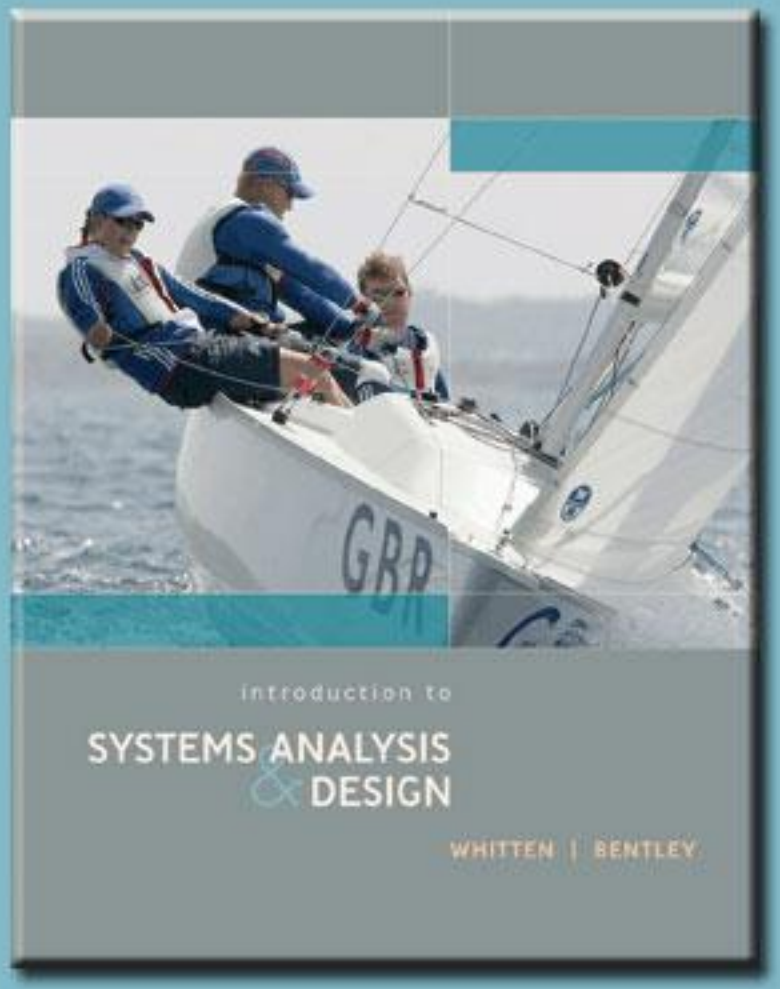


Chapter 9

Object-Oriented Analysis and Modeling Using the UML

Objectives

- Define object modeling and explain its benefits.
- Recognize and understand the basic concepts and constructs of object modeling.
- Define the UML and its various types of diagrams.
- Evolve a business requirements use-case model into a system analysis use-case model.
- Construct an activity diagram.
- Discover objects and classes, and their relationships.
- Construct a class diagram.

Introduction to Object Modeling

Object-oriented analysis (OOA) – an approach used to

1. study existing objects to see if they can be reused or adapted for new uses
2. define new or modified objects that will be combined with existing objects into a useful business computing application

Object modeling – a technique for identifying objects within the systems environment and the relationships between those objects.

Introduction to the UML

Unified Modeling Language (UML) – a set of modeling conventions that is used to specify or describe a software system in terms of objects.

- The UML does not prescribe a method for developing systems—only a notation that is now widely accepted as a standard for object modeling.

Objects & Attributes

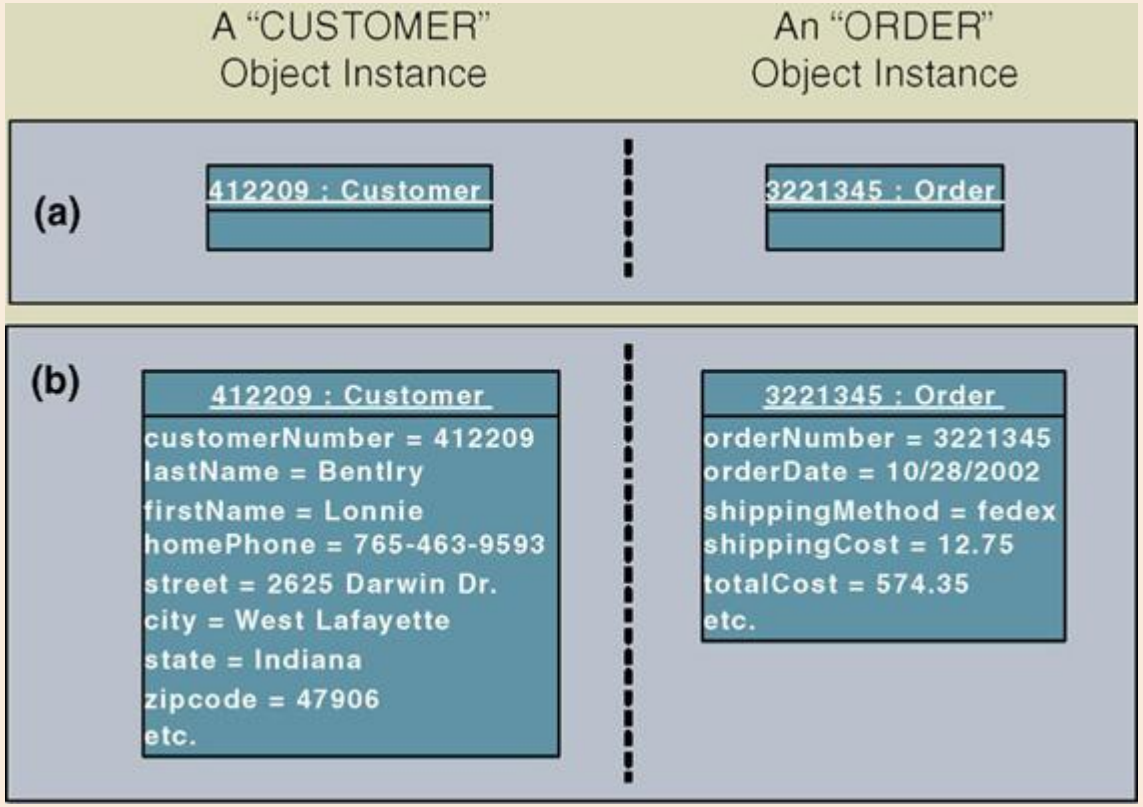
Object – something that is or is capable of being seen, touched, or otherwise sensed, and about which users store data and associate behavior.

- Person, place, thing, or event
- Employee, customer, instructor, student
- Warehouse, office, building, room
- Product, vehicle, computer, videotape

Attribute – the data that represent characteristics of interest about an object.

Objects & Object Instances

Object instance – each specific person, place, thing, or event, as well as the values for the attributes of that object.



Behavior & Encapsulation

Behavior – the set of things that the object can do that correspond to functions that act on the object's data (or attributes).

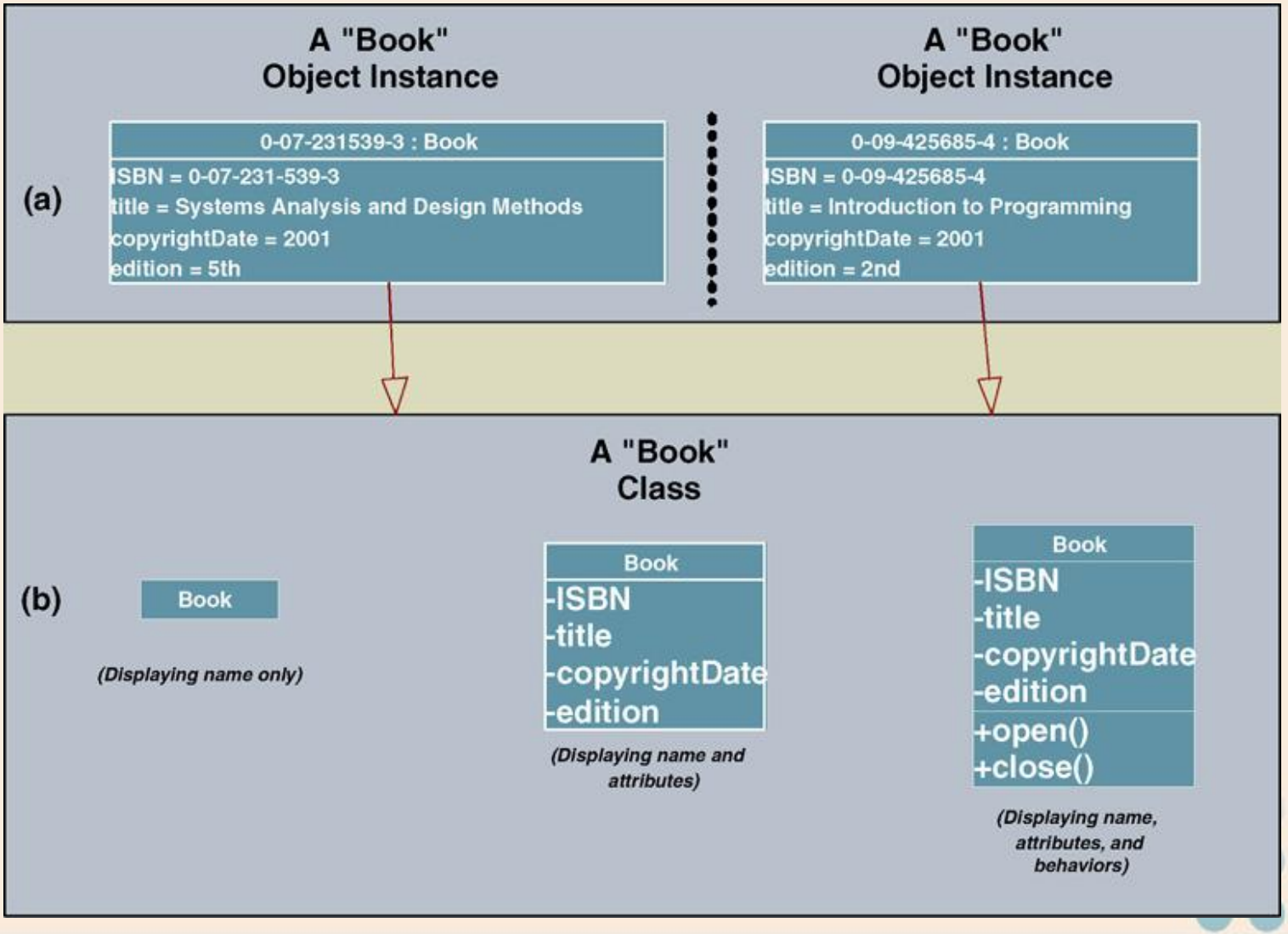
- In object-oriented circles, an object's behavior is commonly referred to as a *method*, *operation*, or *service*.

Encapsulation – the packaging of several items together into one unit.

