**Chapter 4 Business Process and Functional Modeling  
Testbank**

**Multiple Choices**

1. A(n) \_\_\_\_\_ is a formal way of representing how a business system interacts with its environment.

a. use case

b. physical model

c. relationship

d. system boundary

e. trigger

Ans: a

1. Which of the following diagramming symbols represents an actor in a use case diagram?

|  |  |
| --- | --- |
| a. |  |
| b. |  |
| c. |  |
| d. |  |
| e. |  |

Ans: a

1. Which of the following diagramming symbols represents a use case in a use case diagram?

|  |  |
| --- | --- |
| a. |  |
| b. |  |
| c. |  |
| d. |  |
| e. |  |

Ans: b

1. Which of the following diagramming symbols represents a subject boundary in a use case diagram?

|  |  |
| --- | --- |
| a. |  |
| b. |  |
| c. |  |
| d. |  |
| e. |  |

Ans: c

1. Which of the following diagramming symbols represents a generalization relationship in a use case diagram?

|  |  |
| --- | --- |
| a. |  |
| b. |  |
| c. |  |
| d. |  |
| e. |  |

Ans: d

1. A scenario is the same as a(n) \_\_\_\_\_.

a. use case

b. relationship

c. path through a use case

d. collection of use cases

e. role

Ans: c

1. Each use case describes \_\_\_\_\_ function(s) in which users interact with the system.

a. one

b. one or more

c. many

d. zero, one, or more

e. all

Ans: a

1. Jim has documented a use case that describes the functionality of a system as “To compute gross pay, multiply the hours worked that are recorded on the time card from the time clock by the hourly rate that is recorded in the employee master file from the MS SQL server database.” This is an example of a(n) \_\_\_\_\_ use case.

a. overview

b. detail

c. essential

d. real

e. imaginary

Ans: d

1. A(n) \_\_\_\_\_ use case is typically created early in the process of understanding the system requirements as a way of documenting basic information about the use case.

a. overview

b. detail

c. essential

d. real

e. imaginary

Ans: a

1. The importance level of a use case increases for all of the following characteristics of the use case except \_\_\_\_\_.

a. the use case represents an important business process

b. the use case supports revenue generation

c. the technology is proven

d. the functionality is complex

e. the functionality is time-critical

Ans: c

1. The \_\_\_\_\_ relationship in use cases allows use cases to support the concept of inheritance.

a. association

b. extend

c. include

d. generalization

e. none of these

Ans: d

1. The event that causes a use case to begin is called a(n) \_\_\_\_\_.

a. action

b. trigger

c. hammer

d. anvil

e. stakeholder

Ans: b

1. Which of the following relationships describe the communication between the use case and the actors?

a. association relationship

b. extend relationship

c. include relationship

d. generalization relationship

e. none of these

Ans: a

1. Which of the following relationships represent the extension of the functionality of the use case to cover optional behavior?

a. association relationship

b. extend relationship

c. include relationship

d. generalization relationship

e. optional relationship

Ans: b

1. Which of the relationship types in use cases enables functional decomposition?

a. association relationship

b. extend relationship

c. include relationship

d. generalization relationship

e. decompose relationship

Ans: c

1. Individual steps in a use case should be written in the form \_\_\_\_\_.

a. Direct Object, Verb, Subject, Preposition, Indirect Object

b. Direct Object, Subject, Verb, Preposition, Indirect Object

c. Subject, Verb, Preposition, Indirect Object, Direct Object

d. Subject, Verb, Indirect Object, Preposition, Direct Object

e. Subject, Verb, Direct Object, Preposition, Indirect Object

Ans: e

1. If a use case becomes too complex, it should be \_\_\_\_\_.

a. rewritten to simplify it

b. decomposed into a set of use cases

c. written with a series of repeating steps to simplify it

d. written from the perspective of an independent observe to simplify it

e. dropped from the system, as it will be too complex to implement in the final system

