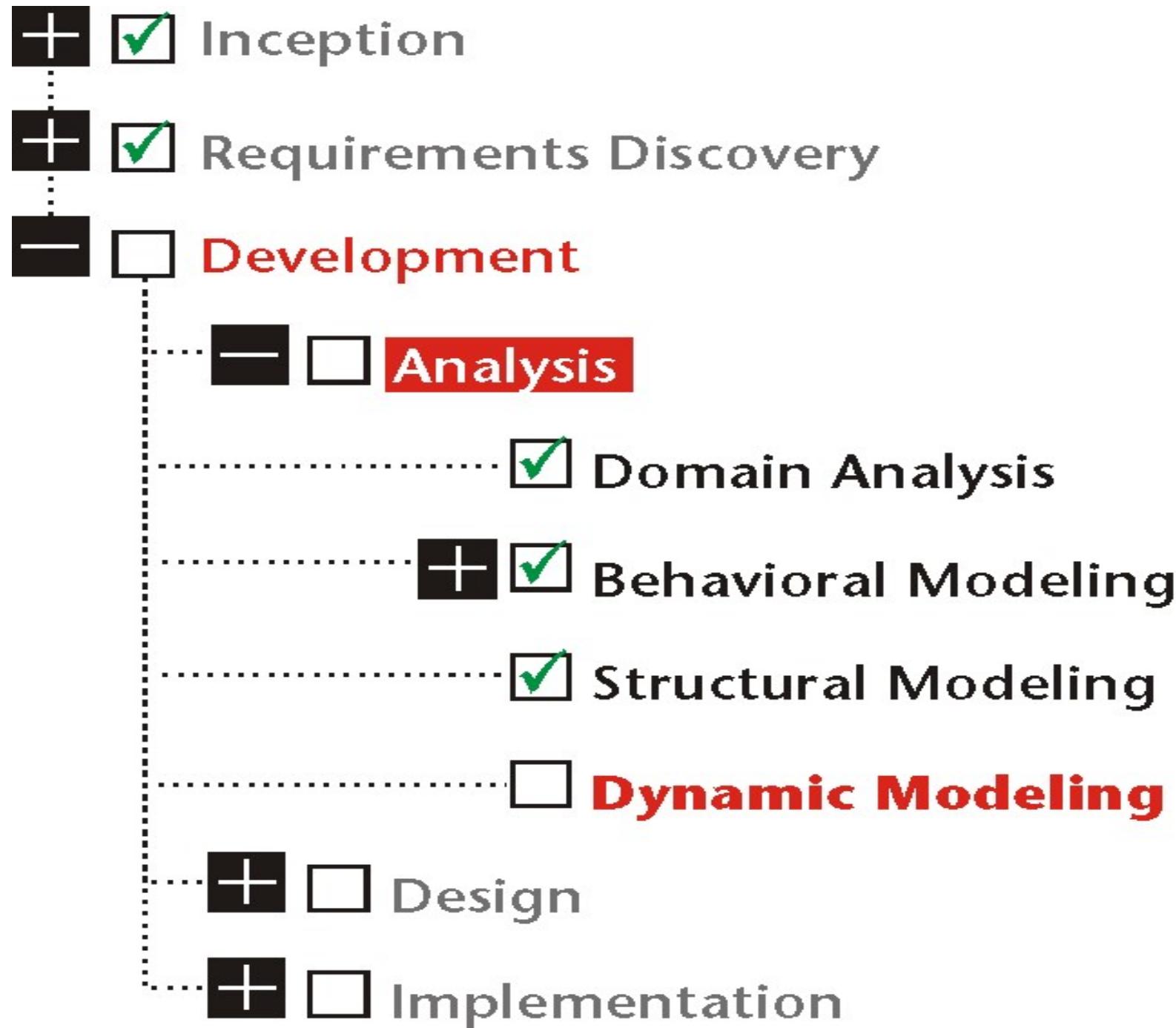
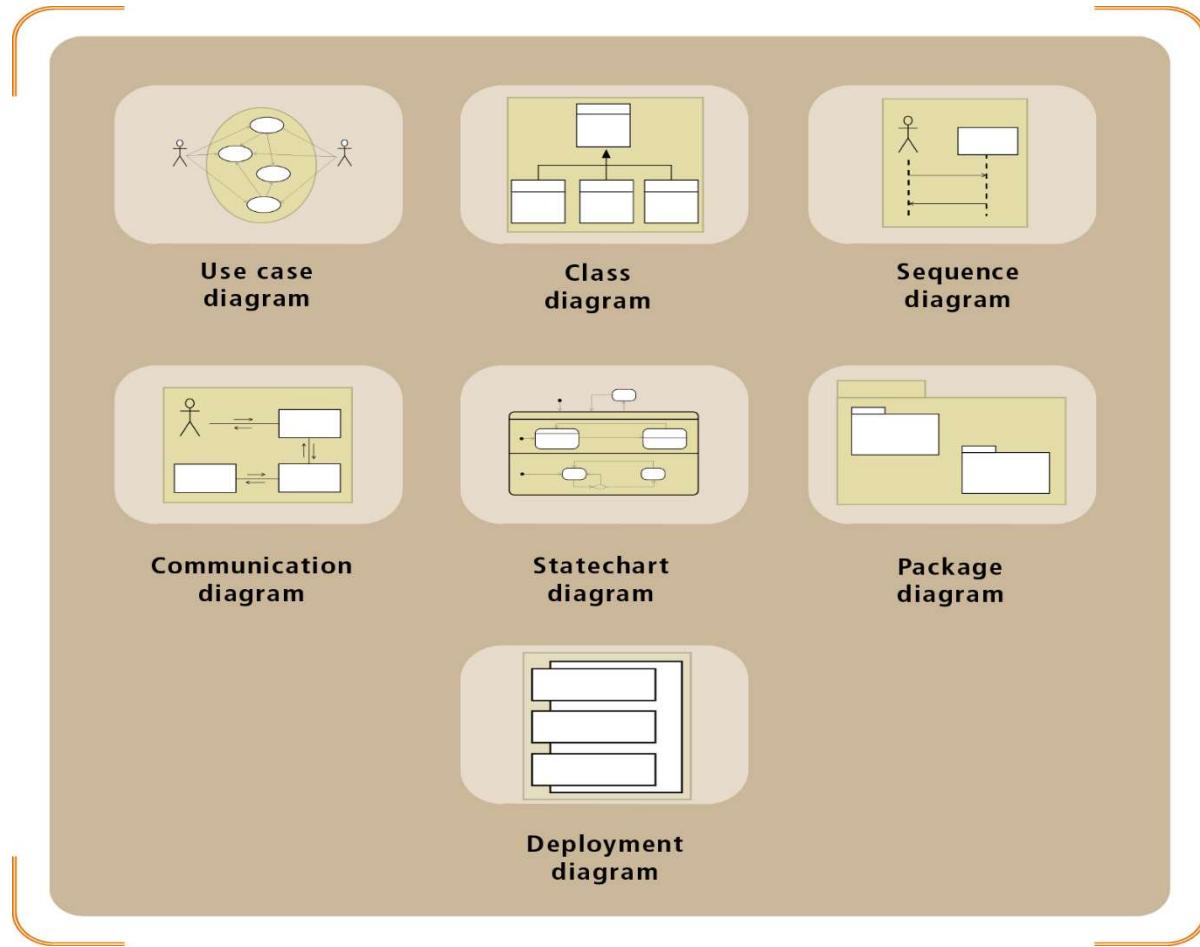


Dynamic Modeling

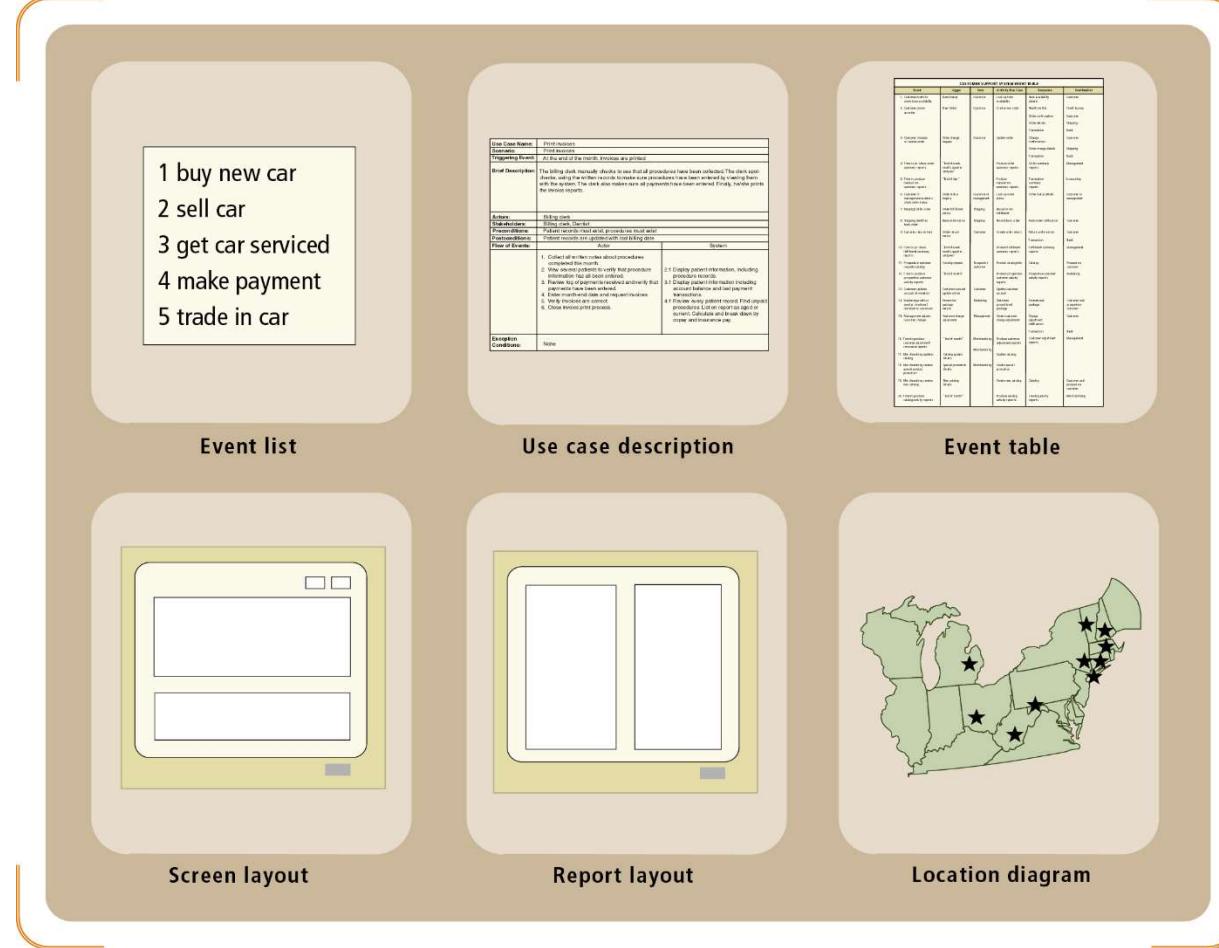
Topics

- Dynamic modeling and its relationship with behavioral and structural modeling.
- How objects interact by exchanging of messages.
- The role of parameters and return values in exchanging messages.
- The significance of methods and their relationships with operations.
- Events and their significance in dynamic modeling.
- How to use sequence diagrams to create a chronological model of interaction among objects.
- How collaboration diagrams represent the organization of objects that must cooperate to reach an objective.
- Statechart diagram, the tool that models the changes of a single class of objects through their lifetime.
- Revisit activity diagram and learn more about its capabilities.