Object Classes

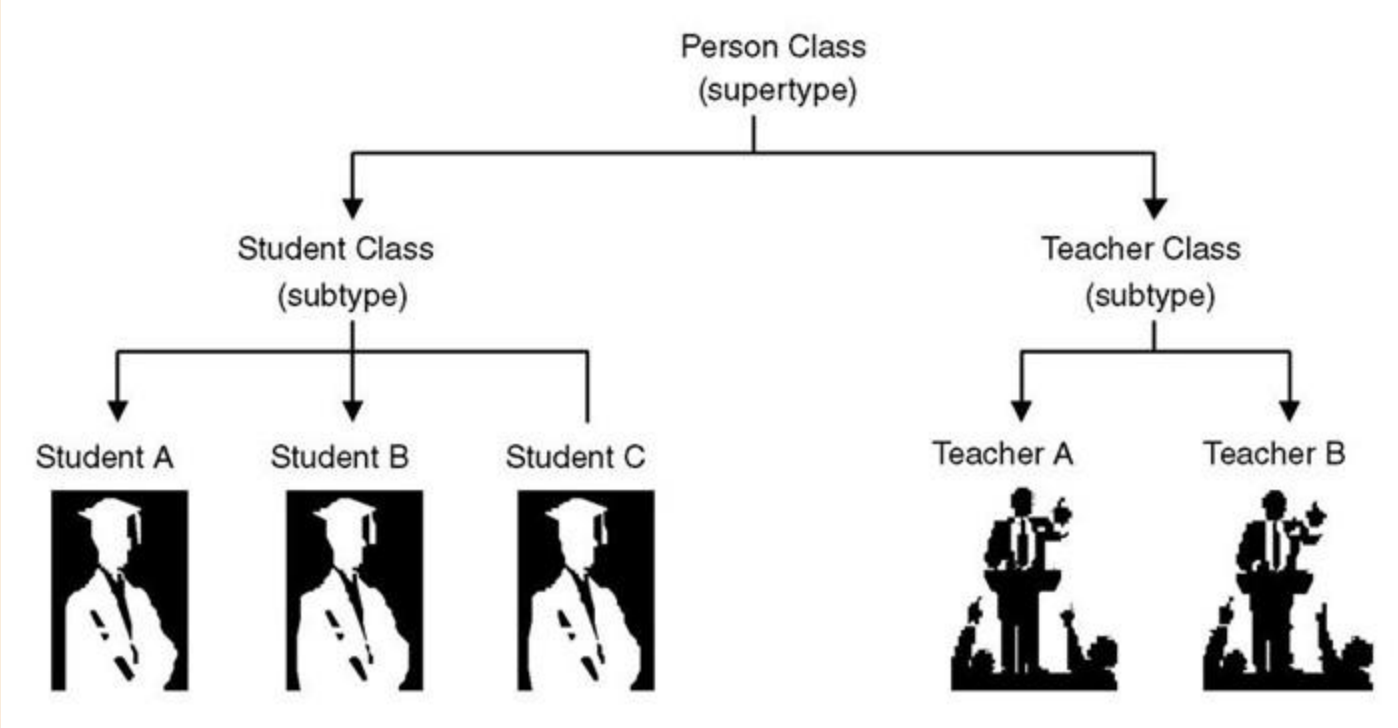
Object Class – a set of objects that share common attributes and behavior. Sometimes referred to as a *class*.

Representing Object Classes in the UML

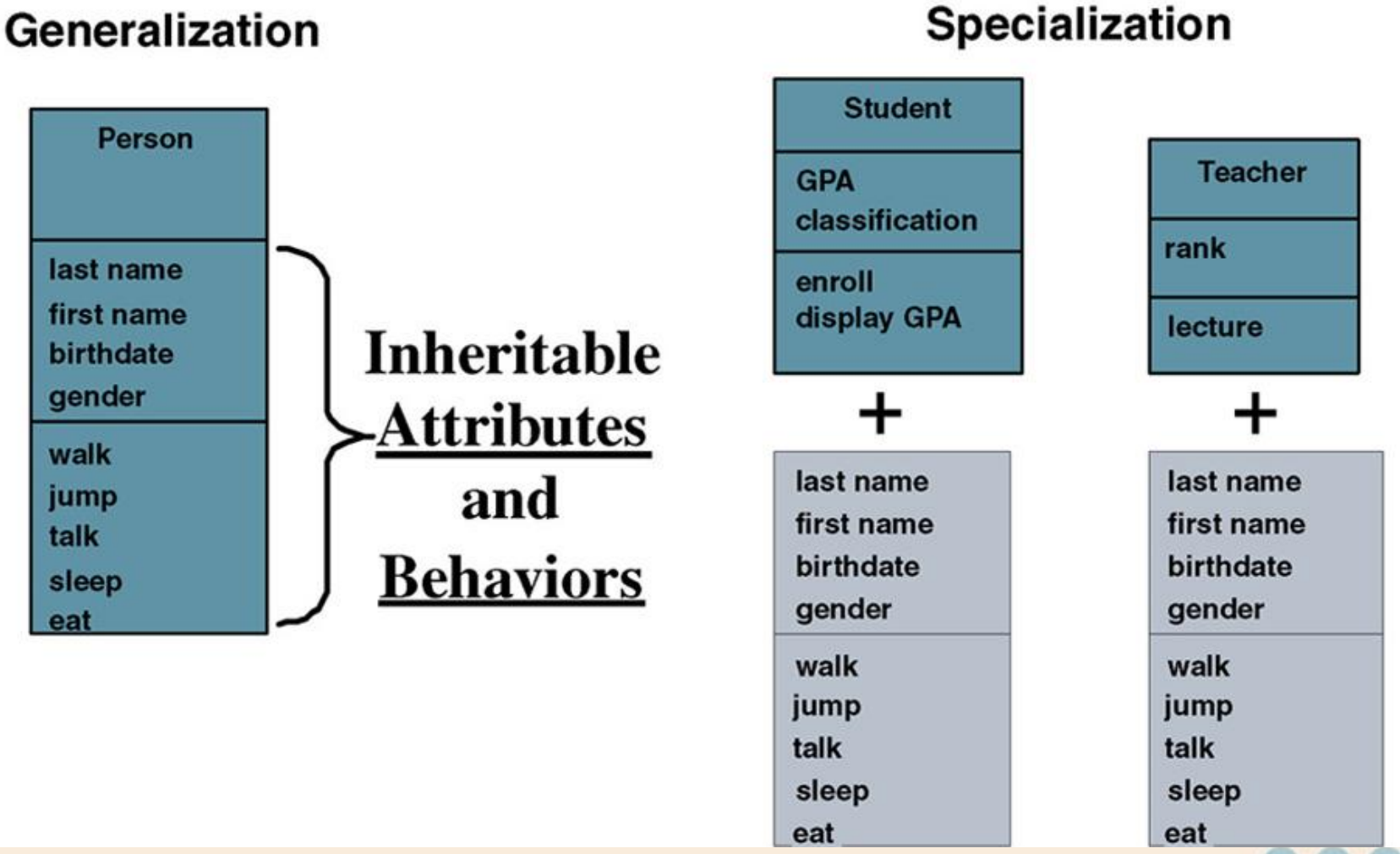


Inheritance

Inheritance – the concept wherein methods and/or attributes defined in an object class can be inherited or reused by another object class.



Inheritance (cont.)



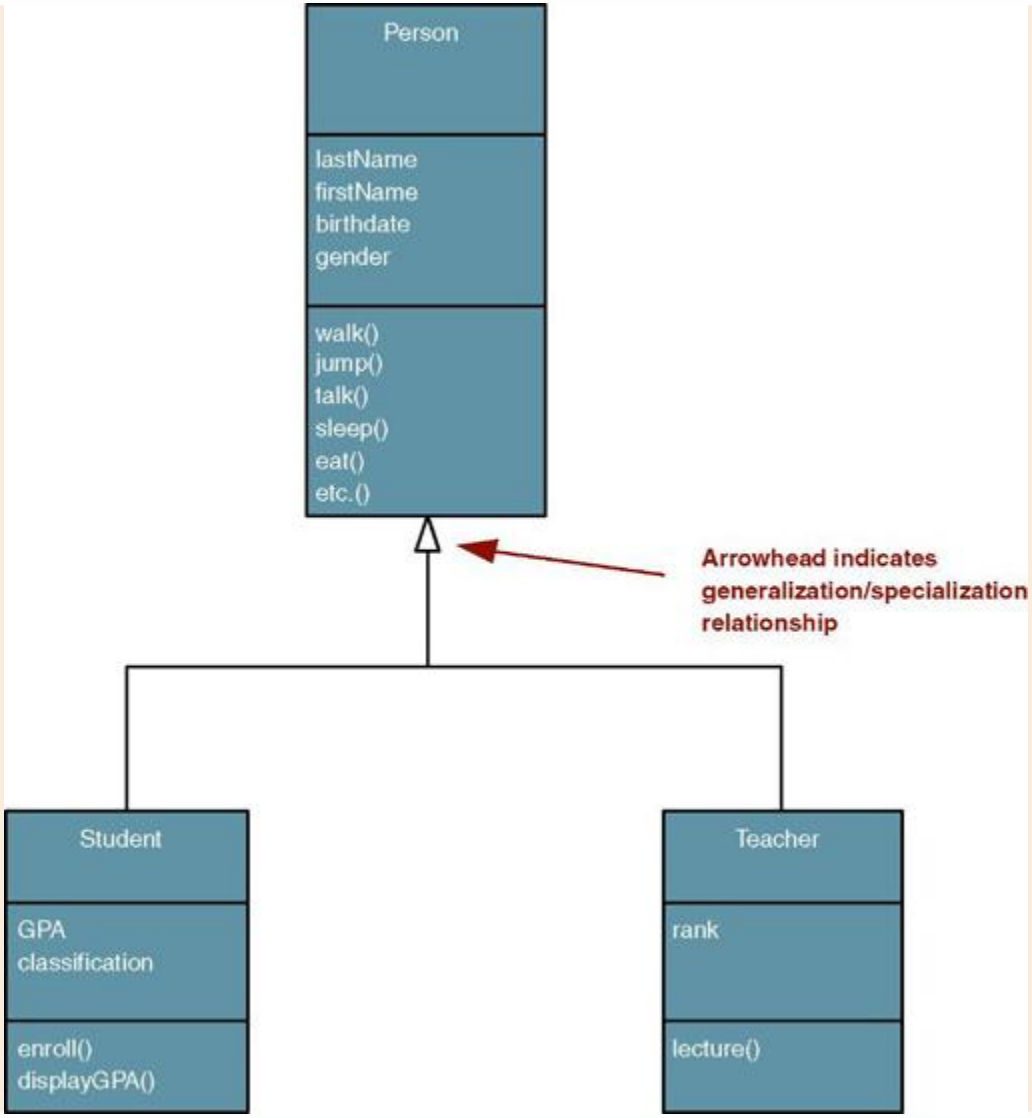
Generalization/Specialization, Supertype, and Subtype

Generalization/specialization – technique wherein attributes and behaviors common to several types of object classes are grouped (or abstracted) into their own class, called a *supertype*.