Ans: b

1. The correct sequence of the major steps in creating use case diagrams is \_\_\_\_\_.

a. identify the major use cases, expand the major use cases, confirm the major use cases, create the use-case diagram

b. identify the major use cases, expand the major use cases, create the use-case diagram, confirm the major use cases

c. create the use-case diagram, identify the major use cases, expand the major use cases, confirm the major use cases

d. create the use-case diagram, identify the major use cases, confirm the major use cases, expand the major use cases

e. identify the major use cases, confirm the major use cases, expand the major use cases, create the use-case diagram

Ans: a

1. When drawing the use-case diagram, an analyst should do the steps in this order: \_\_\_\_\_.

a. draw the use cases on the diagram, identify the system boundary, place the actors on the diagram, and draw the lines connecting the actors to the use cases

b. identify the system boundary, draw the use cases on the diagram, place the actors on the diagram, and draw the lines connecting the actors to the use cases

c. place the actors on the diagram, draw the use cases on the diagram, identify the system boundary, and draw the lines connecting the actors to the use cases

d. identify the system boundary, place the actors on the diagram, draw the use cases on the diagram, and draw the lines connecting the actors to the use cases

e. none of these gives the correct order of steps

Ans: b

1. Object nodes model these objects in an activity diagram. Object nodes are portrayed in an activity diagram as \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

a. arcs

b. ovals

c. diamonds

d. rectangles

e. rounded rectangles

Ans: d

1. A fork node in an activity diagram is used to \_\_\_\_\_\_\_\_\_\_ behavior into a set of parallel or concurrent flows of activities (or actions).

a. combine

b. create

c. initialize

d. split

e. represent

Ans: d

1. The purpose of a walkthrough is to thoroughly test the fidelity of the functional models to the \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ and to ensure that the models are consistent.

a. non-functional models

b. use-case diagrams

c. activity diagrams

d. functional requirements

e. use-case descriptions

Ans: d

1. Activity diagrams, use-case descriptions, and use-case diagrams are three different representations for the \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

a. functional and non-functional model

b. functional model

c. non-functional model

d. design model

e. architectural model

Ans: b

**True/False**

1. Use cases can be used to document both the current (As-Is) system and the future (To-Be) system.

Ans: True

1. Use cases are the primary drivers for all of the UML diagramming techniques.

Ans: True

1. An essential use case describes the specific set of steps to be followed.

Ans: False

1. A use case that describes functionality that is complex and risky would be given a high importance level.

Ans: True

1. The primary actor is the person or thing that starts the execution of a use case.

Ans: True

1. The include relationship represents the optional inclusion of another use case.

Ans: False

1. Inheritance is supported in use cases through the generalization relationship.

Ans: True

1. An alternate or exceptional flow in a use case documents the decomposition of the normal flow of events.

Ans: False

1. The SVDPI form of sentences in use cases aids in the identification of classes.

Ans: True

1. A very complex use case should be broken down into a set of use cases.

Ans: True

1. An actor is represented in a use case diagram by a stick figure of a man.

Ans: True

1. An include relationship is modeled in a use case diagram by an arrow with the word <<include>> above it.

Ans: True

1. An actor in a use case must be a person.

Ans: False

1. An actor in a use case always represents a specific user.

Ans: False

1. A formal way of representing how a business system interacts with its environment is termed a use case.

Ans: True

1. Rectangles are used to represent association relationships in use case diagrams.

Ans: False

1. Each use case can be associated with one or more role(s) that users have in the system.

Ans: False

1. A use case that represents an important business process and involves the use of new technology would likely be given an importance level of medium.

Ans: False

1. A use case that begins at the end of the month has a temporal trigger.

Ans: True

1. The use case Take Order has a temporal trigger if it begins when a customer calls to place an order.

Ans: False

1. As Felix is documenting an order entry system, he discovers that someone can call up to place an order that is not a customer. In this case, the Take Order use case will use the Create Customer use case to capture the customer information, and then the order will be taken. This is an example of the extend relationship between use cases.

Ans: True

1. A use case should be written from the perspective of at least one of the actors associated with the use case.

Ans: False

1. A solid line without arrows between an actor and a use case in a use case diagram represents a one-way flow of communication from the actor to the use case.

Ans: False

1. Use cases are typically written to document the normal flow of events. Exceptions that occur in the normal flow need not be documented as part of the use case description.

