 - Something perceptible by one or more of the senses, especially by vision or touch.
 - Something intelligible or perceptible by the mind. Or we can use Entity instead.
- Business Rule
 - To be discussed in detail later.
- Formula.
 - A statement, especially an equation, of a fact, rule, principle, or other logical relation.
 - A method of doing or treating something that relies on an established model or approach.
- Identifier
 - A symbol that identifies an object.

Patient Management:

Domain Dictionary

Name	Type	Description
Appointment	Process	Scheduling of a <u>patient</u> to receive <u>medical service(s)</u> . Performed by the <u>appointment clerk</u> .
Appointment	Object	The scheduled date and time for providing a <u>medical service</u> to a <u>patient</u> .
Appointment Clerk	Role	Makes <u>appointments</u> for the <u>patient</u> .
Medical Service	Object	Any service of medical nature provided by <u>medical staff</u> to a <u>patient</u> : diagnosis, prescription, administration of drugs, lab tests, etc.
Medical Service	Function	The act of providing a <u>medical service</u> to the <u>patient</u> by <u>medical staff</u> .
...
Referral Source	Role	A <u>primary care physician</u> , an <u>emergency medical worker</u> or an <u>outside hospital</u> that refers a <u>patient</u> for an <u>appointment</u> to receive a <u>medical service</u> . <u>Patient</u> himself or herself can be a referral source.
Registration	Process	Carried out before a set of <u>medical services</u> are performed. The process gathers new or changed personal and insurance information for a new or an existing <u>patient</u> . A hospital <u>ID card</u> may be issued as part of this process. Performed by the <u>registration clerk</u> .
Registration Clerk	Role	Performs <u>registration</u> .

Business Rules

- Business rules are a set of detailed policies, laws, procedures, guidelines and standards under which an enterprise operates. A business rule is a statement that defines or demands adherence to a unit in the set.

Rules Dictionary

- Requirements specify features of the product, while business rules apply beyond any single solution.

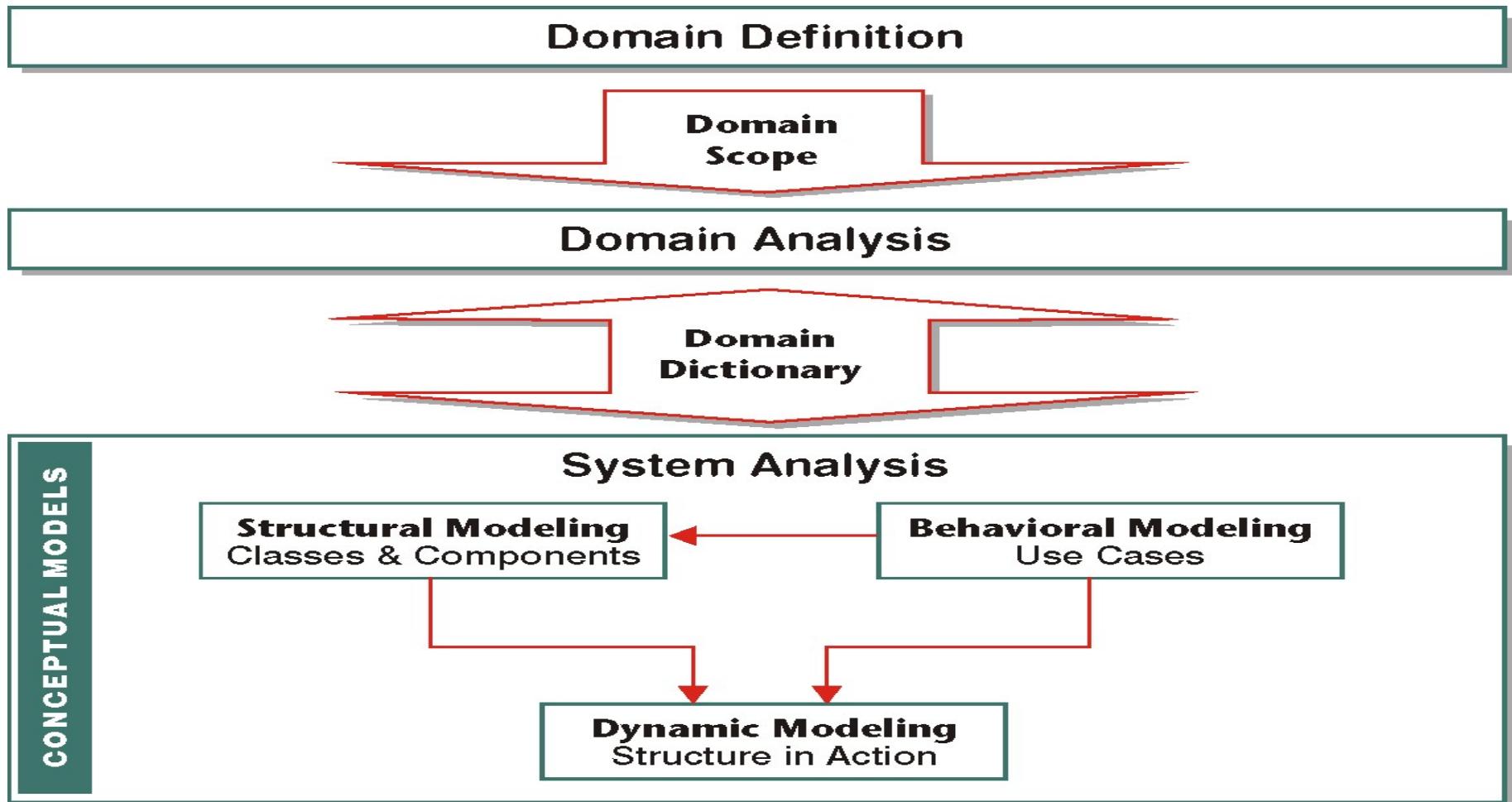
Patient Management:

Business Rules Questionnaire

ID	Definition	True / False	Comment
001	A patient is anybody who suffers from a medical condition and is referred to the hospital by a referral source — a doctor, a medical emergency worker or another hospital.	<input type="checkbox"/> <input type="checkbox"/>	
	...		
003	A patient less than 18-years old must be accompanied by a related adult or an emergency medical worker.	<input type="checkbox"/> <input type="checkbox"/>	
	...		
006	If the patient bill is not paid within 30 days, then the bill is considered overdue.	<input type="checkbox"/> <input type="checkbox"/>	

Next: Conceptual Modeling

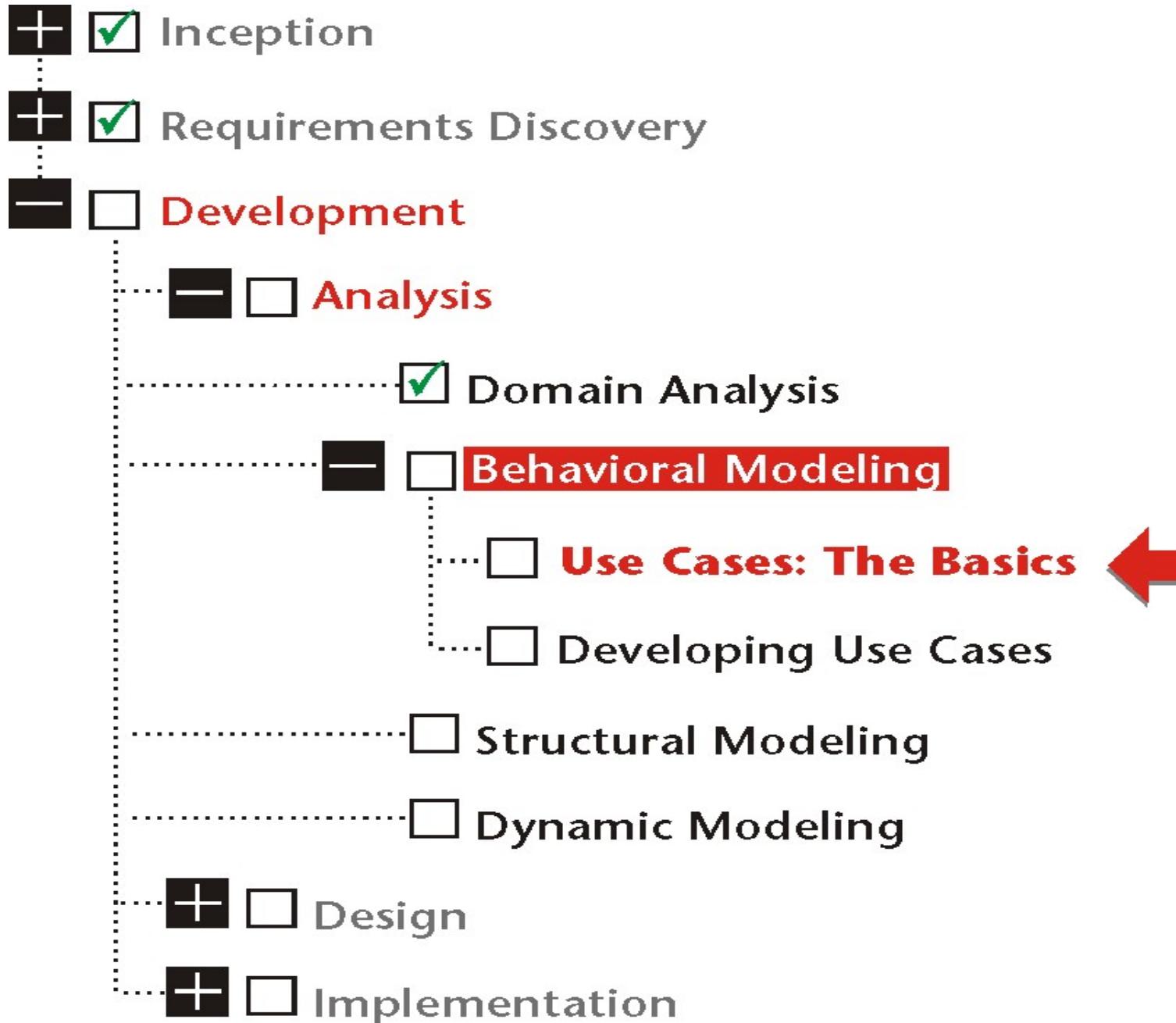
To Conceptual Modeling From Domain Modeling



Behavioral Modeling I: Use Cases: The Basics

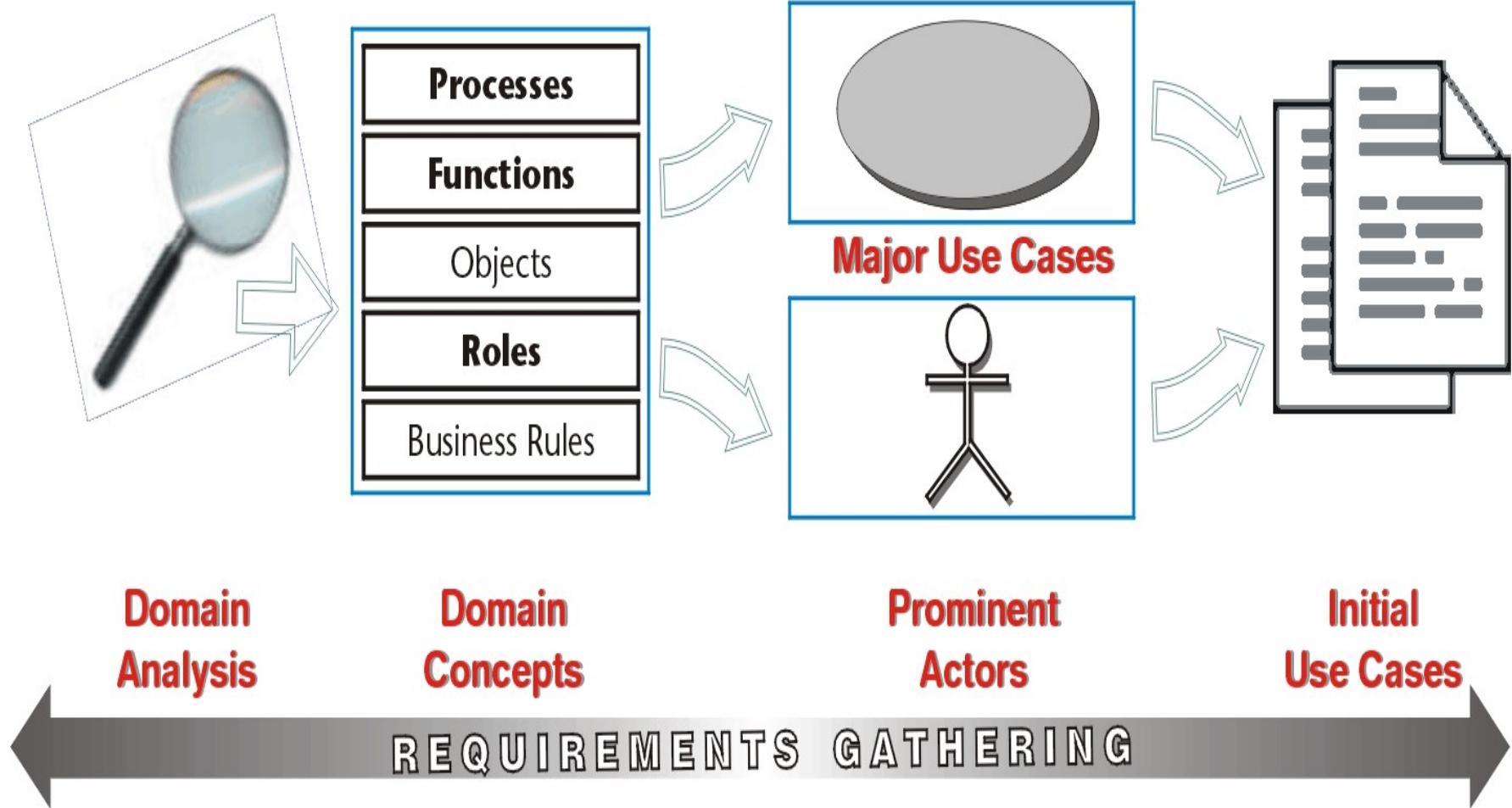
Topics

- What use case modeling is and is not.
- The four components of a use case.
- The basic elements of use case diagram.
- How to transform concepts from domain analysis into use cases.
- Identifying prominent actors.
- Identifying major use cases.
- The context diagram.