Supertype – an entity that contains attributes and behaviors that are common to one or more class subtypes. Also referred to as *abstract* or *parent* class.

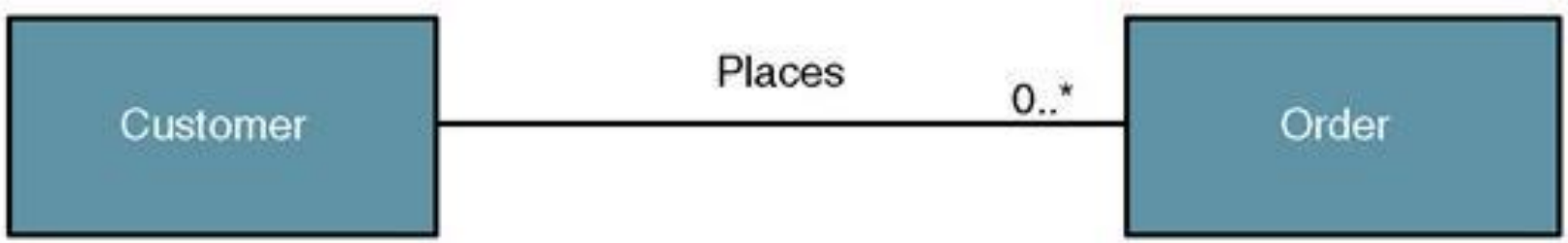
Subtype – an object class that inherits attributes and behaviors from a supertype class and may contain other attributes and behaviors unique to it. Also referred to as a *child* class and, if it exists at the lowest level of the inheritance hierarchy, as *concrete* class.

UML Representation of Generalization/Specialization



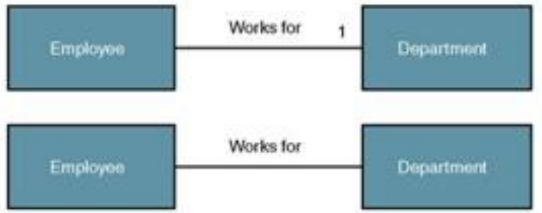
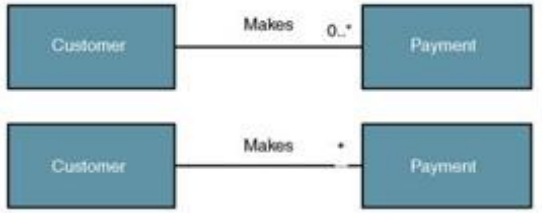
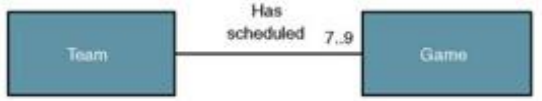
Object/Class Relationships

Object/class relationship – a natural business association that exists between one or more objects and classes.



UML Multiplicity Notations

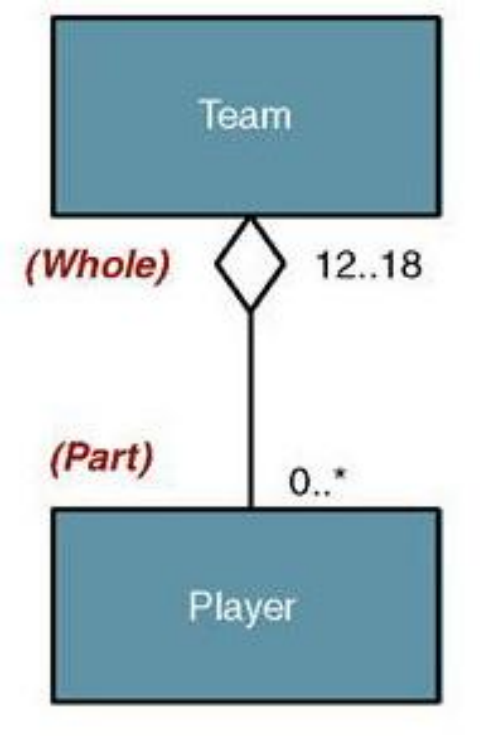
Multiplicity – the minimum and maximum number of occurrences of one object/class for a single occurrence of the related object/class.

Multiplicity	UML Multiplicity Notation	Association with Multiplicity	Association Meaning
Exactly 1	1 or *leave blank*		An employee works for one and only one department.
Zero or 1	0..1		An employee has either one or no spouse.
Zero or more	0..* or *		A customer can make no payment up to many payments.
1 or more	1..*		A university offers at least 1 course up to many courses.
Specific range	7..9		A team has either 7, 8, or 9 games scheduled

Aggregation

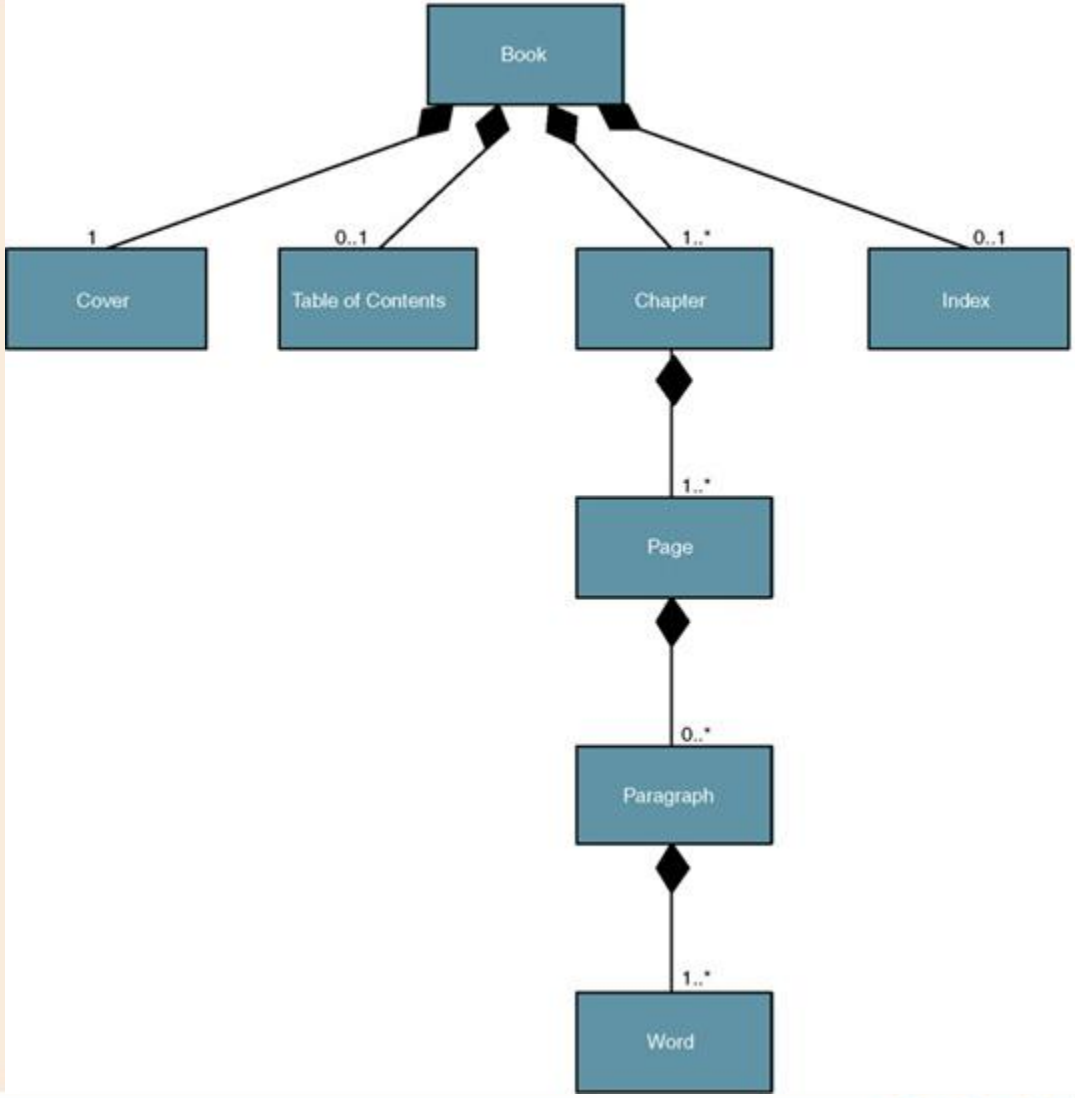
Aggregation – a relationship in which one larger "whole" class contains one or more smaller "parts" classes. Conversely, a smaller "part" class is part of a "whole" larger class

