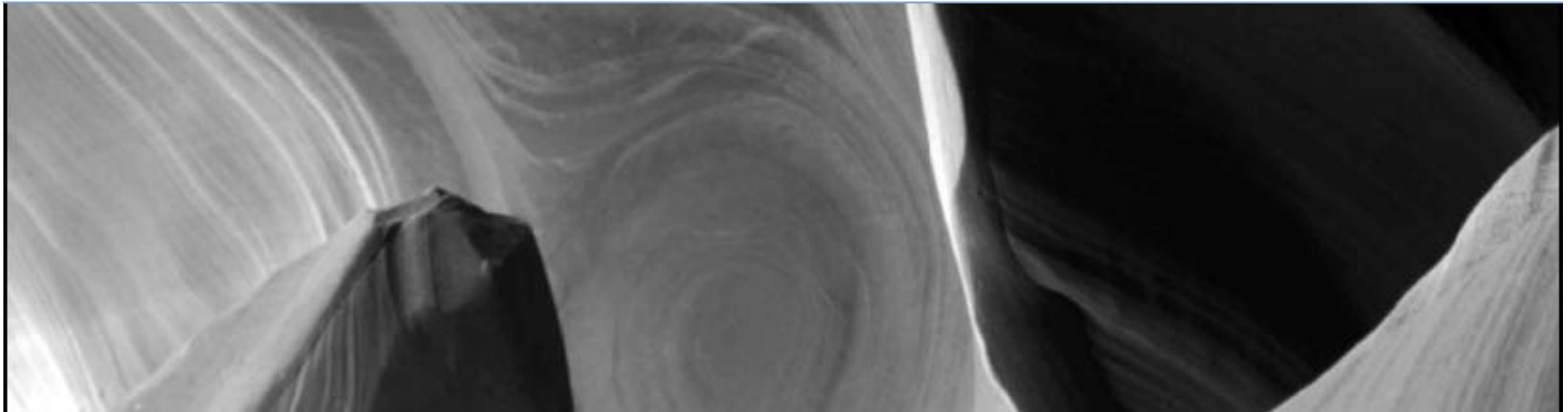# *Systems Analysis and Design in a Changing World, Fifth Edition*

CHAPTER 5

MODELING SYSTEM REQUIREMENTS

Learning Objectives

◆ Explain why identifying use cases is the key to defining functional requirements

◆ Use three techniques for identifying use cases

◆ Write brief, intermediate, and fully developed use case descriptions

Learning Objectives (continued)

◆ Explain how the concept of things in the problem domain also defines requirements

◆ Identify and analyze data entities and domain classes needed in the system

◆ Read, interpret, and create an entity-relationship diagram

◆ Read, interpret, and create a domain model class diagram

Overview

◆ Document functional requirements by creating models

◆ Models created during analysis phase activity – *Define system requirements*

◆ Two concepts help identify functional requirements in the traditional approach and object-oriented approach

- Use cases and the events that trigger them

- Things in the users' work domain

User Goals, Events, and Use Cases

◆ Use Case -- An activity the system performs in response to a user request

◆ Techniques for identifying use cases

- User goal technique

  ◆ Each goal at the elementary business process (EBP) level is a use case

  ◆ EBP – a task performed by one user, in one place in response to a business event, that adds measurable business value, and leaves system and data in consistent state

User Goals, Events, and Use Cases (continued)

- CRUD analysis technique (create, read, update, delete)

- Event decomposition technique

Identifying Use Cases Based on User Goals

User/actor	User goal
Order clerk	Look up item availability Create new order Update order
Shipping clerk	Record order fulfillment Record back order
Merchandising manager	Create special promotion Produce catalog activity report

Figure 5-1

Use Case Based on CRUD Technique

Data entity/class	CRUD	Resulting use case
Customer	Create	Add new customer
	Read/Report	Look up customer Produce customer list List customer orders
	Update	Update customer information
	Delete	Delete inactive customer

Figure 5-2

Systems Analysis and Design in a Changing World, 5th Edition   8

Event Decomposition Technique

◆ Event – an occurrence at a specific time and place and which needs to be remembered

◆ Business events trigger elementary business processes (EBPs)

◆ EBPs are at correct level of analysis for use cases

◆ Identify business events to decompose system into activities/use cases

Types of Events

◆ External

- Outside system

- Initiated by external agent or actor

◆ Temporal

- Occur as result of reaching a point in time

- Based on system deadlines

◆ State

- Something inside system triggers processing need

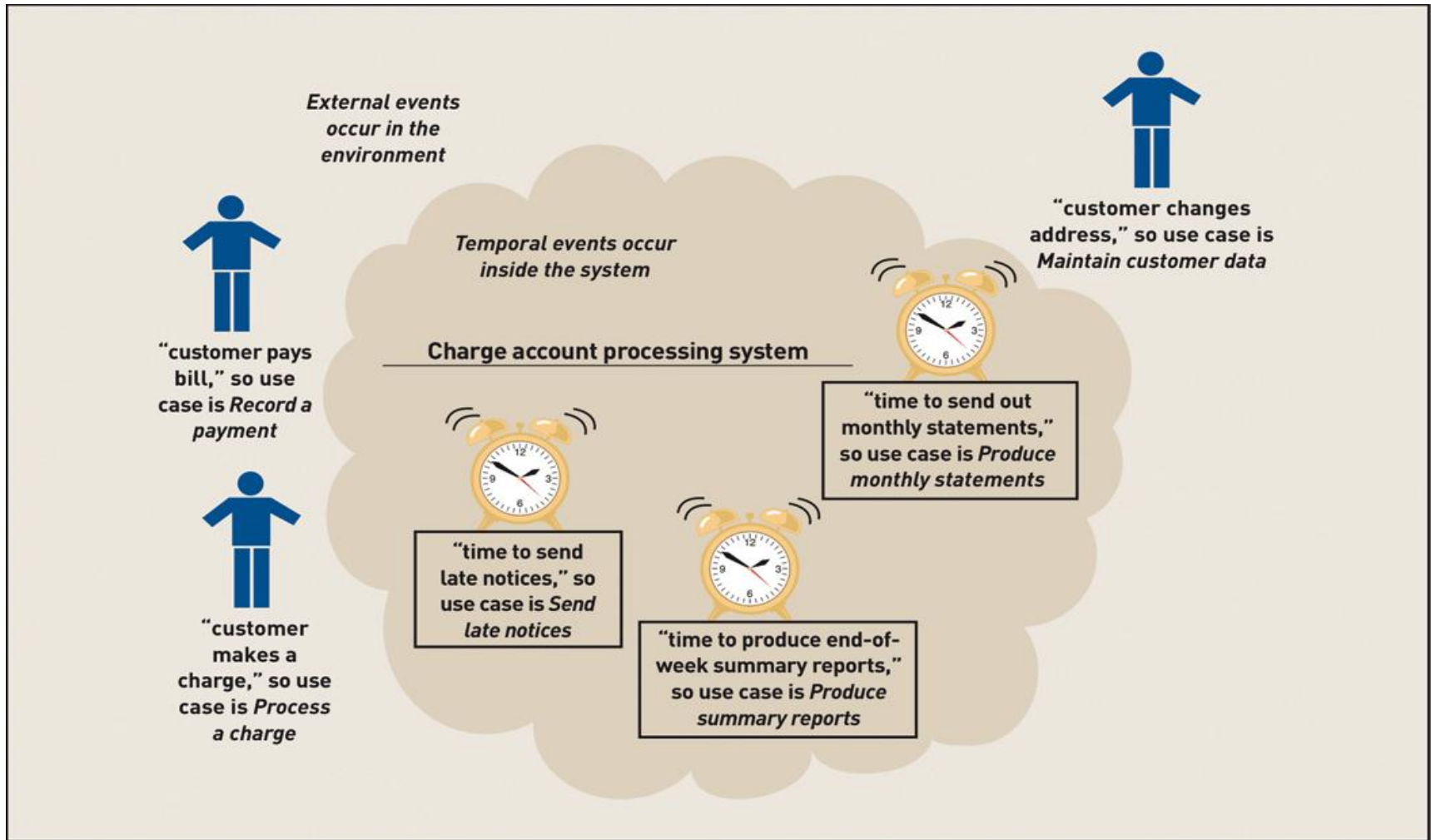# Events Affecting a Charge Account Processing System that Lead to Use Cases

Figure 5-3

External Event Checklist

External events to look for include:
√ External agent wants something resulting in a transaction
√ External agent wants some information
√ Data changed and need to be updated
√ Management wants some information

Figure 5-4

Temporal Event Checklist

Temporal events to look for include:
√ Internal outputs needed
   √ Management reports (summary or exception)
   √ Operational reports (detailed transactions)
   √ Internal statements and documents (including payroll)
√ External outputs needed
   √ Statements, status reports, bills, reminders

Figure 5-5

Systems Analysis and Design in a Changing World, 5th Edition

Identifying Events

◆ Can be difficult to determine

◆ Often confused with conditions and responses

◆ May be useful to trace a transaction's life cycle

◆ Certain events left to design phase

- System controls to protect system integrity

- Perfect technology assumption defers events

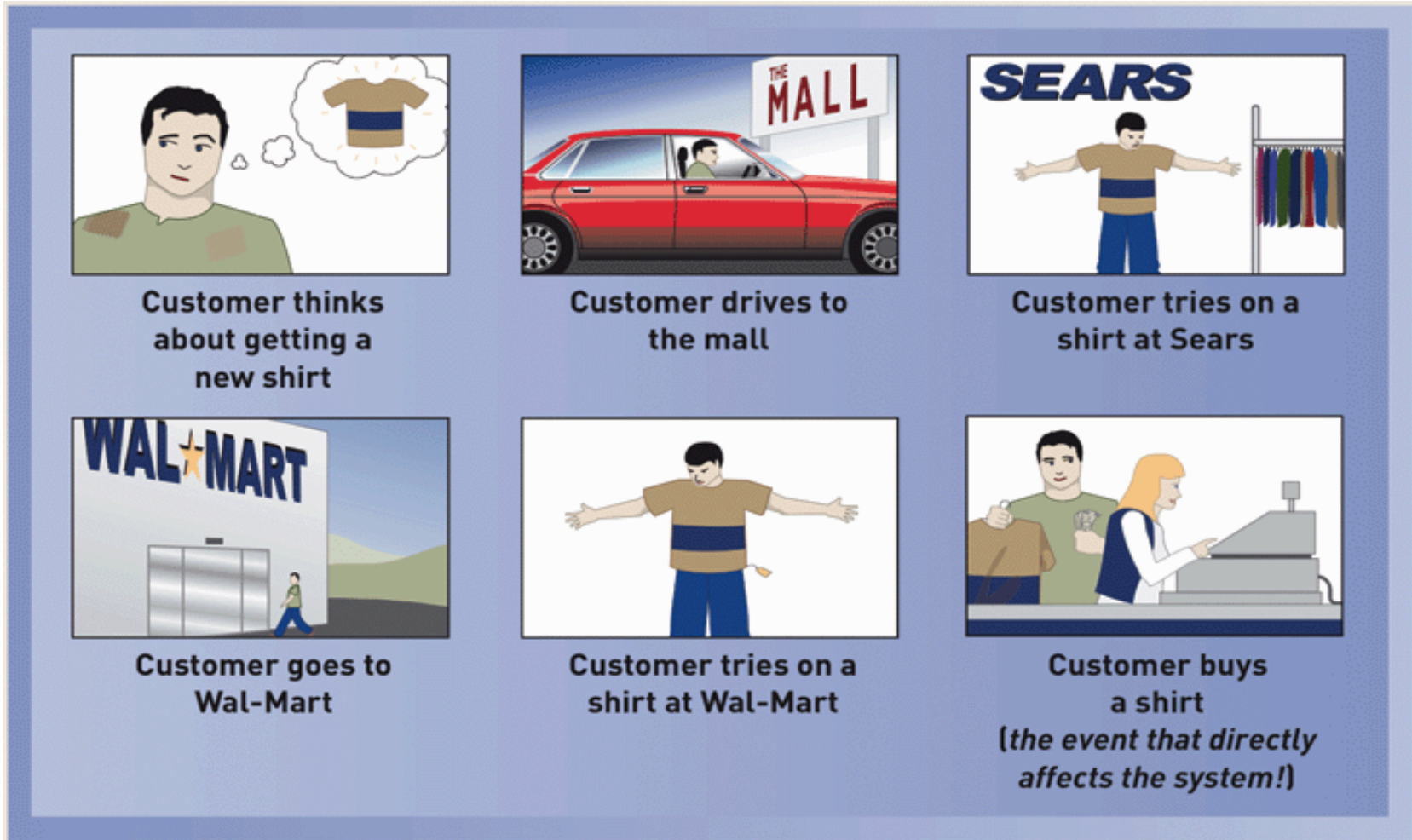# Sequence of Actions that Lead Up to Only One Event Affecting the System
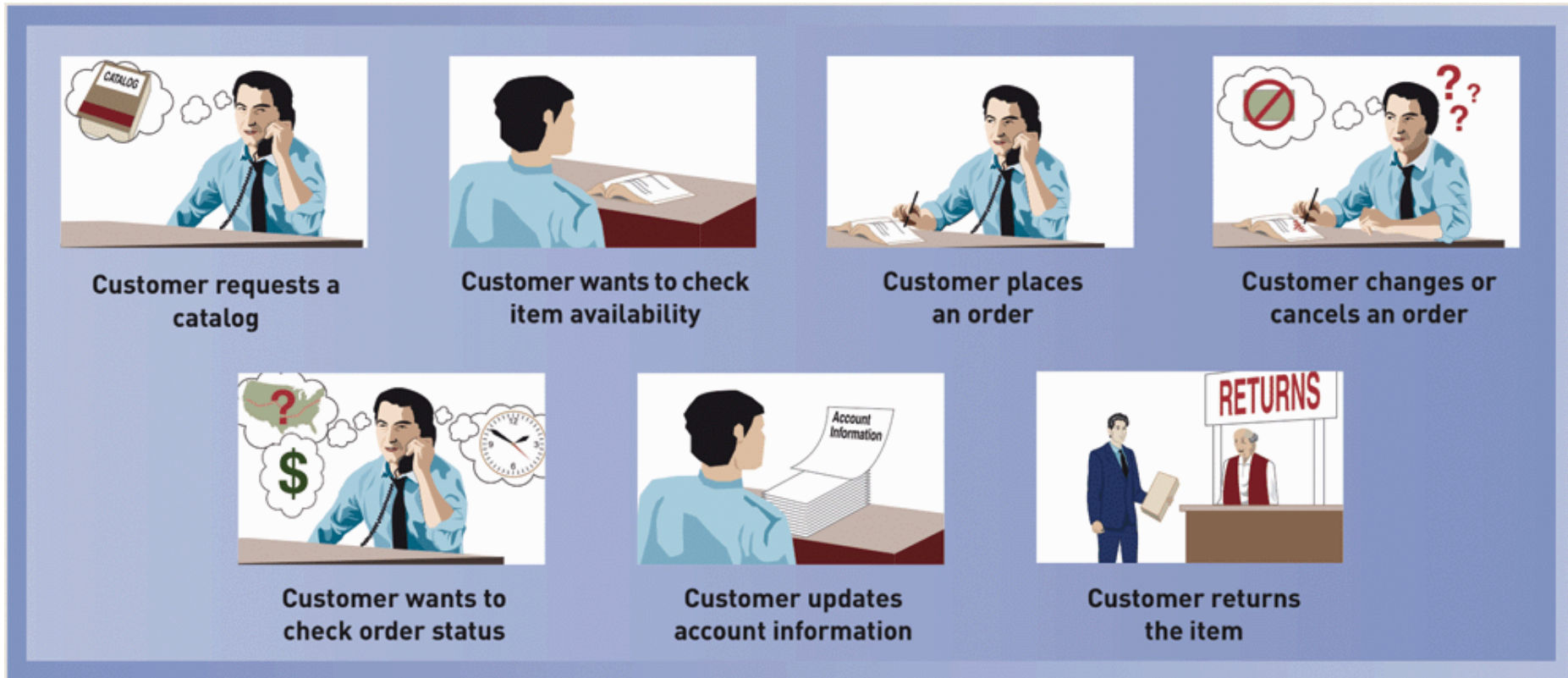