Discovery of Use Cases

Transforming Domain Concepts Into Behavioral Models



Use Case Modeling

- Use cases model the behavior of a system
 - A use case is a unit of system behavior
 - A use case is a contract that formalizes the interaction between stakeholders and the system
 - A use case details the interaction of an actor with a system to accomplish a goal of value to the actor
 - Use cases are technology-independent

What Use Case Modeling Is Not

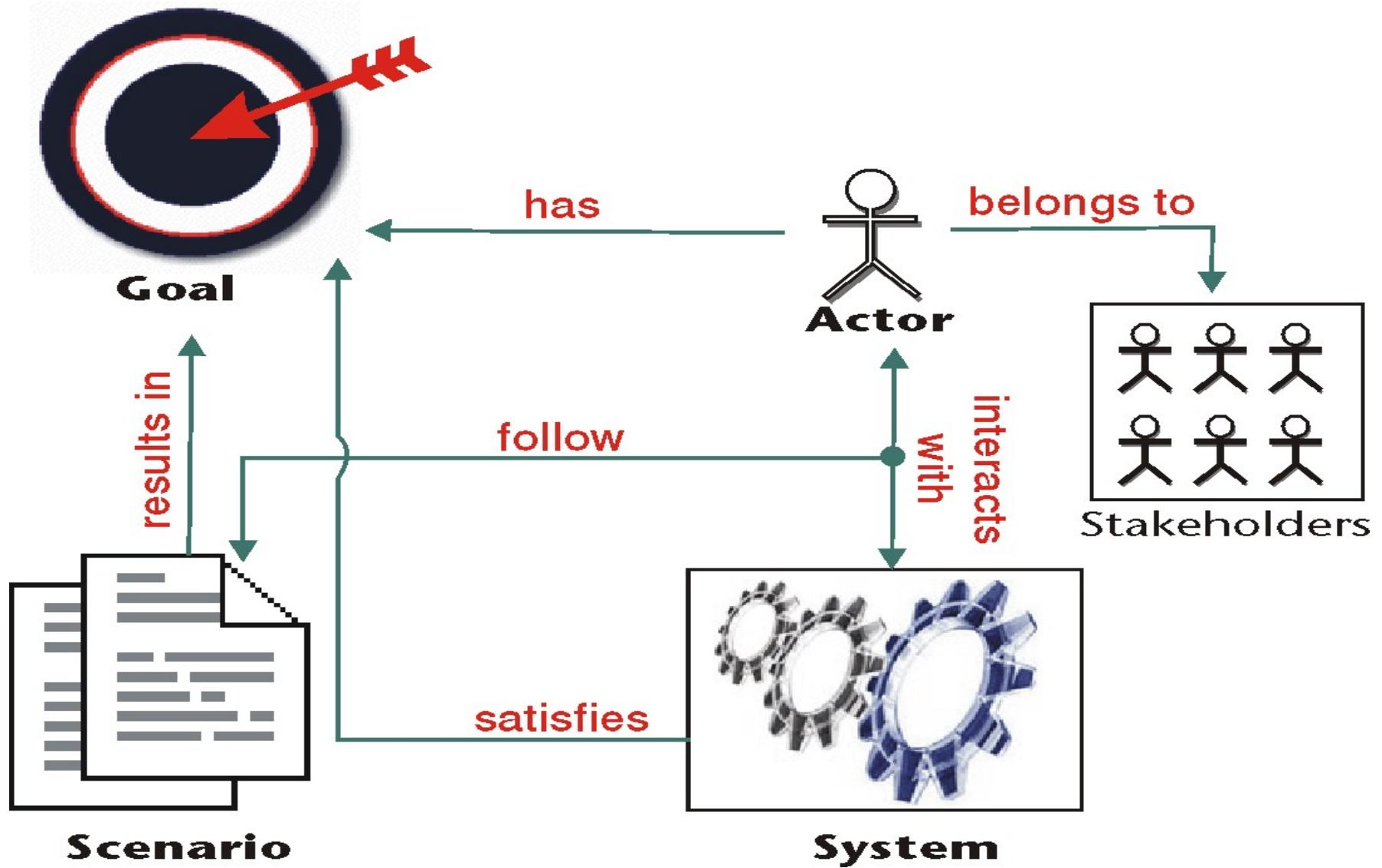
- Use Case modeling is limited to a system's external behavior
 - Use cases do not model the system from *inside*.
 - Use cases are not effective in capturing the non-functional requirements.
 - Use case modeling is *not* the same as functional decomposition.
 - Use cases are not inherently object-oriented.
 - Use cases describe **what** a system accomplishes, not **how**.

Components of Use Cases

- A use case has four components:
 - A goal,
 - Stakeholders (Actors),
 - A system, and
 - A scenario.

Components of a Use Case

Actor(s), System, Goal, Scenario



Goal

- A use case is successful only if its stated goal is completely achieved
- A use case's name is its goal. The name must be active, concise and decisive.
- It is the goal that decides the relevance of activities in a use case.

Stakeholders and Actors

- Stakeholders are those entities whose interests are affected by the success or the failure of the use case.
- An actor is an entity outside the system that interacts with the system to achieve a specific goal.
- A use case must enforce the interests of all stakeholder.

Circle of Stakeholders



Actor

- Actor is a role that any user who has been given the part can play.
- The goal of the **primary actor** is specified by the name of the use case.
- **Supporting** (or secondary) actors support the primary actor in reaching the goal of the use case.
- An actor is identified by a unique name which describes a unique role.

A System

- The system defines the boundaries of a use case
- Two types of systems
 - Real system
 - Grocery store “bricks-and-mortar”
 - Information system
 - Point of Sales System (POS)
- A use case cannot leave a system, but can reach across its boundaries

One Supermarket, Two Systems

The System Is the Boundary of the Use Case



The Real System



The Information System

Purchase Groceries

1. Customer enters the supermarket.
2. Customer picks a shopping cart.
3. Customer selects groceries.
4. Customer brings groceries to cash register.
5. Customer pays for groceries.
6. Customer leaves with groceries.

Checkout Groceries

1. Cust. deposits groceries on checkout counter.
2. Cashier scans items one by one.
3. Cashier calculates the total.
4. Cashier informs customer of total.
5. Cashier receives payment from customer.
6. Cashiers gives receipt to customer.

Purchase Groceries — The Real System Scenario

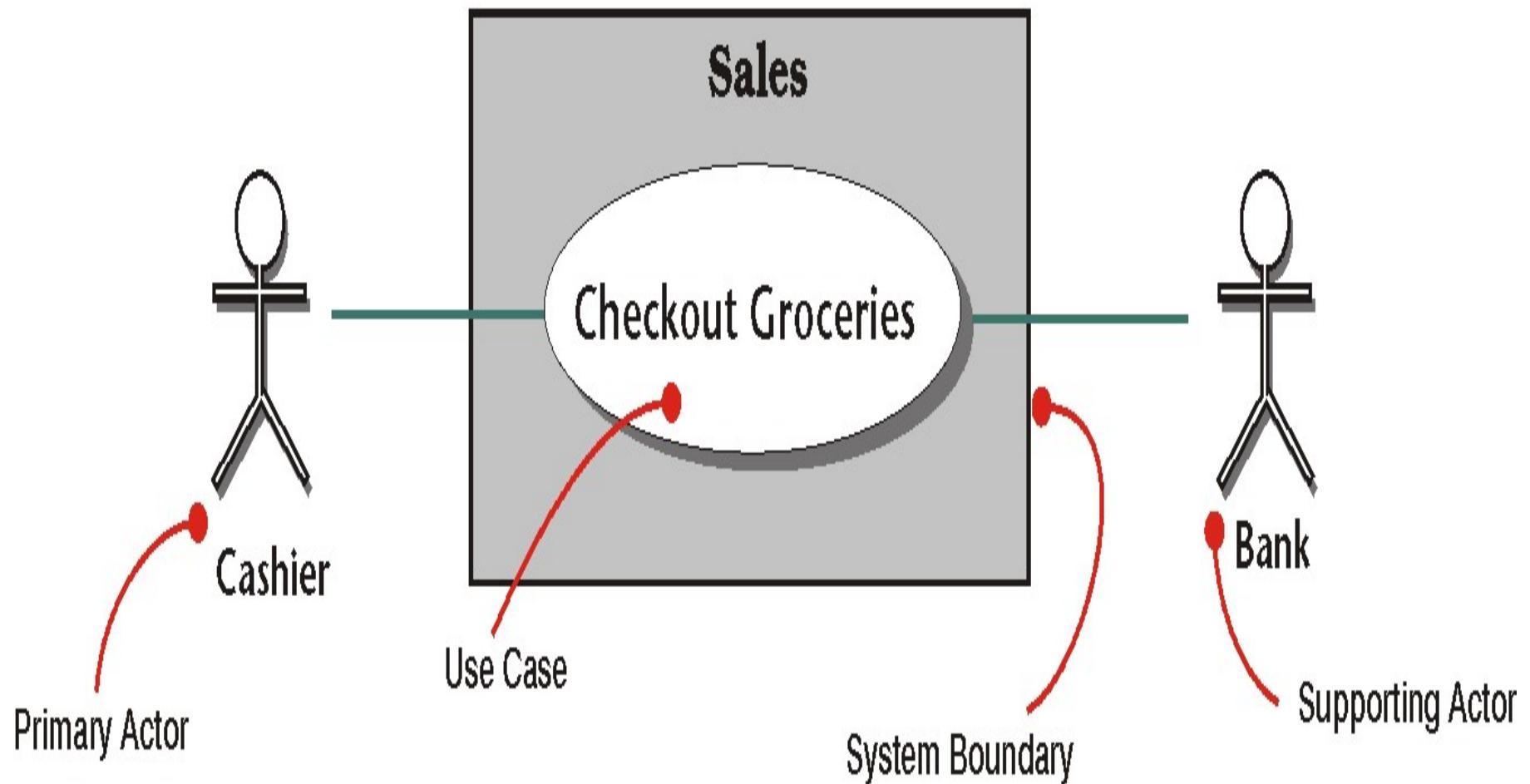
- A customer enters the supermarket. The customer takes a shopping cart or basket and strolls through the supermarket. The customer selects items from the shelves and puts them in the shopping cart or the basket. When finished, the customer brings the items to the cash register. The cashier calculates the total price of the merchandise. The customer pays for the merchandise. The cashier bags the items, issues a receipt to the customer and, if necessary, returns the change. The customer picks up the bags and leaves the supermarket.

Purchase Groceries – The Point-of-Sale System

- The customer deposits groceries on the checkout counter. The cashier scans each item and deposits the item on the bagging counter. When the last item is scanned, the cashier reads the total amount from the system and announces it to the customer. If the customer pays by credit card, the cashier swipes the card through the cash register to charge the amount. The customer then signs the printout. If the customer pays by cash, the cashier returns the change, if any. The cashier then gives a receipt to the customer.

Use Case Diagram

The Interaction Between the Actors & the System



The Basic Elements Of the Use Case Diagram

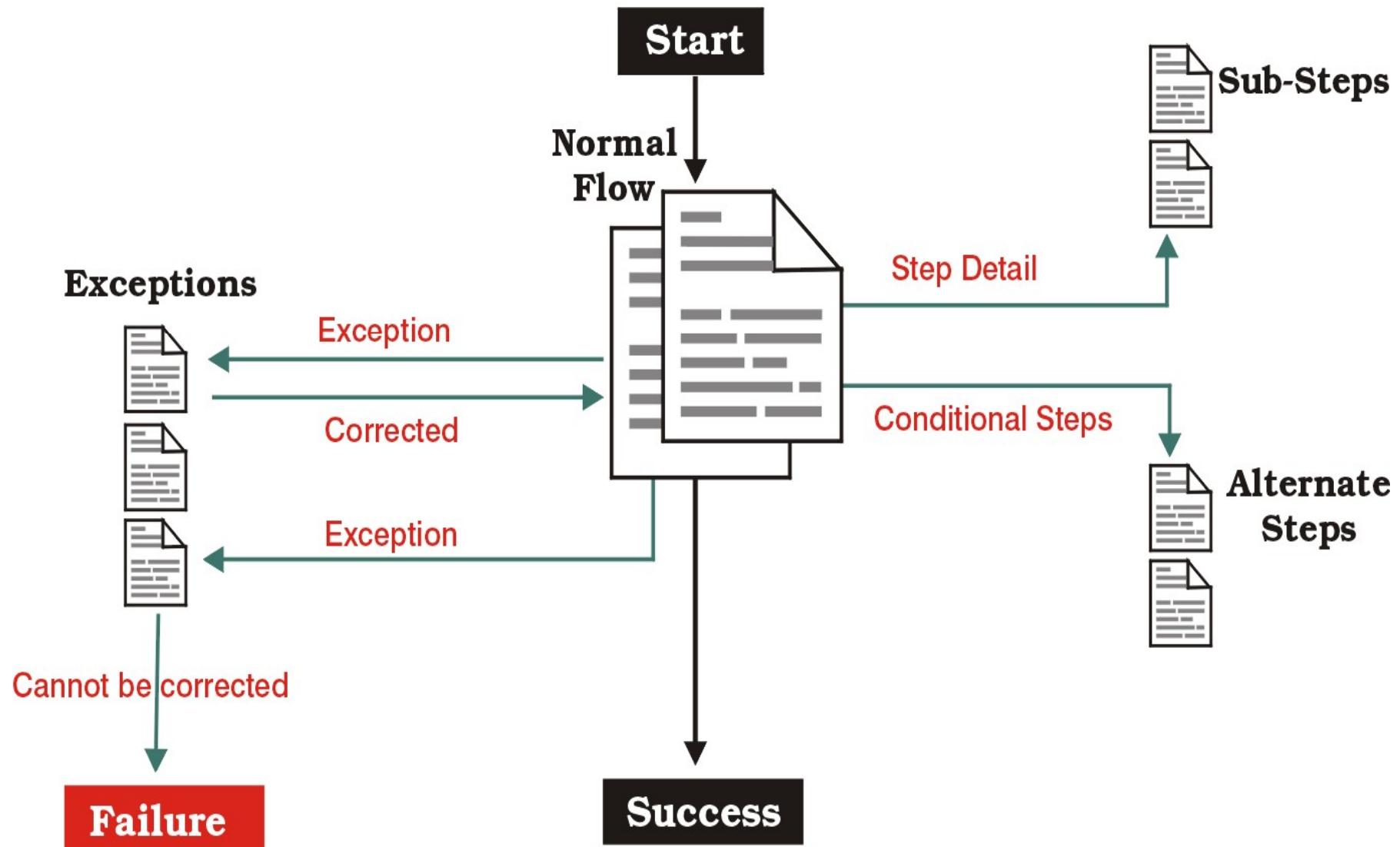
 Actor	<p>The stick figure identifies an actor. More than one actor can be associated with a use case, and one actor can be associated with more than use case.</p>
 System Boundary	<p>The rectangle is the system boundary or the scope. For the use case, any entity outside this boundary can exist only as an actor.</p>
 Use Case	<p>The ellipse (the oval) represents the use case. It resides inside the system boundary. In the diagram,</p>
 Association	<p>The simple line on the left represents association. It shows the communication between an actor and a use case.</p>

A Scenario

- The scenario is an *ordered* sequence of interactions between the actor(s) and the system to accomplish a goal. It consists of:
 - Normal Flow
 - Alternate Flow
 - Sub-Flows
 - Exceptions

Use Case Flows

Classifying the Steps



Steps in a Use Case Scenario

- Steps can be repeated
 - One step or a set of step can be repeated until a certain condition is met. In **Checkout Groceries**, the cashier scans purchase items until there are no more groceries are left to scan.
- A step can call on another use case
 - Each step may call on another use case to complete its function.
- A step is a transaction
 - Each step appears as just an interaction, but it is really a transaction between the actor and the system. That means that in each step:
 - the actor sends a request to the system,
 - the system validates the request,
 - the system changes its state as a result of validation, and then
 - the system responds.

