CS 5150 Software Engineering

Scenarios and Use Cases

William Y. Arms

Scenarios

Scenario

A **scenario** is a **scene** that illustrates some interaction with a proposed system.

A **scenario** is a tool used during requirements analysis to describe a specific use of a proposed system. Scenarios capture the system, as viewed from the outside, e.g., by a user, using specific examples.

Note on terminology

Some authors restrict the word "scenario" to refer to a user's total interaction with the system.

Other authors use the word "scenario" to refer to parts of the interaction.

In this course, the term is used with both meanings.

Describing a Scenario

Some organizations have complex documentation standards for describing a scenario.

At the very least, the description should include:

- A statement of the purpose of the scenario

- The individual user or transaction that is being followed through the scenario

- Assumptions about equipment or software

- The steps of the scenario

Developing a Scenario with a Client

Example of how to develop a scenario with a client

The requirements are being developed for a system that will enable university students to take exams online from their own rooms using a web browser.

Create a scenario for how a typical student interacts with the system.

In the next few slides, the questions in blue are typical of the questions to ask the client while developing the scenario.

Developing a Scenario with a Client: a Typical Student

Purpose: Scenario that describes the use of an online Exam system by a representative student

Individual: *[Who is a typical student?]* Student A, senior at Cornell, major in computer science. *[Where can the student be located? Do other universities differ?]*

Equipment: Any computer with a supported browser. *[Is there a list of supported browsers? Are there any network restrictions?]*

Scenario:

1. Student A authenticates. *[How does a Cornell student authenticate?]*

2. Student A starts browser and types URL of Exam system. *[How does the student know the URL?]*

3. Exam system displays list of options. *[Is the list tailored to the individual user?]*

Developing a Scenario with a Client (continued)

4. Student A selects CS 1234 Exam 1.

5. A list of questions is displayed, each marked to indicate whether completed or not. *[Can the questions be answered in any order?]*

6. Student A selects a question and chooses whether to submit a new answer or edit a previous answer. *[Is it always possible to edit a previous answer? Are there other options?]*

7. *[What types of question are there: text, multiple choice, etc.?]* The first question requires a written answer. Student A is submitting a new answer. The student has a choice whether to type the solution into the browser or to attach a separate file. Student A decides to attach a file. *[What types of file are accepted?]*

Developing a Scenario with a Client (continued)

8. For the second question, the student chooses to edit a previous answer. Student A chooses to delete a solution previously typed into the browser, and to replace it with an attached file. *[Can the student edit a previous answer, or must it always be replaced with a new answer?]*

9. As an alternative to completing the entire exam in a single session, Student A decides to saves the completed questions work to continue later. *[Is this always permitted?]*

10. Student A logs off.

11. Later Student A log in, finishes the exam, submits the answers, and logs out. *[Is this process any different from the initial work on this exam?]*

12. The Student A has now completed the exam. The student selects an option that submits the exam to the grading system. *[What if the student has not attempted every question? Is the grader notified?]*

Developing a Scenario with a Client (continued)

13. Student A now wishes to change a solution. The system does not permit changes once the solution has been submitted. *[Can the student still see the solutions?]*

14. Later Student A logins in to check the grades. *[When are grades made available? How does the student know?]*

15. Student A requests a regrade. *[What are the policies? What are the procedures?]*

Developing a Scenario with a Client (continued)

- Developing a scenario with a client clarifies many functional requirements that must be agreed before a system can be built, e.g., policies, procedures, etc.

- The scenario will often clarify the requirements for the user interface, but the design of the user interface should not be part of the scenario.

Although this scenario is quite simple, many details have been left out.

Scenarios for Analyzing Special Requirements

Scenarios are very useful for analyzing special requirements.

Examples

- **Reversals.** In a financial system, a transaction is credited to the wrong account. What sequence of steps are used to reverse the transaction?

- **Errors.** A mail order company has several copies of its inventory database. What happens if they become inconsistent?

- **Malfeasance.** In a voting system, a voter has houses in two cities. What happens if he attempts to vote in both of them?

Scenarios for error recovery

Murphy's Law: "*If anything can go wrong, it will*". Create a scenario for everything that can go wrong and how the system is expected to handle it.