Figure 5-6

Figure 5-7

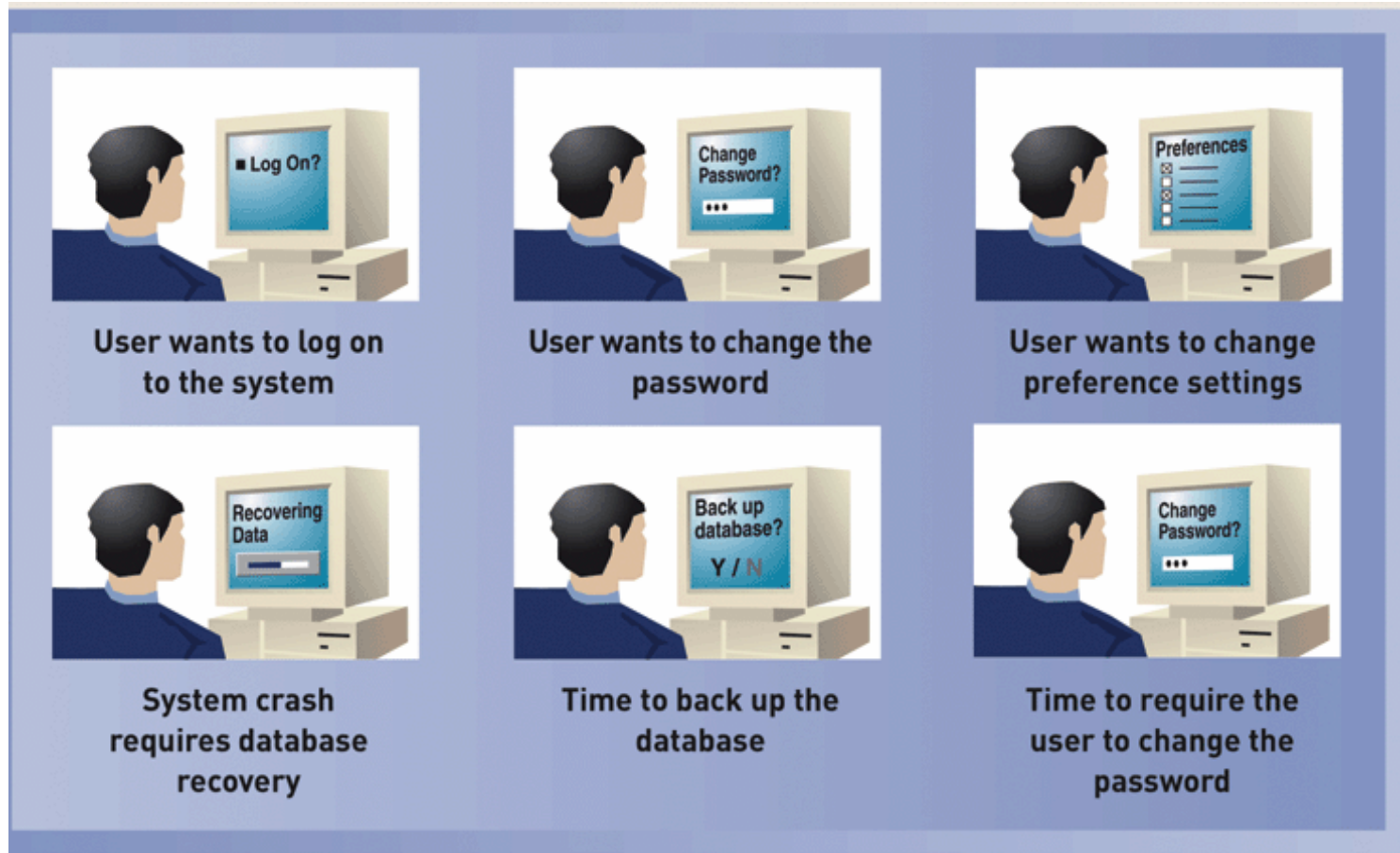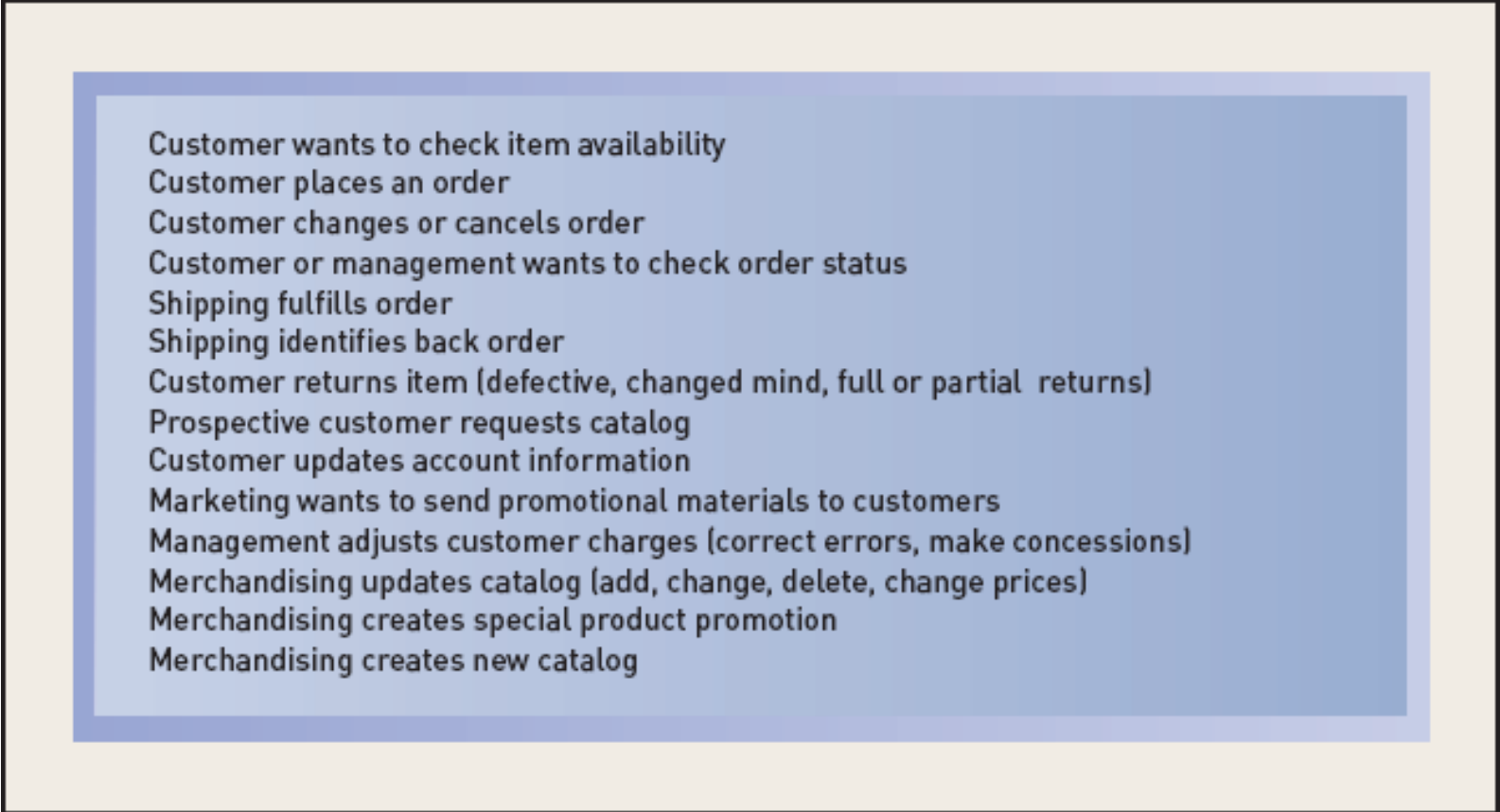# Events Deferred Until the Design Phase

Figure 5-8

Events in the RMO case

- ◆ Important external events involve customers

  - Customer checks item availability, customer places order, customer changes or cancels order

- ◆ Other external events involve departments

  - Shipping fulfills order, marketing sends promotion to customer, merchandising updates catalog

- ◆ Temporal events include periodic reports

  - Time to produce order summary reports, Time to produce fulfillment summary reports

RMO External Events

Customer wants to check item availability
Customer places an order
Customer changes or cancels order
Customer or management wants to check order status
Shipping fulfills order
Shipping identifies back order
Customer returns item (defective, changed mind, full or partial returns)
Prospective customer requests catalog
Customer updates account information
Marketing wants to send promotional materials to customers
Management adjusts customer charges (correct errors, make concessions)
Merchandising updates catalog (add, change, delete, change prices)
Merchandising creates special product promotion
Merchandising creates new catalog

Figure 5-9

RMO Temporal Events

Time to produce order summary reports
Time to produce transaction summary reports
Time to produce fulfillment summary reports
Time to produce prospective customer activity reports
Time to produce customer adjustment/concession reports
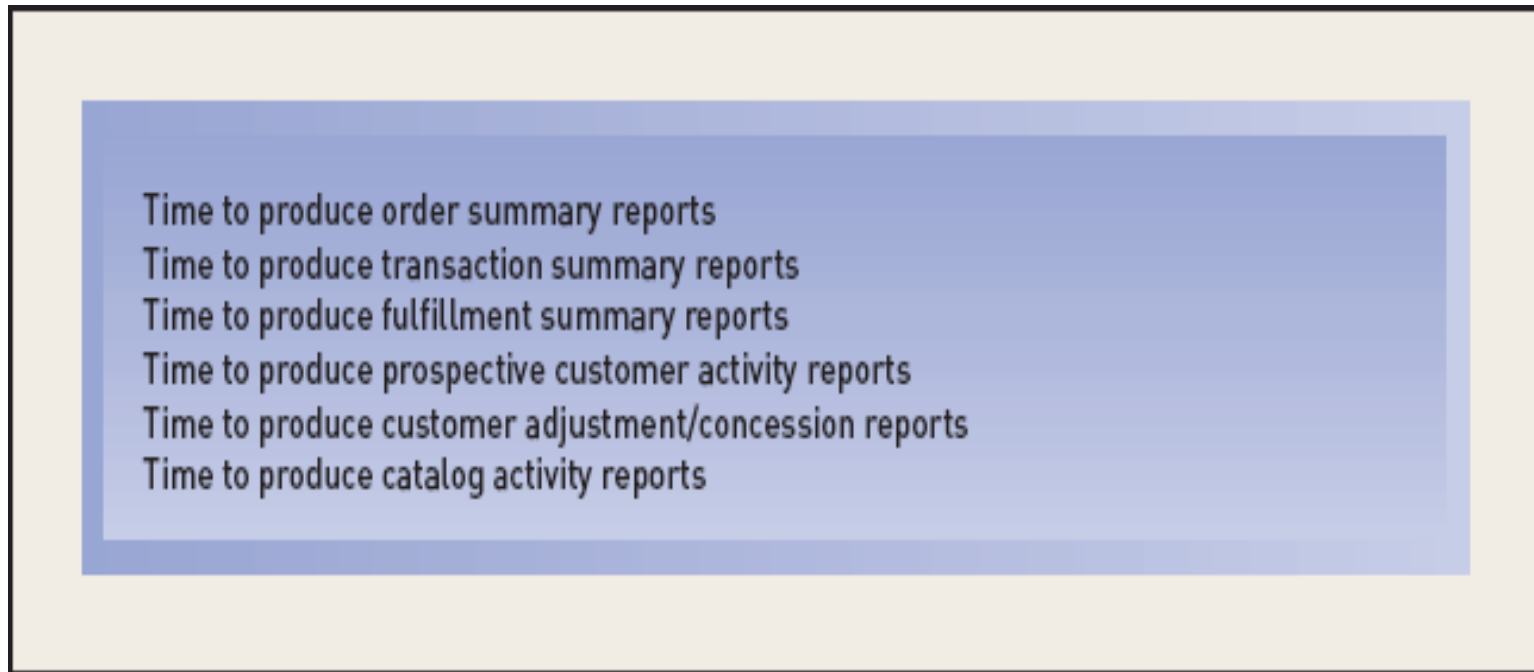Time to produce catalog activity reports