Use Cases in the Modeling Spectrum

- Use cases reside near the **dynamic** edge of the modeling spectrum.

Develop Initial Use Cases

- Components of use case modeling are provided by analyzing and expanding concepts that result from domain analysis.

Walden Hospital: Milestones Achieved

- To illustrate how domain concepts are transformed into use cases, we need concrete examples.
- Let us, then, recap the milestones that our main case history, the Walden project, has achieved up to this point.

Walden Hospital: Milestones Achieved

- **Business Analysis.**
 - The business analyst conducted a broad-based study of Walden Medical Center's business.
- **Problem Definition.**
 - the business analyst identified and scoped the problems that the hospital must solve to save its sagging business.
- **Propose Solutions.**
 - The analyst proposed a capital project for improving all aspects of Walden's operations and infrastructure.
- **Project Initiation.**
 - The hospital charged its newly hired CIO with the task of planning an IS strategy for the medical center.
- **Domain Definition.**
 - Business domains that need the services of an information system are: Patient Management, Medical Records Management, Legal, Drug Inventory & Purchasing, Transportation, Accounting, and many more.
- **Domain Scoping.**
 - The hospital decided that the Patient Management domain must have the highest priority.
- **Domain Analysis & Domain Dictionary.**
 - Within the scope of Patient Management, business concepts were explored, defined, and organized into a preliminary domain dictionary.

Identify Prominent Actors

- The primary candidates for becoming actors are domain concepts classified as “role.”
- Discovering actors is a process of consecutive abstraction.

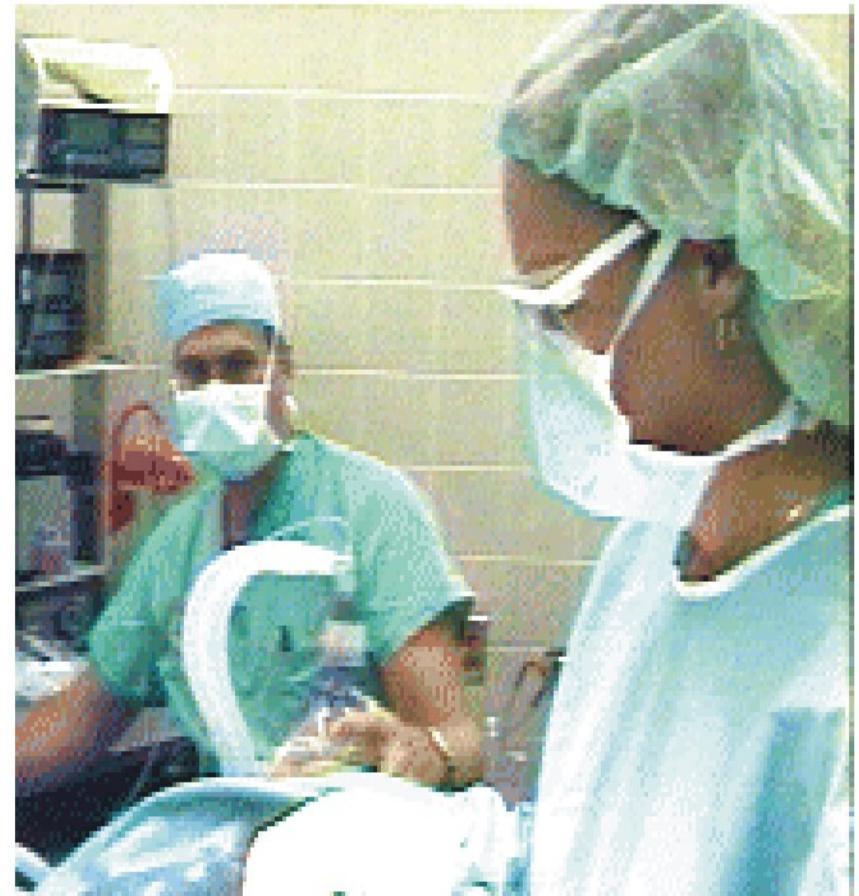
Finding Actors

Does the entity interact with the information system?



Patients

No



Doctors & Nurses

Yes

Patient Management: Domain Dictionary

Name	Description
Appointment Clerk	Makes appointments for the patient.
Billing Clerk	Produces individual patient bills on request; records payments; resolves billing issues.
Doctor	Provides a specialized level of medical services to the patient: diagnosis, procedures and operations, prescriptions and monitoring of medical conditions.
Emergency Medical Worker	Refers the patient to the emergency room. Performs emergency medical services before emergency room.
Lab Technician	Performs a test medical service: X-ray, blood test, MRI, etc.
Medical Staff	Any person who provides a medical service to a patient: a doctor, a nurse, a lab technician, or an emergency medical worker.
Nurse	Helps the doctor in providing medical services. Administers drugs and monitors the patient.
Outside Hospital	Refers patient for an appointment and medical service.
Primary Care Physician	Refers the patient to the hospital to receive medical services.
Referral Source	A primary care physician, an emergency medical worker or an outside hospital that refers a patient for an appointment to receive a medical service. Patient himself or herself can be a referral source.
Registration Clerk	Performs registration.

Identify Major Use Cases

- Major use cases are identified by analyzing business processes and functions

Walden's Major Use Cases

Patient Management: Use Case Summary

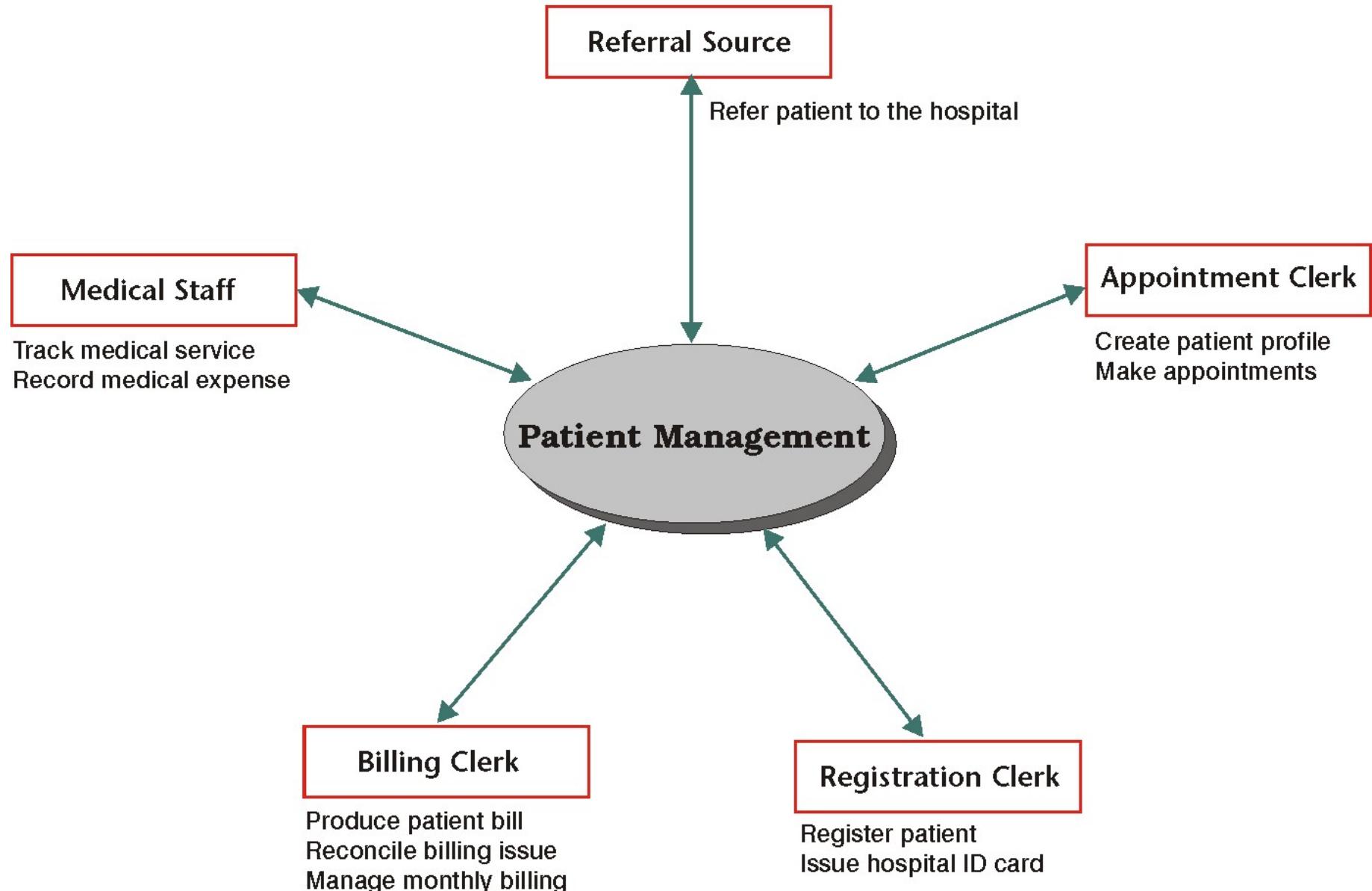
ID	Name	Description	Actors
100	Refer Patient	A referral source refers the patient to the hospital for an appointment to receive a medical service.	Primary Care Physician, Emergency Medical Worker, Another hospital, Patient.
120	Make Appointment	On referral, the appointment clerk schedules a medical service for the patient.	Appointment Clerk
140	Register Patient	Before a medical service, the registration clerk updates personal and insurance information if the patient is new or the relevant information is changed. A hospital ID card is issued if the patient is new or has lost the card.	Registration Clerk
160	Track Medical Service	Hospital renders a medical service to a patient. A medical service covers all activities performed by the medical staff that relate to a patient, from a visit to a doctor to a lab test to hospitalization and discharge. Medical staff records each service along its cost.	Doctor, Nurse, Lab Technician, Emergency Medical Worker
180	Manage Patient Billing	On request, the billing clerk produces a bill for the patient. The clerk also reconciles the patient's account and accepts payments.	Billing Clerk

Develop Context Diagram

- Context diagram represents the interaction of outside entities with a system as a whole.
- Context diagram is composed of three elements:
 - A **system** or subsystem.
 - Entities **outside** the system that interact with it.
 - **Interactions** between outside entities and the system.

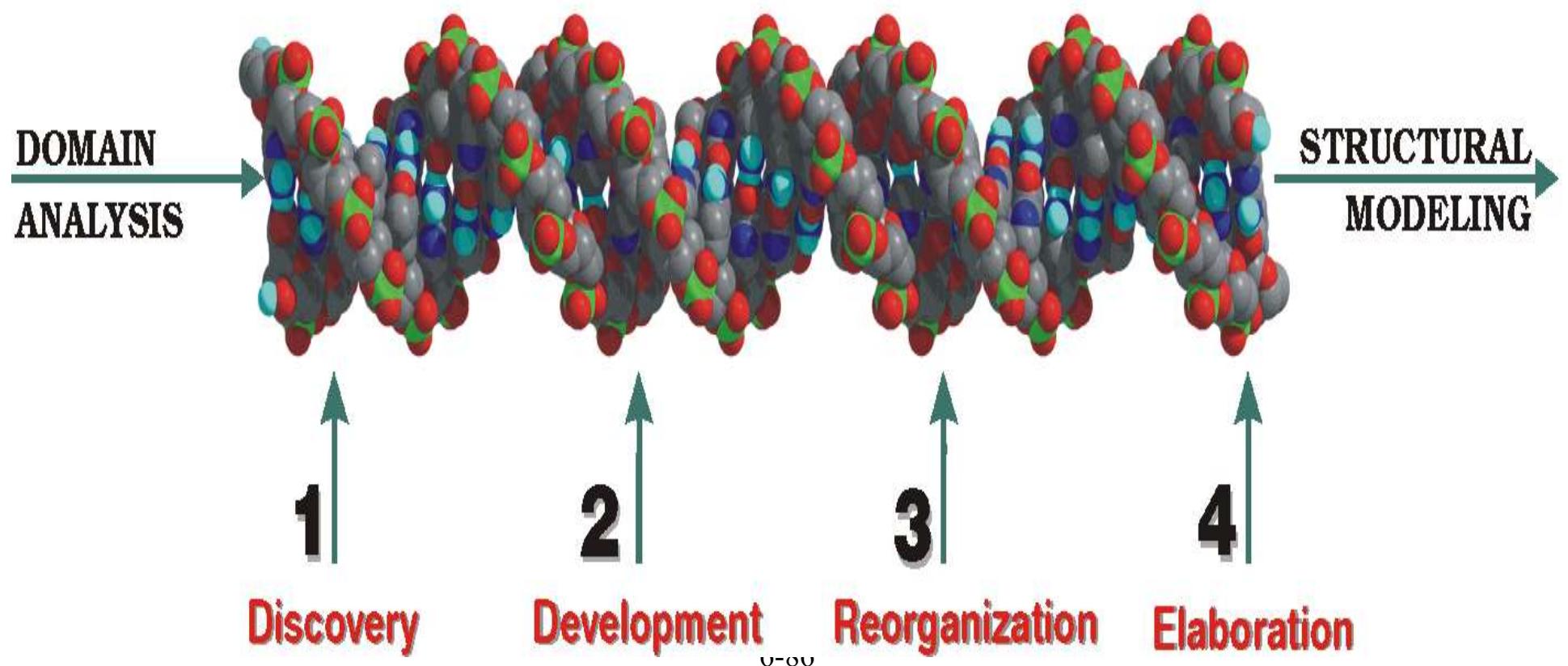
Context Diagram

For Walden's 'Patient Management'