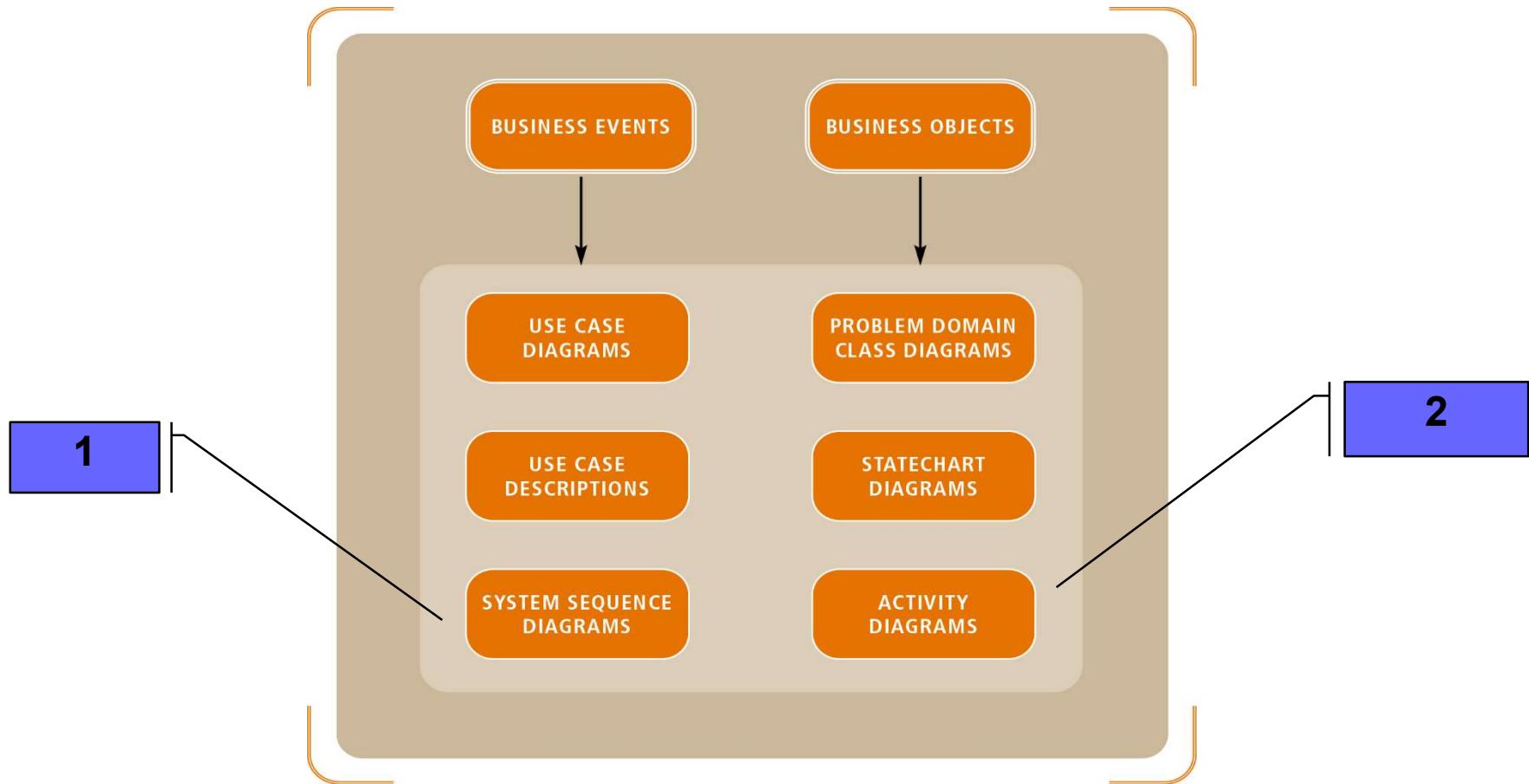




UML Diagrams used for Object-Oriented Analysis and Design



Additional Models used for Requirements and Design Disciplines



Requirements Diagrams With UML Models

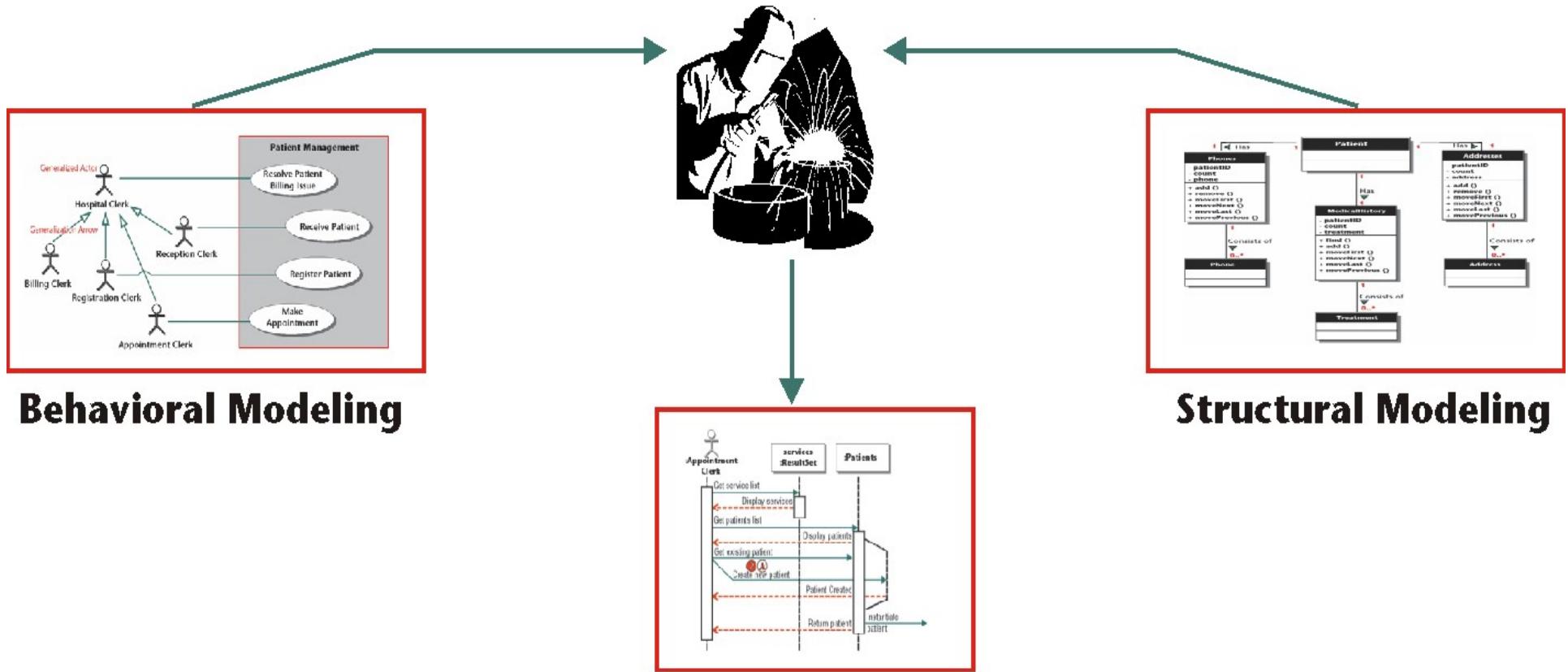
Dynamic Modeling

- Dynamic modeling represents the interaction of the **building blocks of the information system** with each other and with the outside world to satisfy the behavioral requirements of the system.

Dynamic Modeling

- Dynamic modeling is also *interaction*. Unlike the real world in which objects interact in many different ways, residents of a virtual world can only interact through **messages**.
- Dynamic modeling features **objects**.
- Interactions happen in **time**. Therefore, dynamic modeling must show not only who interacts with whom and how, but in what **order**.

Welding Structure to Behavior



Behavioral Modeling

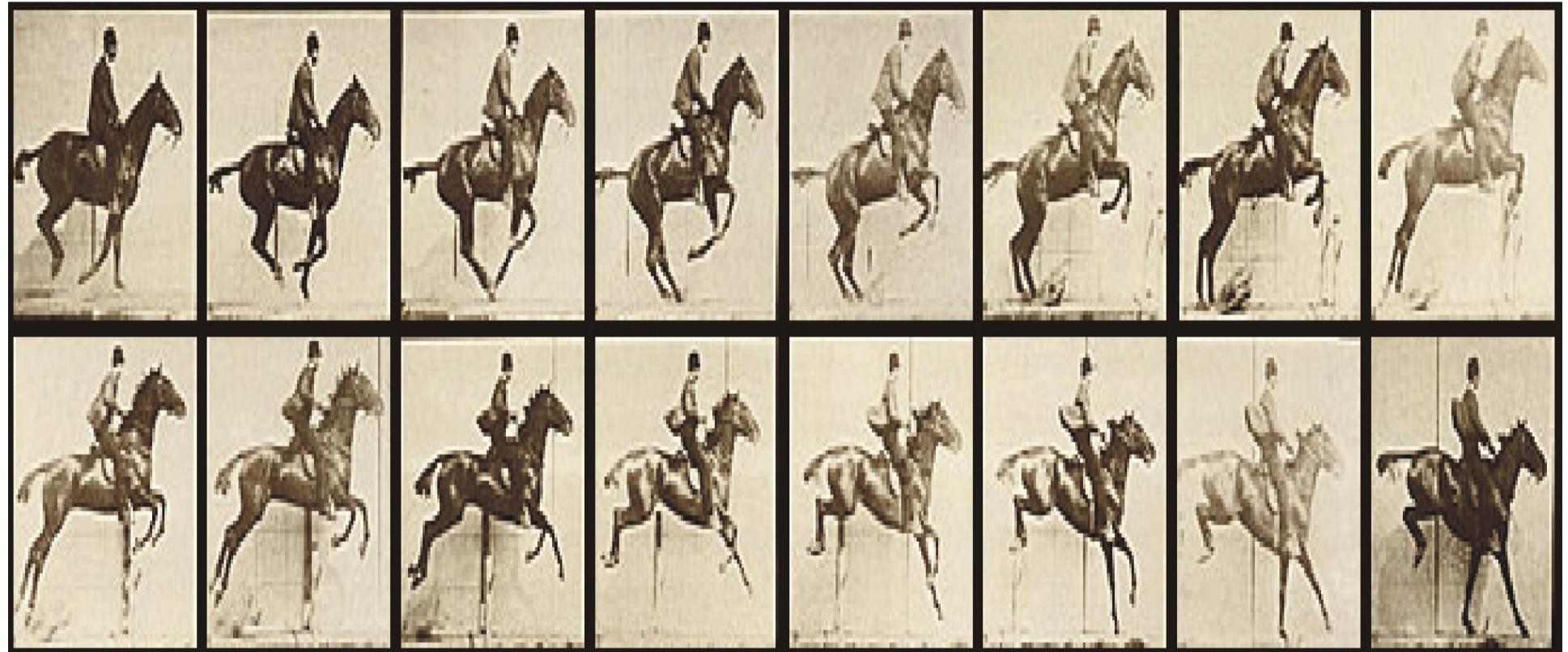
Structural Modeling

Dynamic Modeling

To arrive at a dynamic view of the system, we must place its structure within the context of its behavior.

Dynamic Modeling

Structure in Motion



Dynamic modeling represents **structure in motion**.

Object Interaction

- Dynamic modeling is about *objects*—their interactions and how the objects change through interaction.
- In an object-oriented virtual system the only possible way of interaction is through sending and receiving **messages**.

Object Interaction

- An object is a **black box**.
 - Its inner workings and its knowledge are hidden from the outside world.
 - The users of the system and other objects—may interact with the object only through its **public interface**.
- How do the users interact with the system?
 - The answer is by exchanging **messages**.

Messages

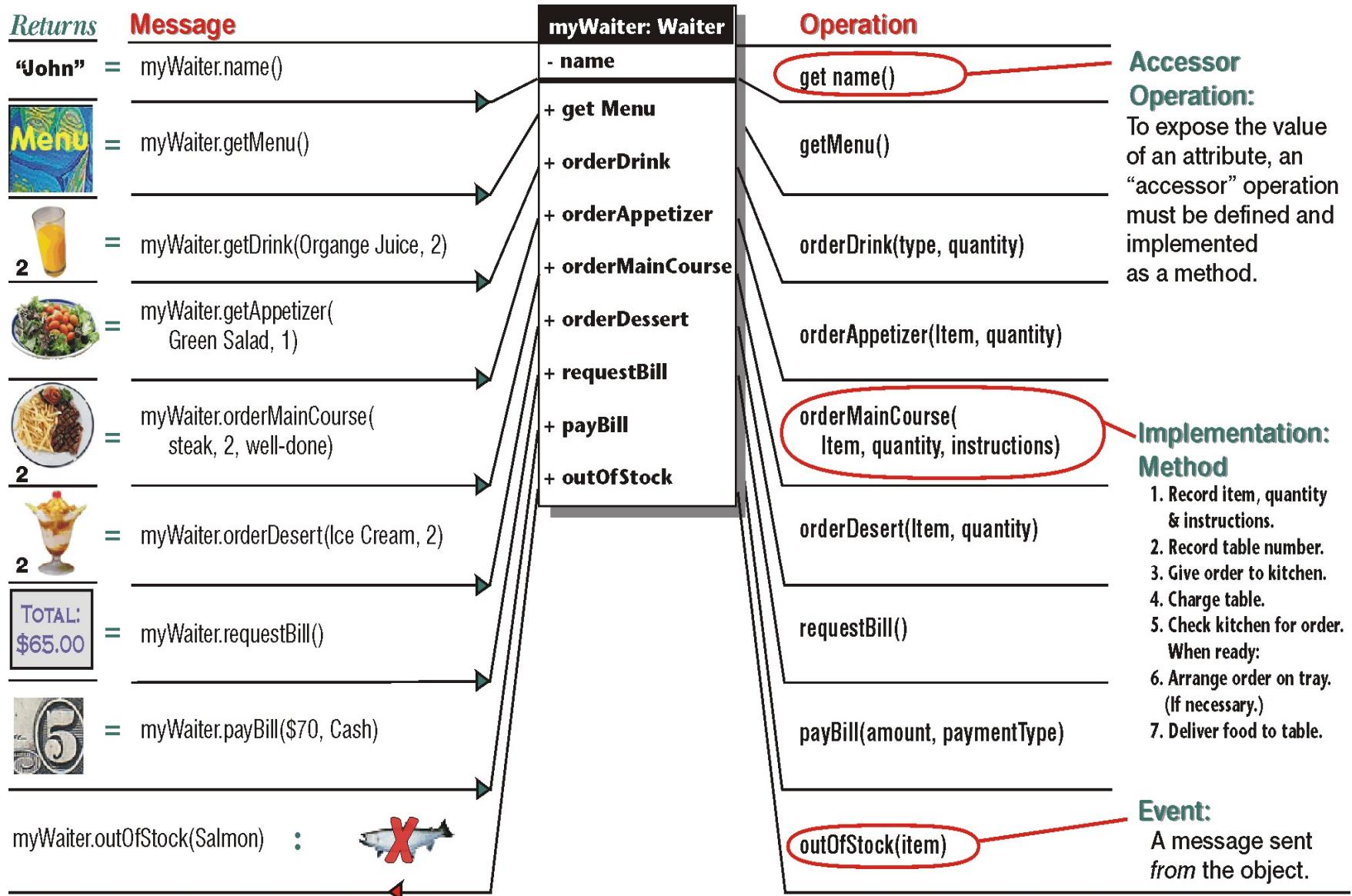
- Messages are instructions and information sent to objects in the expectation that the recipient objects will carry out certain actions.

Restaurant Example

Use Case:	Have Dinner
Scope:	Restaurant
Primary Actor:	Customer
Normal Flow:	<ol style="list-style-type: none">1. Customer is seated at a table2. Customer asks the waiter for the menu.3. Customer orders drink(s).4. Customer orders appetizer(s).5. Customer orders main course(s).6. Customer orders dessert(s).7. Customer asks for the bill.8. Customer pays the bill and receives change (if any).
Alternate Flow / Exceptions:	1.a Customer asks the waiter for his or her name.