Figure 5-10

Events and Use Cases

◆ Event Table – a catalog of use cases listed by event. Contains detailed information

- Trigger – a signal that indicates an event has occurred

- Source – an external agent that initiates event and supplies data for the event

- Response – an output produced by the system

- Destination – an external agent that receives the response
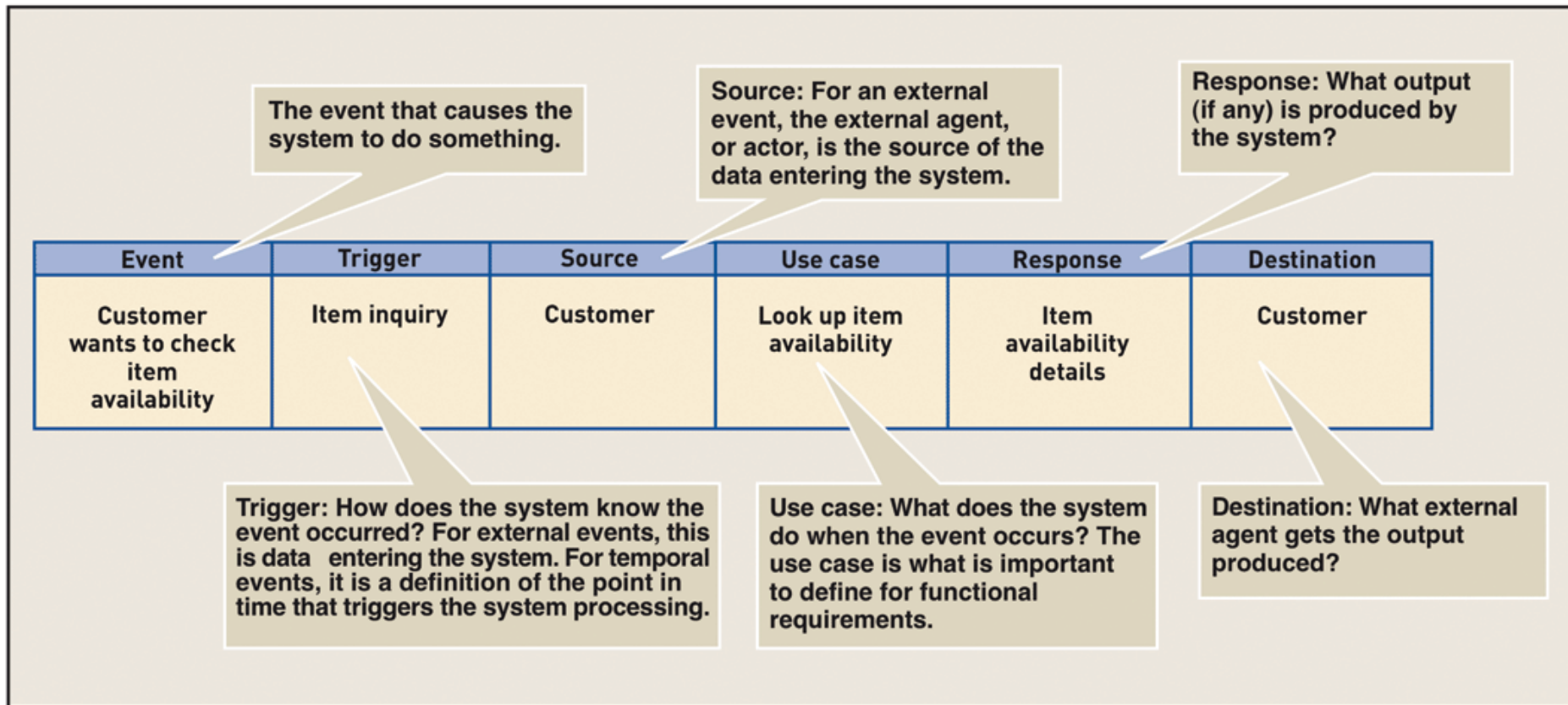
Information about Each Event in an Event Table

The event that causes the system to do something.

Source: For an external event, the external agent, or actor, is the source of the data entering the system.

Response: What output (if any) is produced by the system?

Event	Trigger	Source	Use case	Response	Destination
Customer wants to check item availability	Item inquiry	Customer	Look up item availability	Item availability details	Customer

Trigger: How does the system know the event occurred? For external events, this is data entering the system. For temporal events, it is a definition of the point in time that triggers the system processing.

Use case: What does the system do when the event occurs? The use case is what is important to define for functional requirements.

Destination: What external agent gets the output produced?

Figure 5-11

Systems Analysis and Design in a Changing World, 5th Edition

RMO Event Table

Customer support system event table					
Event	**Trigger**	**Source**	**Use case**	**Response**	**Destination**
1. Customer wants to check item availability	Item inquiry	Customer	Look up item availability	Item availability details	Customer
2. Customer places an order	New order	Customer	Create new order	Real-time link Order confirmation Order details Transaction	Credit bureau Customer Shipping Bank
3. Customer changes or cancels order	Order change request	Customer	Update order	Change confirmation Order change details Transaction	Customer Shipping Bank
4. Time to produce order summary reports	"End of week, month, quarter and year"		Produce order summary reports	Order summary reports	Management
5. Time to produce transaction summary reports	"End of day"		Produce transaction summary reports	Transaction summary reports	Accounting
6. Customer or management wants to check order status	Order status inquiry	Customer or management	Look up order status	Order status details	Customer or management

Figure 5-12

Systems Analysis and Design in a Changing World, 5th Edition 23

Use Case Descriptions

◆ Use case description – a description of the processing steps for a use case

◆ Actor – a person or thing that uses the system. Actors have contact with the system

◆ Scenario or Instance – a particular set of internal steps that represent a unique path of the use case

◆ Three types of descriptions

- Brief description

- Intermediate description

- Fully developed description

Brief Description

Create new order description

When the 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.

Figure 5-13

Intermediate Description

Flow of activities for scenario of *Order Clerk creates telephone order*

Main Flow:

1. Customer calls RMO and gets order clerk.
2. Order clerk verifies customer information. If a new customer, invoke *Maintain customer account information* use case to add a new customer.
3. Clerk initiates the creation of a new order.
4. Customer requests an item be added to the order.
5. Clerk verifies the item and adds it to the order.
6. Repeat steps 4 and 5 until all items are added to the order.
7. Customer indicates end of order; clerk enters end of order; system computes totals.
8. Customer submits payment; clerk enters amount; system verifies payment.
9. System finalizes order.

Exception Conditions:

5. 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.
8. 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.

Figure 5-14

Fully Developed Description

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: Check item availability.
Stakeholders:	Sales department: to provide primary definition. Shipping department: to verify 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 Activities:	Actor	System
	1. Sales clerk answers telephone and connects to a customer.	
	2. Clerk verifies customer information.	2.1 Display customer information.
	3. Clerk initiates the creation of a new order.	3.1 Create a new order.
	4. Customer requests an item be added to the order.	
	5. Clerk verifies the item (Check item availability use case).	5.1 Display item information.
	6. Clerk adds item to the order.	6.1 Add an order item.
	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.	8.1 Complete order. 8.2 Compute totals.
	9. Customer submits payment; clerk enters amount.	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 Maintain customer information 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.

Figure 5-16

"Things" in the Problem Domain

◆ Define system requirements by understanding system information that needs to be stored

◆ Store information about things in the problem domain that people deal with when they do their work
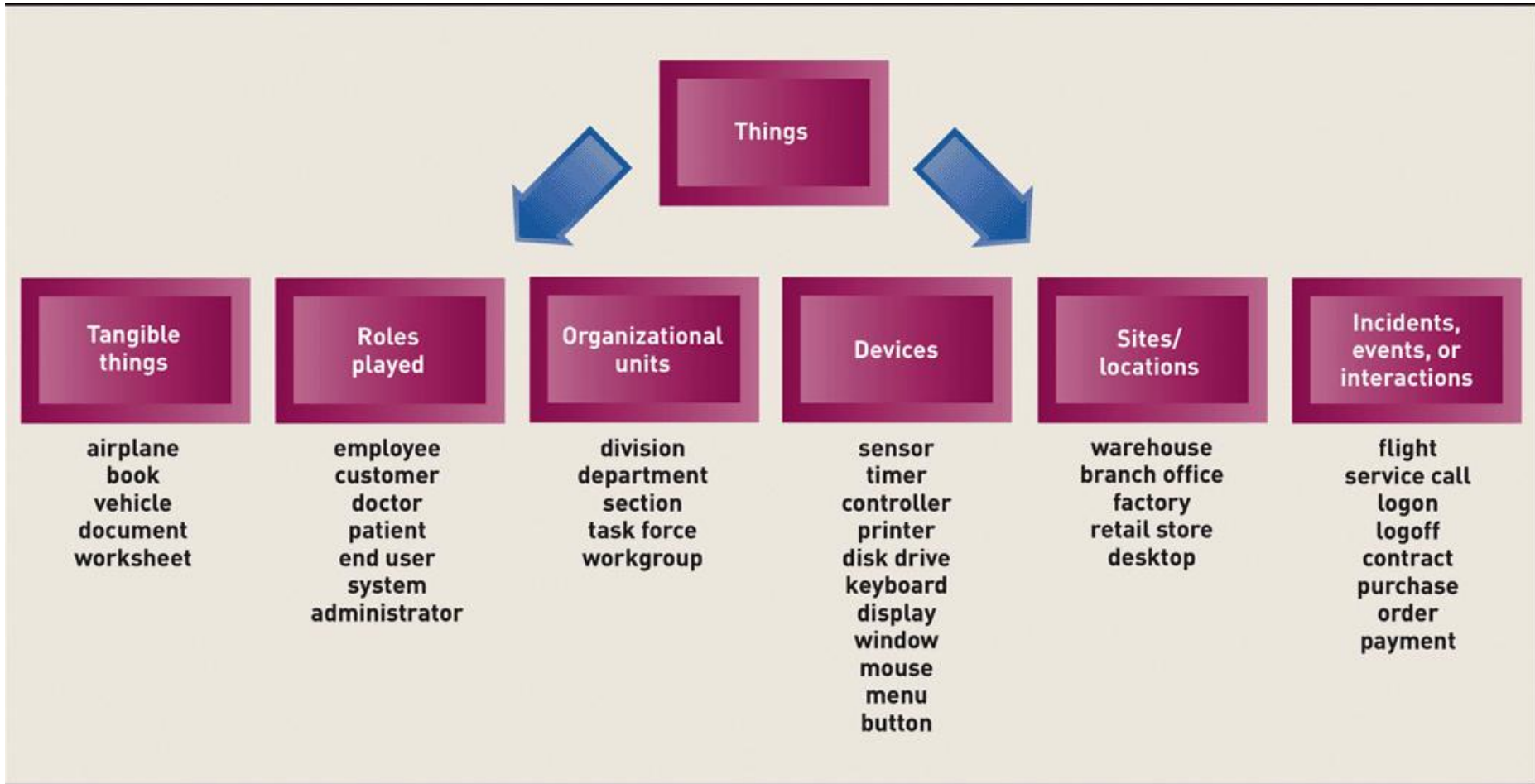
Types of Things

Figure 5-18

Procedure for Developing an Initial List of Things

◆ Step 1: Using the event table and information about each use case, identify all *nouns*

◆ Step 2: Using other information from existing systems, current procedures, and current reports or forms, add items or categories of information needed

◆ Step 3: Refine list and record assumptions or issues to explore

- Questions to include it, exclude it, or research it

RMO Example "Things"

Identified noun	Notes on including noun as a thing to store
Accounting	We know who they are. No need to store it.
Back order	A special type of order? Or a value of order status? Research.
Back-order information	An output that can be produced from other information.
Bank	Only one of them. No need to store.
Catalog	Yes, need to recall them, for different seasons and years. Include.
Catalog activity reports	An output that can be produced from other information. Not stored.
Catalog details	Same as catalog? Or the same as product items in the catalog? Research.
Change request	An input resulting in remembering changes to an order.
Charge adjustment	An input resulting in a transaction.
Color	One piece of information about a product item.
Confirmation	An output produced from other information. Not stored.
Credit card information	Part of an order? Or part of customer information? Research.
Customer	Yes, a key thing with lots of details required. Include.
Customer account	Possibly required if an RMO payment plan is included. Research.
Fulfillment reports	An output produced from information about shipments. Not stored.
Inventory quantity	One piece of information about a product item. Research.
Management	We know who they are. No need to store.
Marketing	We know who they are. No need to store.
Merchandising	We know who they are. No need to store.
Order	Yes, a key system responsibility. Include.
Payment method	Part of an order. Research.
Price	Part of a product item. Research.
Product item	Yes, what RMO includes in a catalog and sells. Include.