Next: Developing Use Cases

Evolution of Use Cases From Discovery to Elaboration

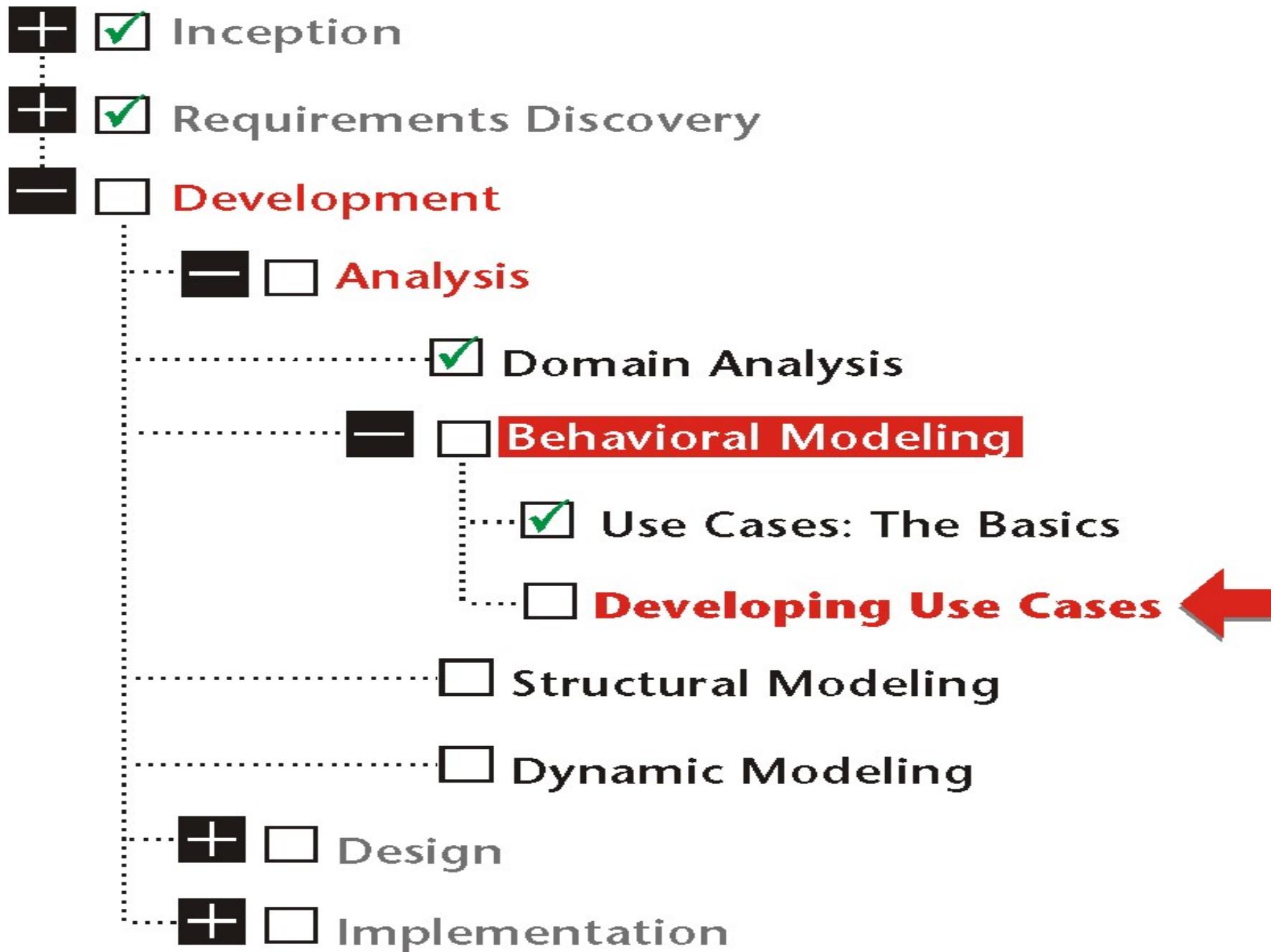


Behavioral Modeling II

Developing Use Cases

Topics

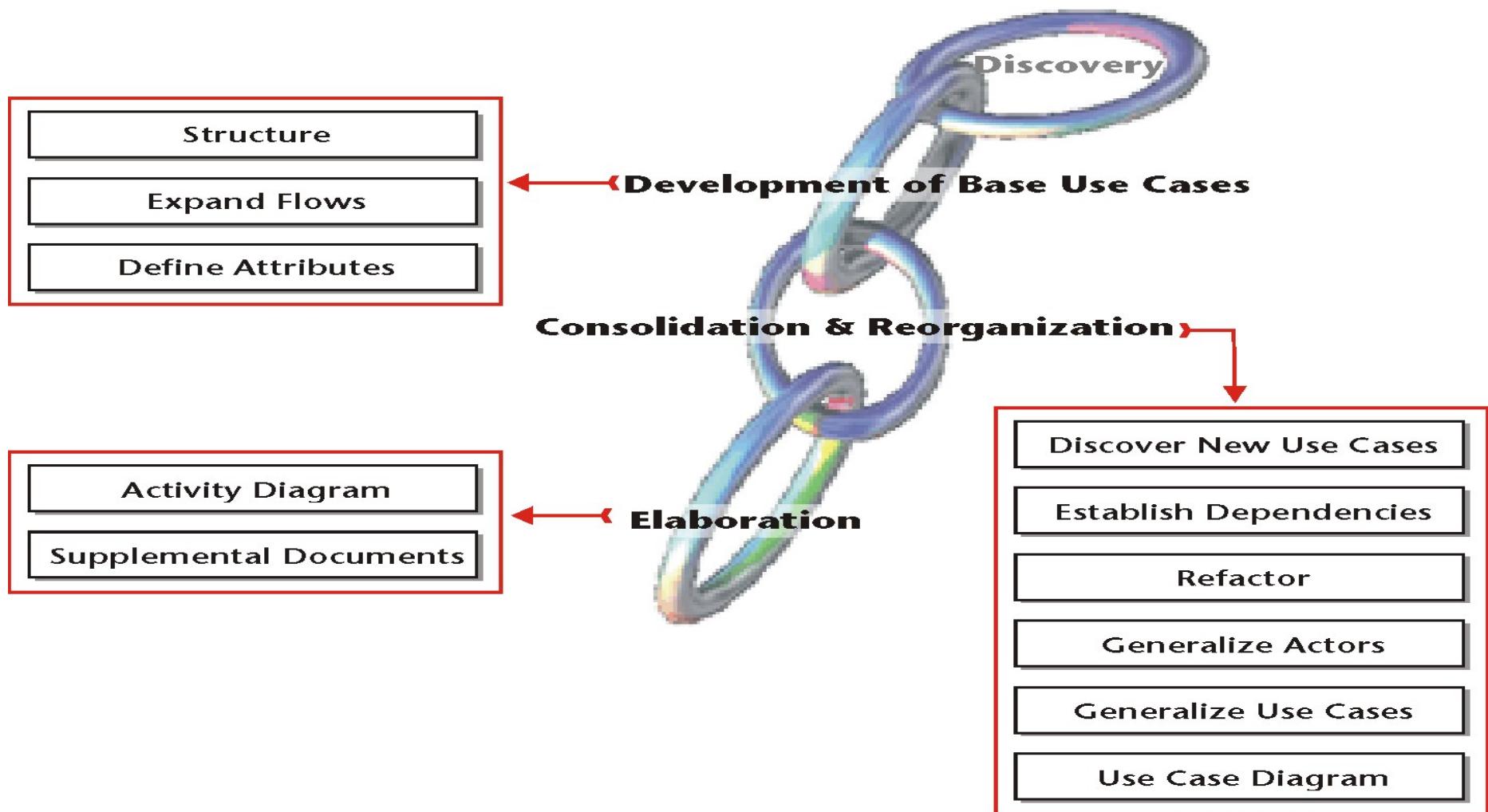
- Structuring and developing use cases through templates.
- When and how to generalize actors.
- When and how to extend the functionality of a use case.
- When and how to reuse use cases.
- When and how to generalize use cases.
- The features and the purpose of use case diagram.
- When and how to join or divide use case.
- Using activity diagram to clarify the logical flow of use cases.
- Use case modeling as a framework for development activities.
- Managing details by creating supplements to use cases.



A Framework for the Development

Use-Case Modeling

Links in a Chain



Develop Base Use Cases

- What a “base” use case is?
 - A base use case is a fully formed, structured use case which serves as a *base* to develop other analysis and design artifacts.

The Template

- The template structures use cases by providing well-defined and ordered fields.

Template Fields

- Template fields represent the building blocks of the use cases, joined in a predefined, orderly manner.

Template Fields

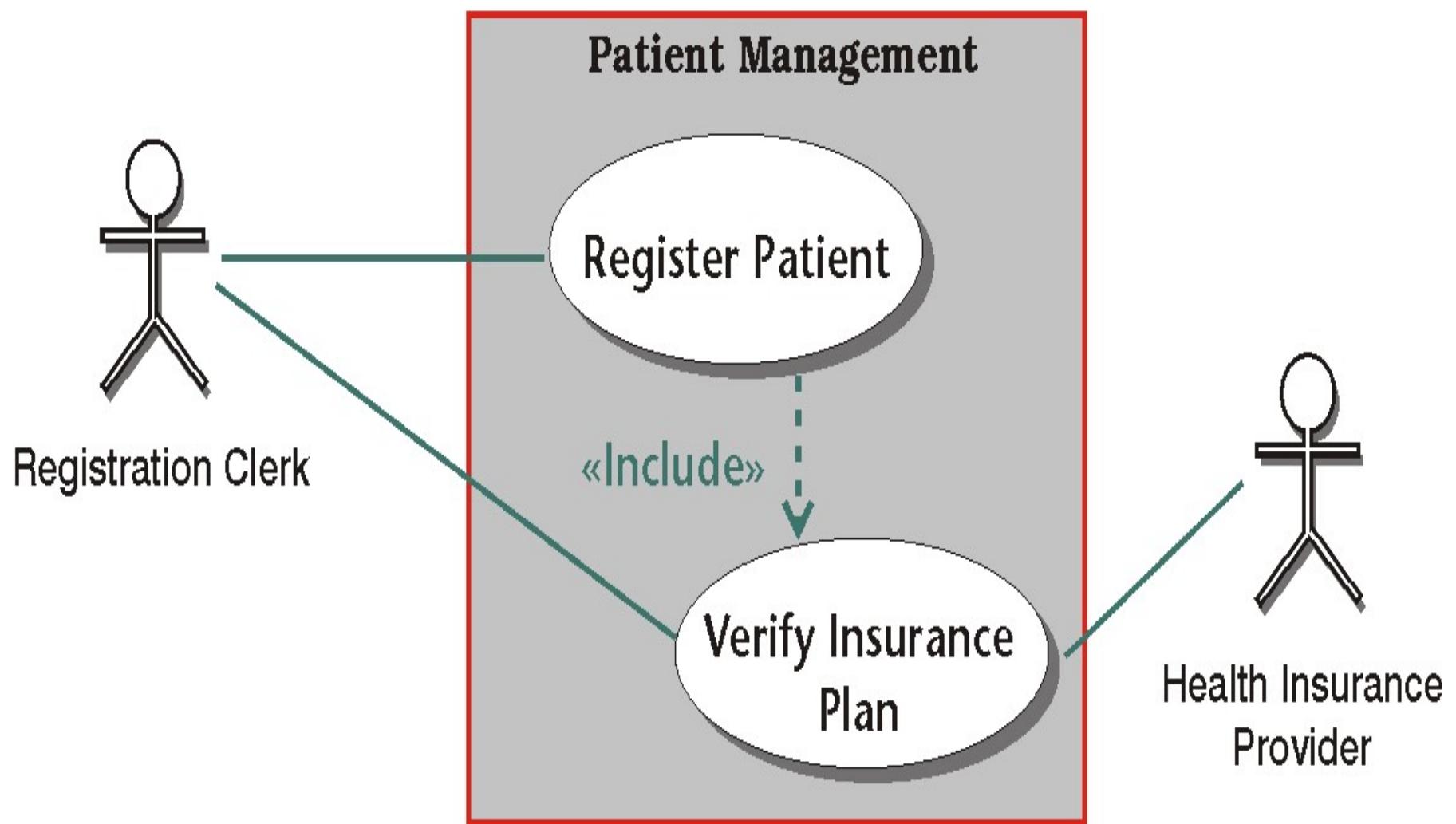
- Name
 - embodies the goal that the use case wants to accomplish.
- ID
 - is *unique* numeric identifier for the use case.
- Scope
 - boundaries of the use case— defined by the system or the subsystem to which it belongs.
- Priority
 - decides the order of design and implementation for use cases.

Template Fields

- Summary
 - a long version of the use case name and a short version of the scenario.
- Primary actor
 - is the actor whose goal identifies and drives the use case.
- Supporting actor
 - assist the primary actor in achieving the goal of the use case.

The Supporting Actor

Helping the Primary Actor to Reach the Goal



Template Fields

- Stakeholder
 - any entity, human or otherwise, who has an interest in the outcome of the use case.
- Precondition
 - defines the state of the system before a use case can start; post-condition defines the state of the system after a use case is complete.
- Trigger
 - the event that starts the use.
- A flow
 - an ordered set of activities that occur as the actors and the system attempt to reach a goal.

Normal Flow

- Normal flow is the best-case scenario

	Normal Flow:	<ol style="list-style-type: none">1. Customer inserts the bank card.2. Customers enters password.3. System verifies password.4. System presents a list of transaction types that the customer may conduct.5. Customer selects a type of transaction.
--	---------------------	--

Sub-Flows

- Sub-flows identify the details of the steps in the normal flow

	Normal Flow:	<ol style="list-style-type: none">1. The registration clerk enters or updates personal data.
	Sub Flows:	<ol style="list-style-type: none">1.1 The registration clerk enters the Social Security Number of the new patient.1.2 The registration clerk enters or updates patient's address.1.3 The registration clerk enters or updates patient's phone number.1.4 The registration clerk enters or updates the name, the address and the phone number of the patient's closest relative.