Modeling Scenarios as Use Cases

Models

Scenarios are useful in discussing a proposed system with a client, but requirements need to be made more precise before a system is fully understood.

This is the purpose of requirements **modeling**.

A **use case** provides such a model.

There is a good discussion of use cases in Wikipedia. The approach used in this course is less complex than the Wikipedia article.

Two Simple Use Cases

BookBorrower

Borrow Book

PressureSensor

Record Pressure

Actor and Use Case Diagram

BookBorrower

PressureSensor

Borrow Book

Record Pressure

- An **actor** is a user of a system in a particular **role**.

  An actor can be human or an external system.

- A **use case** is a a task that an actor needs to perform with the help of the system.

Use Cases and Actors

- Actor is **role**, not an individual
  (e.g., librarian can have many roles)

- Actor must be a **beneficiary** of the use case
  (e.g., not librarian who processes book when borrowed)

In naming actors, choose names that describe the role, not generic names, such as "user" or "client".

Use Cases for Exam System

ExamTaker

Take Exam

Check Grades

Request Regrade

Three separate use cases
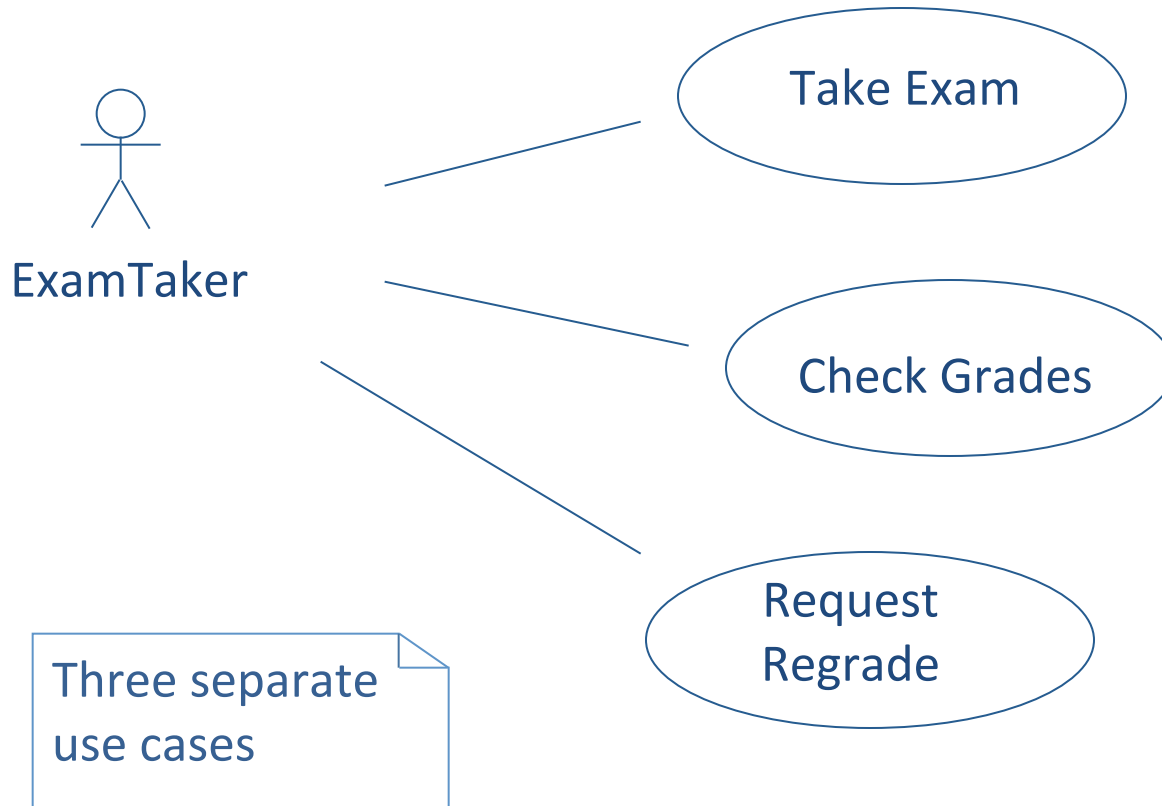
Describing a Use Case

Some organizations have complex documentation standards for describing a use case.