Ans: False

1. An asterisk on a relationship in a use case diagram represents multiplicity of the association.

Ans: True

1. When drawing use case diagrams, higher level use cases are drawn below lower level use cases.

Ans: False

1. The first step in creating use case diagrams is to identify the primary actors and their goals.

Ans: False

1. Object nodes in an activity diagram are depicted using rectangles.

Ans: True

1. Control flows in an activity diagram show the flow of objects into and out of activities.

Ans: False

1. Control flows in an activity diagram are shown using solid-lines with arrows while object flows are shown using dashed lines with arrows.

Ans: True

1. A decision node in an activity diagram is used to represent the actual test condition that is used to determine which of the paths exiting the node is to be traversed.

Ans: True

1. Paths coming out of a decision node are always mutually-exclusive, while the paths coming out of a fork node are sometimes concurrent and sometimes mutually-exclusive.

Ans: False

1. Join nodes in an activity diagram are used to bring concurrent or parallel flows together back into a single flow.

Ans: True

1. A guard condition represents the value of the decision-test, based on which a particular path from the decision node will be traversed.

Ans: True

1. Setting the scope of the activity being modeled is a task that can be accomplished after completing the activity diagram.

Ans: False

1. Activity diagram is a representation for the non-functional model.

Ans: False

1. Activity diagram, use-case diagram and use-case descriptions are three different representations for the non-functional model.

Ans: True

1. Verification and validation through a walkthrough is to uncover and correct errors or faults in the evolving specification.

Ans: False

1. The presenter role in a walkthrough team is more important than the role maintenance oracle.

Ans: False

1. For a walkthrough to be successful, the members of the walkthrough team MUST be fully prepared.

Ans: True

1. When comparing a use-case description to a use-case diagram for functional model verification and validation, there must be one and only one use-case description for each use case, and vice versa..

Ans: True

1. A join node in an activity diagram is used to split behavior into a set of parallel or concurrent flows of activities (or actions).

Ans: False

1. All object-oriented systems development approaches are use-case driven, architecture-centric, and iterative and incremental.

Ans: True

**Short Answer**

1. Write a complete use case description for the interaction that a student would have with your college or university’s class registration system.

Ans: There is no single correct answer to this question, as it will depend on the particular details of your school’s system. Also, it may vary based on the student’s perspective on the system. For example, at most schools athletes go through a different registration process than the rest of the students.

This use case description, though, should document the following to be correct:

* Use case name
* Primary actor
* Use case type
* Stakeholders and interests
* Brief description
* Trigger
* Relationships
* Normal flow of events
* Alternate/exceptional flows (this should include closed classes or others where special permits may be needed, holds on registration because of uncleared financial situations, etc.)

1. How does a project team go about obtaining the information they need to create use case descriptions?

Ans: The project team must work closely with the users as they develop the use cases. Often the project team will use interviews, JAD sessions, and observation to gain this information. In reality, the techniques are similar to those discussed in Chapter 4 of the text. The key will be keeping in mind that a use case is associated with a role that a person plays in the system, not with a specific person.

1. What are the four types of use cases? When are each used?

Ans: Use cases will vary based on the purpose of the use case and the level of detail they contain. Thus, there will be overview versus detail use cases, and essential versus real ones.

Overview use cases allow the analyst and the user to agree on a high-level overview of the requirements. This use case will only document information such as the name, ID number, primary actor, type, and a brief description.

Detail use cases will document all of the information for the use case. These will be developed after all of the overview use cases have been agreed to by the users and the analysts.

An essential use case describes only the minimum information necessary to understand the required functionality of the system.

A real use case goes beyond the essential use case and describes the specific set of steps required to understand the functionality of the system. Essential use cases are implementation independent whereas real use cases are detailed descriptions of how to use the system once it is implemented.

1. What are the five characteristics of a use case that impacts its importance level? Describe each, and indicate what conditions would result in a higher importance level.

Ans: The use case may represent an important business process, which results in a higher importance level. The use case may support the generation of revenue or cost reduction. If so, this would result in a higher importance level. If the technology needed to support the use case is new or risky, that will result in a higher importance level. If the functionality in the use case is complex, risky, and/or time-critical, then the importance level will be higher. If the use case could help increase the level of understanding of the evolving design of the system, and this increase in understanding comes with little relative effort, then the importance level will be higher.