Methods & Messages

How Objects Interact



Accessor Objects

- Accessor operations change or return the value of an **attribute**.
 - The **get** operation returns the value of the attribute to the public.
 - The **set** operation changes the value of the attribute.
- The syntax of an accessor operation is different from that of a normal operation.
 - `getName()`

Parameters

- Parameters, or arguments, specify the data that must be supplied to an object to carry out a specific operation:
- Examples:
 - orderDrink(drink, quantity)
 - orderAppetizer(Appetizer)
 - getMenu()
 - orderMainCourse(whisky, on the rocks, many)

Messages and Operations

- The syntax of a message matches the operation's names and parameters, but a message and its corresponding operation are phrased differently.
 - orderDrink(Drink,Quantity)
 - orderDrink(Orange Juice, 2)
- The option to ignore the argument applies only to the message, *not* to the operation.
 - orderAppetizer(Green Salad)
 - orderAppetizer(appetizer, quantity=1)

Return Value

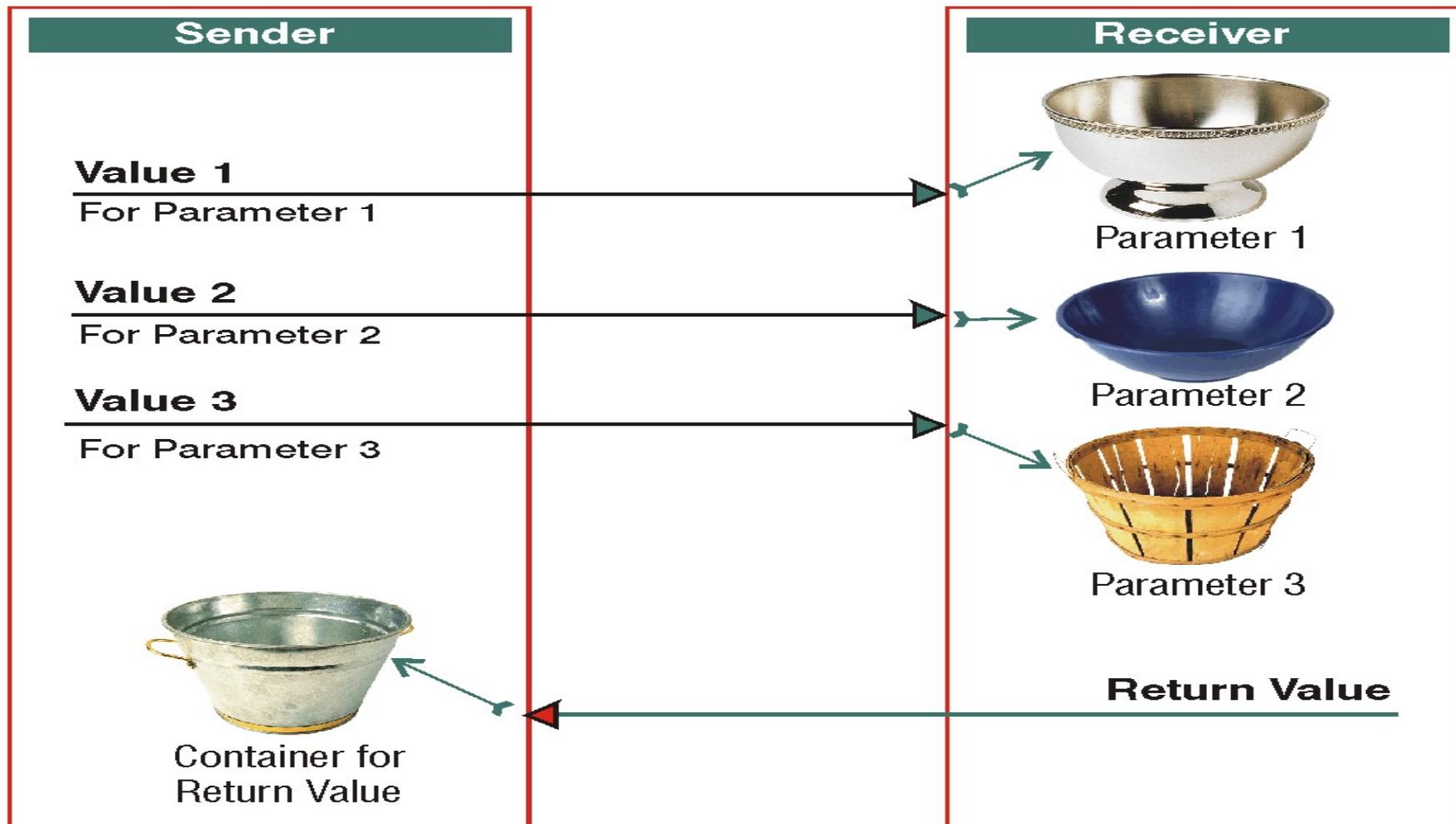
- Return value is the reply that a message may invoke from the receiving object after an operation is complete.

Syntax

- The exact syntax for declaring operations and sending messages is language dependent.
 - [Visibility] [Return Type] [Name](Param 1, . . . , Param n)
 - Public Currency payBill(amount, paymentType)

Variables Are Containers

No Container, No Data



Message

Sender of the message must provide a container for the return value.

Operation

Parameters are containers (or variables) filled by the values from the message.

Methods

- A method is how an operation is implemented or actually carried out by the object responsible for the operation.
 - polymorphism; one operation => more than one method.

Events

- Events are actions by one object that interrupt the existing condition of one or more other objects.

Events

Actions that Interrupt Existing Conditions



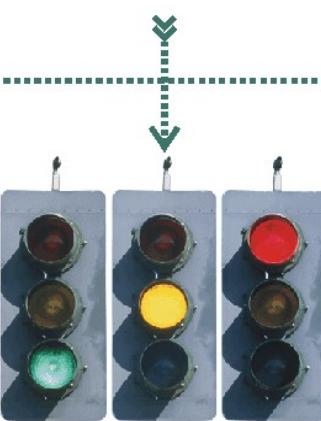
Call

An invocation of an operation.
Like a phone conversation, it is synchronous: both parties remain engaged until the action is complete



Signal

A signal event is disruptive:
it interrupts the normal course of events.
It is also asynchronous.



Change

Any event – call, signal or time – that changes the state of an object or a system is a change event.



Time

Like signals, time events are both interruptive and asynchronous. Unlike signals, they are not triggered by a prior action, but by the passage of time.

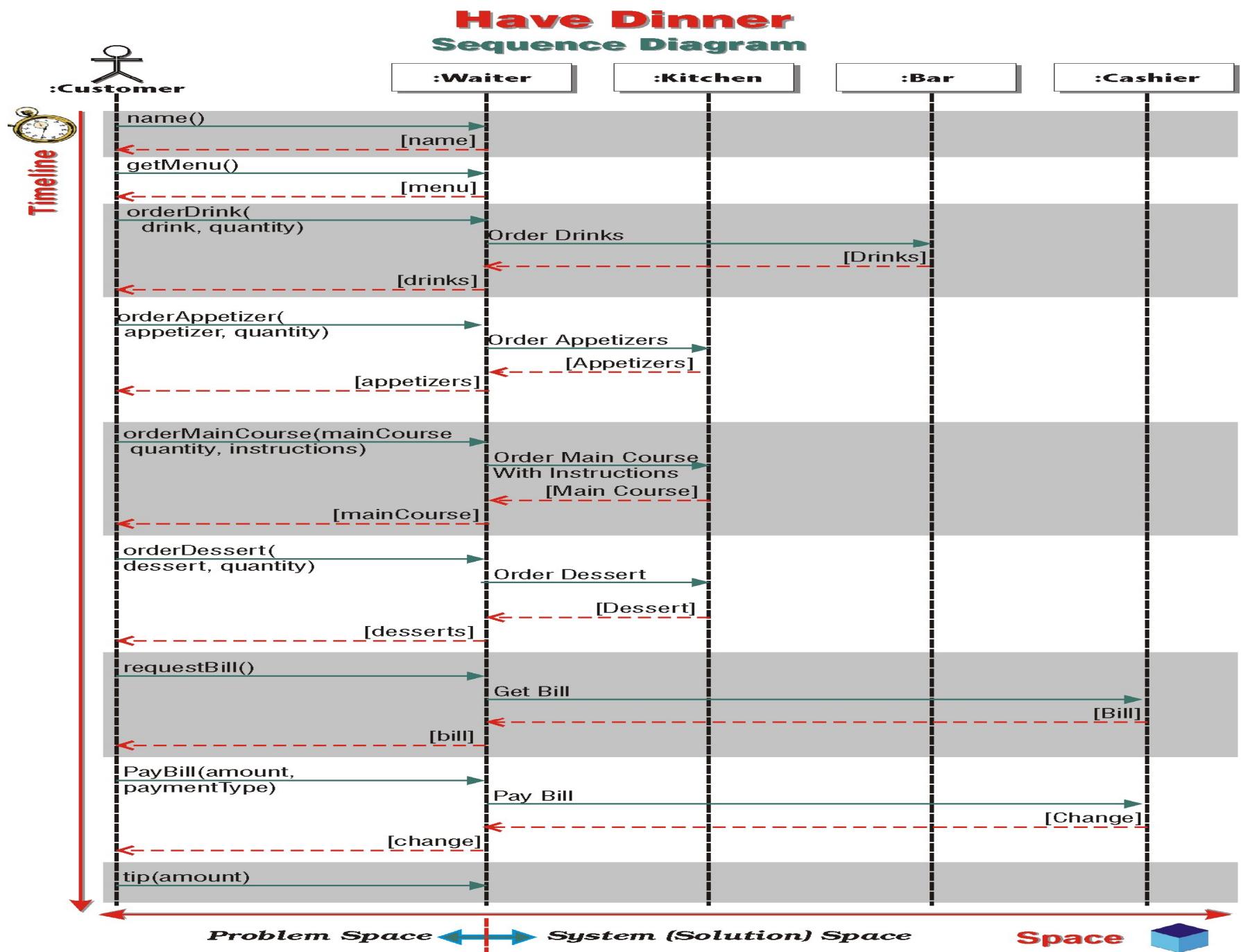


Dynamic Diagrams

- System Sequence diagram
 - Emphasizes the order of interactions with the system (Blackbox) in time.
- Sequence diagram
 - Emphasizes the order of object interactions in time.
- Collaboration diagram
 - Focuses on how the objects are organized and what set of messages they must exchange to satisfy a certain behavior.
- Statechart diagram
 - Traces the results of interactions on the state of objects belonging to a specific class.
- Activity diagram
 - Concentrates on the logical flow of activities.

Sequence Diagram

- Sequence diagram represents the interaction between objects, or between actors and objects ordered in time.
- A sequence diagram is composed of a timeline, objects that interact across this timeline, and the messages that they exchange.



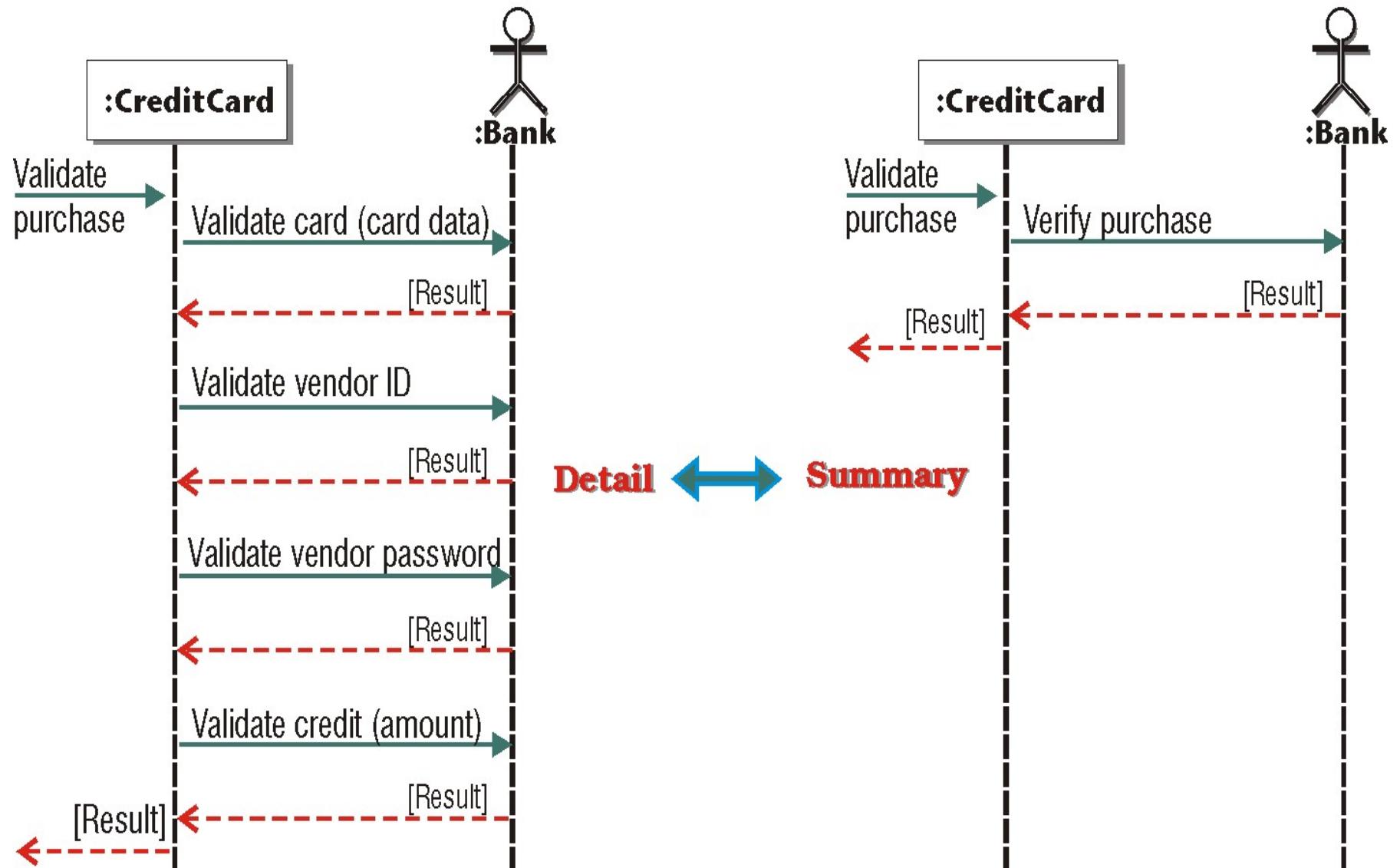
Sequence Diagram

- The **objects** across the top of the diagram are interacting with each other by passing **messages** back and forth (horizontal).
- An actor is the user whose interactions with the system are described in each of the **use cases** (horizontal).
- The **object lifetime** represents time and starts with the top-most message (vertical).