At the very least, the description should include:

- The name of the use case, which should summarize its purpose

- The actor or actors

- The flow of events

- Assumptions about entry conditions

Outline of Take Exam Use Case

Name of Use Case: Take Exam

Actor(s): ExamTaker

Flow of events:

1. ExamTaker connects to the Exam server.

2. Exam server checks whether ExamTaker is already authenticated and runs authentication process if necessary.

3. ExamTaker selects a exam from a list of options.

4. ExamTaker repeatedly selects a question and either types in a solution, attaches a file with a solution, edits a solution or attaches a replacement file.

# Outline Specification of Use Case (continued)

Flow of events (continued):

5.   ExamTaker either submits completed exam or saves current state.

6.   When a completed exam is submitted, Exam server checks that all questions have been attempted and either sends acknowledgement to ExamTaker, or saves current state and notifies ExamTaker of incomplete submission.

7.   ExamTaker logs out.

Entry conditions:

1.   ExamTaker must have authentication credentials.

2.   Computing requirements: supported browser.

Use Cases for Exam System (continued)

Instructor
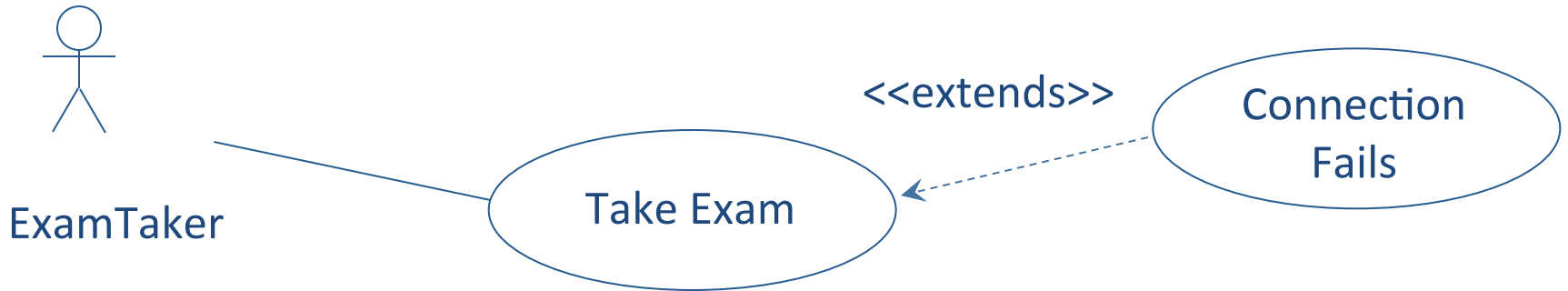
Set Exam

Grade

Regrade
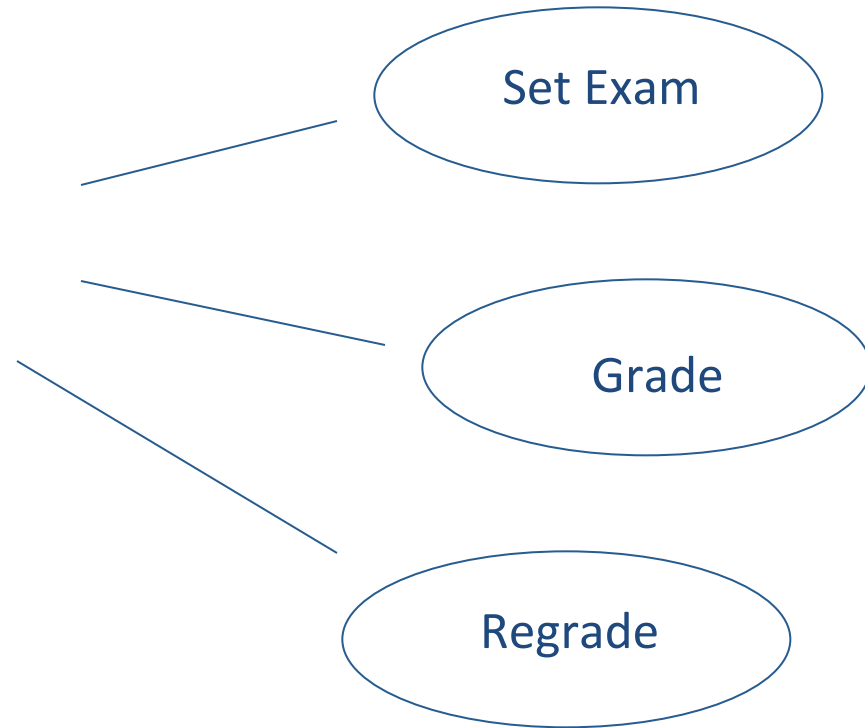
Note that actor is a role. An individual can be an ExamTaker on one occasion and an Instructor at a different time.