Alternate Flow and Exceptions

- Alternate steps identify remedies; exceptions signify failure

Alternate Flow/ Exceptions:	<p>3.a Patient is new. Reception clerk directs the patient to registration...</p> <p>3.bPatient is not new but personal or insurance data has changed. Reception clerk directs the patient to registration...</p> <p>3.cPatient has lost the hospital ID card. Reception clerk directs the patient to registration...</p>

Non-Behavioral Requirements

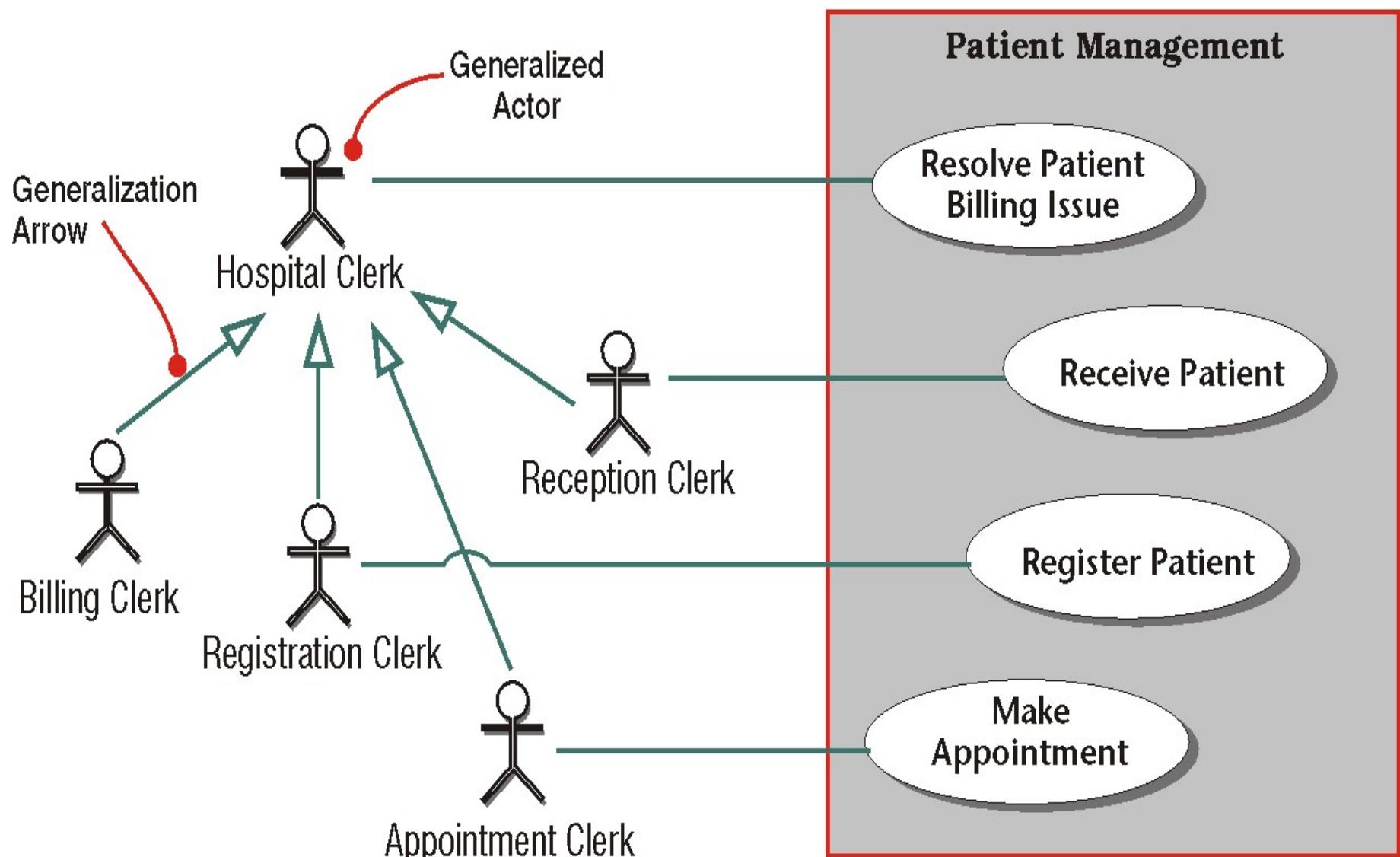
- Only when a non-behavioral requirements applies to a specific use case, the requirement is specified in the template.

Template Fields

- Open Issues
 - questions that must be resolved before the use case can be judged as complete.
- Audit fields
 - help us to keep track of the evolution of the use case.
- Custom Fields
 - specifies an attribute or requirement that is specific to one use case or a set of use cases within the system.

Actor Generalization

Creating a 'Super-Role'



Actor Dictionary

Actor	Description	Abstract	Use Case (s)
Appointment Clerk	Makes appointment for the patient to receive medical service.		Make Appointment
Billing Clerk	Maintains patient billing.		Enter Bulk Payment
Hospital Clerk	Generalizes: <ul style="list-style-type: none">▪ Appointment Clerk▪ Billing Clerk▪ Reception Clerk▪ Registration Clerk	✓	Resolve Patient Billing Issue
Reception Clerk	Receives patient on arrival at the hospital. Verifies registration. Arranges for the patient to receive medical service.		Receive Patient
Registration Clerk	Enters or updates patient's personal and payment data. Issues a hospital card, if necessary.		Register Patient

Dependencies: Include and Extend

- An extend relationship is one in which a use case is created to extend the functionality of a base use case.

	Alternate Flow/ Exceptions:	<p>2.a The patient is not new and insurance data has not changed. Registration clerk does not update the insurance data by default.</p> <p>2.b The patient wants to pay the entire bill or the co-payments by a credit card. Registration clerk verifies the credit card (<u>Extend</u>: 142 - Verify Credit Card) and records credit card information.</p>
--	--	---

Include Relationship

- An include relationship is one in which one use case uses the functionality of another, independent, use case.

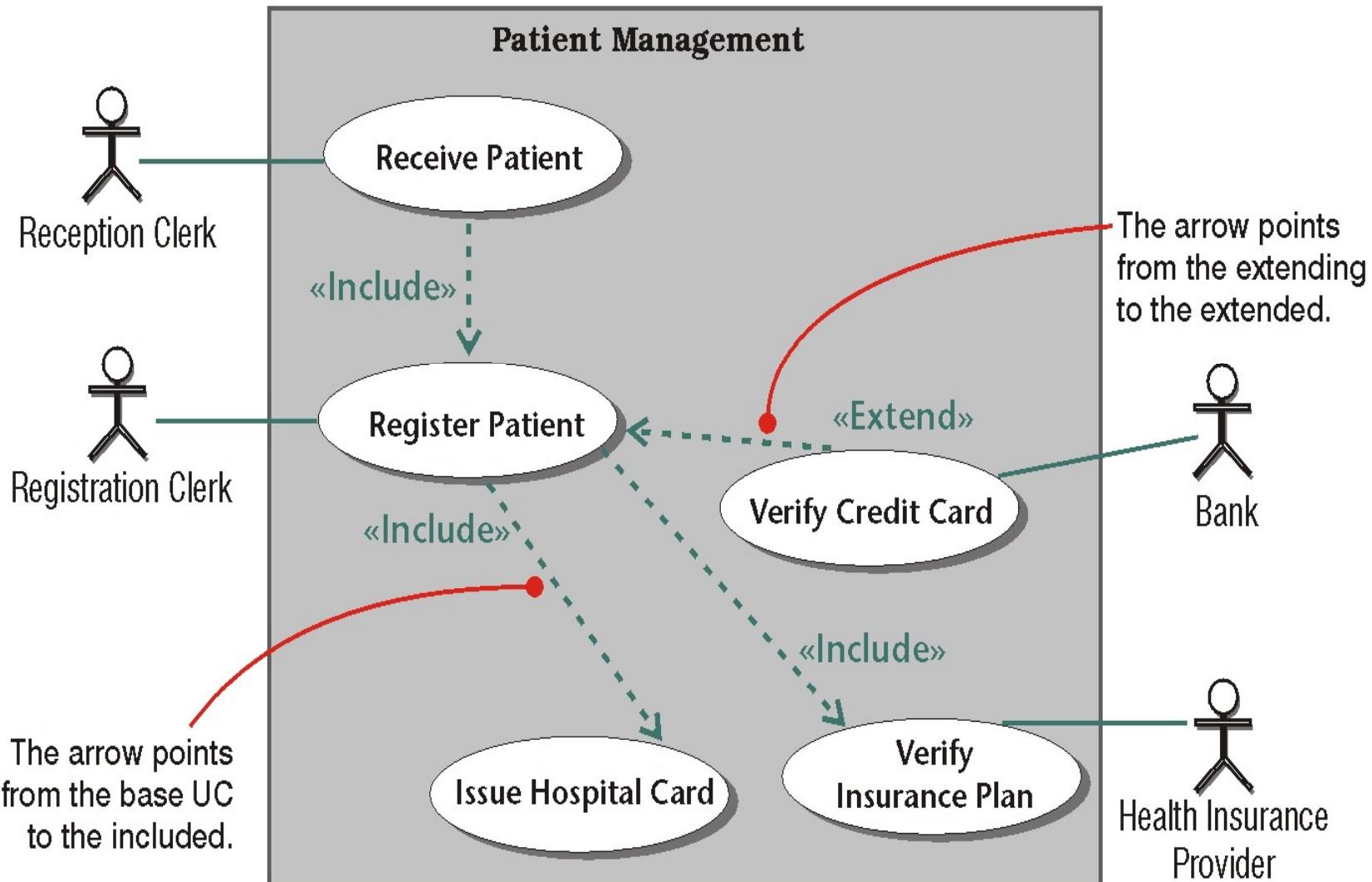
	Normal Flow:	<p>...</p> <p>3. Reception clerk verifies that patient has been registered and registration is valid.</p> <p>...</p>
	Alternate Flow/ Exceptions:	<p>3.a Patient is new. Reception clerk directs the patient to registration. (<u>Include:</u> 140 - Register Patient.)</p> <p>...</p>

Use Case Diagram for Dependencies

- In a use case diagram, dependency type is indicated by the direction of an arrow.

Use Case Dependencies

Include & Extend



Base UC	Arrow's Direction	Referenced UC
Extended UC Register Patient	←	Extending UC Verify Credit Card
Including UC Receive Patient	→	Included UC Register Patient

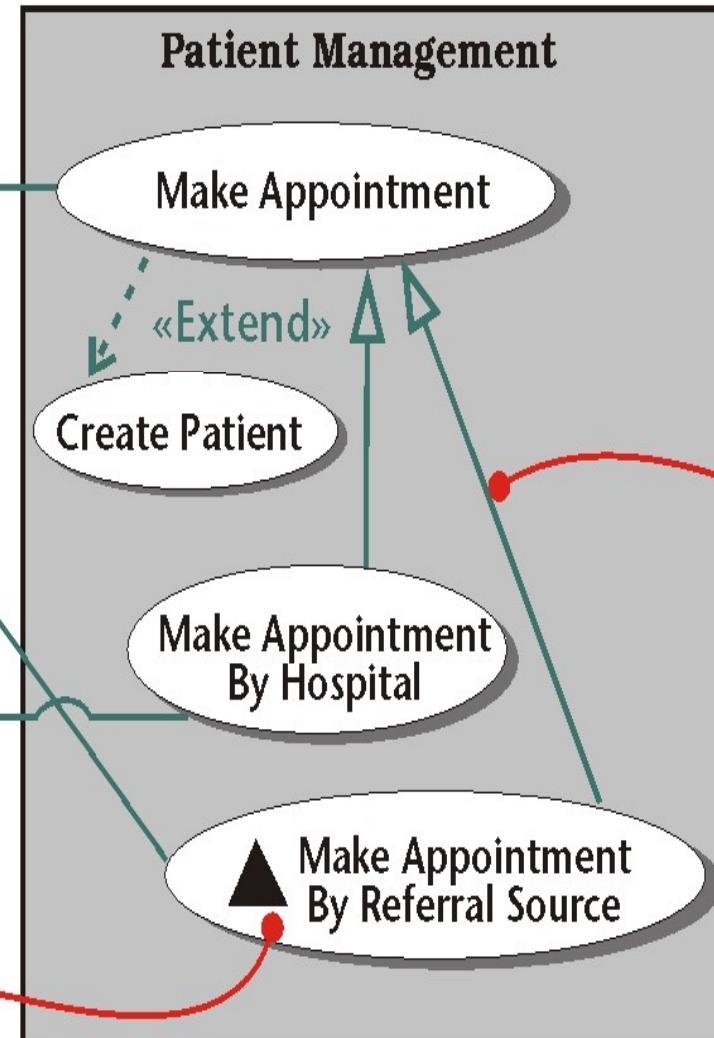
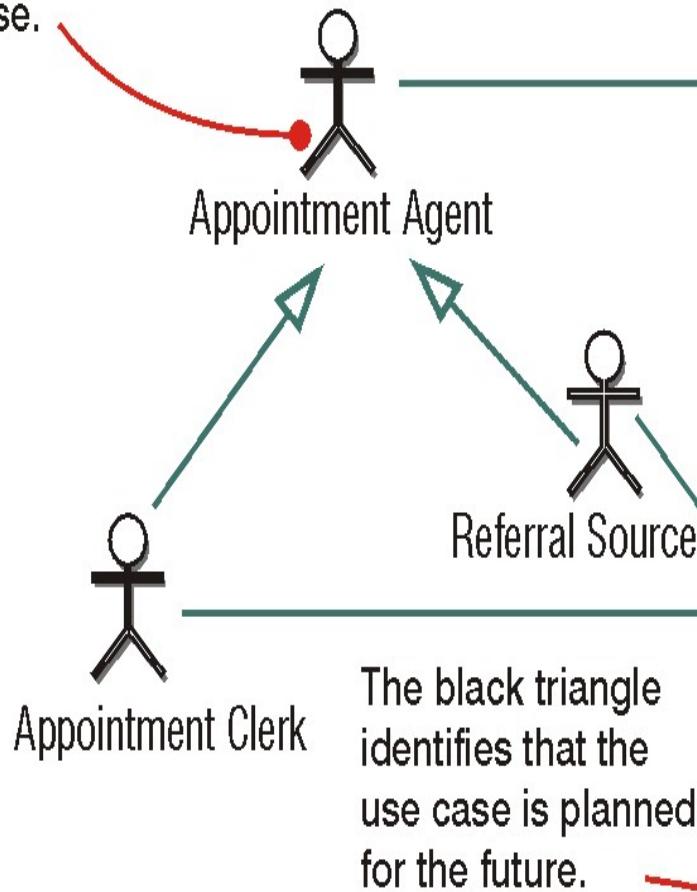
Use Case Generalization

- We generalize use cases when they achieve the same goal by different means.

Use Case Generalization

When the Same Goal Is Achieved by Different Means

A new, abstract actor for the abstract use case.