Elements of Sequence Diagram

- Actor Instance: outside the system.
 - **Instance Name:Actor Name**
 - **aCustomer:Customer**
- Class Instance: inside the system
 - **Instance Name:Class Name**
- Timeline---dotted vertical line.
- Object Lifetime: A hollow box on the **timeline** identifies the **lifetime** of an object.
- Instantiation: specifies when the **lifetime** of the instance starts.
- Destruction: specifies when the **lifetime** of the instance **ends** and the object is destroyed.
- Message & Timeline Forking: Specifies **alternates**.
- Message to Self & Looping: Specifies **repeats**.

Summarizing Transactions In Sequence Diagram



Object Lifetime

- The lifetime of an object specifies when its instantiated, how long it exists and when it is destroyed.

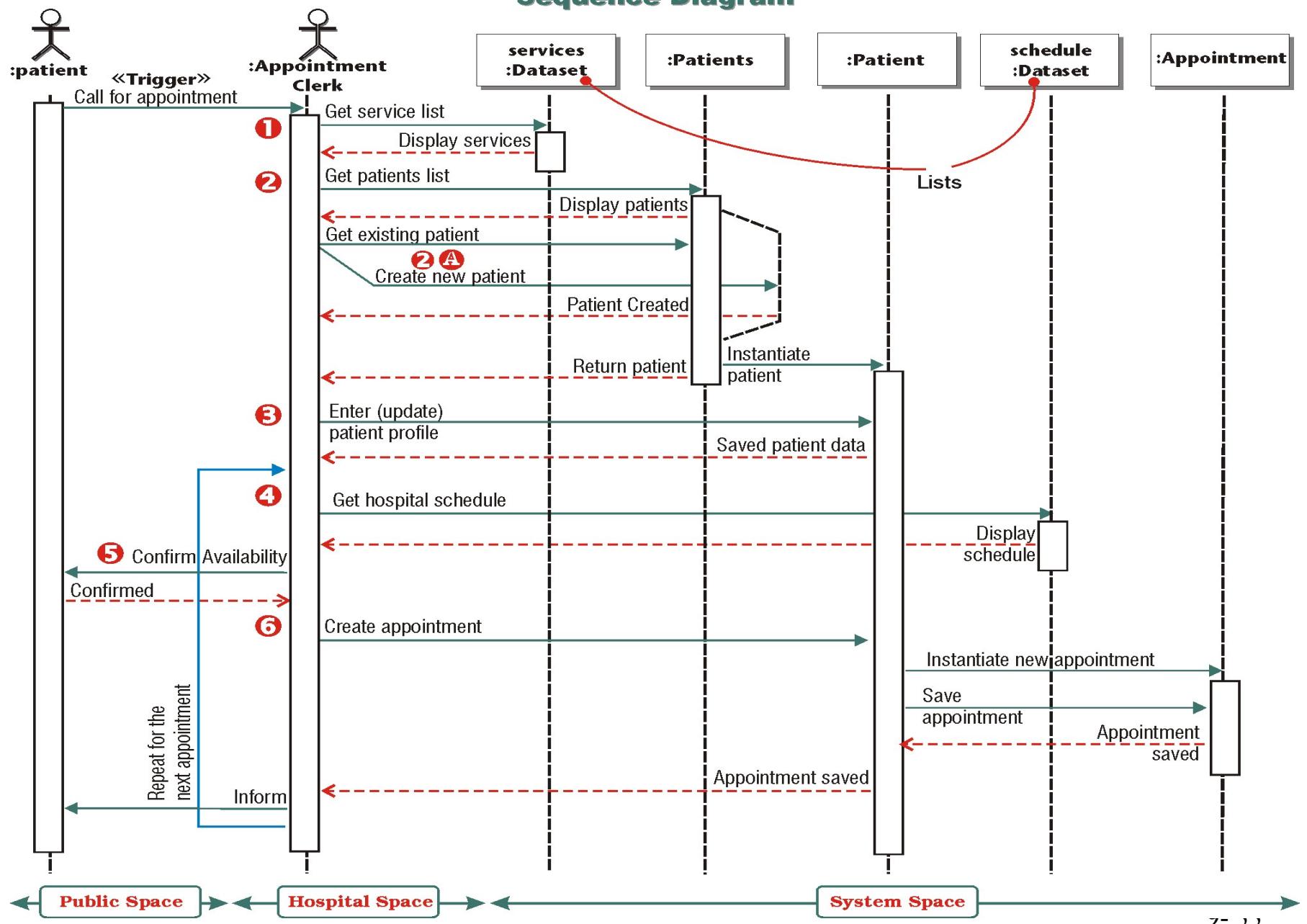
Make Appointment

Use Case

Normal Flow:	<ol style="list-style-type: none">1. Appointment clerk verifies that the needed medical service is provided by the hospital.2. Appointment clerk records patient's personal and contact data.3. Appointment clerk records information about the referral source.4. Appointment clerk consults hospital's schedule to find a free slot for the required medical service.5. Appointment clerk verifies that the patient is available for the appointment. → Loop 1: Repeat steps 4-5 until hospital's schedule matches patient's availability.6. Appointment clerk makes the appointment. → Loop 2: Repeat steps 4-6 for each appointment.
Alternate Flow/ Exceptions:	<p>2.a Patient is not on file. Create new patient. (<u>Extend</u>: 141 - Create Patient.)</p>

Make Appointment

Sequence Diagram



Tips for Drawing Sequence Diagram

- Draw a sequence diagram for every **use case**.
- Paste the text of the use case **scenario** into a note.
- Map your use **case text** to the **messages** being passed on the sequence diagram.
- Line up the text and message **arrows**.
- Include **alternates** and **loops**.

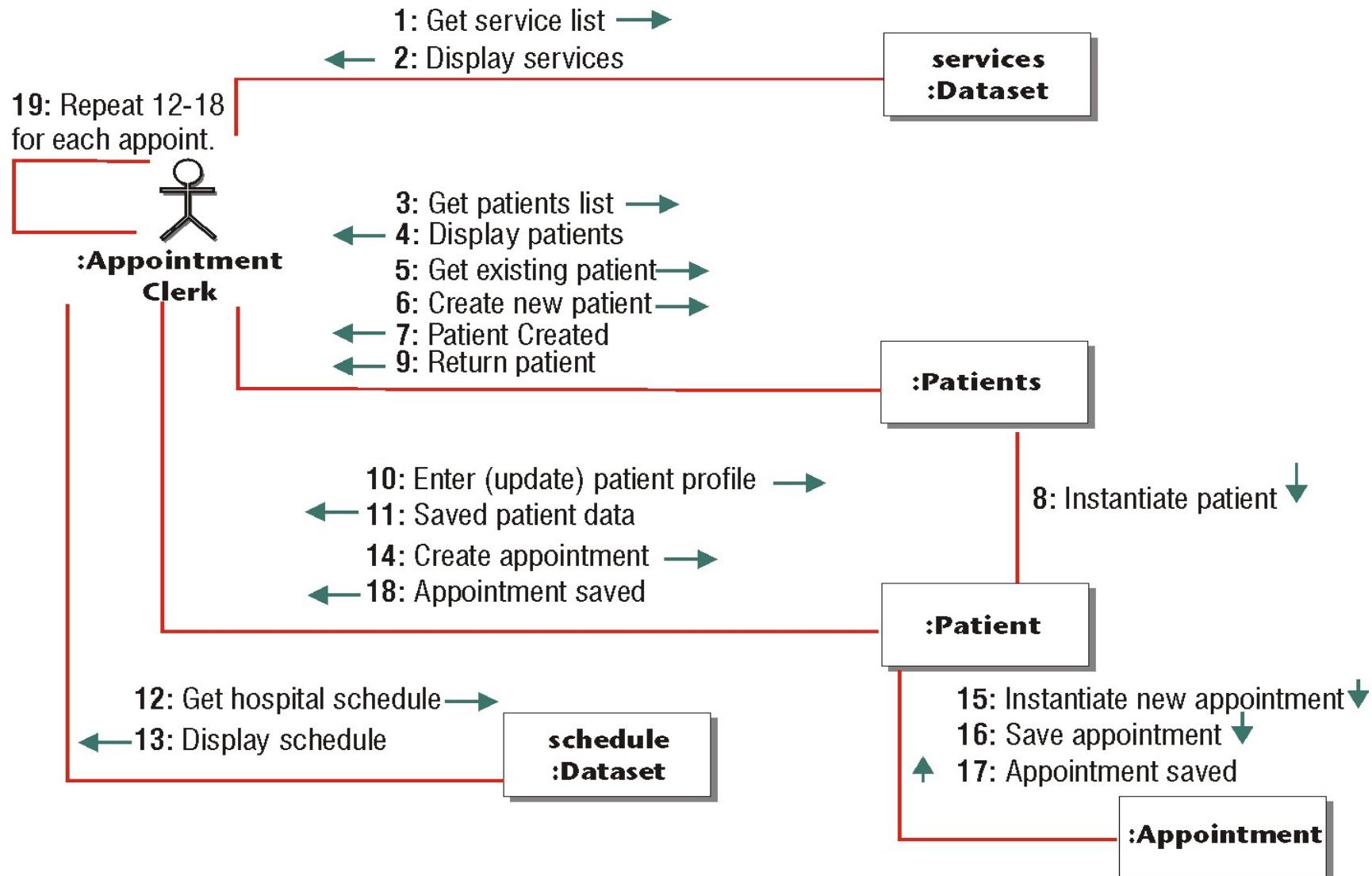
Note: messages, methods, operations represent the behavior of a class.

Collaboration Diagram

- Collaboration diagram is a dynamic diagram that focuses on the organization of objects that must cooperate, by exchanging messages, to satisfy a certain behavior.
- Collaboration = Communication.

Make Appointment

Collaboration Diagram

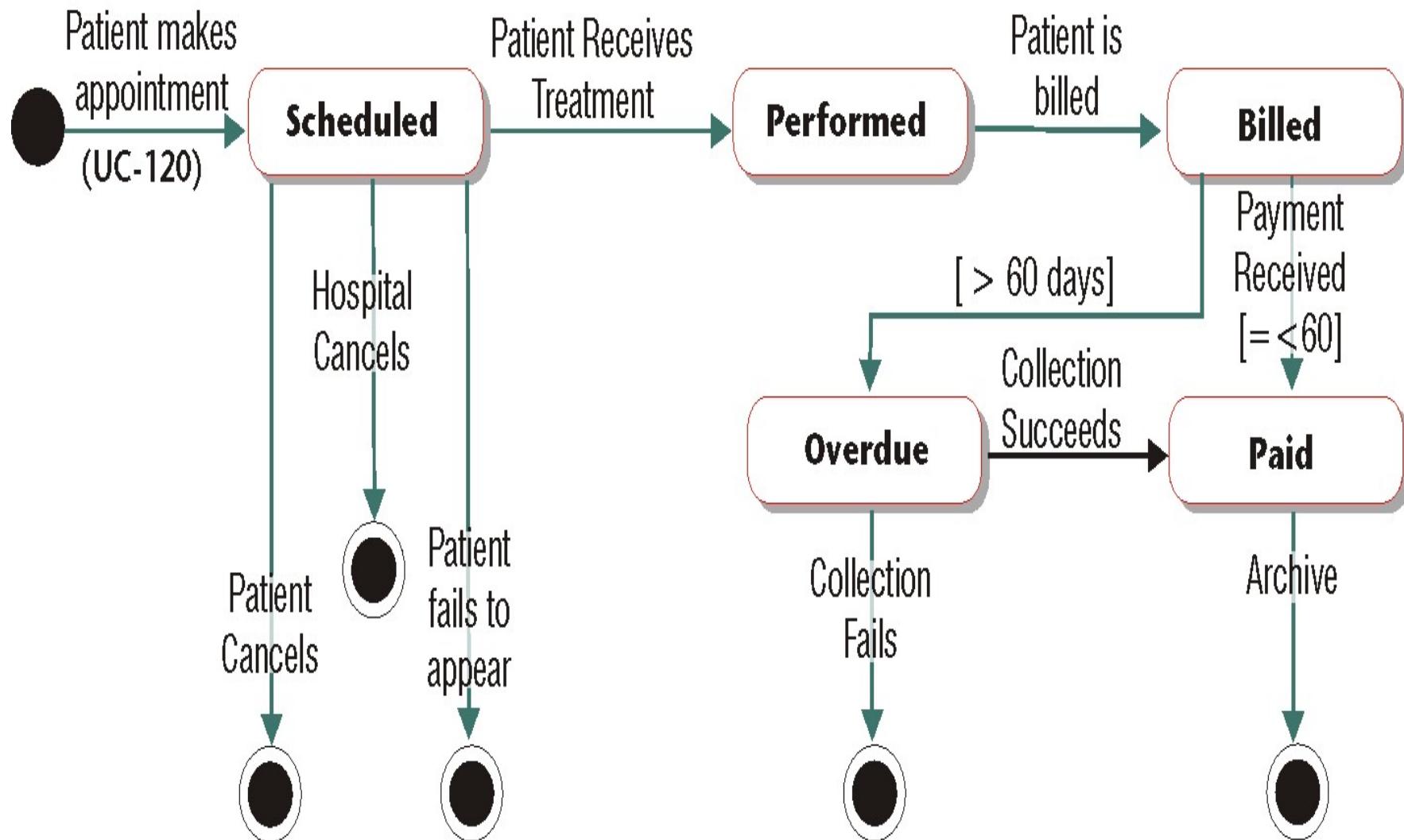


Statechart Diagram

- Statechart diagram represents **milestones** in the lifetime of an object when its state changes.
- A state is an object's **condition** at a certain stage and from a certain viewpoint. It is a snapshot of the object in a usually important point in time.
- Statechart diagram is composed of the states of an object and the flow of events that change its state.