Figure 5-19

Characteristics of Things

◆ Relationship

- Naturally occurring association among specific things

- Occur in two directions

- Number of associations is cardinality or multiplicity

  ◆ Binary, unary, ternary, n-ary

◆ Attribute

- One specific piece of information about a thing
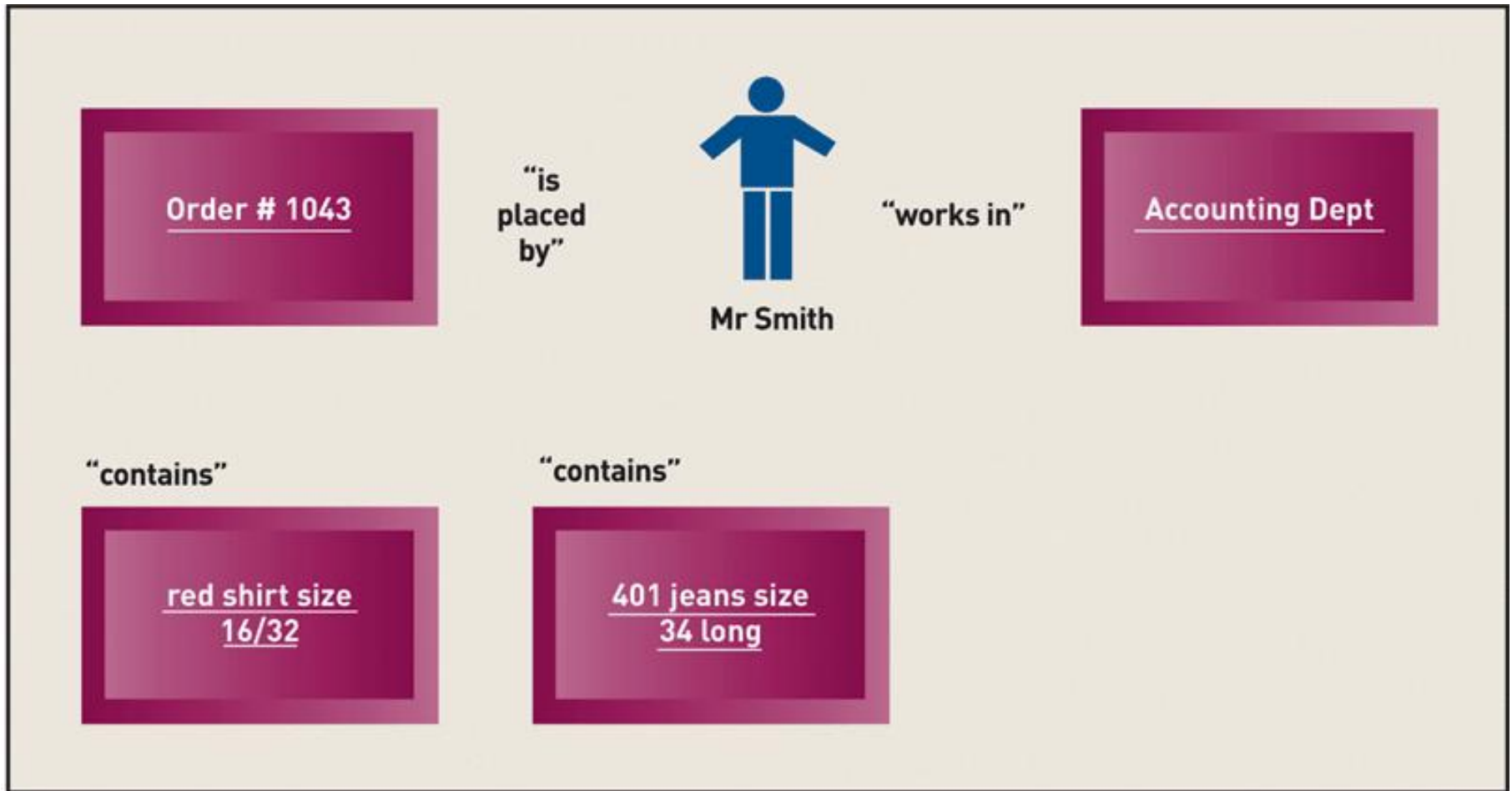
Relationships Naturally Occur Between Things

Figure 5-20
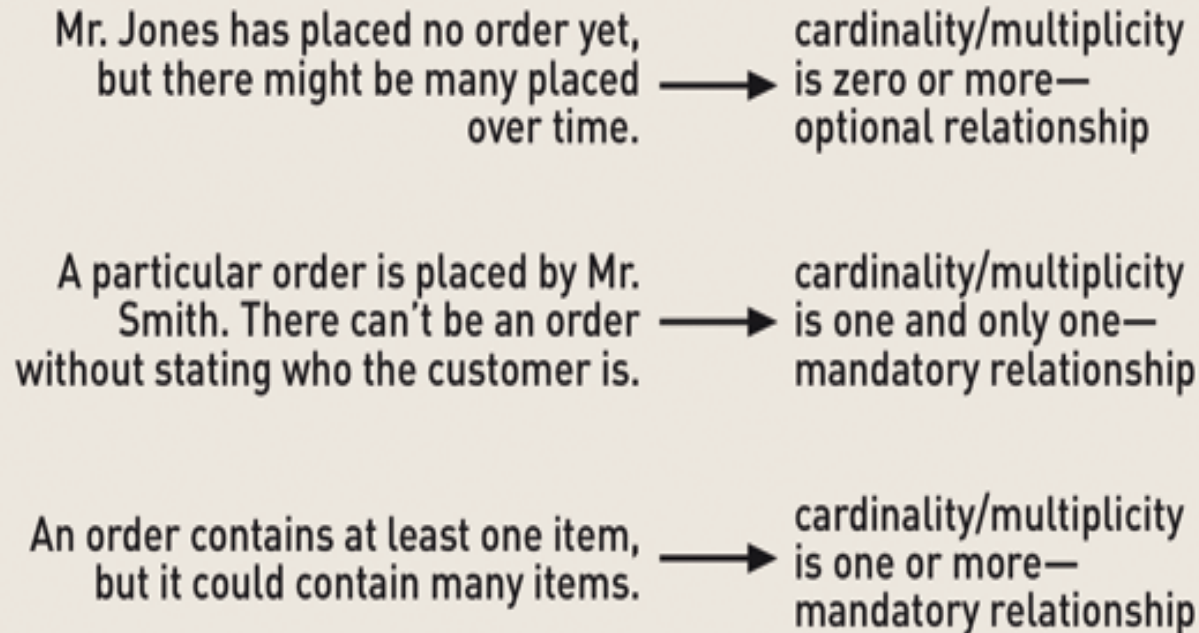
Cardinality/Multiplicity of Relationships

Mr. Jones has placed no order yet, but there might be many placed over time. ⟶ cardinality/multiplicity is zero or more—optional relationship

A particular order is placed by Mr. Smith. There can't be an order without stating who the customer is. ⟶ cardinality/multiplicity is one and only one—mandatory relationship

An order contains at least one item, but it could contain many items. ⟶ cardinality/multiplicity is one or more—mandatory relationship

Figure 5-21

Attributes and Values

All customers have these attributes:	Each customer has a value for each attribute:		
Customer ID	101	102	103
First name	John	Mary	Bill
Last name	Smith	Jones	Casper
Home phone	555-9182	423-1298	874-1297
Work phone	555-3425	423-3419	874-8546

Figure 5-22

Data Entities

◆ Things system needs to store data about in traditional IS approach

◆ Modeled with entity-relationship diagram (ERD)

◆ Requirements model used to create the database design model for relational database

The Entity-Relationship Diagram (ERD)
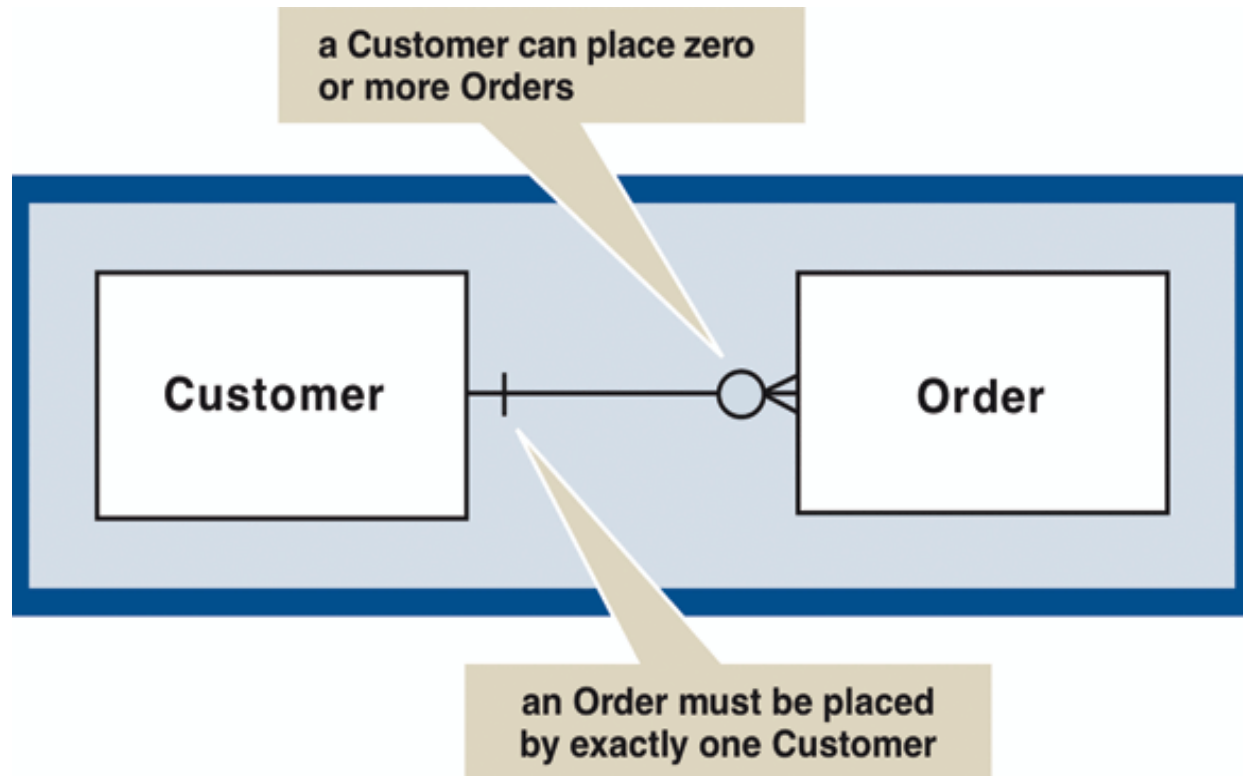
Figure 5-23

Cardinality Symbols of Relationships for ERD
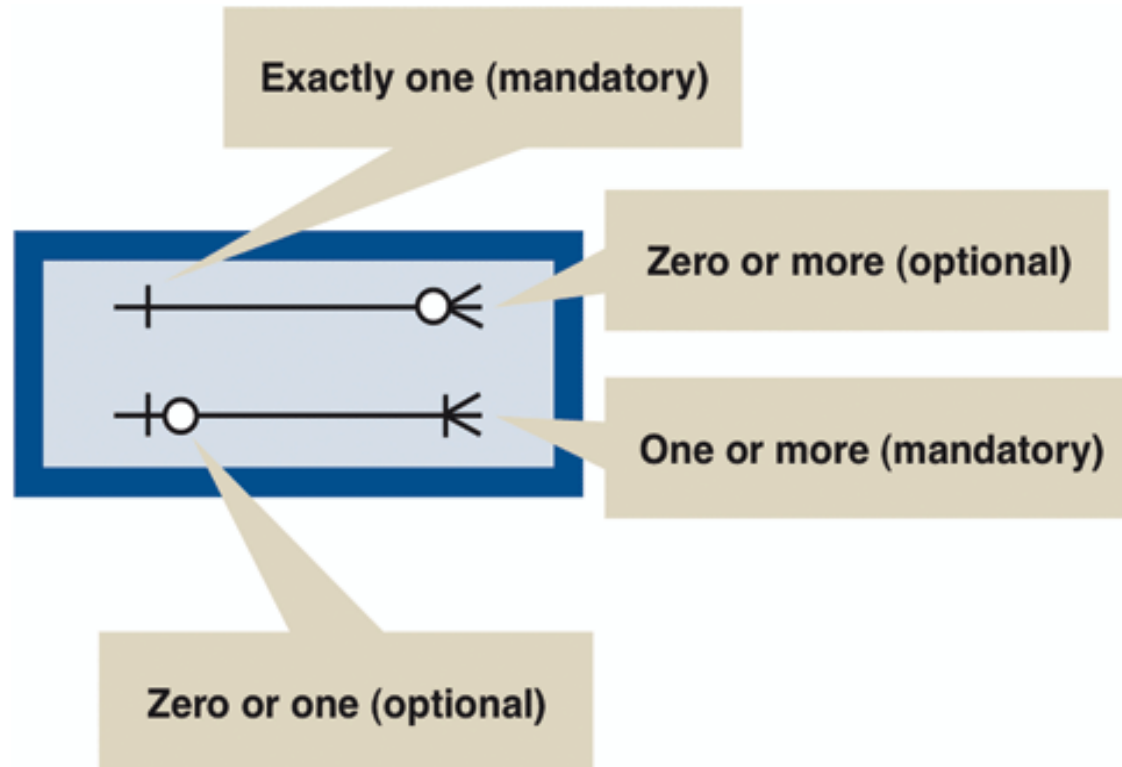
Figure 5-24