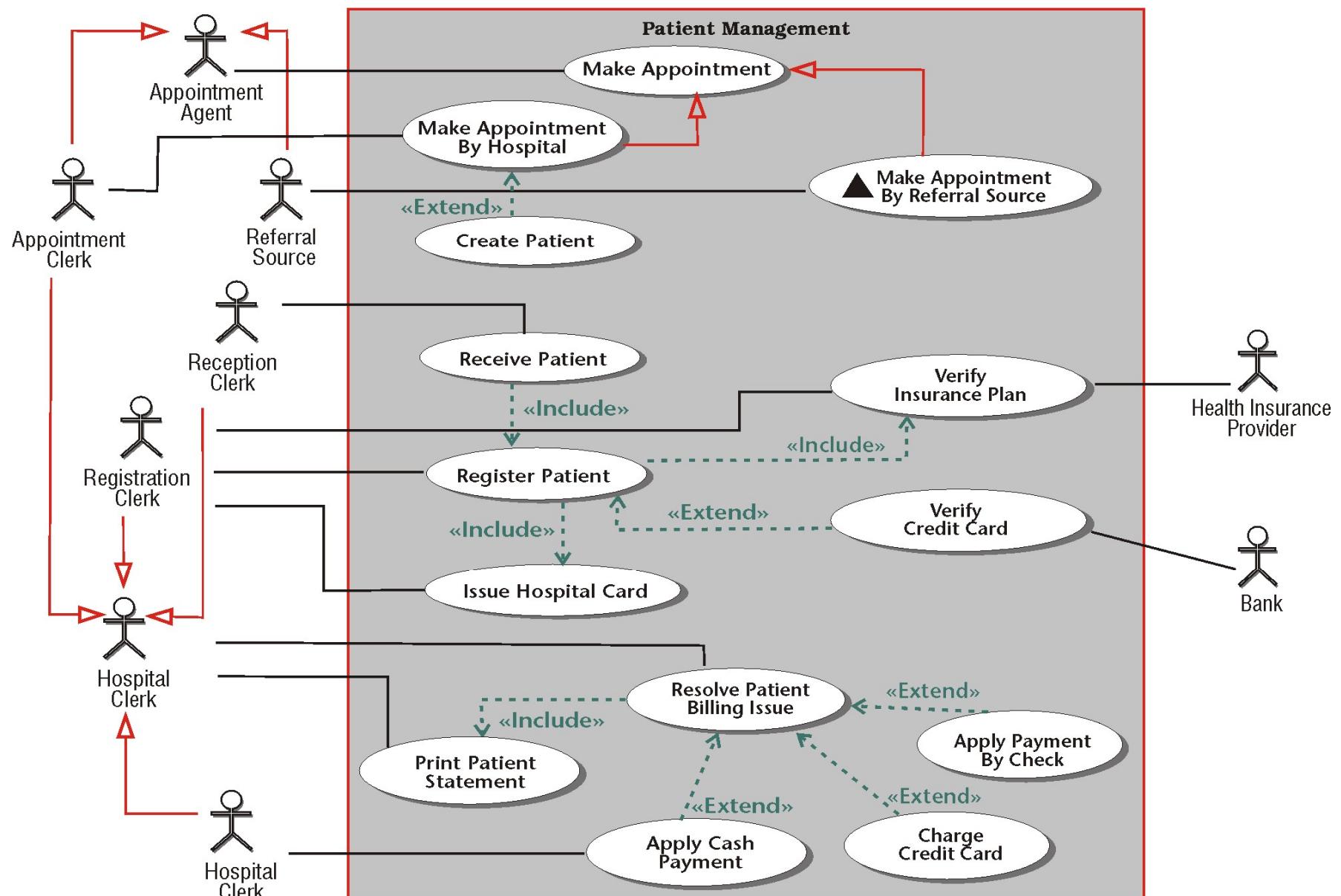
The arrow points from the child use case to the parent use case.

Use Case Diagram

- Use case diagram is a meta-model that portrays associations among actors, use cases and the system.

Use Case Diagram

A Meta-Model for the 'Big Picture'

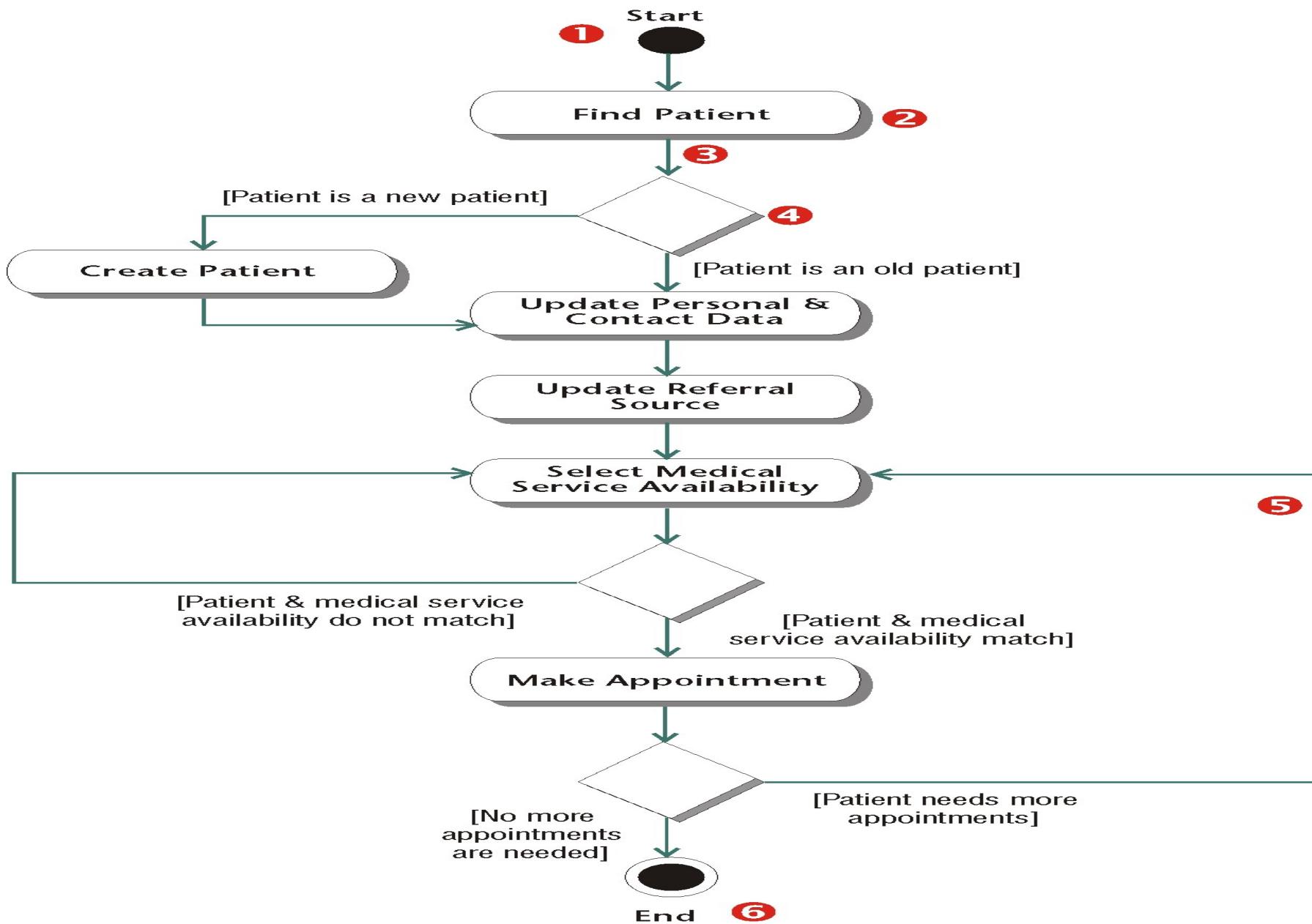


Activity Diagram

- Activity diagram depicts the flow from activity to activity. It presents a visual, dynamic view of the system and its components.

Activity Diagram

The Logical Flow of Make Appointment



Uses of Use Cases

- Use cases provide a crucial framework for analysis, design, implementation and deployment activities.

Uses of Use Cases

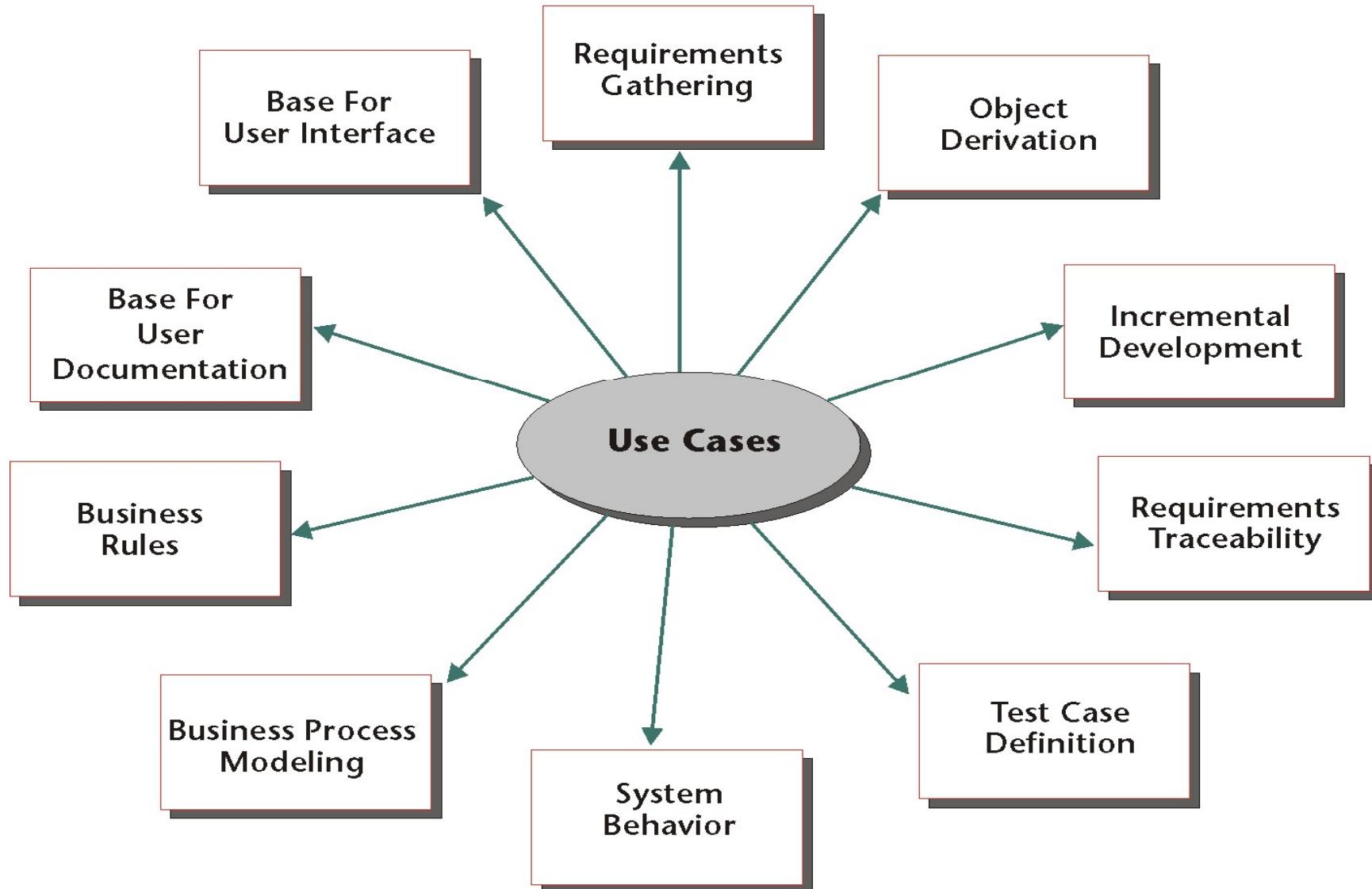
- Requirements Gathering
 - Use cases provide the base tools for gathering requirements within a meaningful context.
- Requirements Traceability
 - Use cases and their supporting documents are the prime sources for tracing requirements.
- Business Rules
 - Use cases are the framework for gathering business rules.
- System Behavior
 - The external behavior of any open system can be captured effectively through use cases.

Uses of Use Cases

- Object Derivation
 - By launching a cycle of gathering requirements from the use cases, we can arrive at many of the objects that would form the structure of the system.
- Incremental Development
 - By prioritizing use cases and their dependencies, we can build a system incrementally.
- Base for User Interface
 - Use cases describe the basic messages that the actor and the system must exchange to achieve a goal.
- Test Case Definition
 - Use cases are the conceptual blueprints for functional test cases.
- Base for User Documentation
 - Use cases are built to describe the interaction between a user type and a system.
- Business Process Modeling
 - Use cases can be used to model business processes, prior to, after, or independent from an information system.

Uses of Use Case Modeling

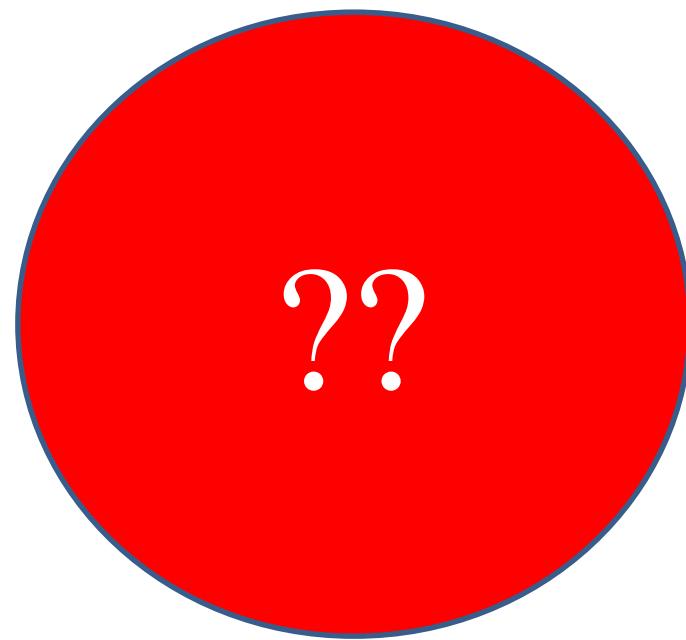
A Framework for Development

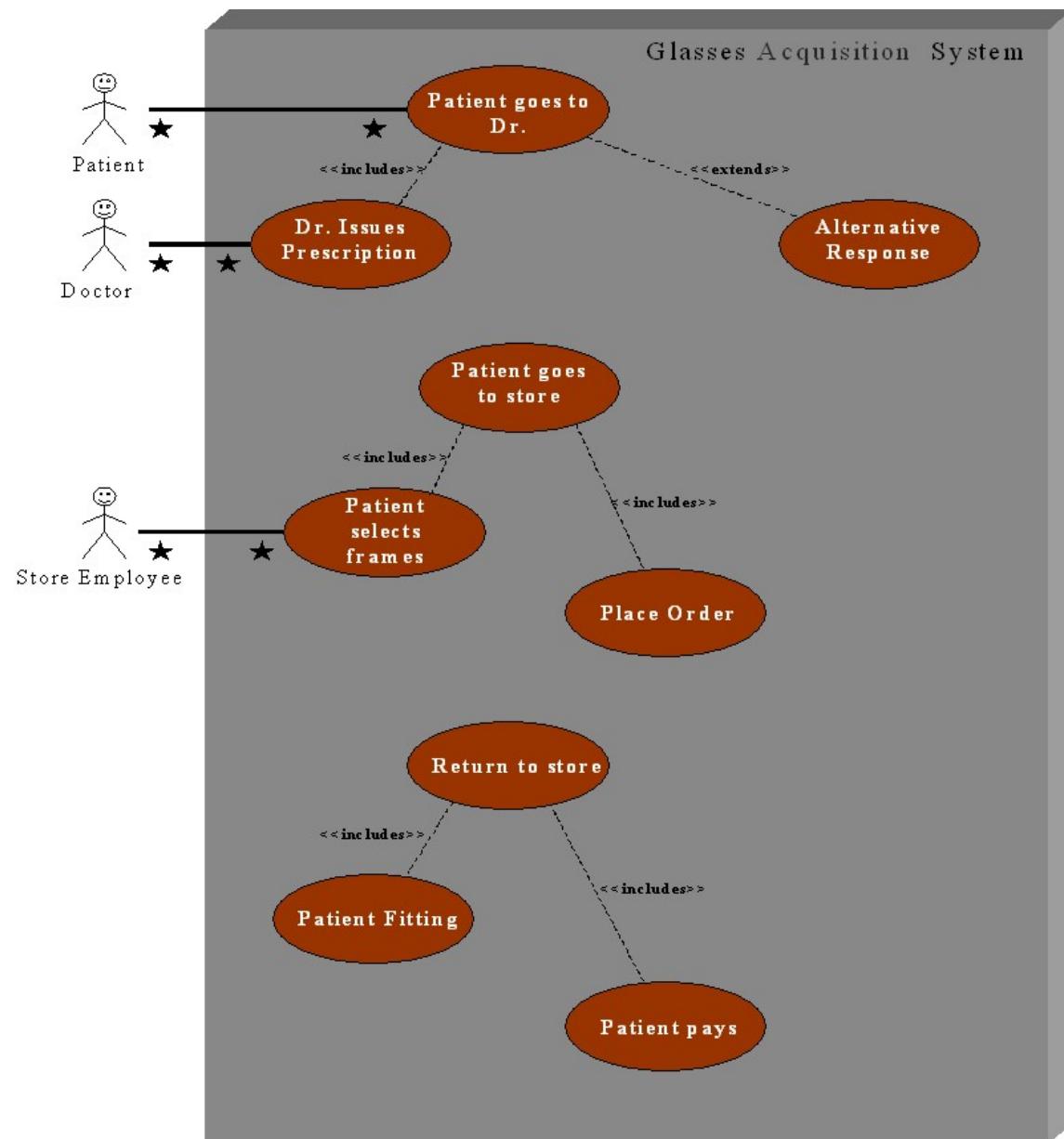


Case Study #1

- Draw a use-case diagram for the process of buying glasses from the viewpoint of the patient. The first step is to see an eye doctor who will give you a prescription. Once you have a prescription, you go to an optical dispensary, where you select your frames and place the order for your glasses. Once the glasses have been made, you return to the store for a fitting and pay for the glasses.