- In UML 2.0 the notation for aggregation has been dropped



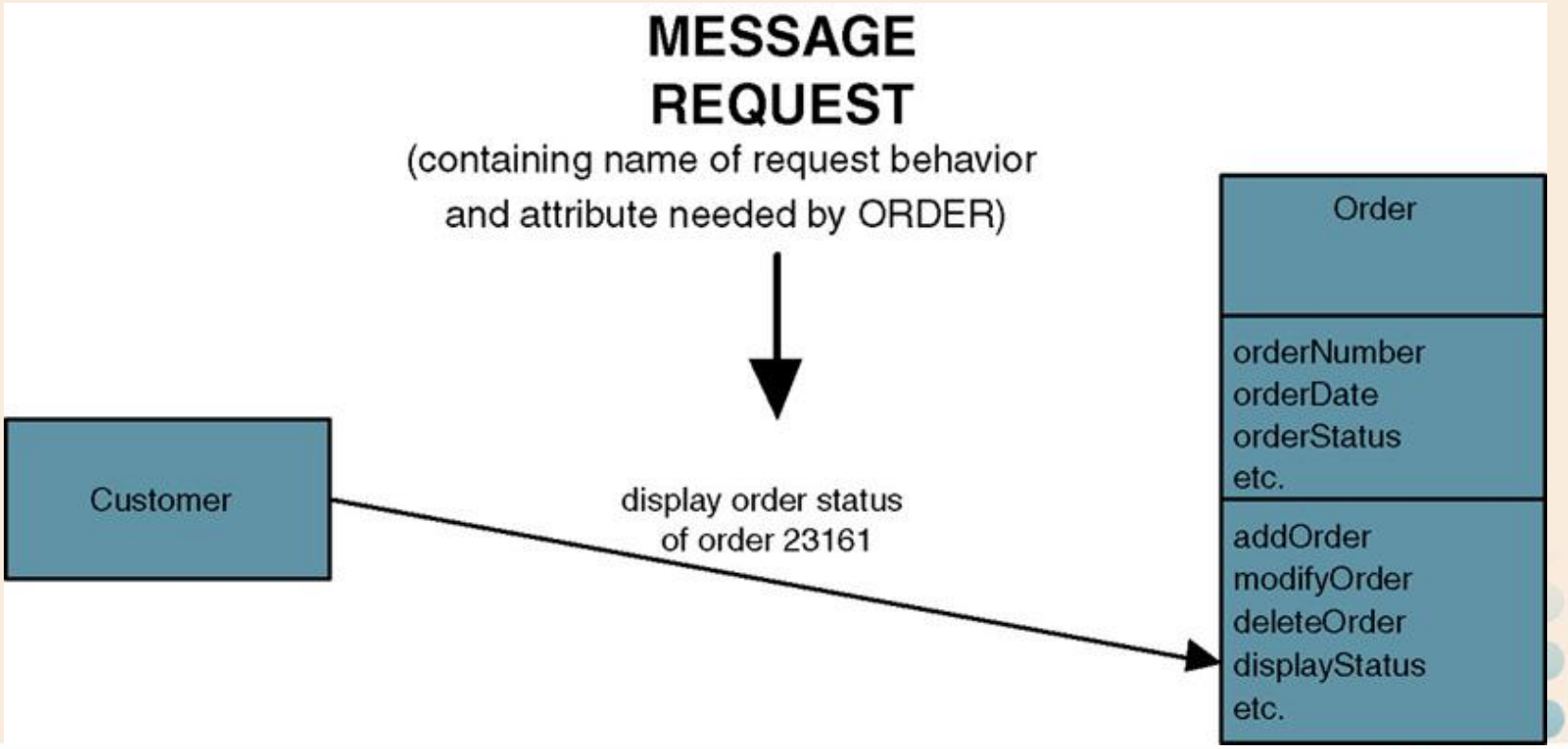
Composition

Composition – an aggregation relationship in which the "whole" is responsible for the creation and destruction of its "parts." If the "whole" were to die, the "part" would die with it.



Messages

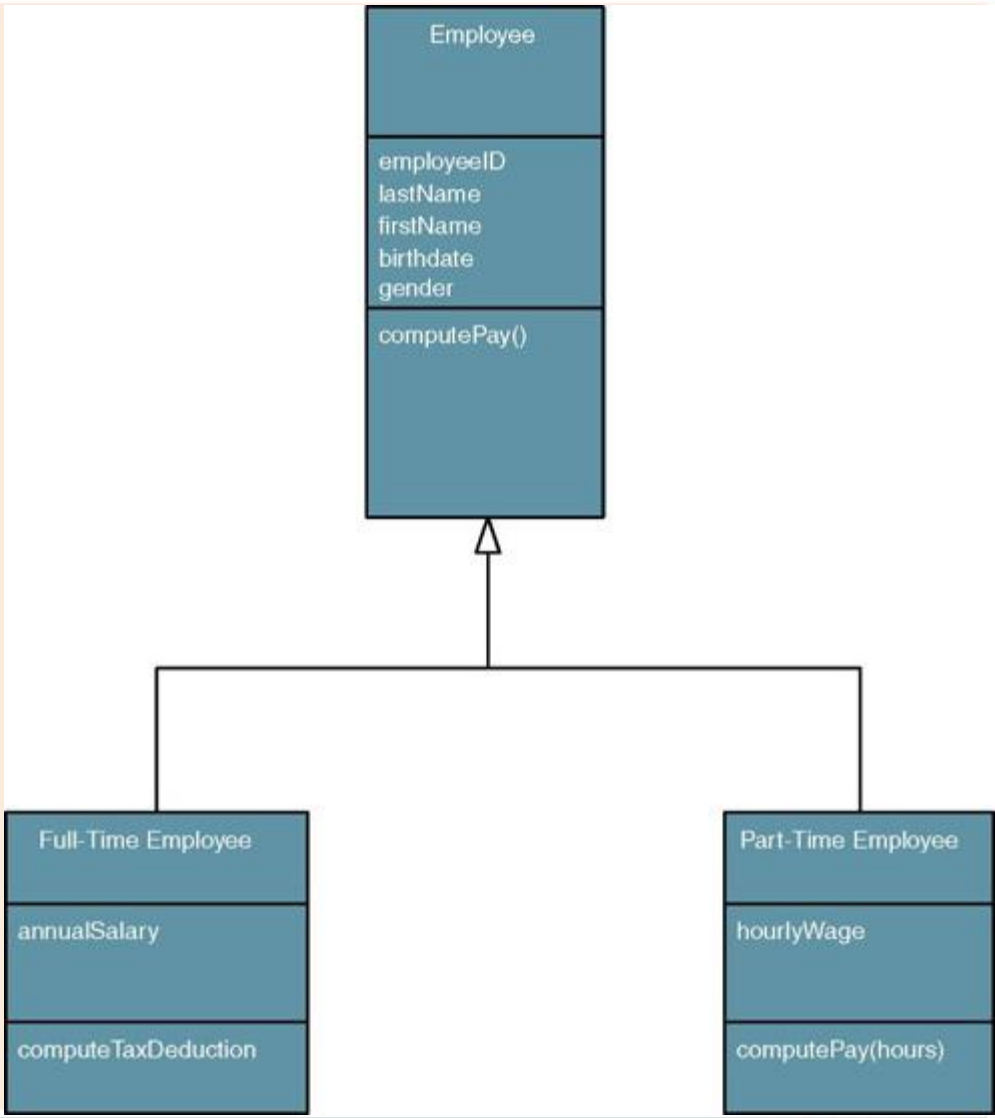
Message – communication that occurs when one object invokes another object's method (behavior) to request information or some action



Polymorphism

Polymorphism – the concept that different objects can respond to the same message in different ways.

Override – a technique whereby a subclass (subtype) uses an attribute or behavior of its own instead of an attribute or behavior inherited from the class (supertype).



UML 2.0 Diagrams

Diagram	Description
Use Case	Depicts interactions between the system and external systems and users. In other words it graphically describes who will use the system and in what ways the user expects to interact with the system. The use-case narrative is used in addition to textually describe the sequence of steps of each interaction.
Activity	Depicts sequential flow of activities of a use case or business process. It can also be used to model logic with the system.
Class	Depicts the system's object structure. It shows object classes that the system is composed of as well as the relationships between those object classes.
Object	Similar to a class diagram, but instead of depicting object classes, it models actual object instances with current attribute values. The object diagram provides the developer with a "snapshot" of the system's object at one point in time.
State Machine	Models how events can change the state of an object over its lifetime, showing both the various states that an object can assume and the transitions between those states.
Composite Structure	Decomposes internal structure of class, component, or use case.

UML 2.0 Diagrams (cont.)

Diagram	Description
Sequence	Graphically depicts how objects interact with each other via messages in the execution of a use case or operation. It illustrates how messages are sent and received between objects and in what *sequence*.
Communication	(Collaboration diagram in UML 1.X) Depicts interaction of objects via messages. While a sequence diagram focuses on the timing or sequence of messages, a communication diagram focuses on the structural organization of objects in a network format.
Interaction Overview	Combines features of sequence and activity diagrams to show how objects interact within each activity of a use case.
Timing	Another interaction diagram that focuses on timing constraints in the changing state of a single object or group of objects. Especially useful when designing embedded software for devices.
Component	Depicts the organization of programming code divided into components and how the components interact.
Deployment	Depicts the configuration of software components within the physical architecture of the system's hardware "nodes."
Package	Depicts how classes or other UML constructs are organized into packages (corresponding to Java packages or C++ and .NET namespaces) and the dependencies of those packages.

The Process of Object Modeling

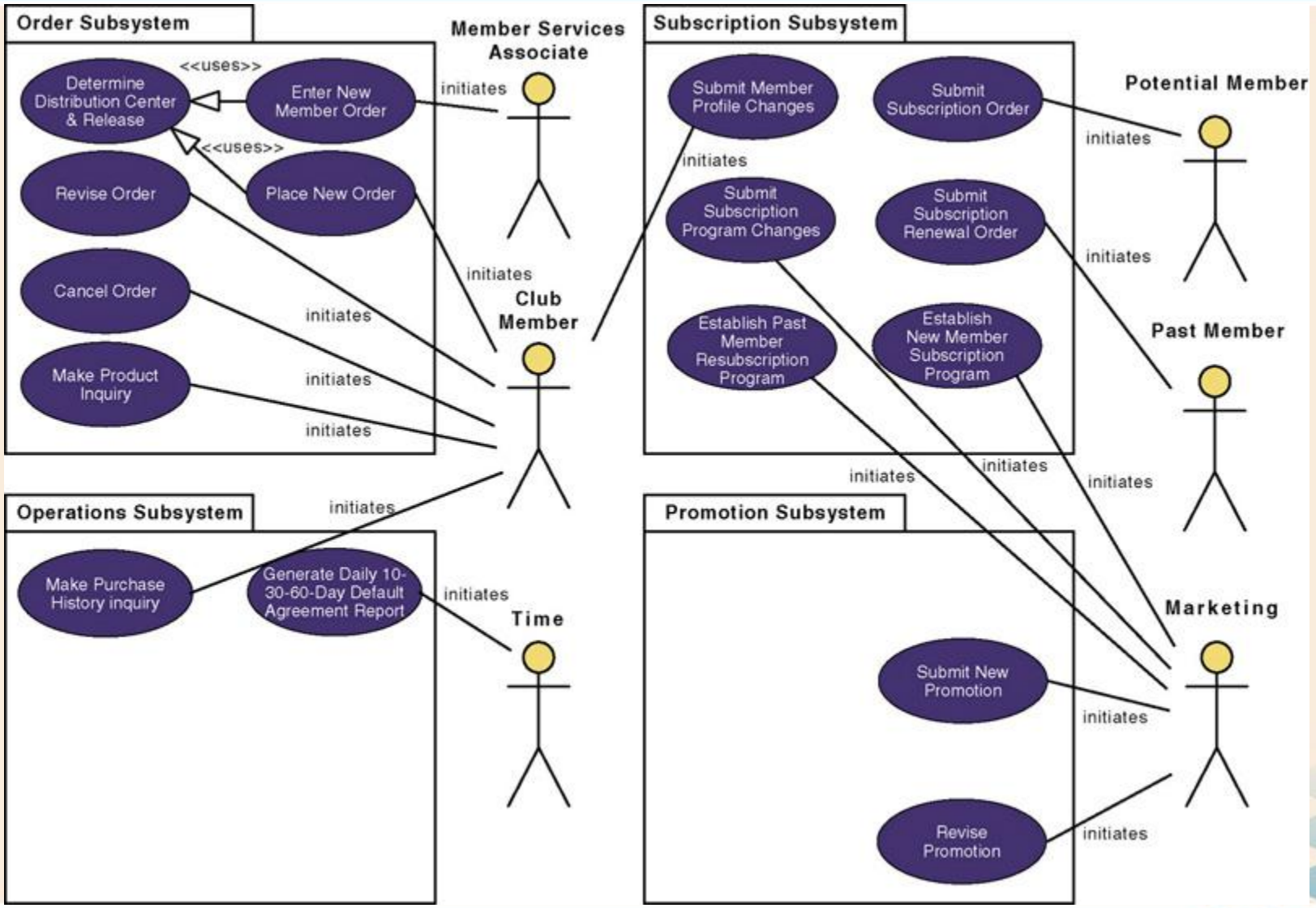
1. Modeling the functions of the system.
2. Finding and identifying the business objects.
3. Organizing the objects and identifying their relationships.