Relationships Between Use Cases: <<includes>>

ExamTaker

Take Exam

Check Grades

Authenticate

<<includes>>

<<includes>>
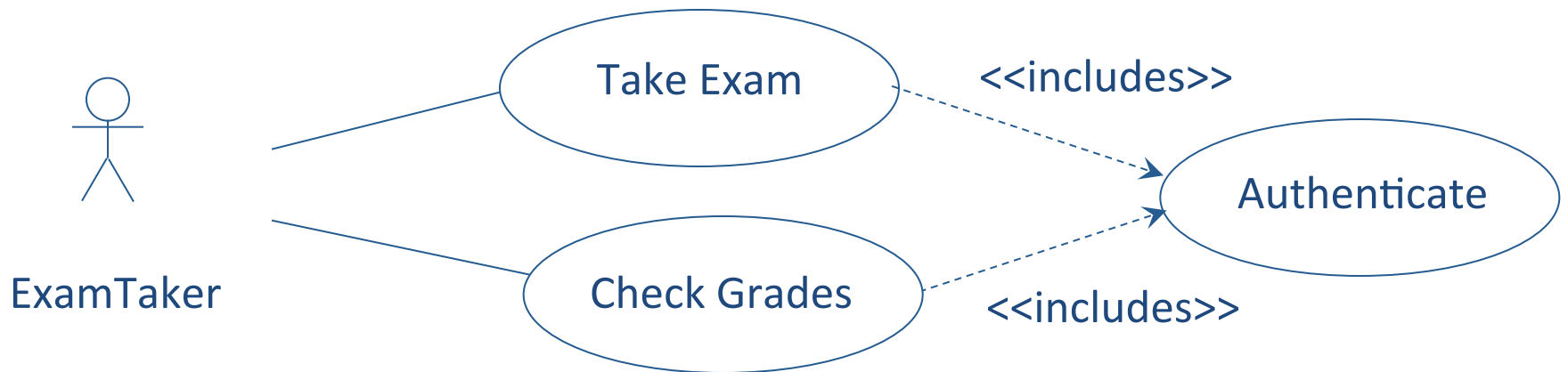
The Authenticate use case may be used in other contexts

Relationships Between Use Cases: <<extends>>

<<extends>>

ExamTaker — Take Exam ← <<extends>> — Connection Fails

<<includes>>   is used for use cases that are in the flow of events of the main use case.

<<extends>>   is used for exceptional conditions, especially those that can occur at any time.

Scenarios and Use Cases in the Development Cycle

Scenarios and use cases are both intuitive -- easy to discuss with clients

Scenarios are a tool for requirements analysis.

- They are useful to validate use cases and in checking the design of a system.

- They can be used as test cases for acceptance testing.

Use cases are a tool for modeling requirements.

- A set of use cases can provide a framework for the requirements specification.