- Draw the UML USE-CASE Diagram

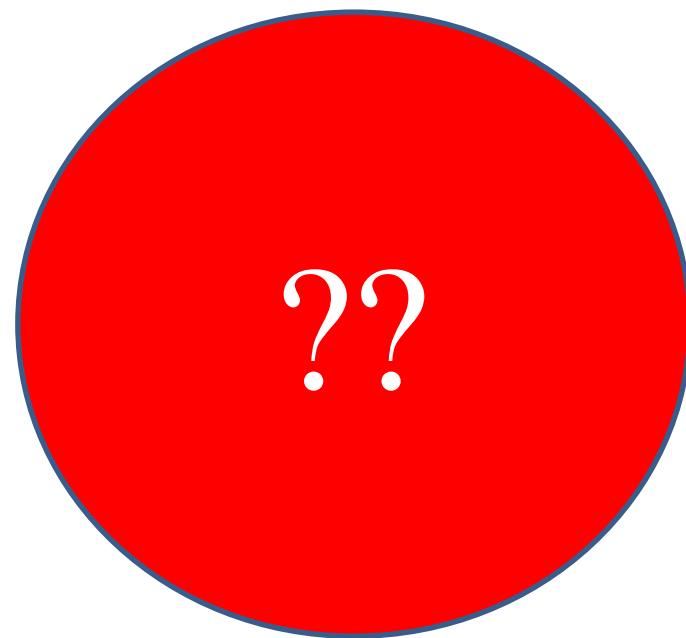


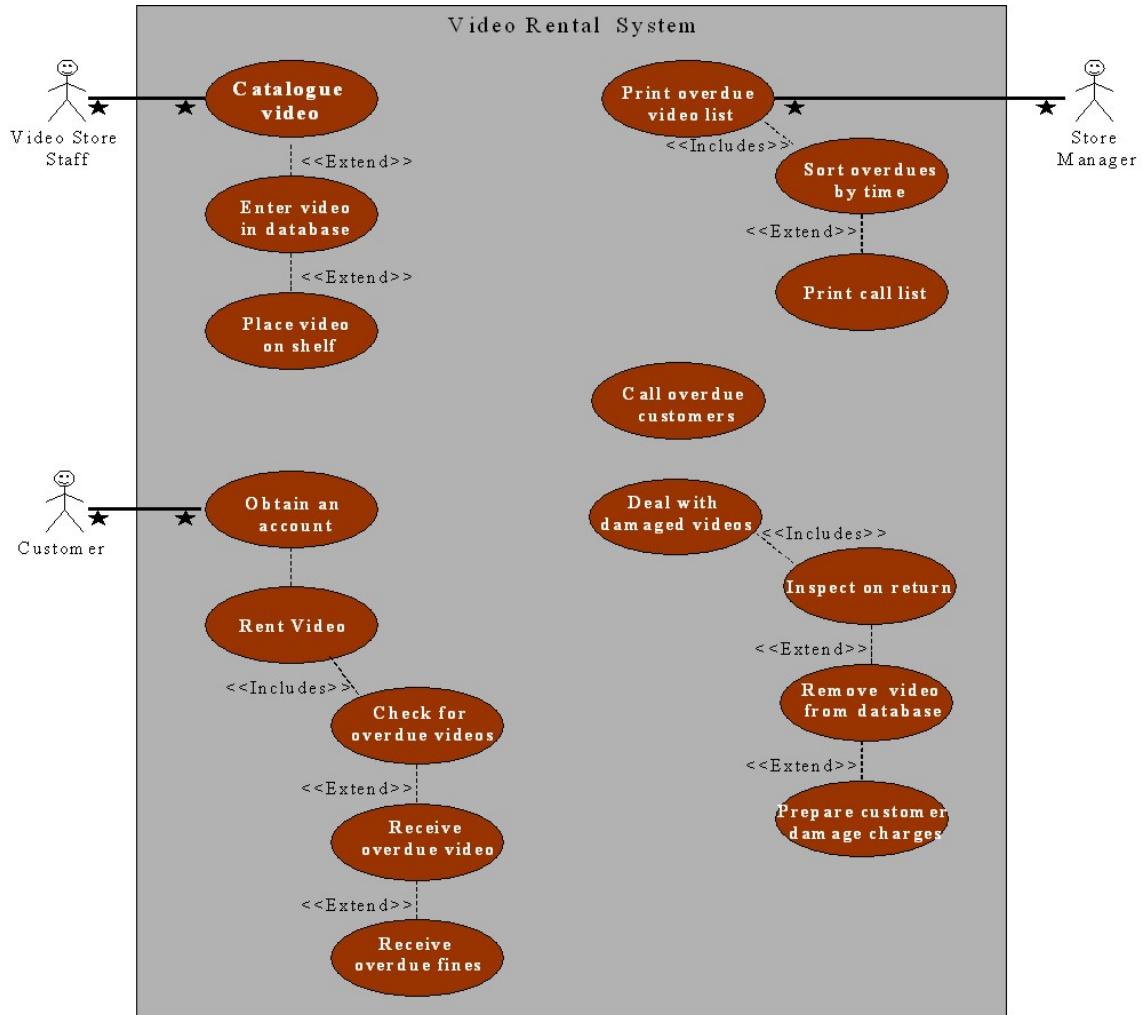


Case Study #2

- Draw a use-case diagram for the following system. A Video Store (AVS) runs a series of fairly standard video stores. Before a video can be put on the shelf, it must be cataloged and entered into the video database. Every customer must have a valid AVS customer card in order to rent a video. Customers rent videos for three days at a time. Every time a customer rents a video, the system must ensure that they do not have any overdue videos. If so, the overdue videos must be returned and an overdue fee paid before customer can rent more videos. Likewise, if the customer has returned overdue videos, but has not paid the overdue fee, the fee must be paid before new videos can be rented. Every morning, the store manager prints a report that lists overdue videos. If a video is two or more days overdue, the manager calls the customer to remind them to return the video. If a video is returned in damaged condition, the manager removes it from the video database and may sometimes charge the customer.

- Draw the UML USE-CASE Diagram

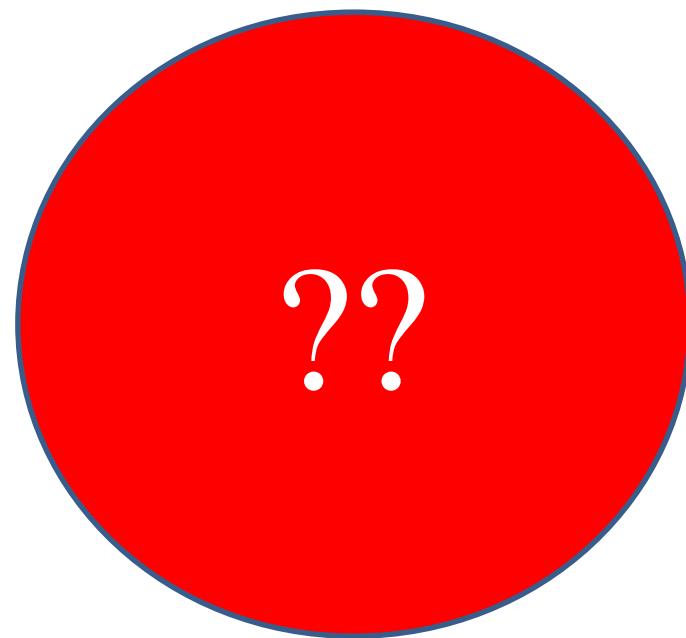


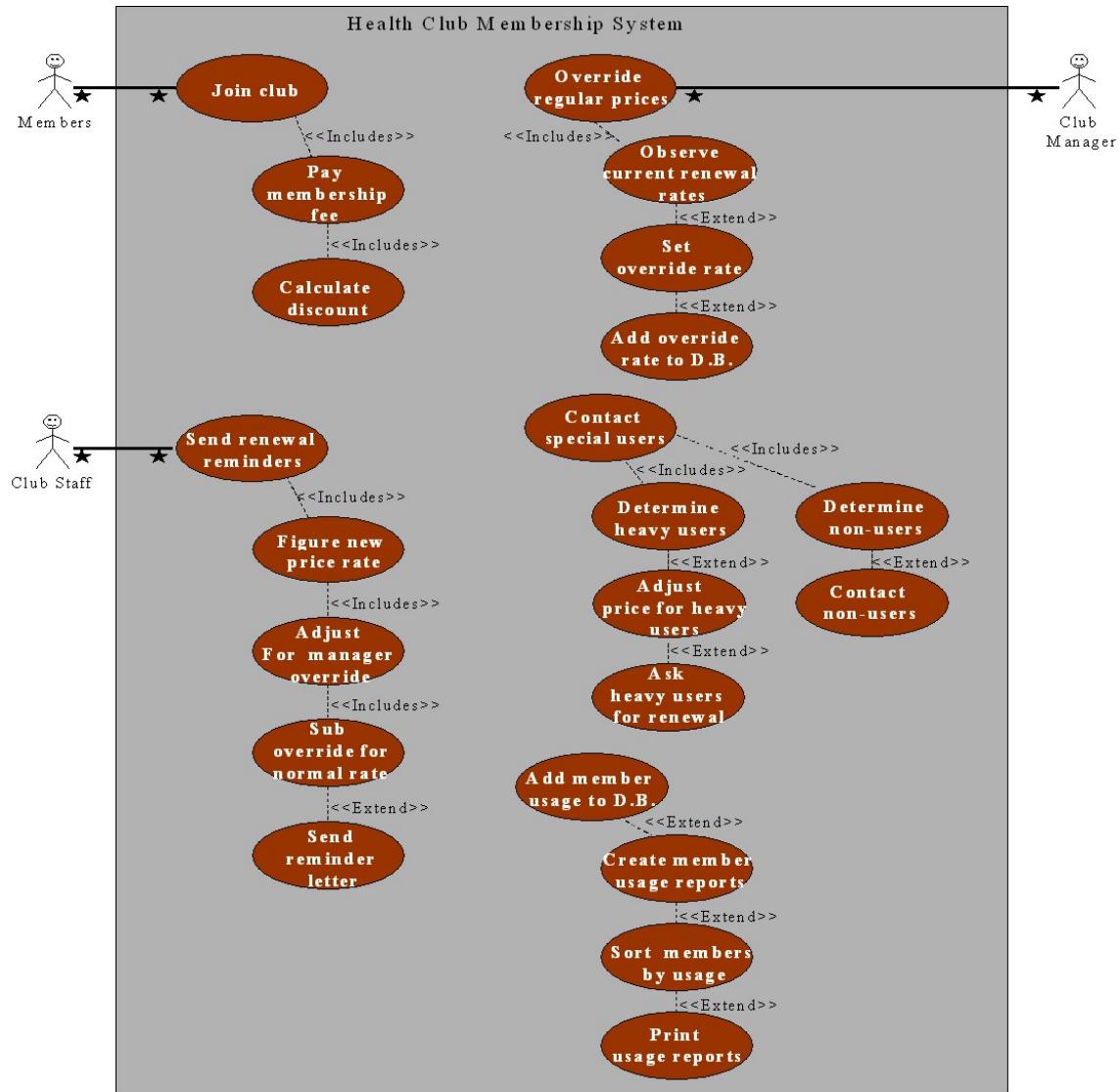


Case Study #3

- Draw a use-case diagram for a health club membership system. When members join the health club, they pay a fee for a certain length of time. Most memberships are for one year, but memberships as short as two months are available. Throughout the year, the health club offers a variety of discounts on their regular membership prices (e.g., two memberships for the price of one for Valentine's Day). It is common for members to pay different amounts for the same length of membership. The club wants to mail out reminder letters to members asking them to renew their memberships one month before their memberships expire. Some members have become angry when asked to renew at a much higher rate than their original membership contract, so the club wants to track the price paid so that the manager can override the regular prices with special prices when members are asked to renew. The system must track these new prices so that renewals can be processed accurately. One of the problems in the health club industry is the high turnover rate of members. While some members remain active for many years, about half of the members do not renew their memberships. This is a major problem, because the health club spends a lot in advertising to attract each new member. The manager wants the system to track each time a member comes into the club. The system will then identify the heavy users, and generate a report so the manager can ask them to renew their memberships early, perhaps offering them a reduced rate for early renewal. Likewise, the system should identify members who have not visited the club in more than a month, so the manager can call them and attempt to reinterest them in the club