Construction the Analysis Use-Case Model

System analysis use case – a use case that documents the interaction between the system user and the system. It is highly detailed in describing what is required but is free of most implementation details and constraints.

1. Identify, define, and document new actors.
2. Identify, define, and document new use cases.
3. Identify any reuse possibilities.
4. Refine the use-case model diagram (if necessary).
5. Document system analysis use-case narratives.

Revised System Use-Case Model Diagram



Use-Case Narrative

Member Services System

Author(s): K. Dittman **Date:** 11/01/06 **Version:** 1.00

Use Case Name:	Place New Order
Use Case ID:	MSS-SUC002.00
Priority:	High
Source:	Requirement—MSS-R1.00 Requirements Use Case—MSS-BUC002.00

Use Case Type
Business Requirements: ☐
System Analysis: ☑ (1)

Primary Business Actor:	Club Member (Alias—Active Member, Member)
Primary System Actor:	Club Member (Alias—Active Member, Member) (2)
Other Participating Actors:	• Warehouse (Alias—Distribution Center) (external receiver) • Accounts Receivable (external server)
Other Interested Stakeholders:	• Marketing—Interested in sales activity in order to plan new promotions. • Procurement—Interested in sales activity in order to replenish inventory. • Management—Interested in order activity in order to evaluate company performance and customer (member) satisfaction.
Description:	This use case describes the event of a club member submitting a new order for SoundStage products via the World Wide Web. The member selects the items he or she wishes to purchase. Once the member has completed shopping, the member's demographic information as well as account standing will be validated. Once the products are verified as being in stock, a packing order is sent to the warehouse for it to prepare the shipment. For any product not in stock, a back order is created. On completion, the member will be sent an order confirmation.
Precondition:	The individual submitting the order must be an active club member. The member must log in to the system (provide identification) to enter an order.
Trigger:	This use case is initiated when the member selects the option to enter a new order.

Typical Course of Events:

Actor Action	System Response
Step 1: The member requests the option to enter a new order.	**Step 2:** The system responds by displaying the catalog of the SoundStage products.
Step 3: The member browses the available items and selects the ones he or she wishes to purchase, along with the quantity.	**Step 4:** Once the member has completed the selections, the system retrieves from file and presents the member's demographic information (shipping and billing addresses).
Step 5: The member verifies demographic information (shipping and billing addresses). If no changes are necessary, the member responds accordingly (to continue).	**Step 6:** For each product ordered, the system verifies the product availability and determines an expected ship date, determines the price to be charged to the member, and determines the cost of the total order. If an item is not immediately available, it indicates the product is back-ordered or that it has not been released for shipping (for preorders). If an item is no longer available, that is indicated also. The system then displays a summary of the order to the member for verification.
Step 7: The member verifies the order. If no changes are necessary, the member responds accordingly (to continue).	**Step 8:** The system checks the status of the member's account. If satisfactory, the system prompts the member to select the desired payment option (to be billed later or pay immediately with a credit card).
Step 9: The member responds by selecting the desired payment option.	**Step 10:** The system displays a summary of the order, including the desired payment option, to the member for verification.
Step 11: The member verifies the order. If no changes are necessary, the member responds accordingly (to continue).	**Step 12:** The system records the order information (including back orders if necessary). **Step 13:** Invoke abstract use case *MSS-AUC001.00, Determine Appropriate Distribution Center and Release Order to Be Filled.* (3) **Step 14:** Once the order is processed, the system generates an order confirmation and displays it to the member as well as sending it to the member via e-mail.

Use-Case Narrative (cont.)

Alternate Courses:	**Alt-Step 3:** The member enters search criteria to retrieve a specific item or to display a reduced list of items to browse and order from. **Alt-Step 5:** If changes are required, the member updates the appropriate shipping, billing, or e-mail addresses and tells the system to store them accordingly. The system will validate the changes and, if successful, will store the new information to file. **Alt-Step 7:** If the order requires changes, the member can delete any item no longer wanted or change the order quantity. Once the member has completed the order changes, the system reprocesses the order (**go to step 6**). If the member requests to do additional shopping, **go to step 3**. If the member needs to change the demographic information, **go to step 5**. **Alt-Step 8:** If the member's account is not in good standing, display to the member the account status, the reason the order is being held, and what actions are necessary to resolve the problem. In addition, an e-mail is sent to the member with the same information. The system prompts the member to hold the order for later processing or cancel the order. If the member wishes to hold the order, the system records the order information and places it in hold status and then displays the SoundStage main page. If the member chooses to cancel the order, the system clears the inputted information and then displays the SoundStage main page. Terminate the use case. **Alt-Step 10:** If the member selects the option to pay by credit card, the system prompts the member to enter the credit card information (number and expiration date) and reminds the member that the billing address on file must match the billing address of the credit card provided. The member enters the required information and requests that the system continue. The system validates the credit card account provided. If the account cannot be validated, the system notifies the member and requests an alternative means of payment. If the member cannot provide an alternative means at this time, he or she can choose either to hold or to cancel the order. If the member wishes to hold the order, the system records the order information and places it in hold status and then displays the SoundStage main page. If the member chooses to cancel the order, the system clears the inputted information and then displays the SoundStage main page. Terminate the use case. **Alt-Step 11:** If the order requires changes, the member can delete any item no longer wanted or change the order quantity. Once the member has completed the order changes, the system reprocesses the order (**go to step 6**). If the member requests to do additional shopping, **go to step 3**. If the member needs to change the demographic information, **go to step 5**. **Alt-Step 12:** If all items ordered are on back order, the order is not released to the distribution center.
Conclusion:	This use case concludes when the member receives a confirmation of the order.
Postcondition:	The order has been recorded and, if the ordered products were available, released to the distribution center. For any product not available a back order has been created.
Business Rules:	• Member must have a valid e-mail address to submit online orders. • Member is billed for products only when they are shipped.
Implementation Constraints and Specifications:	• Use case must be available to the member 24 × 7. • Frequency—It is estimated that this use case will be executed 3,500 times per day. It should support up to 50 concurrent members.
Assumptions:	• Product can be transferred among distribution centers to fill orders. • Procurement will be notified of back orders by a daily report (separate use case). • The member responding to a promotion or using credits may affect the price of each ordered item. • The member can cancel the order at any time.
Open Issues:	None

Abstract Use-Case Narrative

Member Services System

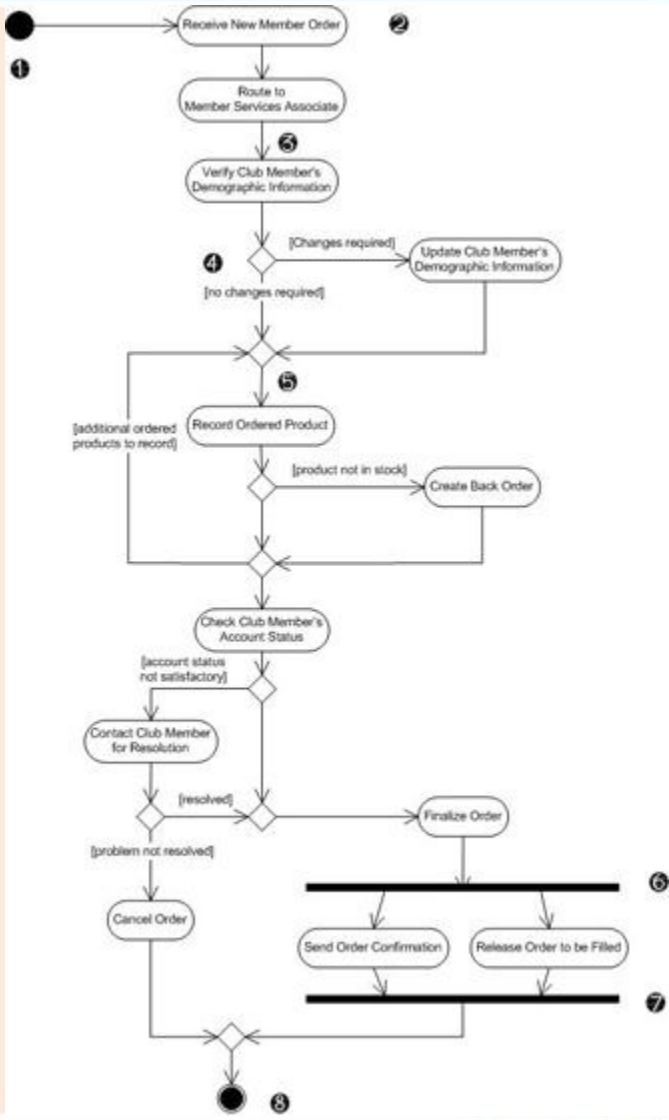
Author(s): K. Dittman **Date:** 11/01/06 **Version:** 1.00

Use Case Name:	Determine Appropriate Distribution Center and Release Order to Be Filled.
Use Case ID:	MSS-AUC001.00
Priority:	High
Source:	MSS-SUC002.00 MSS-SUC003.00 (2)
Participating Actors:	• Warehouse (Alias — Distribution Center) (external receiver)
Description:	This use case describes the event of selecting the distribution center that services the shipping address provided by the club member for a particular order. The order information (packing order) is then sent (released) to that distribution center to be filled.
Precondition:	The order is ready to be released to the appropriate distribution center.
Typical Course of Events:	**Step 1:** The system selects the appropriate distribution center based on the state and zip code of the shipping address. **Step 2:** Once the distribution center has been selected, a packing order containing the items to ship is formatted. **Step 3:** The packing order is transmitted to the distribution center (shipping and receiving system) to be used to prepare the shipment.
Alternate Courses:	**Alt-Step 1:** If the shipping address is an international address, route the packing order to the Indianapolis, IN, location.
Postcondition:	The packing slip has been transmitted (released) to the appropriate distribution center.