Expanded ERD with Attributes Shown
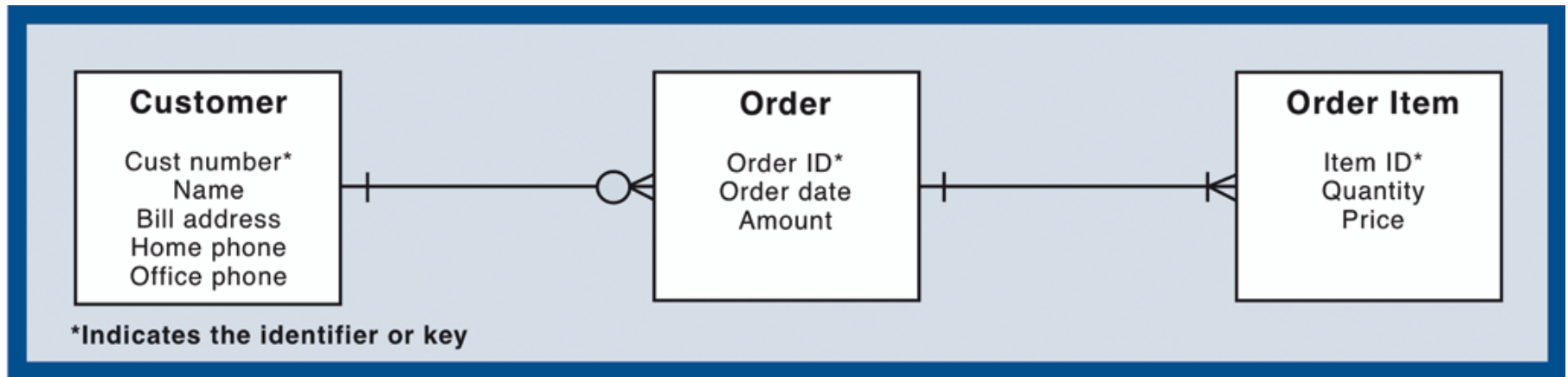
Figure 5-25

Customers, Orders, and Order Items
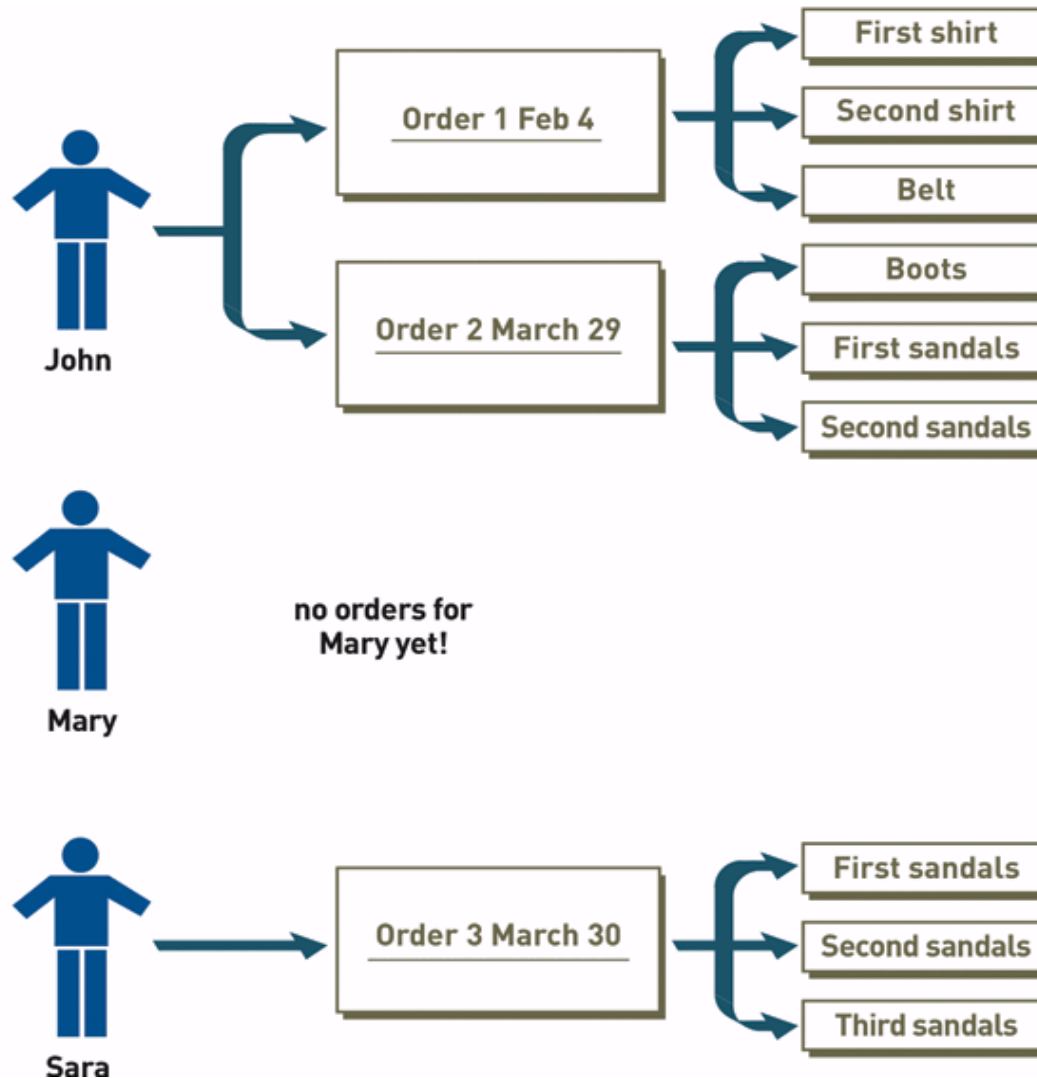
Figure 5-26

ERD with Many-to-Many Relationship
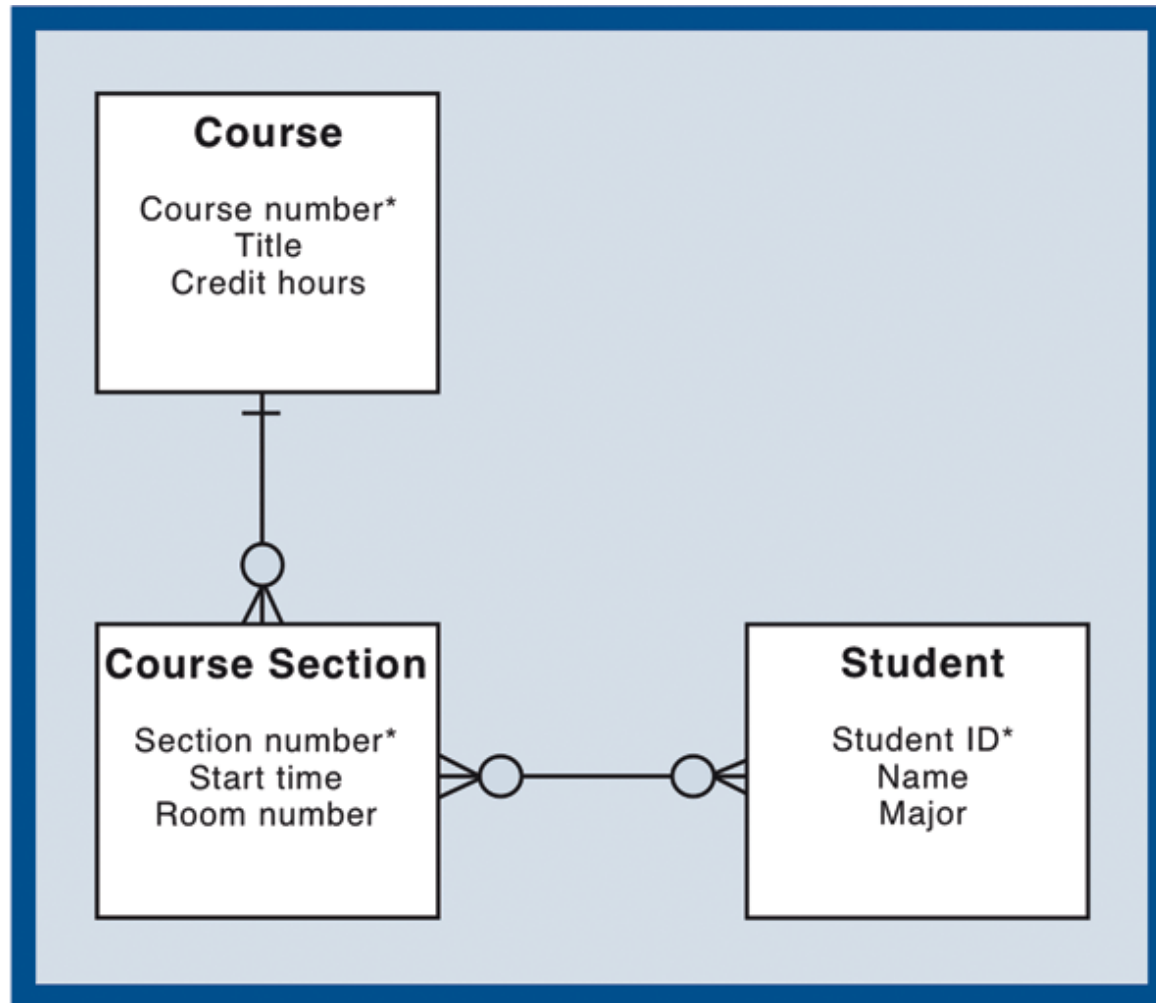
Figure 5-27

Many-to-Many Relationship Converted to Associative Entity to Store Grade Attribute
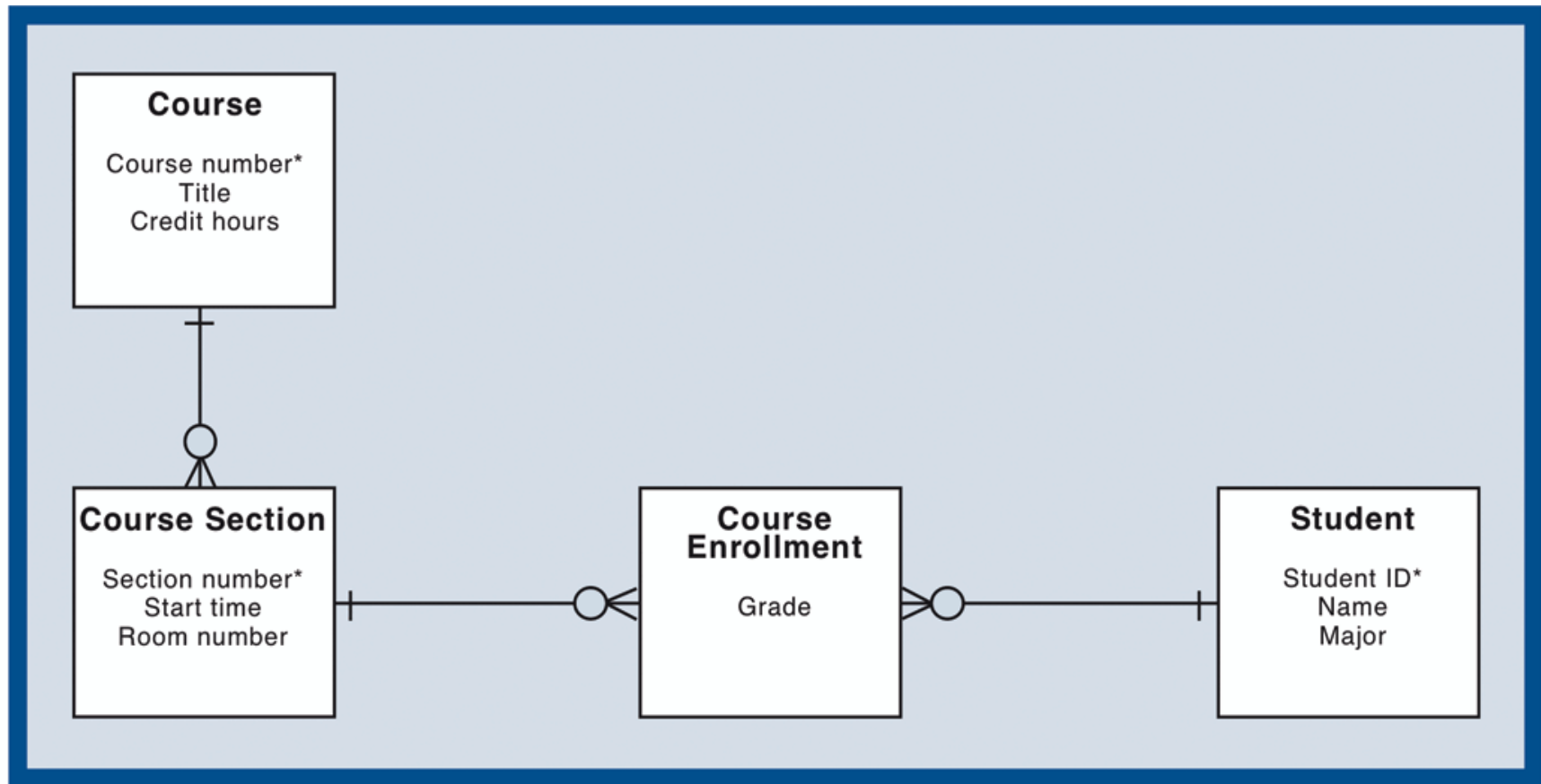
Figure 5-28