Treatment

Statechart Diagram for an Object



Elements of Statechart Diagram

- Initial State (Starting Point)
- State
- Transition & Event
- Final State (Termination Point)

The Value of Statechart Diagram

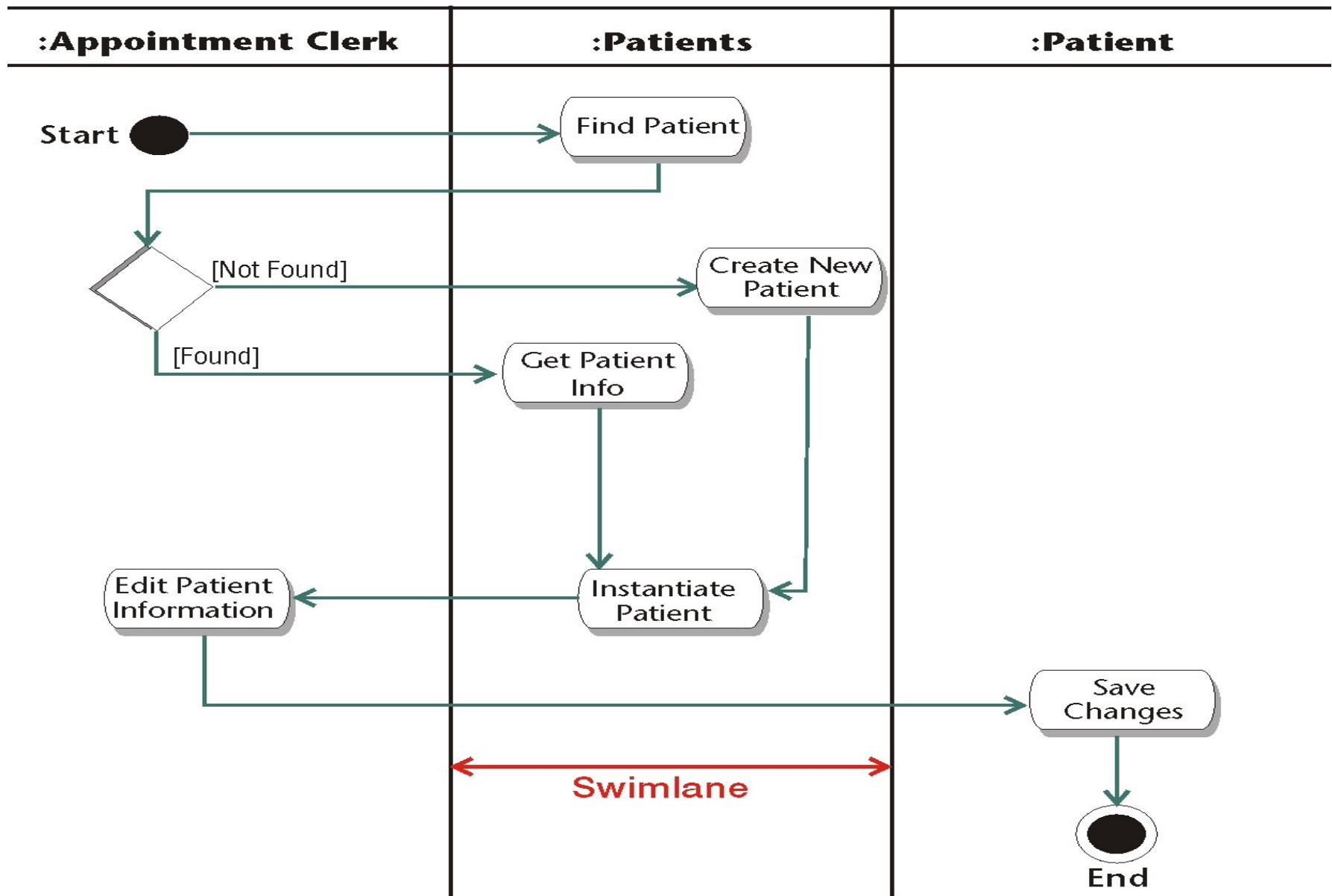
- Statechart diagram is the only dynamic model that can illustrate the milestones in the lifetime of *one* class of objects in its entirety.

Activity Diagram

- Activity diagram provides the most lucid tool for modeling the *logical* flow of activities that takes place between the system and the outside world or within the system among its components.
- Activity diagram is concerned with **the logical flow of activities** in an almost pure form.

Enter Patient Data

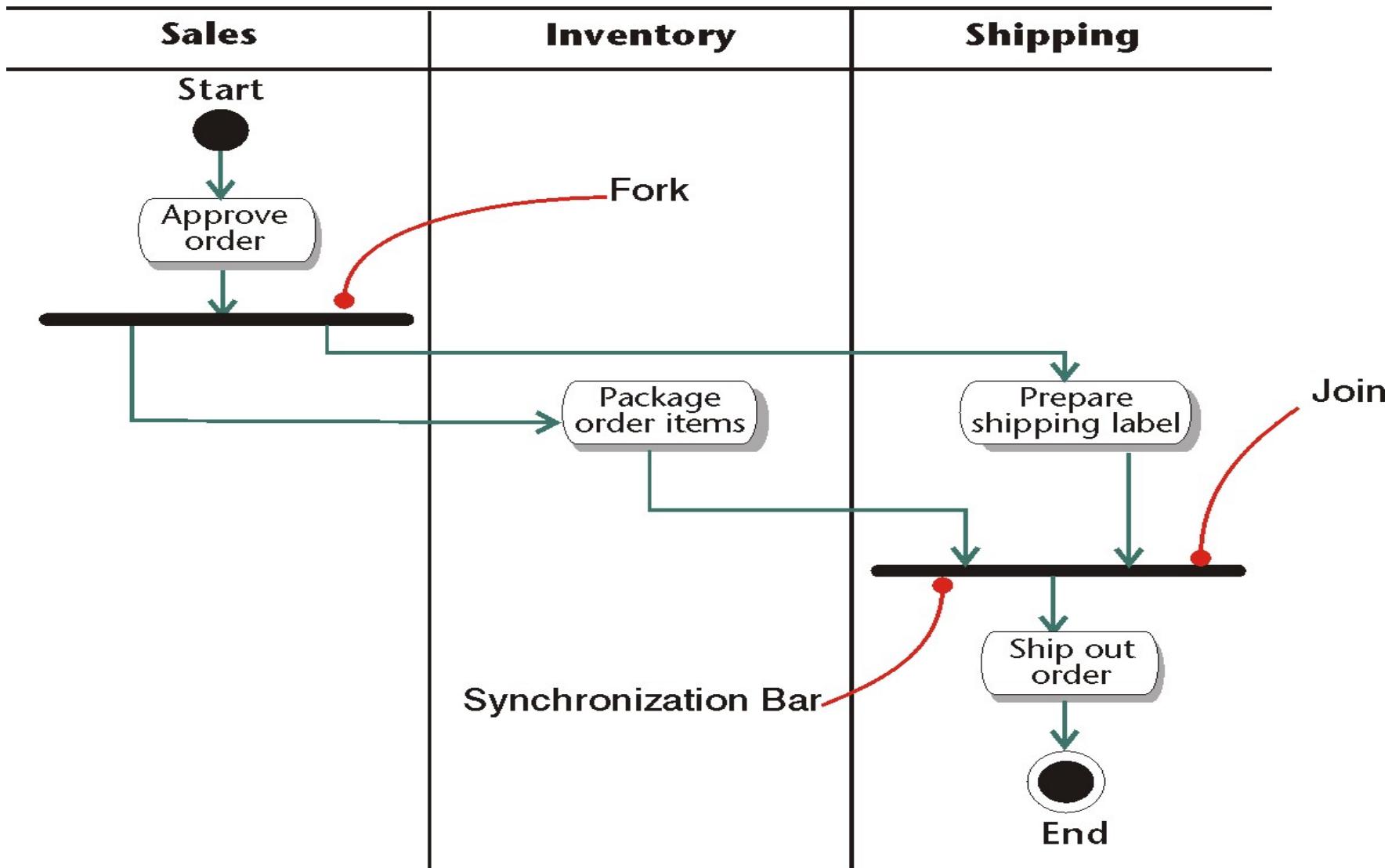
Activity Diagram



Ship Order

Activity Diagram

For Workflow Modeling



Activity Diagram

detailed discussions

Activity Diagrams

- **Activity Diagrams:**

- Unobtrusively observe business processes
- Diagram all information gathered
- Sample diagram: representation of workflow
 - Identify agents to create the appropriate swimlanes
 - Represent steps of workflow with appropriate ovals
 - Connect activity ovals with arrows to show direction
 - Use decision symbol to represent either/or situation
 - Use synchronization bars for parallel paths



Rocky Mountain Outfitters—Customer Order Form

Name and address of person placing order.
(Please verify your mailing address and make correction below.)
Order Date / /

Name _____

Address _____ Apt. No _____

City _____ State _____ Zip _____

Phone: Day () _____ Evening () _____

Gift Order or Ship To: (Use only if different from address at left.)

Name _____

Address _____ Apt. No _____

City _____ State _____ Zip _____

Gift Address for this Shipment Only Permanent Change of Address

Gift Card Message _____

Delivery Phone () _____

Item No.	Description	Style	Color	Size	Sleeve Length	Qty	Monogram	Style	Price Each	Total

Method of Payment

Check/Money Order Gift Certificate(s) AMOUNT ENCLOSED \$ _____

American Express MasterCard VISA Other

Account Number

--	--	--	--	--	--	--	--	--	--	--

MO / YR
Expiration Date

MERCHANDISE TOTAL _____

Regular FedEx shipping \$4.50 per U.S. delivery address
(Items are sent within 24 hours for delivery in 2 to 4 days)