Use Case Type
Abstract: ☑ (1)
Extension: ☐

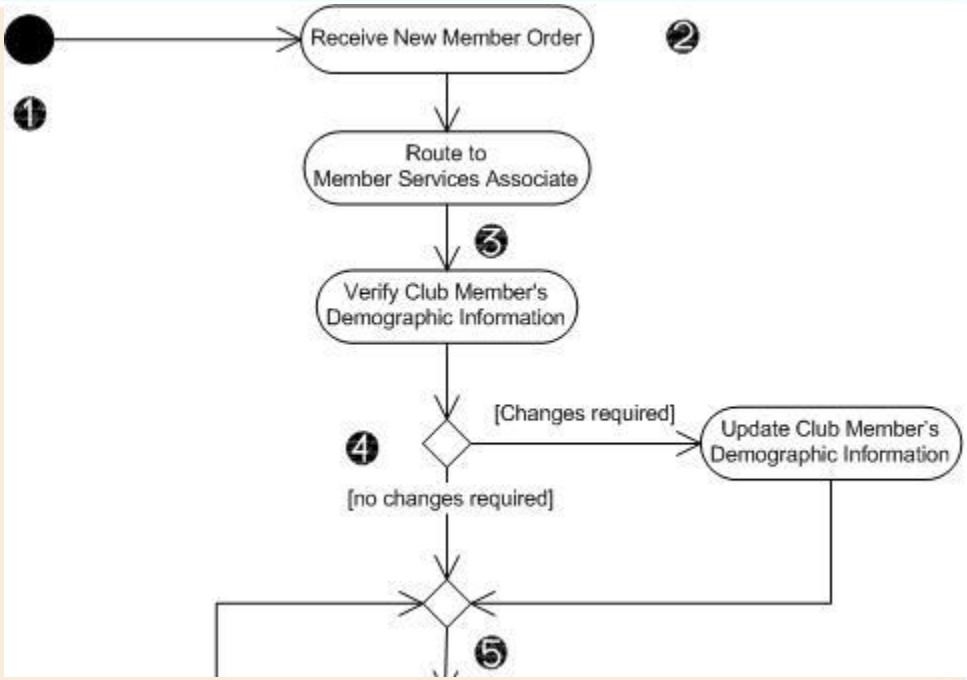
Modeling Use-Case Activities

Activity diagram – a diagram that can be used to graphically depict the flow of a business process, the steps of a use case, or the logic of an object behavior (method).



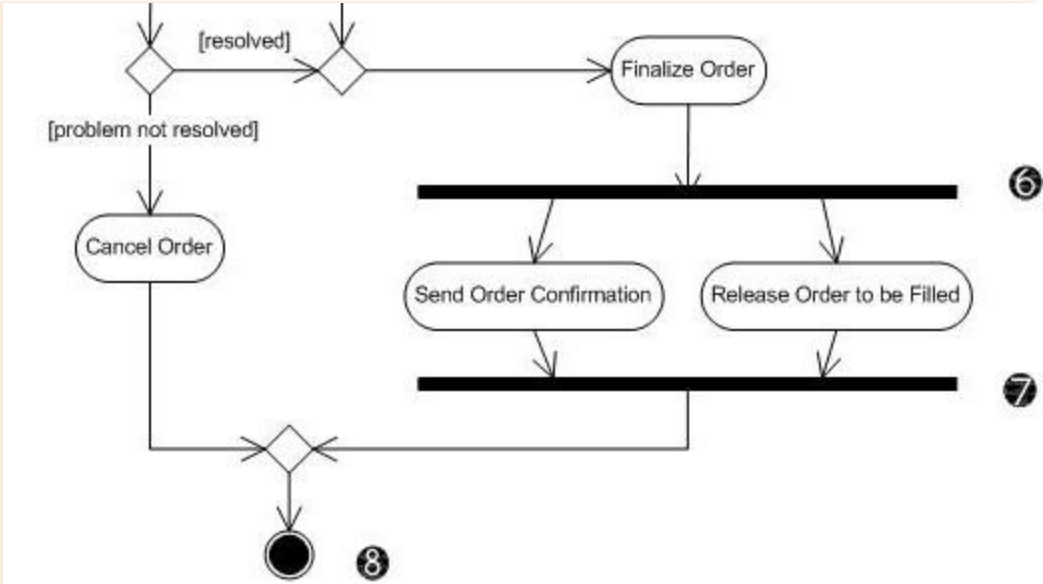
Activity Diagram Notations

1. **Initial node** - solid circle representing the start of the process.
2. **Actions** – rounded rectangles representing individual steps. The sequence of actions make up the total activity shown by the diagram.
3. **Flow** - arrows on the diagram indicating the progression through the actions. Most flows do not need words to identify them unless coming out of decisions.
4. **Decision** - diamond shapes with one flow coming in and two or more flows going out. The flows coming out are marked to indicate the conditions.
5. **Merge** - diamond shapes with multiple flows coming in and one flow going out. This combines flows previously separated by decisions. Processing continues with any one flow coming into the merge.



Activity Diagram Notations (cont.)

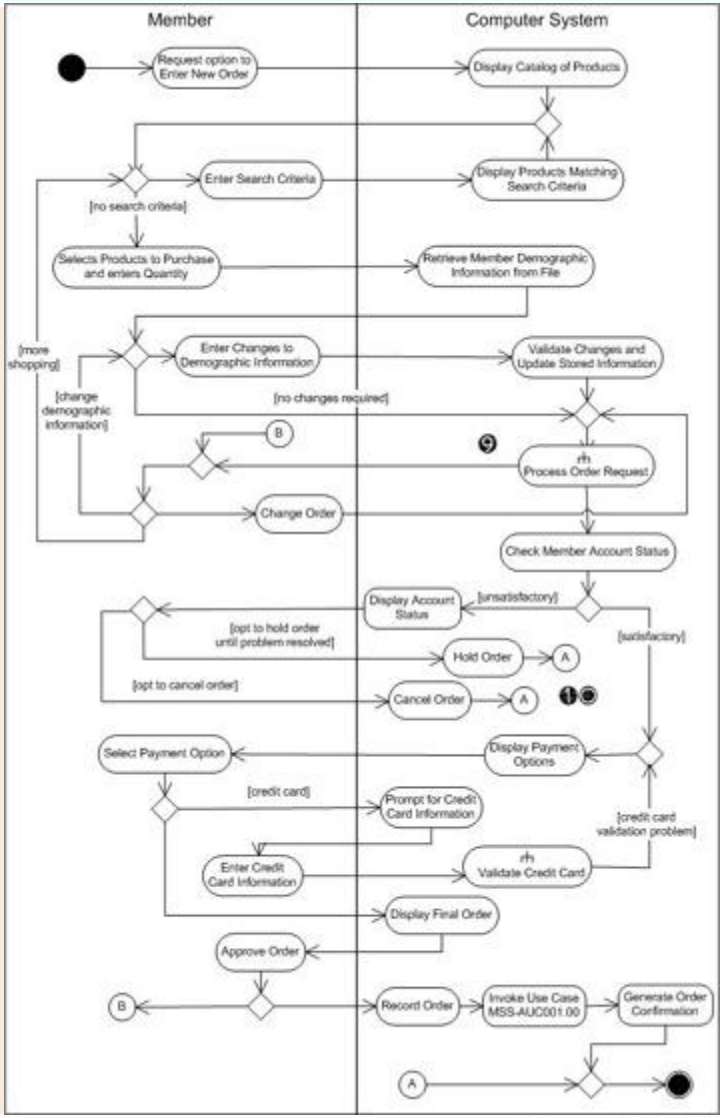
6. **Fork** – a black bar with one flow coming in and two or more flows going out. Actions on parallel flows beneath the fork can occur in any order or concurrently.



7. **Join** – a black bar with two or more flows coming in and one flow going out, noting the end of concurrent processing. All actions coming into the join must be completed before processing continues.
8. **Activity final** – the solid circle inside the hollow circle representing the end of the process.

Activity Diagram with Partitions

9. **Subactivity indicator** – the rake symbol in an action indicates that this action is broken out in another separate activity diagram. This helps you keep the activity diagram from becoming overly complex.
10. **Connector** – A letter inside a circle gives you another tool for managing complexity. A flow coming into a connector jumps to the flow coming out of a connector with a matching letter.



Guidelines for Constructing Activity Diagrams

- Start with one initial node as a starting point.
- Add partitions if it is relevant to your analysis.
- Add an action for each major step of the use case (or each major step an actor initiates.
- Add flows from each action to another action, a decision point, or an end point. For maximum precision of meaning, each action should have only one flow coming in and one flow going out with all forks, joins, decisions, and merges shown explicitly.
- Add decisions where flows diverge with alternating routes. Be sure to bring them back together with a merge.
- Add forks and joins where activities are performed in parallel.
- End with a single notation for activity final.

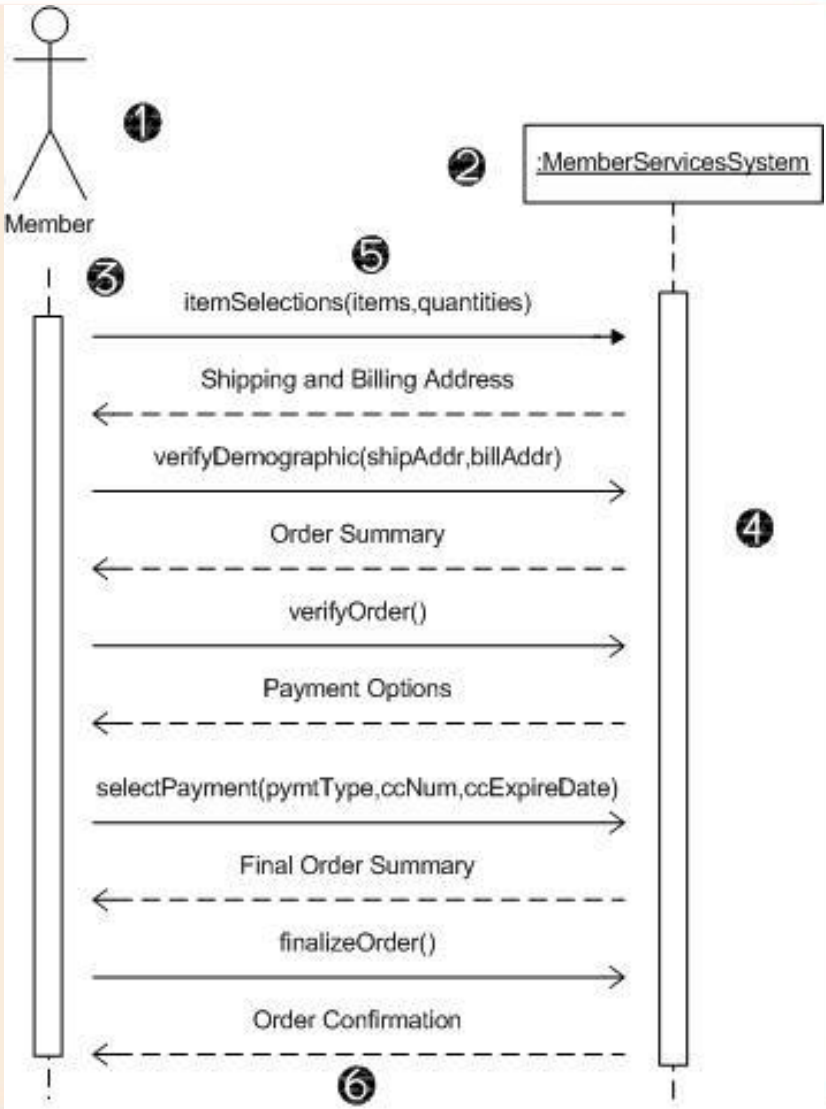
Drawing System Sequence Diagrams

System sequence diagram - a diagram that depicts the interaction between an actor and the system for a use case scenario.