RMO Customer Support System ERD
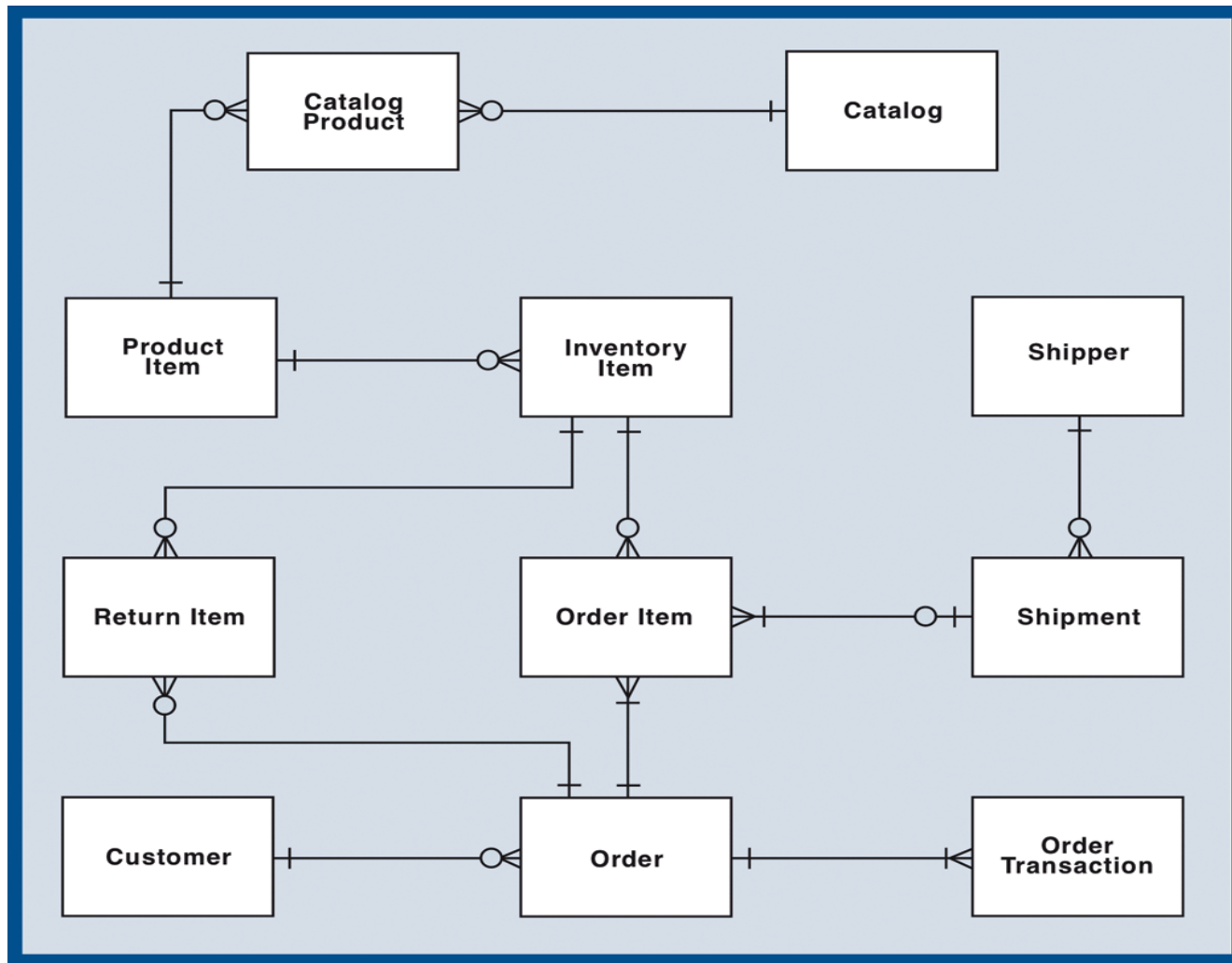
Figure 5-29

The Domain Model Class Diagram

◆ Unified Modeling Language (UML) diagram

◆ Domain model class diagram

- Models things in the users' work domain

- Used to define requirements for OO (very similar to entities in ERD)

UML Class Symbol
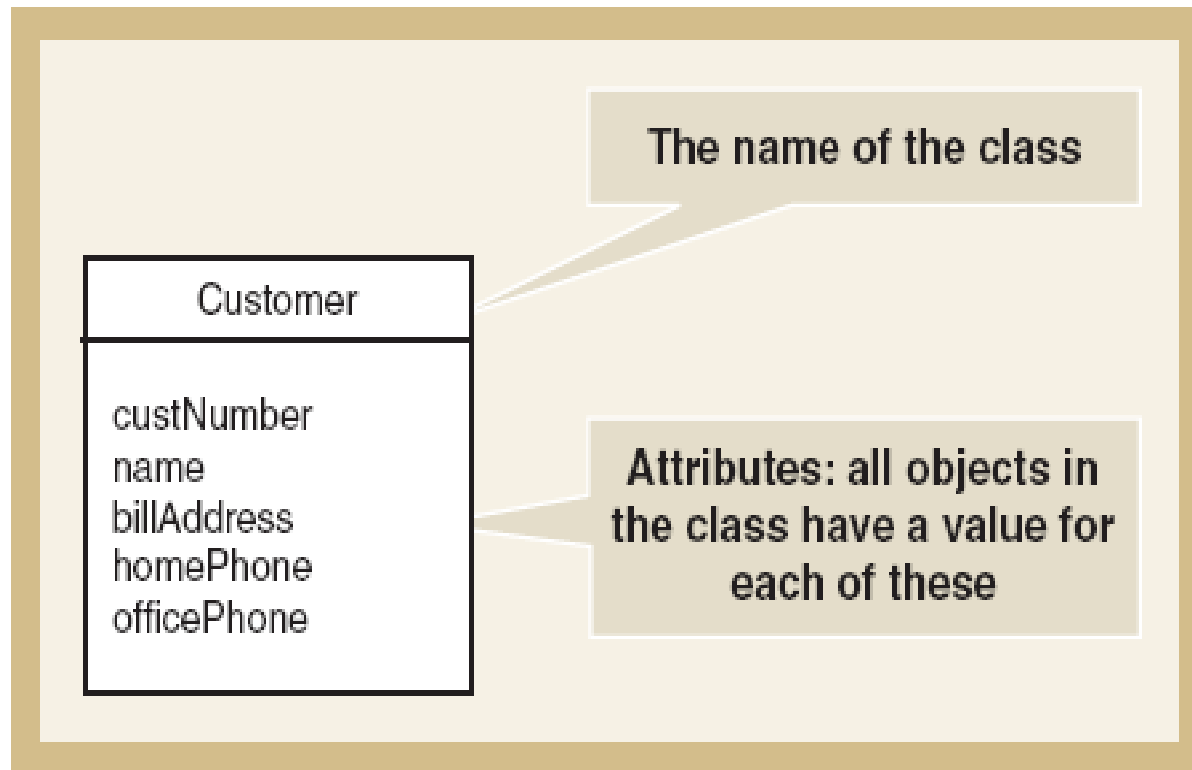
Figure 5-30

Simple Domain Model Class Diagram
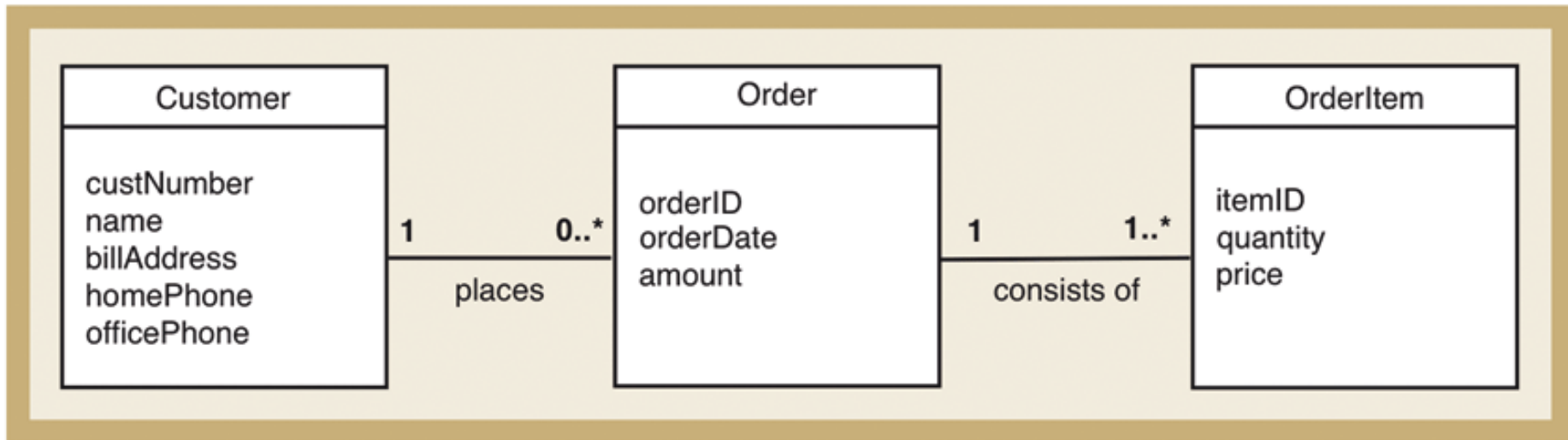
Simple Domain Model Class Diagram (continued)

◆ No methods shown in domain model

- Domain classes are not software classes

◆ Very similar to ERD

- UML and domain model can be used in place of ERD in traditional approach

Multiplicity of Associations
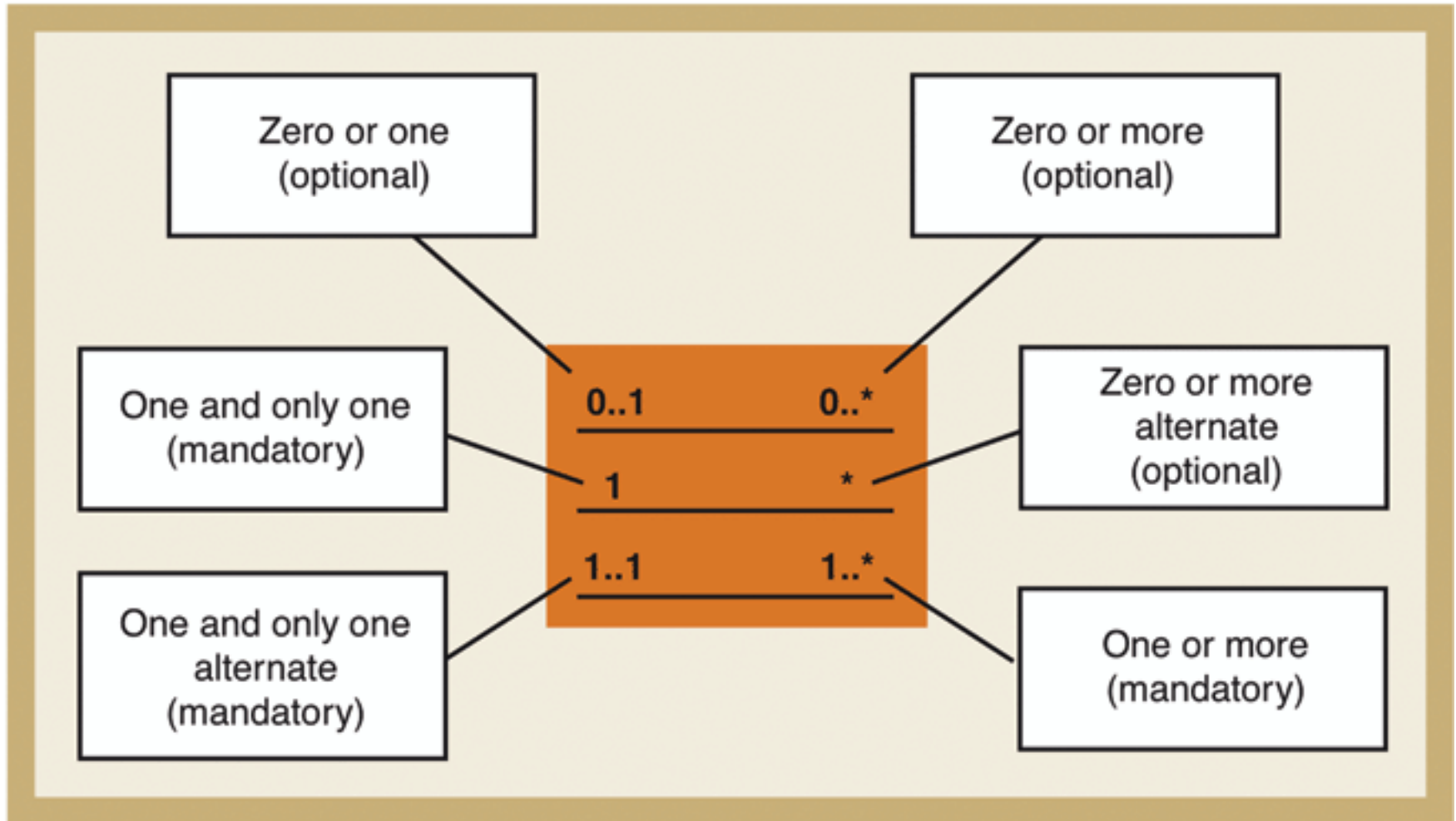
Figure 5-32