Please add \$4.50 per each additional U.S. delivery address

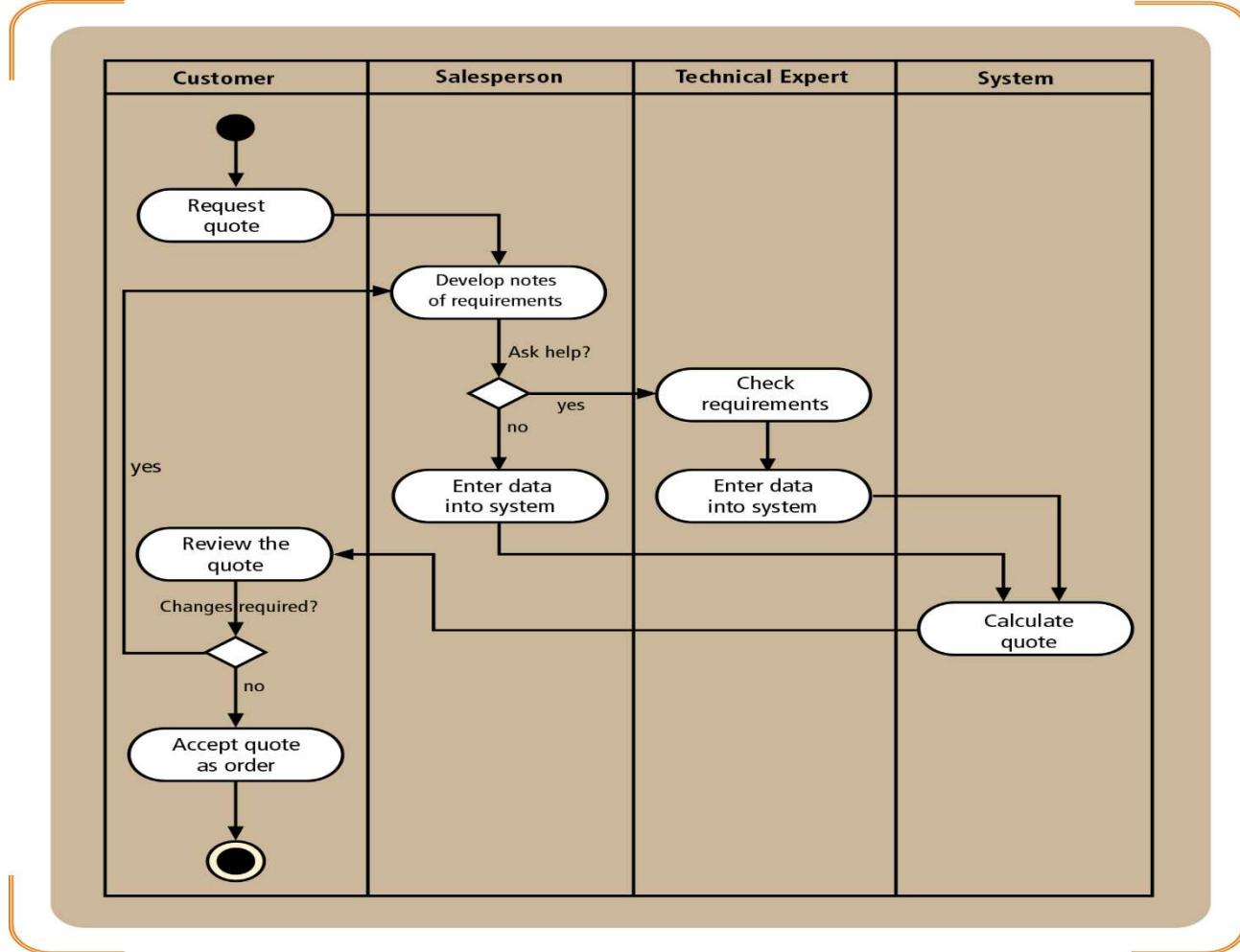
FedEx Standard Overnight Service

Any additional freight charges

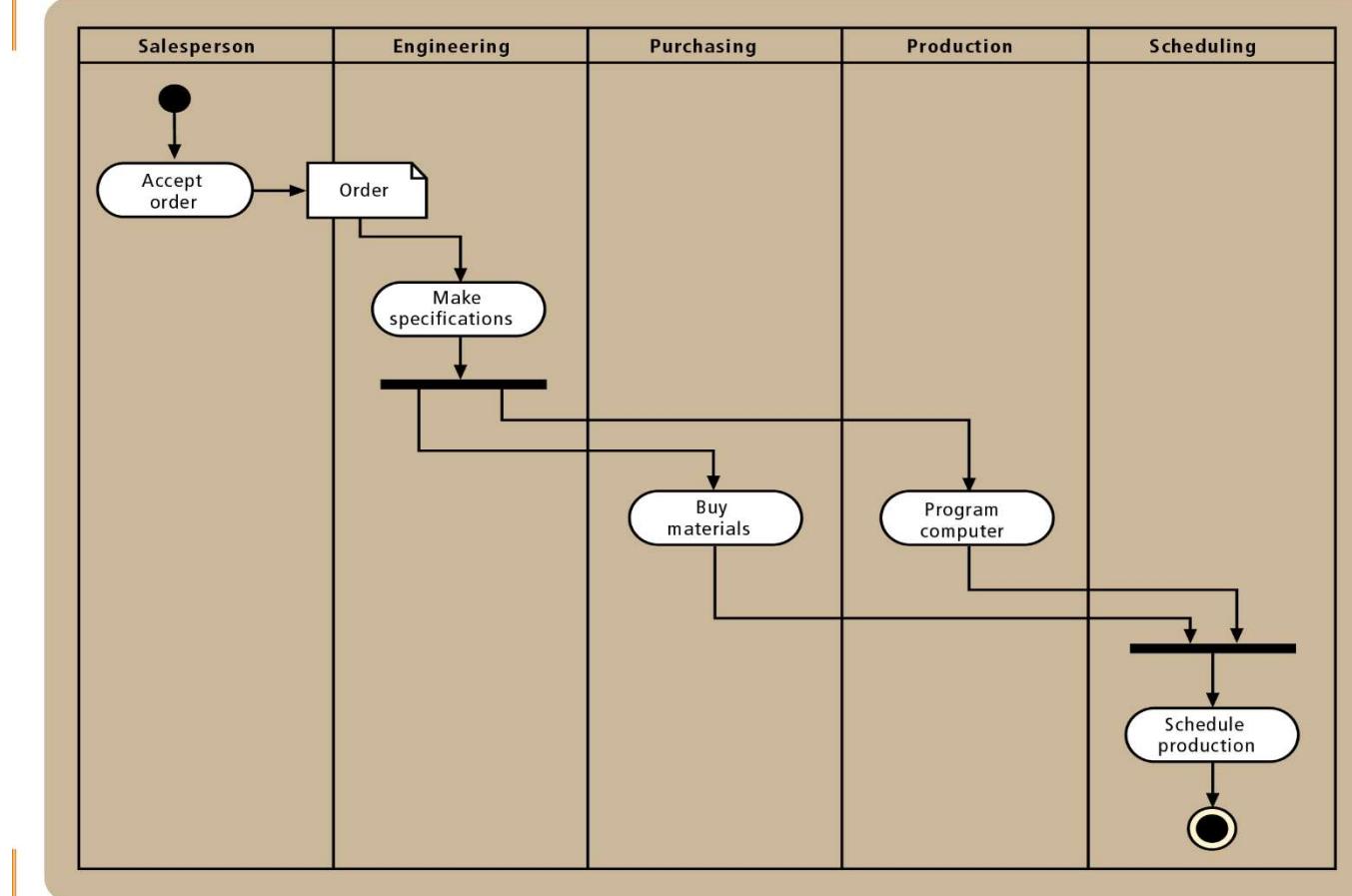
International Shipping (see shipping information on back)

Signature

A Sample Order Form for Rocky Mountain Outfitters



A Simple Activity Diagram to Demonstrate a Workflow



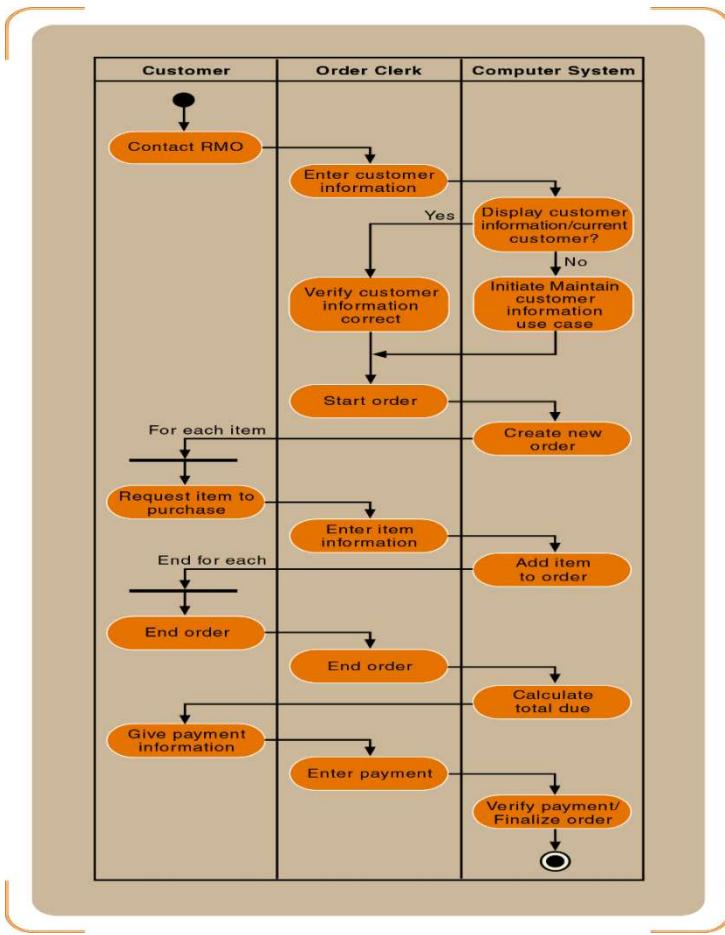
An Activity Diagram Showing Concurrent Paths

Avtivity Diagrams

- Activity Diagram Description
 - Document the workflows of business processes
 - Document flow of activities for use case scenarios
 - Form basis of system sequence diagrams (SSDs)

Use Case Name:	Create new order	
Scenario:	Create new telephone order	
Triggering Event:	Customer telephones RMO to purchase items from the catalog.	
Brief Description:	When customer calls to order, the order clerk and system verify customer information, create a new order, add items to the order, verify payment, create the order transaction, and finalize the order.	
Actors:	Telephone sales clerk	
Related Use Cases:	Includes: <i>Check item availability</i>	
Stakeholders:	Sales department: to provide primary definition Shipping department: to verify that information content is adequate for fulfillment Marketing department: to collect customer statistics for studies of buying patterns	
Preconditions:	Customer must exist. Catalog, Products, and Inventory items must exist for requested items.	
Postconditions:	Order and order line items must be created. Order transaction must be created for the order payment. Inventory items must have the quantity on hand updated. The order must be related (associated) to a customer.	
Flow of Events:	Actor 1. Sales clerk answers telephone and connects to a customer. 2. Clerk verifies customer information. 3. Clerk initiates the creation of a new order. 4. Customer requests an item be added to the order. 5. Clerk verifies the item (<i>Check item availability</i> use case). 6. Clerk adds item to the order. 7. Repeat steps 4, 5, and 6 until all items are added to the order. 8. Customer indicates end of order; clerk enters end of order. 9. Customer submits payment; clerk enters amount.	System 3.1 Create a new order. 5.1 Display item information. 6.1 Add an order item. 8.1 Complete order. 8.2 Compute totals. 9.1 Verify payment. 9.2 Create order transaction. 9.3 Finalize order.
Exception Conditions:	2.1 If customer does not exist, then the clerk pauses this use case and invokes <i>Maintain customer information</i> use case. 2.2 If customer has a credit hold, then clerk transfers the customer to a customer service representative. 4.1 If an item is not in stock, then customer can a. choose not to purchase item, or b. request item be added as a back-ordered item. 9.1 If customer payment is rejected due to bad-credit verification, then a. order is canceled, or b. order is put on hold until check is received.	

Fully Developed Description of Telephone Order Scenario for Create New Order Use Case

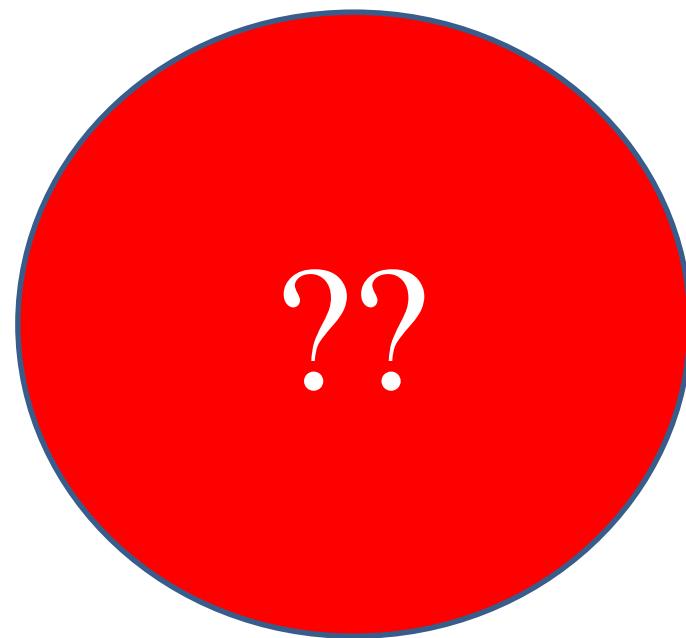


Activity Diagram of the Telephone Order Scenario

Case Study

- Develop the activity diagram based on the following narrative.
- The purpose of the Open Access Insurance System is to provide automotive insurance to car owners. Initially, prospective customers fill out an insurance application, which provides information about the customer and his or her vehicles. This information is sent to an agent, who sends it to various insurance companies to get quotes for insurance. When the responses return, the agent then determines the best policy for the type and level of coverage desired and gives the customer a copy of the insurance policy proposal and quote.

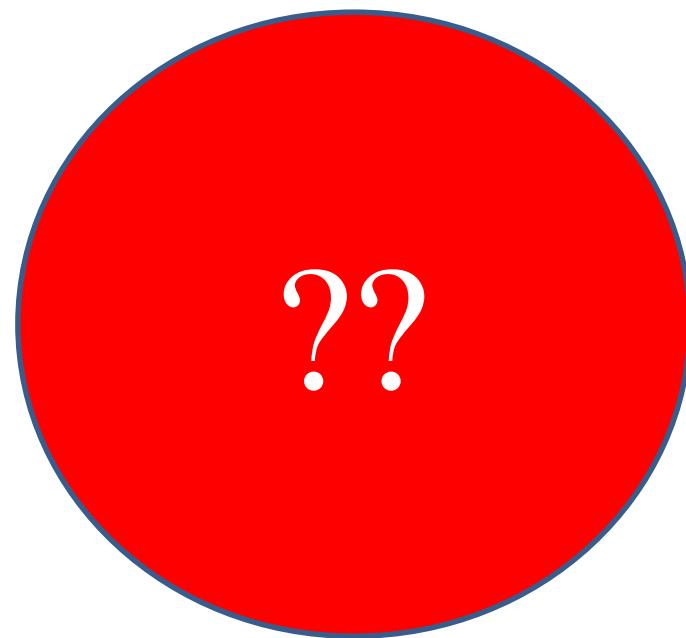
- Develop the activity diagram



Case Study

- Develop the activity diagram based on the following narrative.
- The purchasing department handles purchase requests from other departments in the company. People in the company who initiate the original purchase request are the "customers" of the purchasing department. A case worker within the purchasing department receives that request and monitors it until it is ordered and received. Case workers process requests for the purchase of products under \$1,500, write a purchase order, and then send it to the approved vendor. Purchase requests over \$1,500 must first be sent out for bid from the vendor that supplies the product. When the bids return, the case worker selects one bid. Then, he or she writes a purchase order and sends it to the vendor.

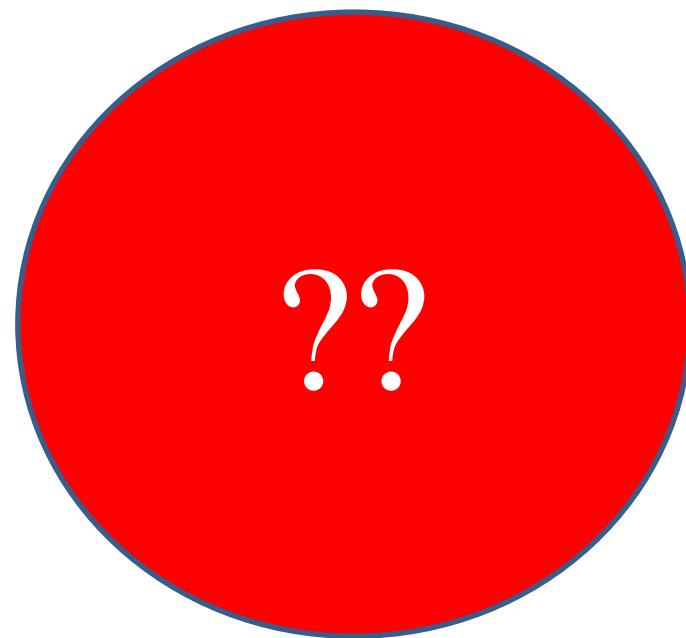
- Develop the activity diagram



Case Study

- Develop the activity diagram to cancel an order for the OutdoorPowerEquipmentDepot.

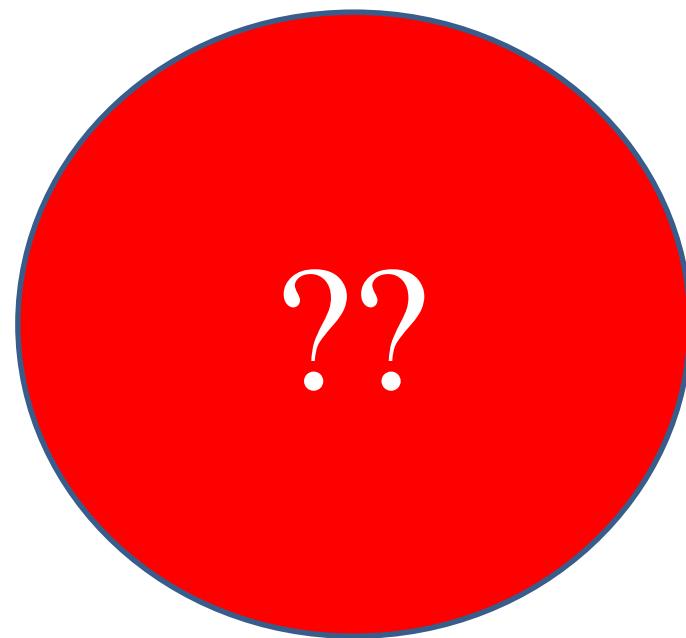
- Develop the activity diagram



Case Study

- Develop the activity diagram to place an order for the OutdoorPowerEquipmentDepot.