- helps identify high-level messages that enter and exit the system

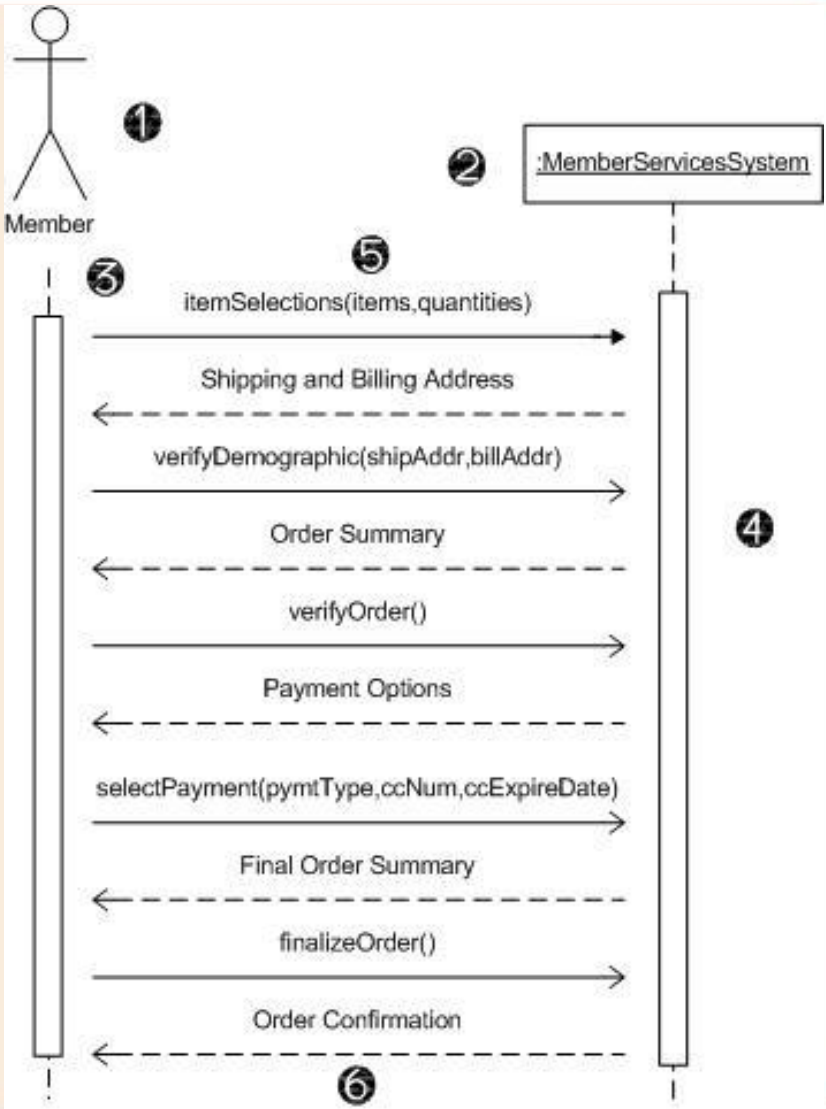
System Sequence Diagram Notations

1. **Actor** - the initiating actor of the use case is shown with the use case actor symbol.
2. **System** – the box indicates the system as a "black box" or as a whole. The colon (:) is standard sequence diagram notation to indicate a running "instance" of the system.
3. **Lifelines** – the dashed vertical lines extending downward from the actor and system symbols, which indicate the life of the sequence.
4. **Activation bars** – the bars set over the lifelines indicate period of time when participant is active in the interaction.



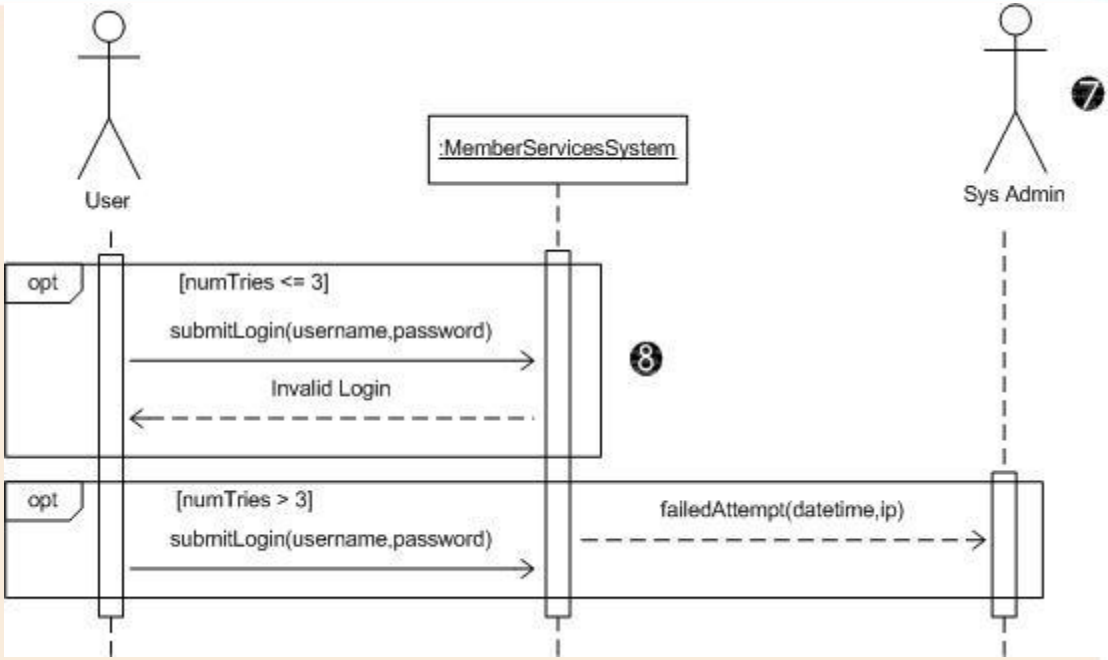
System Sequence Diagram Notations (cont.)

5. **Input messages** - horizontal arrows from actor to system indicate the message inputs. UML convention for messages is to begin the first word with a lowercase letter and add additional words with initial uppercase letter and no space. In parentheses include parameters, following same naming convention and separated with commas.
6. **Output messages –** horizontal arrows from system to actor shown as dashed lines. Since they are web forms, reports, e-mails, etc. these messages do not need to use the standard notation.



System Sequence Diagram Notations (cont.)

7. **Receiver Actor** – other actors or external systems that receive messages from the system can be included.
8. **Frame** – a box can enclose one or more messages to divide off a fragment of the sequence. These can show loops, alternate fragments, or optional (opt) steps. For an optional fragment the condition shown in square brackets indicates the conditions under which the steps will be performed.



Guidelines for Constructing System Sequence Diagrams

- Identify which scenario of use case you will depict. Purpose is to discover messages, not to model logic. So more important to clearly communicate a single scenario.
- Draw a rectangle representing the system as a whole and extend a lifeline under it.
- Identify each actor who directly provides an input to the system or directly receives an output from the system. Extend lifelines under the actor(s).
- Examine use case narrative to identify system inputs and outputs. Ignore messages inside system. Draw each external message as a horizontal arrow from the actor's lifeline to the system or from the system to the actor. Label inputs according to UML convention.
- Add frames to indicate optional messages with conditions. Frames can also indicate loops and alternate fragments.
- Confirm that the messages are shown in the proper sequence from top to bottom.

Finding and Identifying the Business Objects

1. Find the Potential Objects
   - Review each use case to find nouns that correspond to business entities or events.
2. Select the Proposed Objects
   - Not all nouns represent business objects.
     - Is it a synonym of another object?
     - Is it outside the scope of the system?
     - Is it a role without unique behavior, or an external role?
     - Is it unclear or in need of focus?
     - Is it an action or an attribute that describes another object?

Partial Use-Case Narrative with Nouns Highlighted

DESCRIPTION:	This use case describes the event of a member submitting a new order for SoundStage products via the world wide web. The member selects the items they wish to purchase. Once they have completed their shopping, the member's demographic information as well as their account standing will be validated. Once the products are verified as being in stock, a packing order is sent to the distribution center for them to prepare the shipment. For any product not in stock, a back order is created. On completion, the member will be sent an order confirmation.	
PRE-CONDITION:	The individual submitting the order must be an active club member. The member must login in to the system (provide identification) to enter an order.	
TRIGGER:	This use case is initiated when the member selects the option to enter a new order.	
TYPICAL COURSE OF EVENTS:	**Actor Action**	**System Response**
	Step 1: The member requests the option to enter a new order.	**Step 2**: The system responds by displaying the catalogue of the SoundStage products.
	Step 3: The Member browses the available items and selects the ones they wish to purchase along with the quantity.	**Step 4**: Once the member has completed their selections the system retrieves from file and presents the member's demographic information (shipping and billing addresses).
	Step 5: The member verifies demographic information (shipping and billing addresses). If no changes are necessary they respond accordingly (to continue).	**Step 6**: For each product ordered, the system verifies the product availability and determines an expected ship date, determines the price to be charged to the member, and determines the cost of the total order. If an item is not immediately available it indicates that the product is backordered or that it has not been released for shipping (for pre-orders). If an item is no longer available that is indicated also. The system then displays a summary of the order to the member for verification.
	Step 7: The member verifies the order. If no changes are necessary they respond accordingly (to continue).	**Step 8**: The system checks the status of the member's account. If satisfactory, the system prompts the member to select the desired payment option (to be billed later or pay immediately with a credit card).