- Use cases are the basis for system and program design, but are often hard to translate into class models

Use Cases with Several Actors
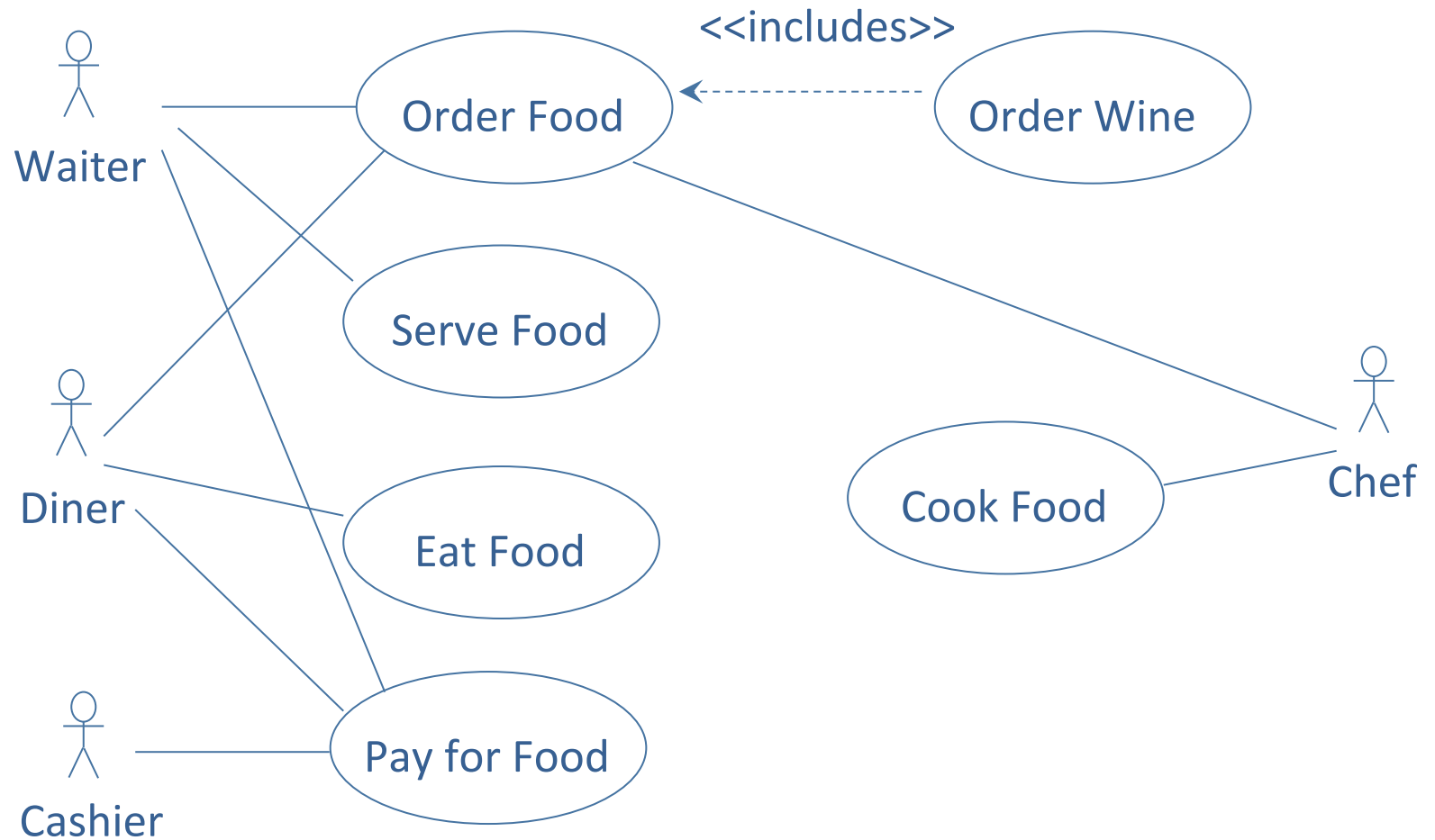
This restaurant example is based on a use case diagram from Wikipedia.

Use Case Diagrams

Use case diagrams

A use case diagram shows the **relationships** between actors and and their interactions with a system.

It does not show the **logic** of those interactions.

An Old Examination Question

The Pizza Ordering System

The Pizza Ordering System allows the user of a web browser to order pizza for home delivery. To place an order, a shopper searches to find items to purchase, adds items one at a time to a shopping cart, and possibly searches again for more items.

When all items have been chosen, the shopper provides a delivery address. If not paying with cash, the shopper also provides credit card information.

The system has an option for shoppers to register with the pizza shop. They can then save their name and address information, so that they do not have to enter this information every time that they place an order.