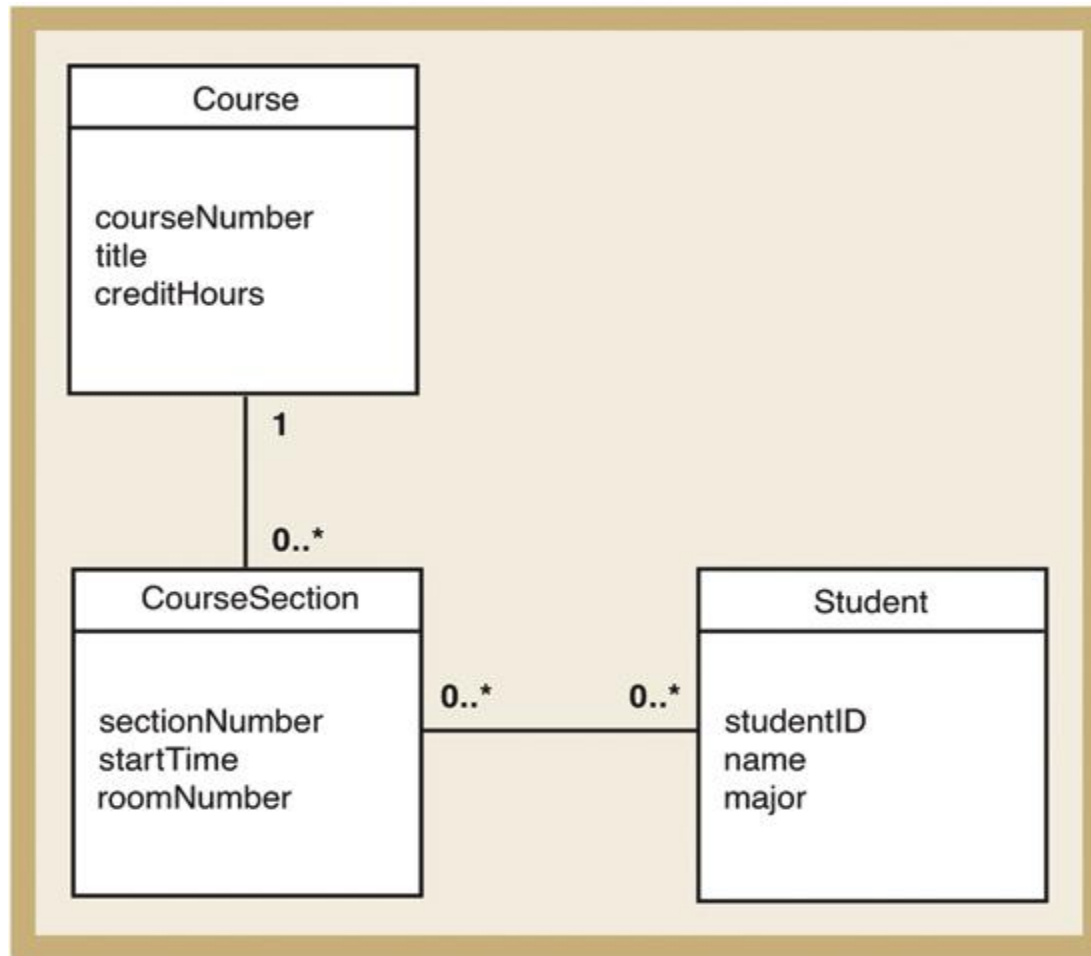# University Course Enrollment Domain Model Class Diagram

Figure 5-33
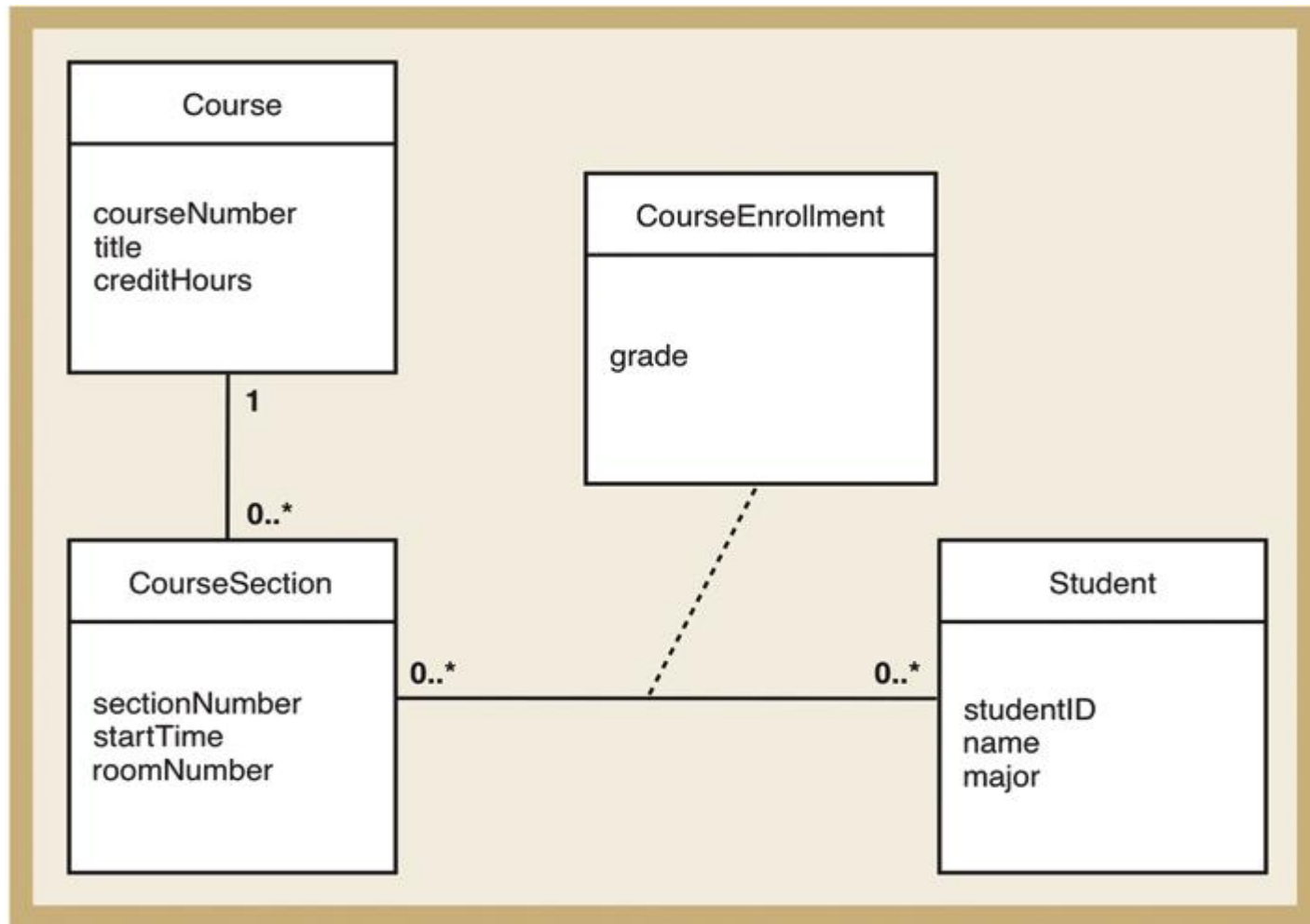
Refined Model with Association Class and Grade Attribute

Figure 5-34

More Complex Class Concepts

◆ Generalization/specialization hierarchies

- General superclasses to specialized subclasses

- Inheritance allows subclasses to share characteristics of their superclasses

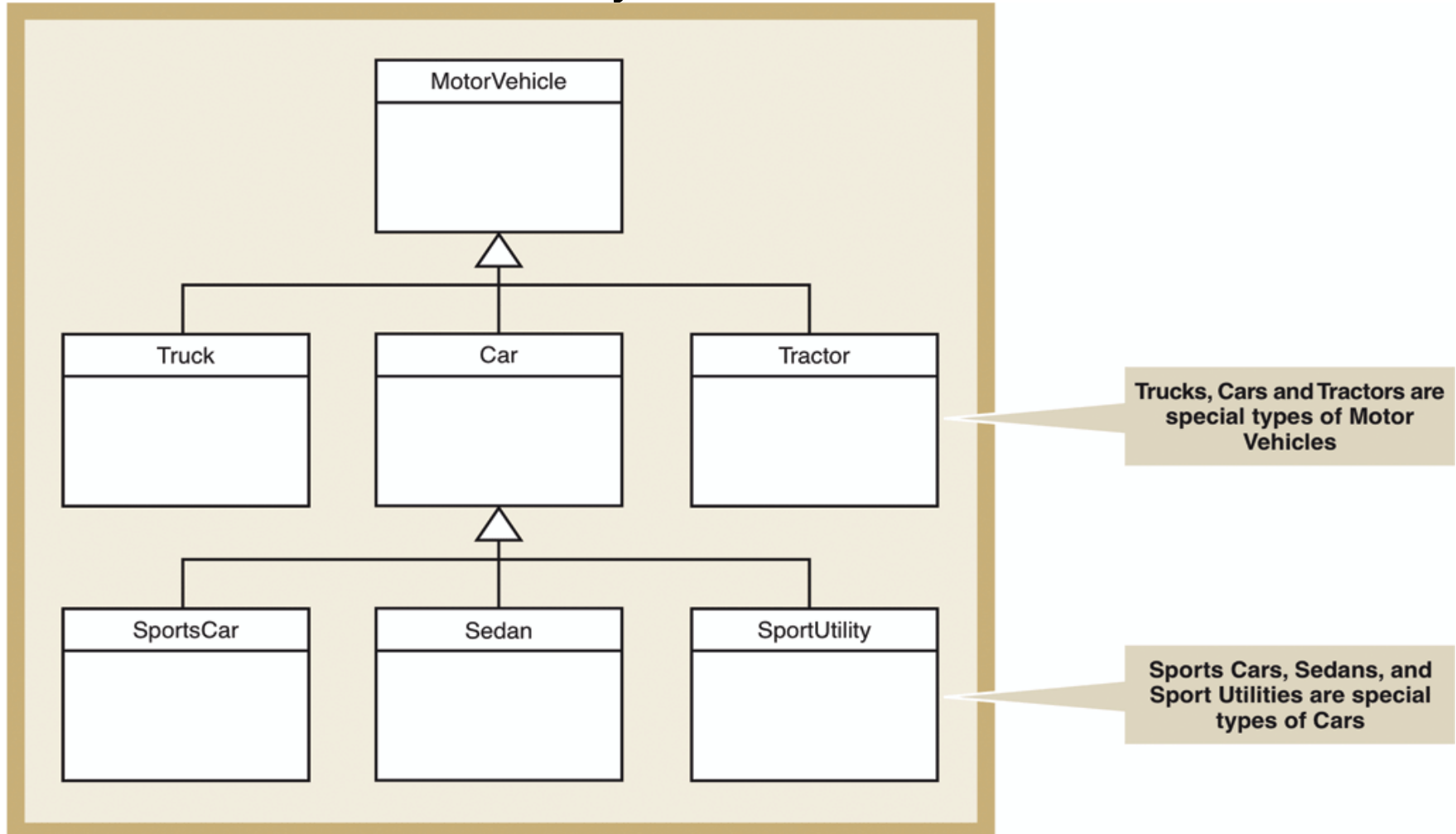# A Generalization/Specialization Class Hierarchy for Motor Vehicles

Trucks, Cars and Tractors are special types of Motor Vehicles

Sports Cars, Sedans, and Sport Utilities are special types of Cars

Figure 5-35

A Generalization/Specialization Class Hierarchy for RMO Orders

Web Orders, Telephone Orders, and Mail Orders are special types of Orders
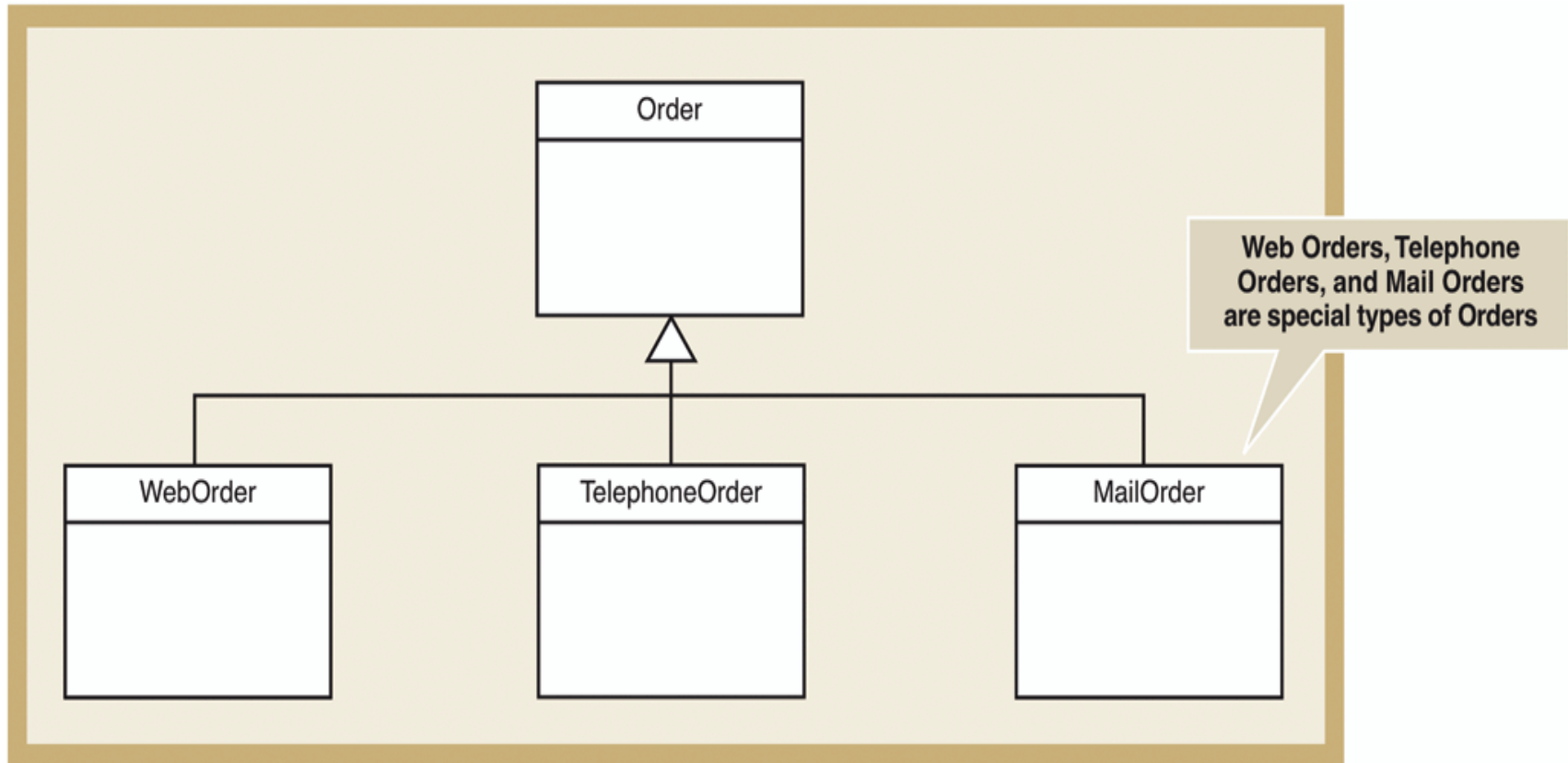
Figure 5-36

Whole-Part Hierarchies

◆ Whole-part hierarchies – hierarchies that structure classes by components

◆ Aggregation – whole-part relationships between and object and its removable parts

- Parts can exist separately

- Like car and its tires

◆ Composition – whole-part relationships between and object and its non-removable parts.