- Draw the UML USE-CASE Diagram

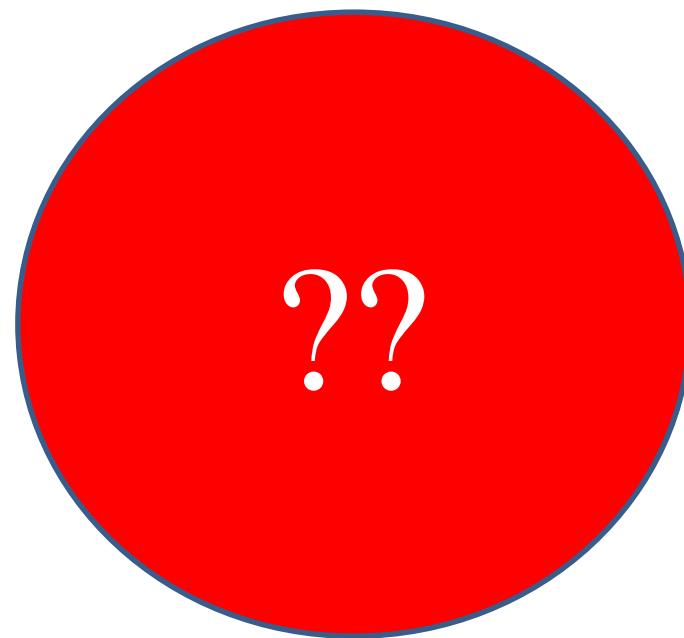


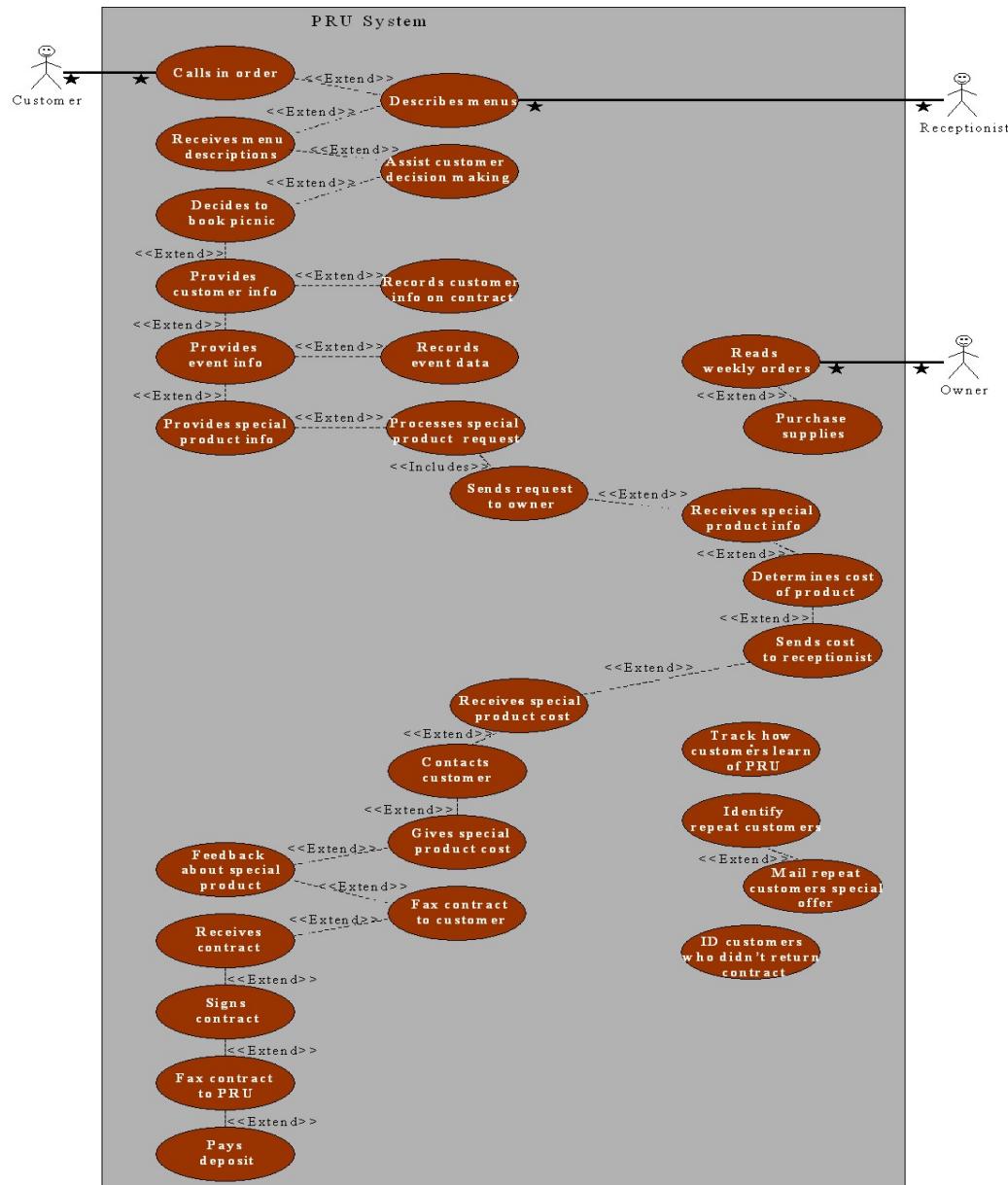


Case Study #4

- Draw a use-case diagram for the following system. Picnics R Us (PRU) is a small catering firm with five employees. During a typical summer weekend, PRU caters fifteen picnics with twenty to fifty people each. The business has grown rapidly over the past year and the owner wants to install a new computer system for managing the ordering and buying process. PUR has a set of ten standard menus. When potential customers call, the receptionist describes the menus to them. If the customer decides to book a picnic, the receptionist records the customer information (e.g., name, address, phone number, etc.) and the information about the picnic (e.g., place, date, time, which one of the standard menus, total price) on a contract. The customer is then faxed a copy of the contract and must sign and return it along with a deposit (often a credit card or by check) before the picnic is officially booked. The remaining money is collected when the picnic is delivered. Sometimes, the customer wants something special (e.g., birthday cake). In this case, the receptionist takes the information and gives it to the owner who determines the cost; the receptionist then calls the customer back with the price information. Sometimes the customer accepts the price, other times, the customer requests some changes that have to go back to the owner for a new cost estimate. Each week, the owner looks through the picnics scheduled for that weekend and orders the supplies (e.g., plates) and food (e.g., bread, chicken) needed to make them. The owner would like to use the system for marketing as well. It should be able to track how customers learned about PUR, and identify repeat customers, so that PUR can mail special offers to them. The owner also wants to track the picnics on which PUR sent a contract, but the customer never signed the contract and actually booked a picnic.

- Draw the UML USE-CASE Diagram

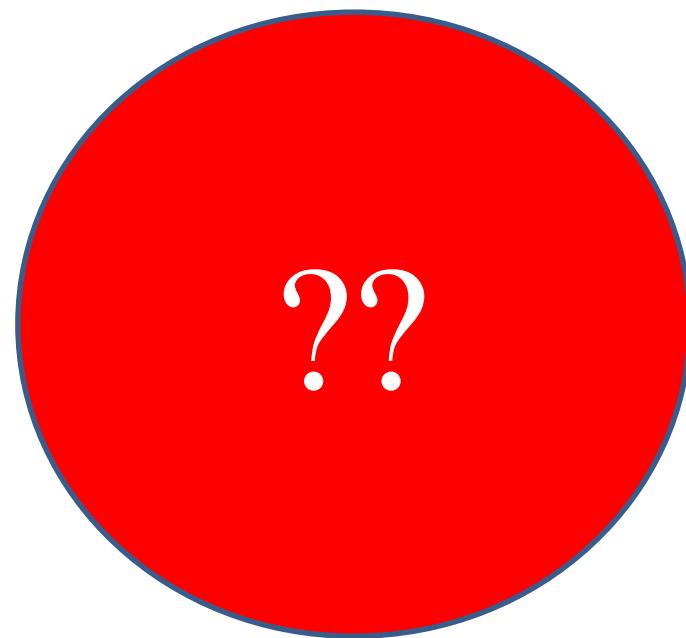


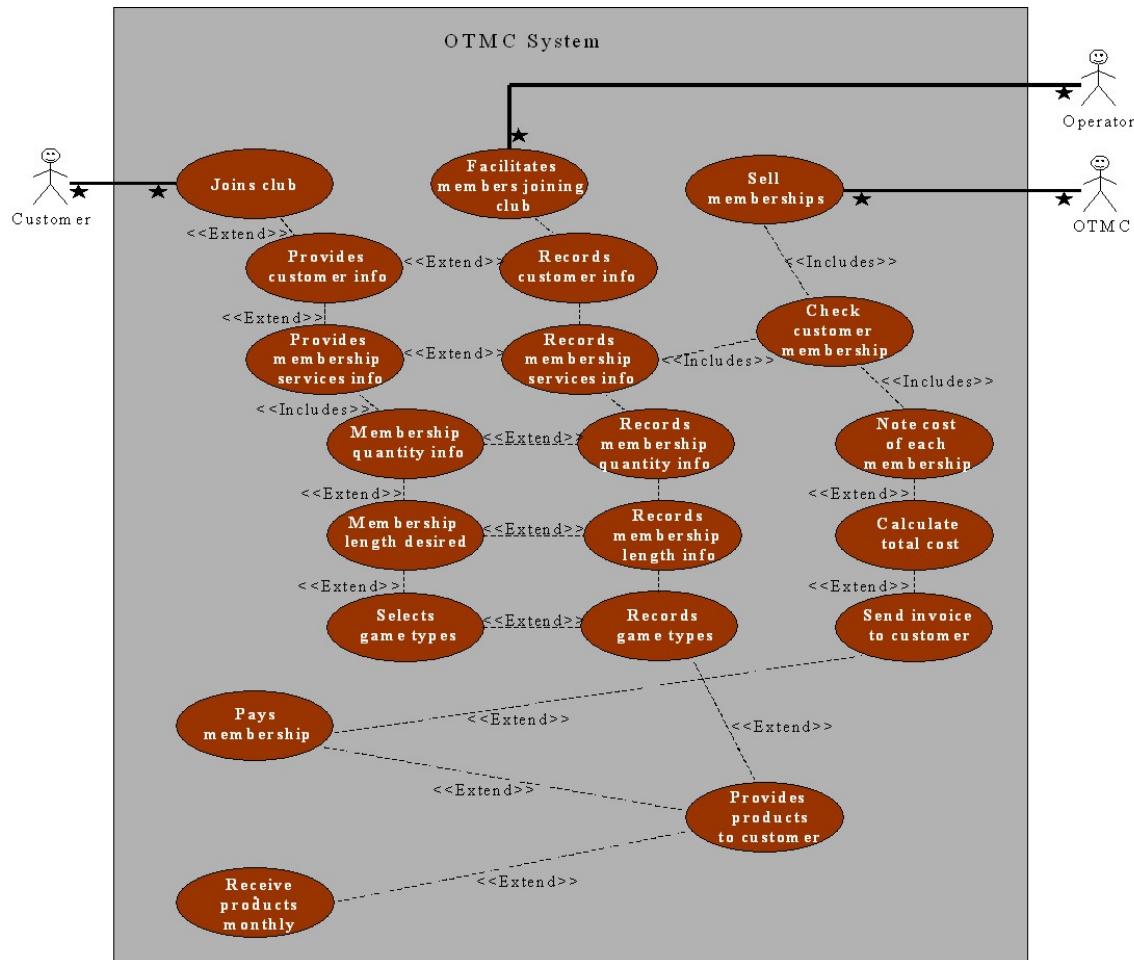


Case Study #5

- Draw a use-case diagram for the following system. Of-the-Month Club (OTMC) is an innovative young firm that sells memberships to people who have an interest in certain products. People pay membership fees for one year and each month receive a product by mail. For example, OTMC has a coffee-of-the-month club that sends members one pound of special coffee each month. OTMC currently has six memberships (coffee, wine, beer, cigars, flowers, and computer games) each of which costs a different amount. Customers usually belong to just one, but some belong to two or more. When people join OTMC, the telephone operator records the name, mailing address, phone number, e-mail address, credit card information, start date, and membership service(s) (e.g., coffee). Some customers request a double or triple membership (e.g., two pounds of coffee, three cases of beer). The computer game membership operates a bit differently from the others. In this case, the member must also select the type of game (action, arcade, fantasy/ science-fiction, educational, etc.) and age level. OTMC is planning to greatly expand the number of memberships it offers (e.g., video games, movies, toys, cheese, fruit, and vegetables) so the system needs to accommodate this future expansion. OTMC is also planning to offer 3-month and 6-month memberships.

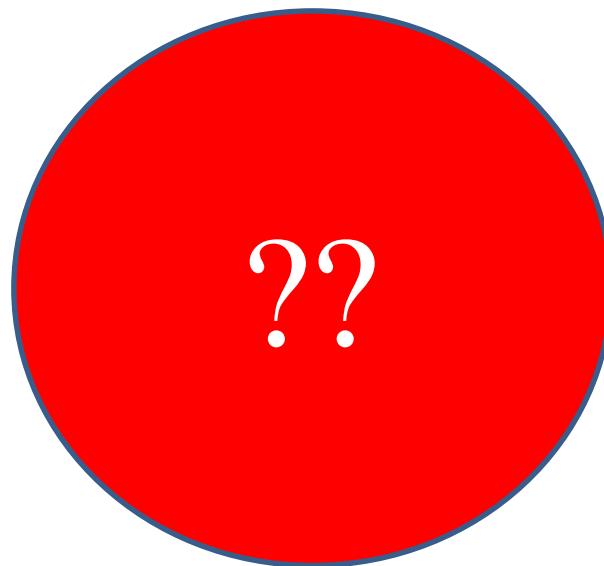
- Draw the UML USE-CASE Diagram





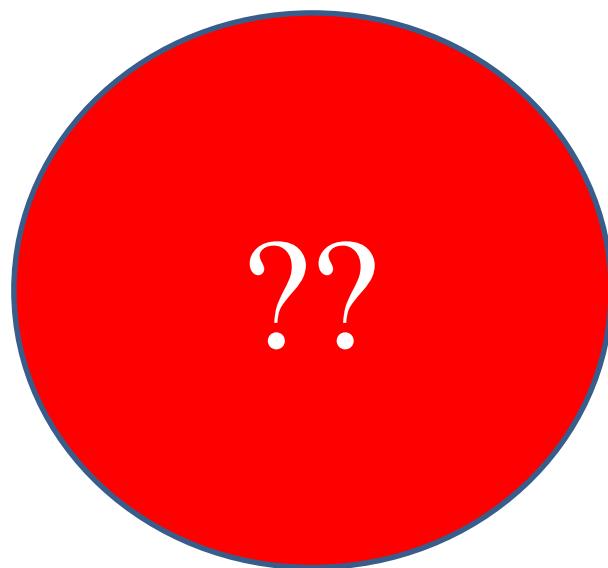
OutdoorPowerEquipmentDepot – Case Study

- Create the complete Use-Case Diagram for the Customer



OutdoorPowerEquipmentDepot – Case Study

- Create Use-Case Diagram for Manager & Account Specialist



OutdoorPowerEquipmentDepot – Case Study

- Create the COMPLETE Use-Case Diagram