1. Distinguish between the extend and include relationships in use case diagrams. When would you use each of these relationships?

Ans: An *extend relationship* documents the extension of the functionality of the use case to incorporate optional behavior. For example, if the use case Register for Classes describes the registration process at your university, the use case Clear Financial Hold would only be executed for students who have financial holds on their registration, and the relationship between these two use cases would be an extend relationship. An *include relationship* represents the mandatory inclusion of another use case, which would happen if you had decomposed a particularly complex use case into several simpler ones. The text points out that this type of relationship allows for functional decomposition.

1. Distinguish between the association and generalization relationships in use case diagrams. When would you use each of these relationships?

Ans: An *association relationship* documents the communication between the use case and the actors that use the use case. All actors involved in the use case will be documented with the association relationship. The *generalization relationship* allows use cases to support the concept of inheritance, since use cases can inherit the behaviors of other use cases. For example, Athlete Registration and New Student Registration can both have generalization relationships with Register for Classes, which would be used by the typical student.

1. Explain the difference between normal flows and subflows that would be documented in a use case.

Ans: The *normal flow* of events describes the steps that are normally executed in a use case. The steps in a normal flow are listed in the order in which they are performed. In some cases it is recommended that the steps in a normal flow be decomposed to simplify the normal flow. Thus, *subflows* are used to help keep the normal flow of events as simple as possible. Subflows represent the decomposition of normal flow steps. At times, subflows could be replaced with a separate use case, if it makes sense.

1. Explain the usefulness of alternate flows and when they are used.

Ans:*Alternate or exceptional flows* are those that can happen, but they are not the normal flow. They are documented separately to keep the normal flow of events as simple as possible. For example, in a student registration system, you may have alternate flows for students with financial or academic holds on registration, or those who have to deal with a closed class situation before they can complete their registration.

1. What is the SVDPI form for steps in a use case, and why is it important to use this format when writing use cases? Provide an example of a sentence written in this format.

Ans: SVDPI represents Subject-Verb-Direct Object and optionally Preposition-Indirect Object. This form of the sentence is useful for identifying classes and operations from the use case description. An example may be something like “The Student contacts the academic department to obtain a closed-class waiver.”

1. List and briefly describe the four major steps in writing effective use-case descriptions and use-case diagrams.

Ans: Identify the major use cases

In this step you find the boundary of the system, identify the primary actors, list the goals for those actors, identify and write the overviews for the major use cases, and review any current use cases.

Expand the major use cases

In this step choose one of the use cases to expand, fill in the details, write the normal flow for the events in that use case, decompose the flow into subflows if needed, list the possible alternate or exceptional flows, and describe how the actor or system should react when an alternate or exceptional flow occurs.

Confirm the major use cases

Review the set of use cases, and revise as needed, and then start at the top again.

Create the use-case diagram

Draw the system boundary, place the use cases on the diagram, place the actors on the diagram, and then draw the associations. Revise as needed.

1. What is an actor? What is their role in a system?

Ans: An actor is a role played by an individual while they are interacting with the system, or it may be another system that interacts with the current system. An actor is not a specific person. Actors can provide input to the system, receive output from the system, or both. A primary actor is an actor who triggers the use case.

1. Distinguish between logical models and physical models.

Ans: Logical models are models that describe the business domain’s activities without suggesting how they are conducted. Logical models are also referred to as problem domain models. Examples of logical models include activity diagrams and use case diagrams. Logical models are constructed in the analysis phase while physical models are constructed in the design phase. In the design phase, logical models are refined into physical models, which provide information that is ultimately used to build the system. Logical models allow the analyst to focus on the business operations without getting distracted by the implementation details.

1. Describe the terms action, activity and object nodes that are found in an activity diagram.

Ans: Action is a simple non-decomposable piece of the overall behavior that is being modeled. On the other hand, an activity is used to represent a set of actions. An activity can further be decomposed into other activities or actions. Actions and activities can represent either computerized or manual behavior. They are depicted using rounded rectangles. Object nodes model objects in the activity diagram. They are represented using rectangles. Object nodes represent the flow of information from one activity to another activity.