- Parts cannot exist separately

- Like Hand is composed of fingers and thumb
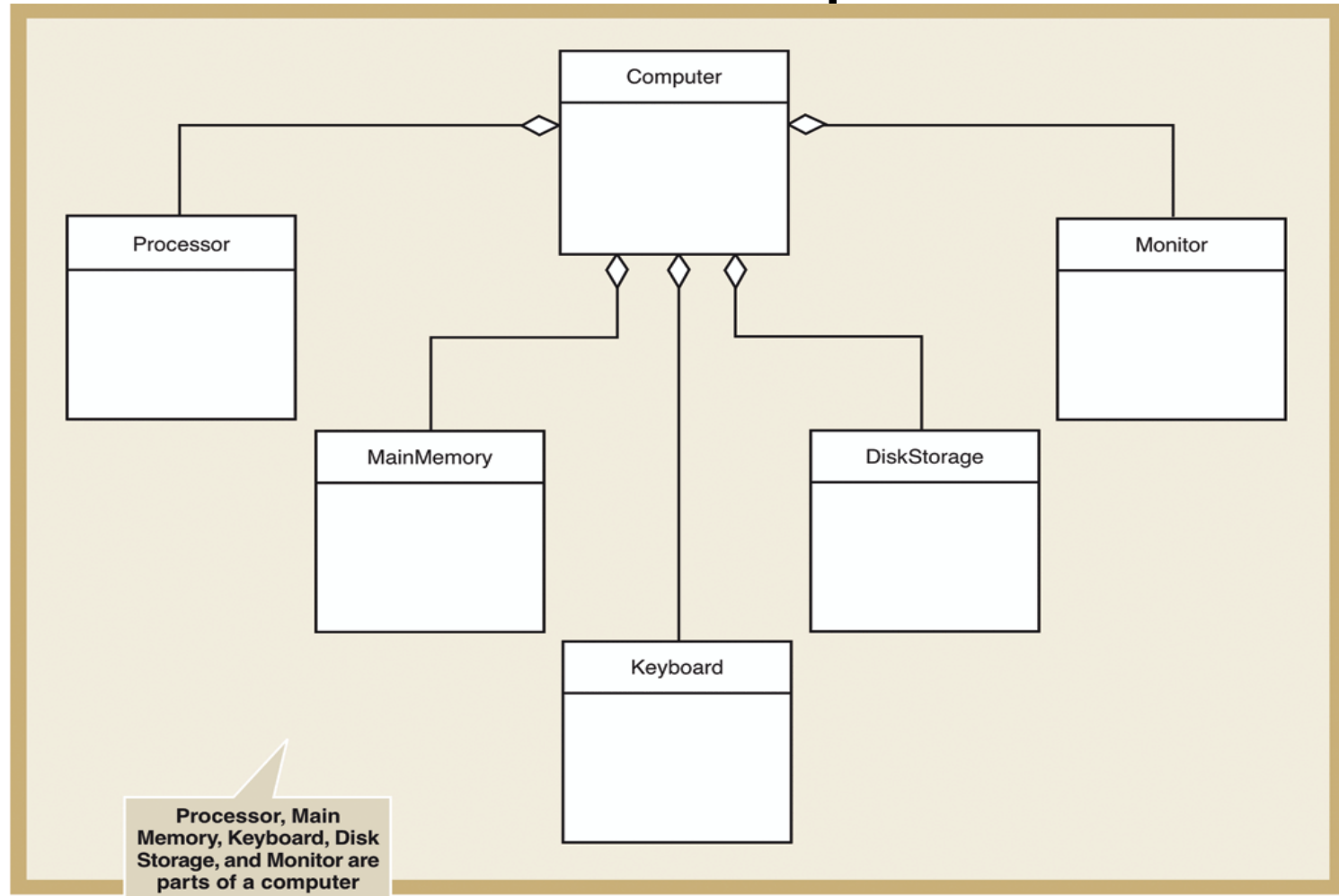
Whole-Part Aggregation Relationships
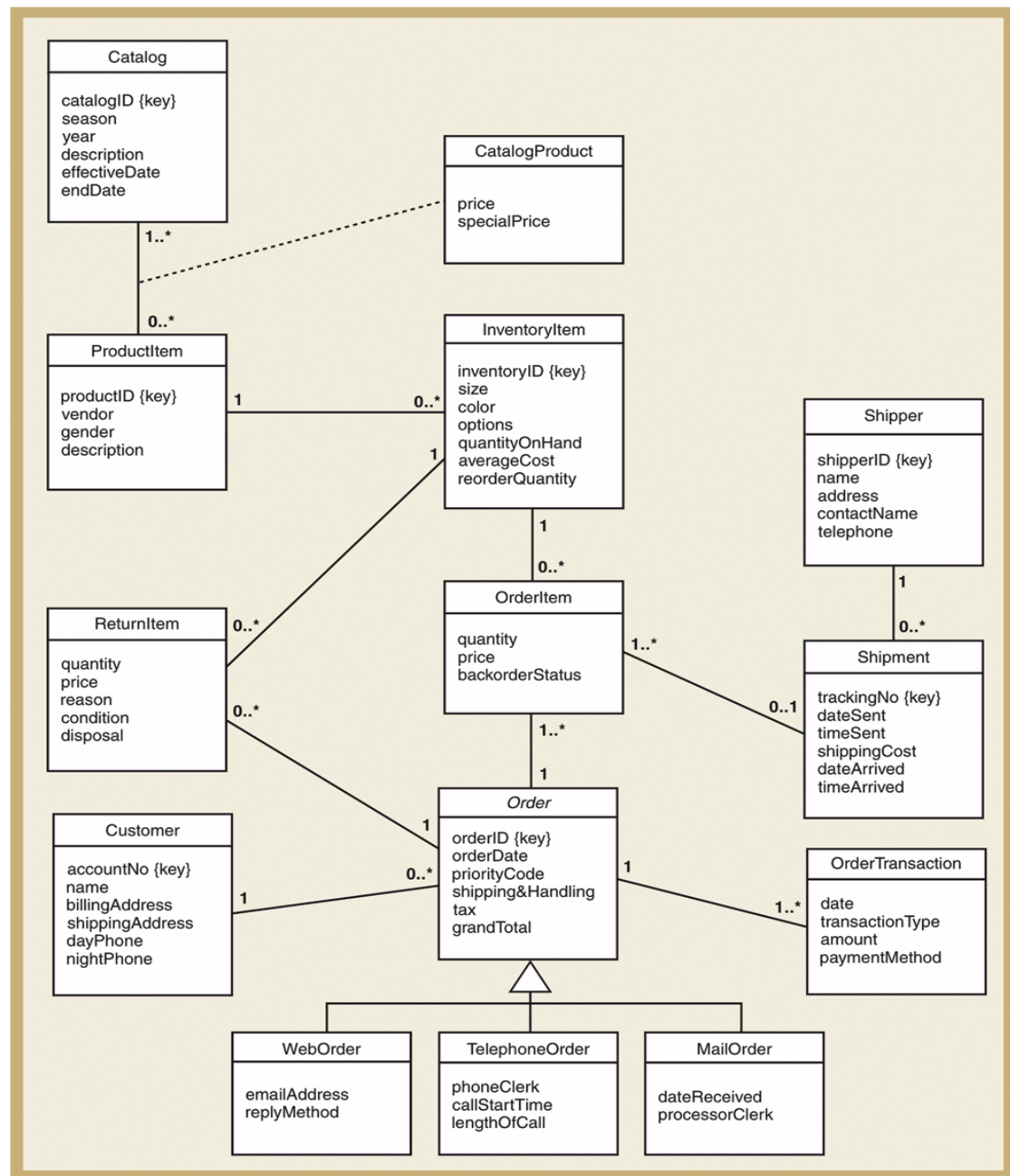

Figure 5-37

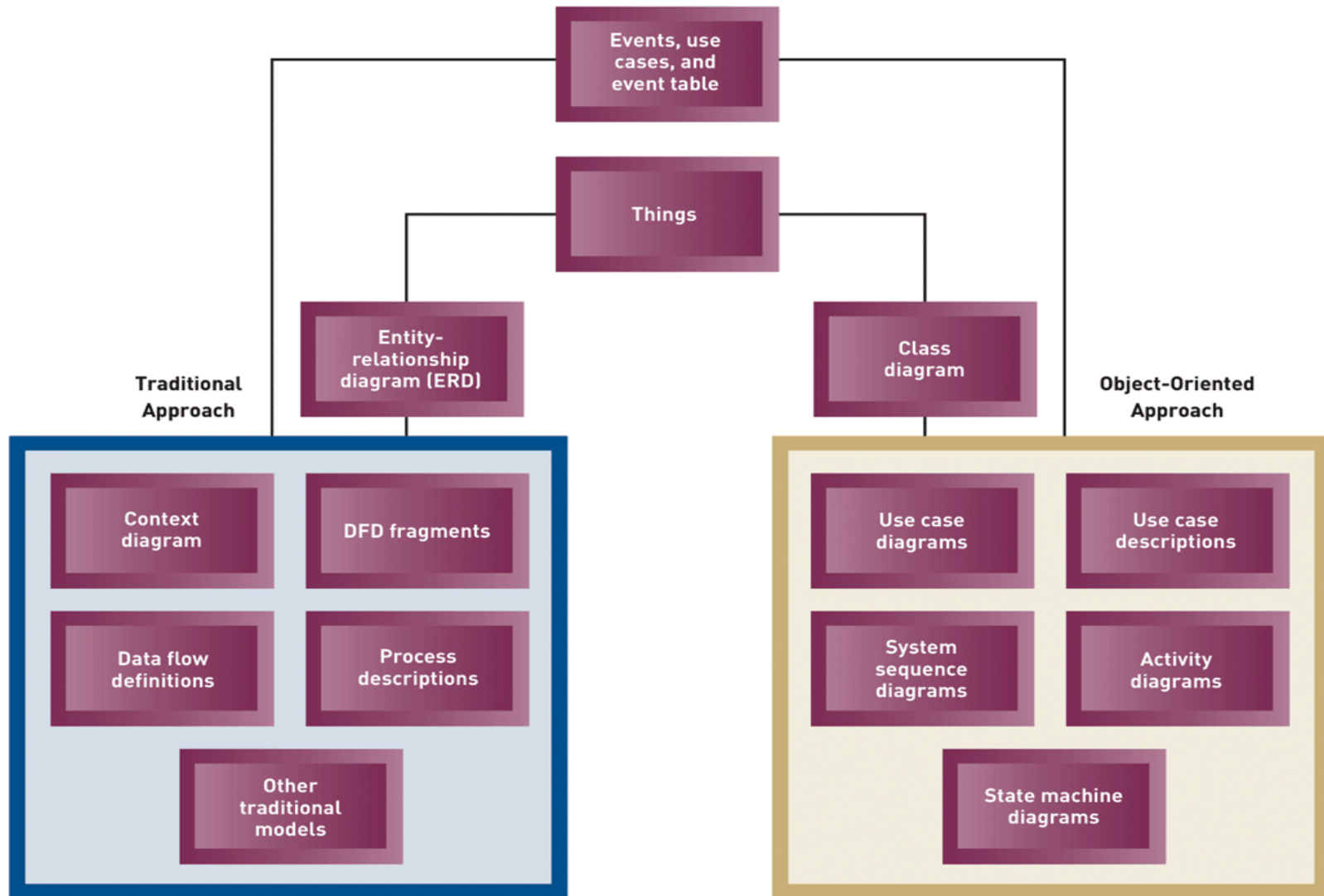# RMO Domain Model Class Diagram

Figure 5-38

Where You Are Headed

Figure 5-39

Systems Analysis and Design in a Changing World, 5th Edition   57

Summary

- ◆ Analysis phase – defines system requirements

- ◆ Models created to further learning process, reduce complexity, communicate with team members, and document requirements

- ◆ Key early step in modeling is to identify and list

  - Events that require a use case in the system

  - Things users deal with in work environment

Summary (continued)

◆ Use cases (activities) are identified from user goals and business events that trigger elementary business processes

◆ Business events are memorable, can be described, and occur at a specific time and place

- External events, temporal events, and state events

◆ Event table records event, trigger, source, use case, response, and destination

- A catalog of information about each use case

Summary (continued)

◆ "Things" are what user deals with and system remembers, such as customer placing an order

◆ Traditional approach uses entity-relationship diagrams (ERD) for data entities, attributes of data entities, and relationships between entities

◆ Object-oriented approach uses UML class diagrams for classes, attributes, methods of class, and associations among classes

● Domain model class diagram