- Develop the activity diagram



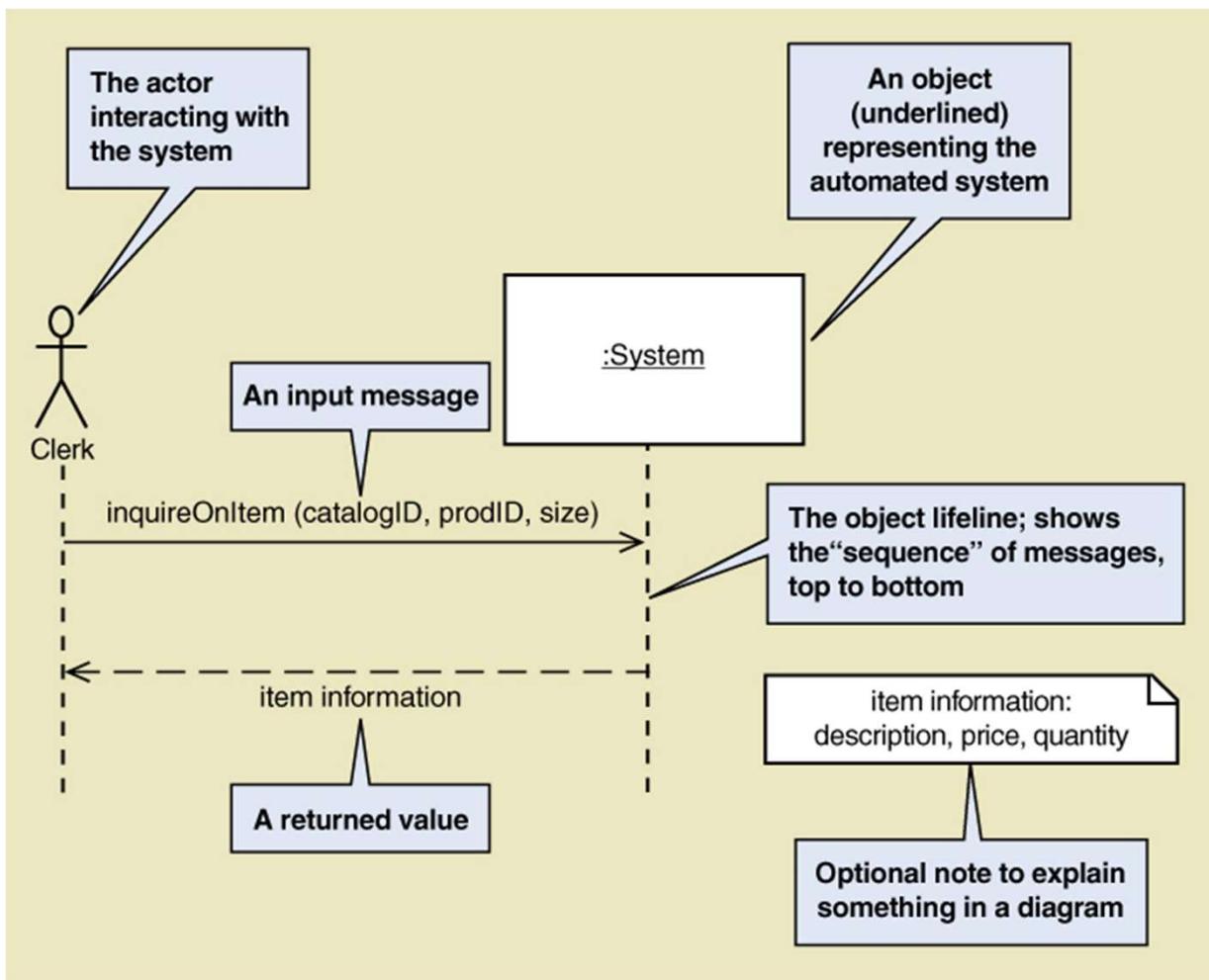
System Sequence Diagram (SSD)

detailed discussions

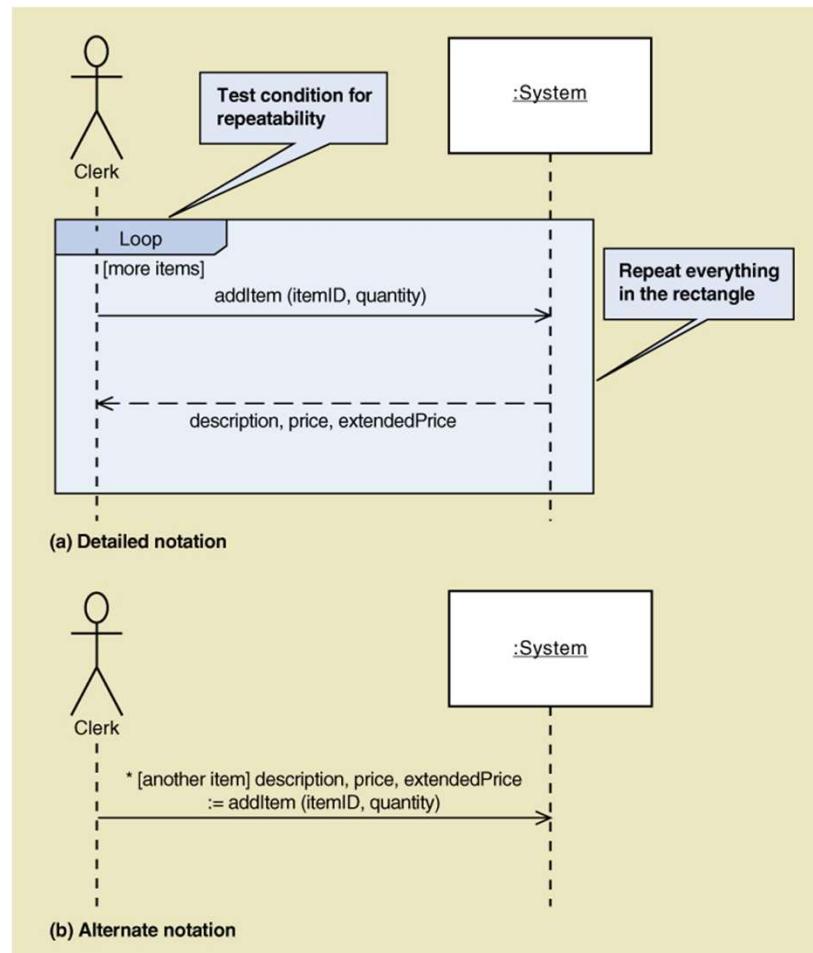
System Sequence Diagram (SSD)

- A UML sequence diagram
- Special case for a sequence diagram
 - Only shows actor and one object
 - The one object represents the complete system
 - Shows input & output messaging requirements for a use case
- Actor, :System, object lifeline
- Messages

System Sequence Diagram (SSD) Notation



SSD Message Examples with Loop Frame

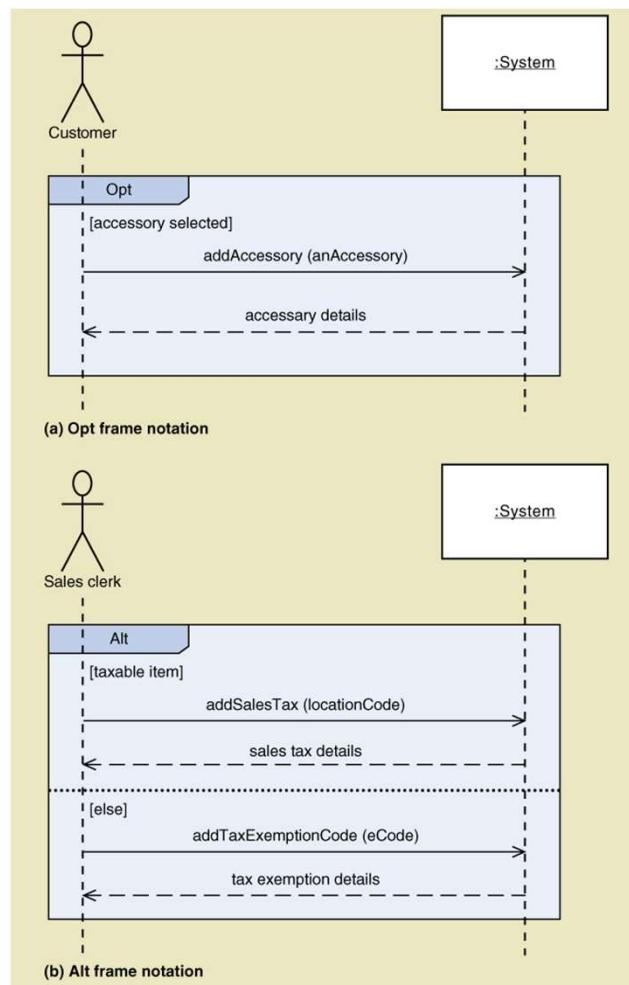


Message Notation for SSD

- *[true/false condition] return-value := message-name (parameter-list)*
 - An asterisk (*) indicates repeating or looping of the message
 - Brackets [] indicate a true/false condition. This is a test for that message only. If it evaluates to true, the message is sent. If it evaluates to false, the message isn't sent.
 - Message-name is the description of the requested service written as a verb-noun.
 - Parameter-list (with parentheses on initiating messages and without parentheses on return messages) shows the data that are passed with the message.
 - Return-value on the same line as the message (requires :=) is used to describe data being returned from the destination object to the source object in response to the message.

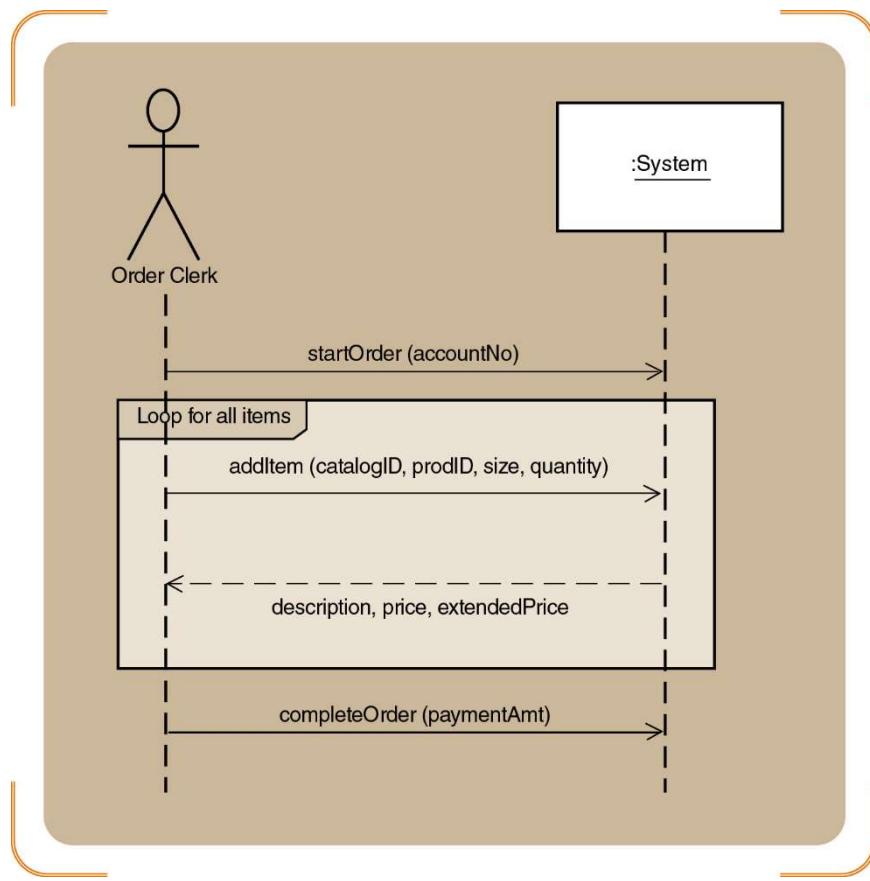
SSD Message Examples

- Opt Frame (optional)
- Alt Frame (if-else)