Potential Object List

- Accounts receivable
- Actions
- Active Member
- Available Items
- Back Order
- Back Ordered
- Billing Addresses
- Catalog
- Club Member
- Company Performance
- Credit Card
- Credit Card Expiration Date
- Credit Card Number
- Credits
- Customer Satisfaction
- Daily report
- Demographic Information
- Distribution Center
- E-Mail
- E-Mail Addresses
- Event
- Expected Ship Date
- External Receiver
- External Server
- File
- Hold Status
- Identification
- In Stock
- Individual
- Inventory
- Items
- Main Page
- Management
- Marketing
- Member
- Member Account Standing
- Member's Account Status
- New Order
- New Promotions
- Option
- Order Activity
- Order Confirmation
- Order Total Cost
- Ordered Products
- Packing Order
- Payment Option
- Preorders
- Price to Be Charged
- Problem
- Procurement
- Product Availability
- Product Ordered
- Promotion
- Purchase
- Quantity
- Reason
- Requirement
- Requirements Use Case
- Sales Activity
- Search Criteria
- Selections
- Shipment
- Shipping Address
- Shopping
- SoundStage Products
- Summary of the Order
- System
- Warehouse
- World Wide Web

Cleaning Up List of Candidate Objects

Potential Object		Reason
New Order	√	MEMBER ORDER
New Promotions	√	PROMOTION
Option	x	Potential Interface item to be addressed in object-oriented design
Order Activity	x	Potential Interface item to be addressed in object-oriented design (report)
Order Confirmation	x	Potential Interface item to be addressed in object-oriented design
Order Total Cost	x	Attribute of MEMBER ORDER
Ordered Products	√	MEMBER ORDERED PRODUCT
Packing Order	x	Potential Interface item to be addressed in object-oriented design
Payment Option	x	Attribute of MEMBER ORDER
Preorders	√	Type of MEMBER ORDER
Price to Be Charged	x	Attribute of MEMBER ORDERED PRODUCT
Problem	x	Needs better focus – probably will be a comments attribute in MEMBER ORDER
Procurement	x	Not relevant for current project
Product Availability	x	Attribute of PRODUCT
Product Ordered	x	Synonym of MEMBER ORDERED PRODUCT
Promotion	√	PROMOTION
Purchase	x	Synonym of MEMBER ORDER
Quantity	x	Attribute of MEMBER ORDERED PRODUCT
Reason	x	Needs better focus – probably will be a comments attribute in MEMBER ORDER
Requirement	x	Not relevant for current project
Requirements Use Case	x	Not relevant for current project
Sales Activity	x	Potential Interface item to be addressed in object-oriented design (report)
Search Criteria	x	Potential Interface item to be addressed in object-oriented design
Selections	x	Synonym of MEMBER ORDERED PRODUCT
Shipment	x	Not relevant for current project – responsibility of shipping and receiving
Shipping Address	√	Type of ADDRESS
Shopping	x	Potential Interface item to be addressed in object-oriented design
SoundStage Products	x	Synonym of DISTRIBUTION CENTER
Summary of the Order	x	Potential Interface item to be addressed in object-oriented design
System	x	Not relevant for current project
Warehouse	x	Synonym of DISTRIBUTION CENTER
World Wide Web	x	Potential Interface item to be addressed in object-oriented design

Proposed Object List

Proposed Object List
ACTIVE MEMBER
BILLING ADDRESS
CLUB MEMBER
CREDIT CARD ACCOUNT
DISTRIBUTION CENTER
EMAIL ADDRESS
MEMBER
MEMBER ORDER
PROMOTION
MEMBER ORDERED PRODUCT
PRE-ORDER
PROMOTION
SHIPPING ADDRESS
-PLUS-
AGREEMENT
AUDIO TITLE
FORMER MEMBER
GAME TITLE
INACTIVE MEMBER
MERCHANDISE
RETURN
TITLE
TRANSACTION
VIDEO TITLE

Organizing the Objects and Identifying their Relationships

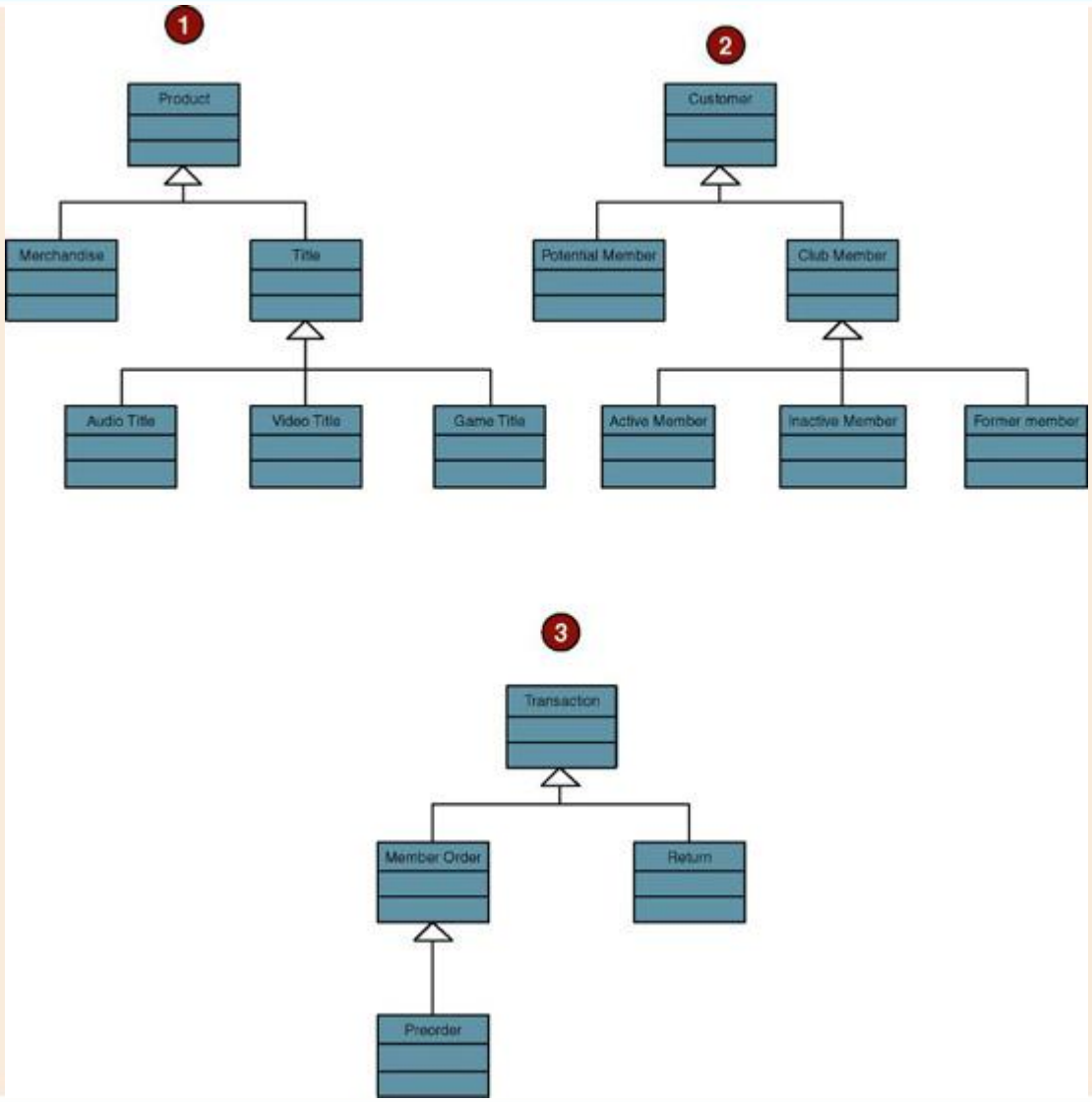
1. Identifying Associations and Multiplicity
2. Identifying Generalization/Specialization Relationships
3. Identifying Aggregation Relationships
4. Prepare the Class Diagram

Class diagram – a graphical depiction of a system's static object structure, showing object classes that the system is composed of as well as the relationships between those object classes.

Object Association Matrix

	CLUB MEMBER	MEMBER ORDER	MEMBER ORDERED PRODUCT	PRODUCT
CLUB MEMBER		Places zero to many	Has purchased zero to many	XX
MEMBER ORDER	Is placed by one and only one		Contains one to many	XX
MEMBER ORDERED PRODUCT	Was purchased by one and only one	Is part of one and only one		Relates to one and only one
PRODUCT	XX	XX	Sold as zero to many	

Generalization/Specialization Hierarchies



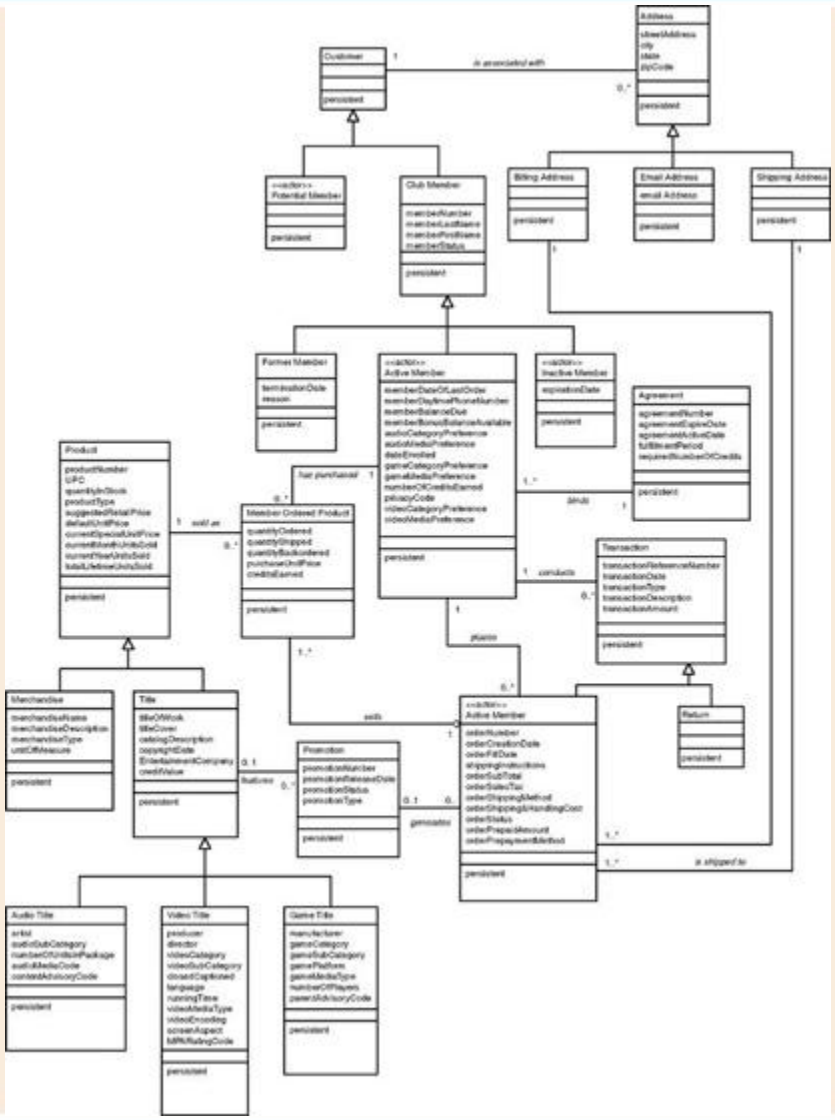
Persistent and Transient Object Classes

Persistent class – a class that describes an object that outlives the execution of the program that created it.

- Stored permanently as in a database

Transient object class – a class that describes an object that is created temporarily by the program and lives only during that program's execution.

Class Diagram



Refer to Figure 9-24 in text for a more readable copy