1. Distinguish between control flows and object flows in an activity diagram.

Ans: Control flows model the paths of execution through a business process. Control flows can only be attached to actions or activities, and are depicted using a solid line with an arrowhead depicting the direction of flow. Object flows model the flow of objects through the business process. Since activities or actions modify or transform objects, object flows are necessary to show the actual objects that are used by and modified by these actions and activities. Object flows are depicted using dashed lines with arrows, and must have an activity at one end and an object at the other end.

1. What is a walkthrough?

Ans.: A walkthrough is essentially a peer review of a product. In the case of the functional models, a walkthrough is a review of the different models and diagrams created during functional modeling. This review typically is completed by a team of individuals that comes from the development team and the client. The purpose of a walkthrough is to thoroughly test the fidelity of the functional models to the functional requirements and to ensure that the models are consistent. That is, a walkthrough uncovers errors or faults in the evolving specification. However, a walkthrough does not correct errors—it simply identifies them. Error correction is to be accomplished by the team after the walkthrough is completed.

1. What are the different roles played during a walk-through?

Ans.: There are specified roles that different members of the walkthrough team can play. The first is the presenter role. This should be played by the individual who is primarily responsible for the specific representation being reviewed. This individual presents the representation to the walkthrough team. The second role is recorder, or scribe. The recorder should be a member of the analysis team. This individual carefully takes the minutes of the meeting by recording all significant events that occur during the walkthrough. In particular, all errors that are uncovered must be documented so that the analysis team can address them. The third role is to have someone who raises issues regarding maintenance of the representation. Due to the emphasis on reusability in object-oriented development, this role becomes particularly crucial.

Finally, someone must be responsible for calling, setting up, and running the walkthrough meetings.

1. Briefly describe the set of rules that are used to ensure the three representations (activity diagrams, use-case descriptions, and use-case diagrams) for the functional models are consistent among themselves.

Ans.: There are three different representations for the functional model: activity diagrams, use-case descriptions, and use-case diagrams. A set of rules have been developed to ensure that these three representations are consistent among themselves.

First, when comparing an activity diagram to a use-case description, there should be at least one event recorded in the normal flow of events, subflows, or alternate/exceptional flows of the use-case description for each activity or action that is included on an activity diagram, and each event should be associated with an activity or action.

Second, all objects portrayed as an object node in an activity diagram must be mentioned in an event in the normal flow of events, subflows, or alternate/exceptional flows of the use-case description.

Third, sequential order of the events in a use-case description should occur in the same sequential order of the activities contained in an activity diagram.

Fourth, when comparing a use-case description to a use-case diagram, there must be one and only one use-case description for each use case, and vice versa.

Fifth, all actors listed in a use case description must be portrayed on the use-case diagram. Furthermore, each one must have an association link that connects it to the use case and must be listed with the association relationships in the use-case description. In some organizations, we should also include the stakeholders listed in the use-case description as actors in the use-case diagram.

Sixth, all other relationships listed in a use-case description (include, extend, and generalization) must be portrayed on a use-case diagram.

Finally, there are many diagram-specific requirements that must be enforced. For example, in an activity diagram a decision node can be connected to activity or action nodes only with a control flow, and for every decision node there should be a matching merge node. Every type of node and flow has different restrictions.

1. Briefly describe the purposes of use case modeling in software development.

Ans: All object-oriented systems development approaches are use-case driven, architecture-centric, and iterative and incremental. A use case is a formal way of representing the way a business system interacts with its environment. Essentially, a use case is a high-level overview of the business processes in a business information system. From a practical perspective, use cases represent the entire basis for an object-oriented system. Use cases can document the current system (i.e., as-is system) or the new system being developed (i.e., to-be system). Given that object-oriented systems are use-case driven, use cases also form the foundation for testing and user-interface design. From an architecture-centric perspective, use case modeling supports the creation of an external or functional view of a business process in that it shows how the users view the process rather than the internal mechanisms by which the process and supporting systems operate.