Develop a use case diagram, for a use case for placing an order, *PlaceOrder*. The use case should show a relationship to two previously specified use cases, *IdentifyCustomer*, which allows a user to register and log in, and *PaybyCredit*, which models credit card payments.

An Old Examination Question

Correct solution

<<includes>>

PaybyCredit

PlaceOrder

Shopper

<<includes>>

Is link is optional

IdentifyCustomer
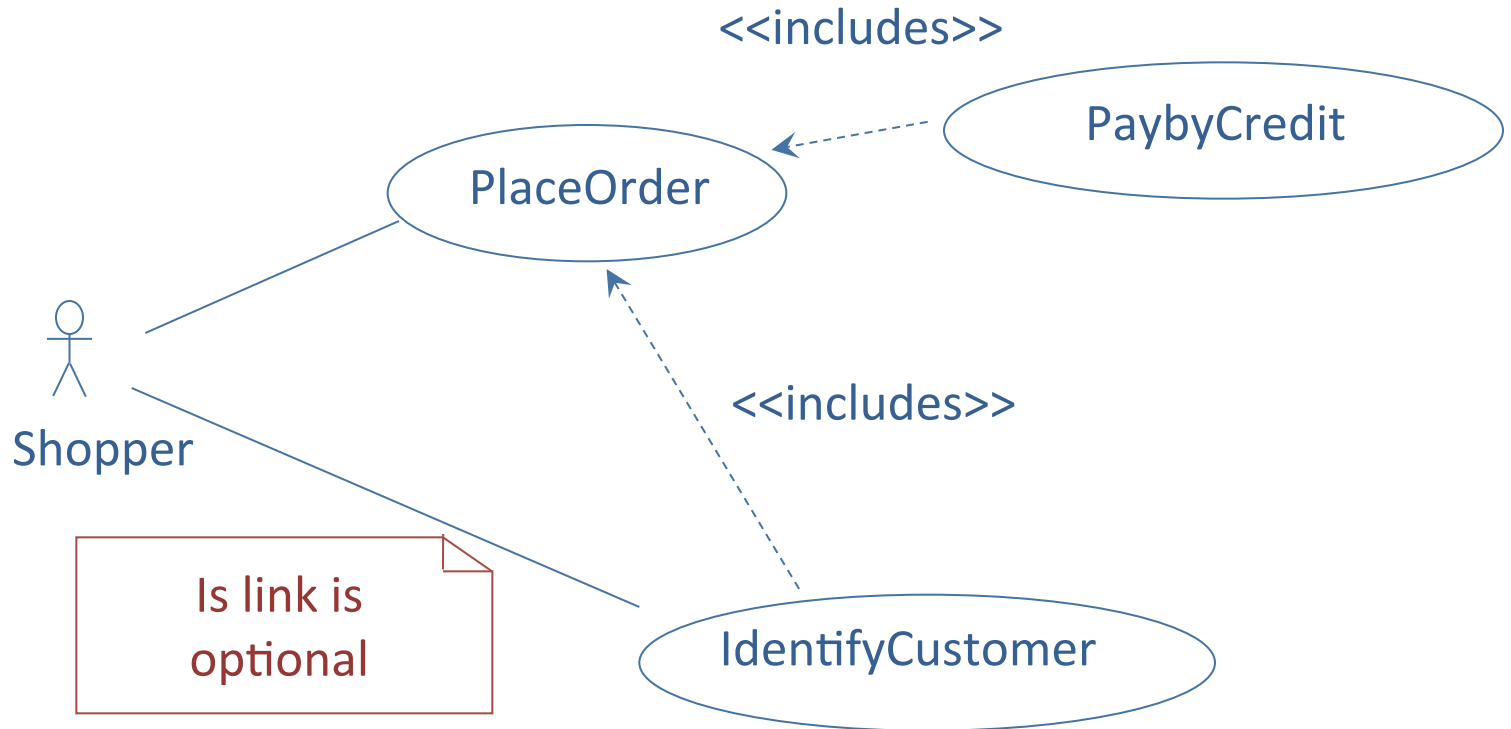
An Old Examination Question

Wrong Solution