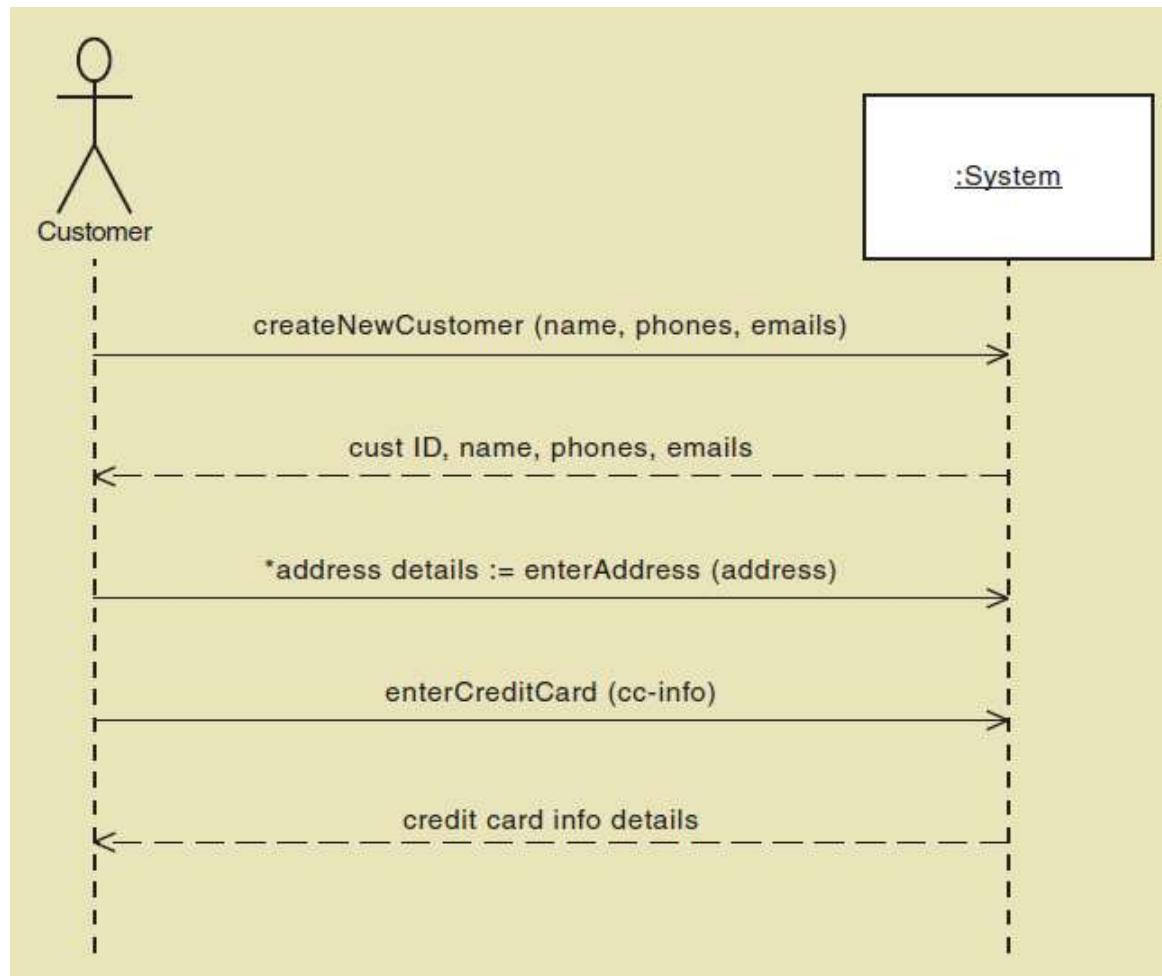
Steps for Developing SSD

- Begin with detailed description of use case
 - Fully developed form
 - Activity diagrams
- (4) step process for turning activity diagram into SSD
 - [1] Identify the input messages
 - [2] Describe messages from external actor to system
 - [3] Identify/apply special conditions to input messages
 - Iteration/loop frame
 - Opt or Alt frame
 - [4] Identify and add the output return messages

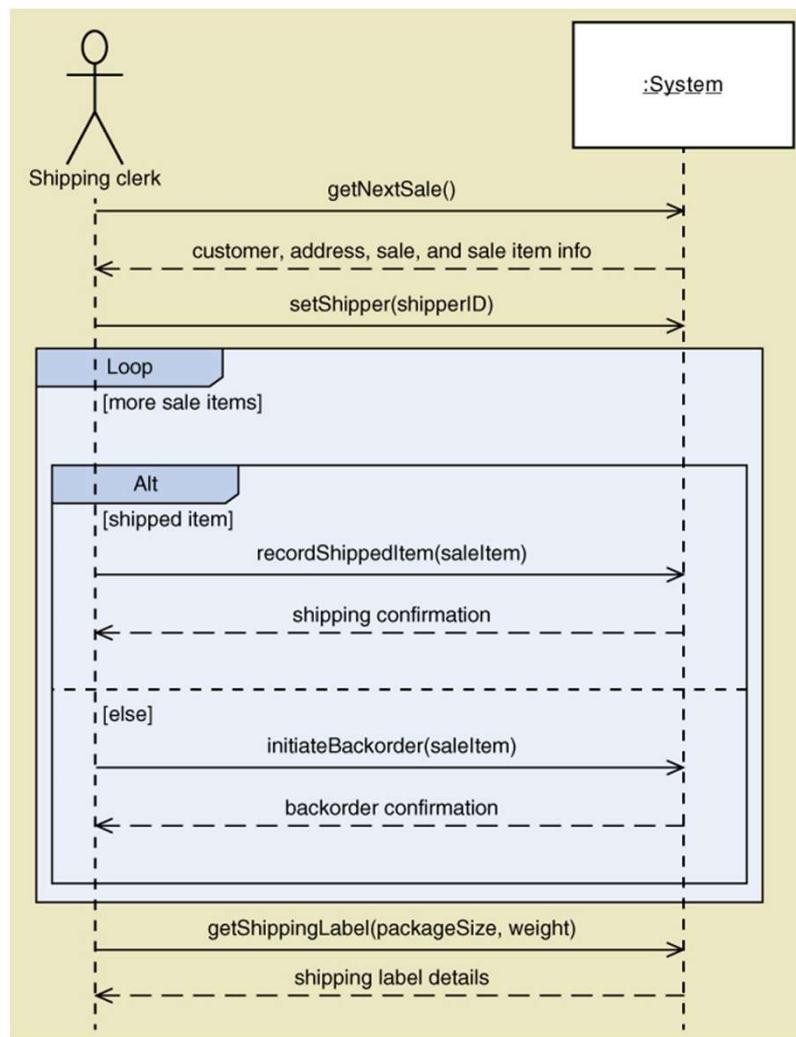


An SSD of the Simplified Telephone Order Scenario for the Create New Order Use Case

SSD for *Create customer account* Use case

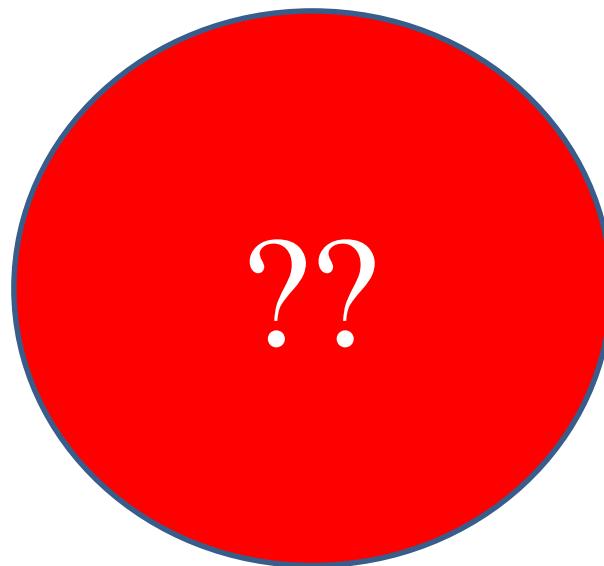


SSD for *Ship items* Use Case



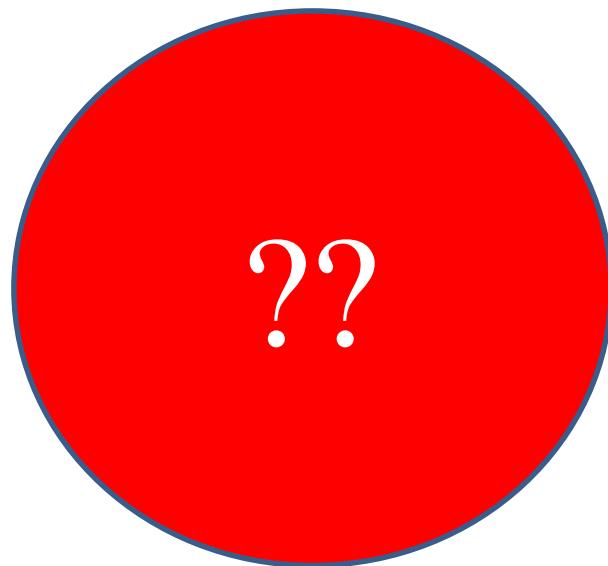
OutdoorPowerEquipmentDepot – Case Study

- Create System Sequence Diagram for Manager to Add Equipment



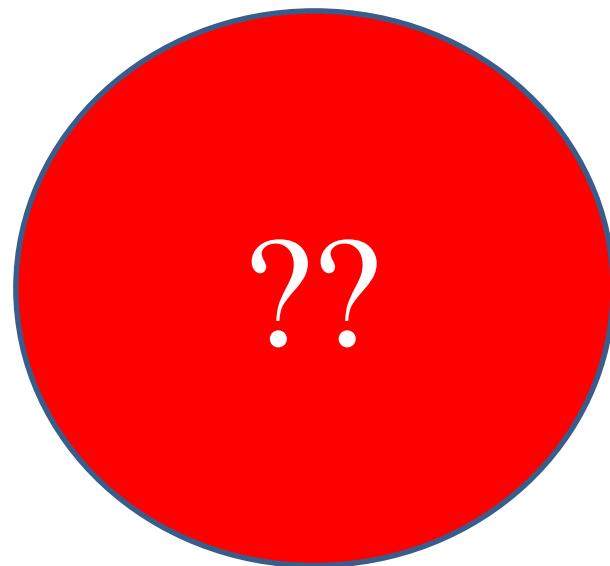
OutdoorPowerEquipmentDepot – Case Study

- Create System Sequence Diagram for Manager to Update Equipment



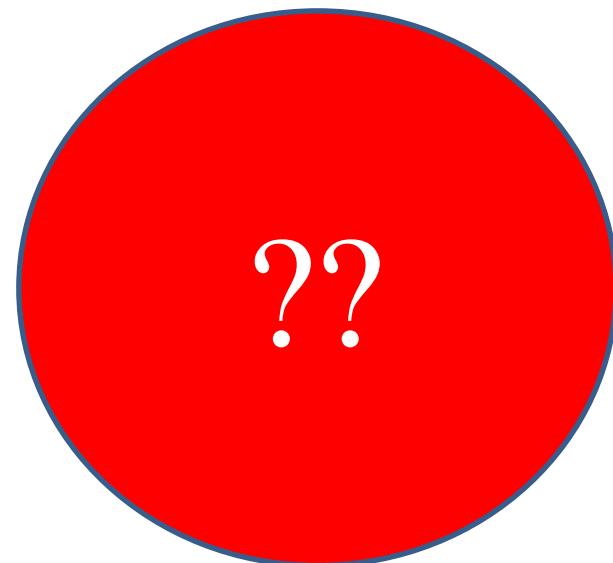
OutdoorPowerEquipmentDepot – Case Study

- Create System Sequence Diagram for Customer to become Member



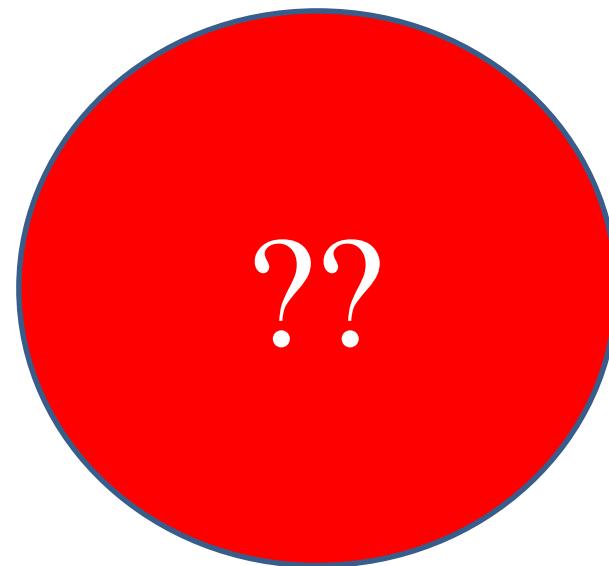
OutdoorPowerEquipmentDepot – Case Study

- Create System Sequence Diagram for Customer to Buy Equipment



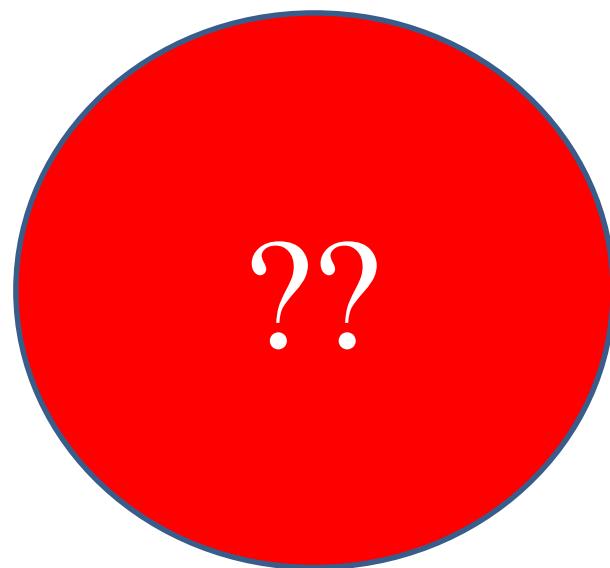
OutdoorPowerEquipmentDepot – Case Study

- Create System Sequence Diagram for Customer to Cancel Order



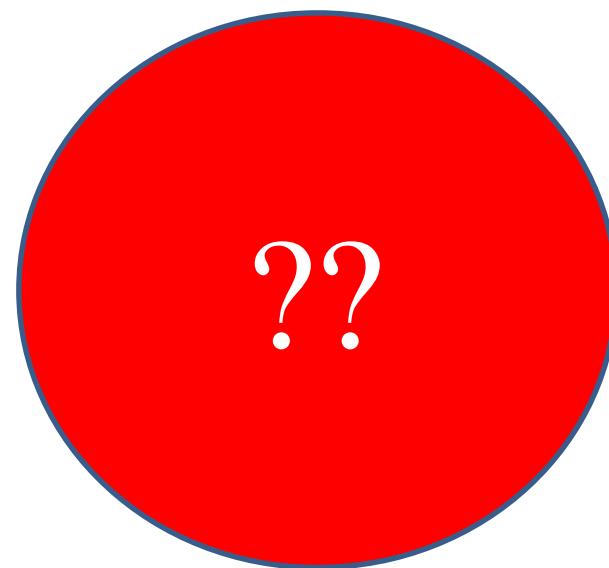
OutdoorPowerEquipmentDepot – Case Study

- Create System Sequence Diagram for Customer to Check status for Order



OutdoorPowerEquipmentDepot – Case Study

- Create System Sequence Diagram for Customer to Register and Create Login



OutdoorPowerEquipmentDepot – Case Study

- Create System Sequence Diagram for Customer to Rent Equipment

