

48024 Object Oriented Design

Self-Study Module: CRC Modelling and Class Diagrams

This module consists of a series of exercises for developing your skills in finding and modelling objects and classes from a problem domain.

The learning objectives are:

- To develop skill in distinguishing between objects and classes
- To develop skill in finding classes from the problem domain to model a system
- To build CRC cards from use-case scenarios
- To gain skill in understanding and creating class diagrams

You will need a deck of playing cards and the rules for Blackjack (otherwise known as 21 or Pontoon).

References:

- Simon Bennett, Steve McRobb, Ray Farmer, "Object-Oriented Systems Analysis and Design using UML", McGraw-Hill, 2nd edition, 2002

Contents:

Exercise 1 - Playing Blackjack	2
Exercise 2 – CRC cards	3
Exercise 3 - Construct a class diagram from CRC cards	5
Exercise 4 - Construct a class diagram from use cases.....	6

Exercise 1 - Playing Blackjack

This is a warm-up exercise to refresh your knowledge of objects and classes in a concrete way. This exercise is to be done in a small group. You will play a game of Blackjack and identify the key classes, objects and interactions.

The rules of basic Blackjack are given below – read through them.

Nominate one person as the dealer. Deal out two cards to each player.

Play the game a few times, and then discuss the following:

- What are the key classes?
- What are the important characteristics of each class - that is, what information do you need to know about each class? Write down a list of characteristics for each class.
- What are the objects that exist for this particular game? Give each object a name and decide which type of class it is.
- What objects interact with each other?

You should finish with a set of classes, objects and interactions between objects.

Rules of Blackjack

Blackjack is one of the most famous table games around the world. Here we present the very basic rules to play the game.

Card values

A Jack, a Queen, or a King is worth 10 points.

An Ace is worth one point or 11 points; the player chooses the value to get the best score. An Ace can change its value from 11 to 1 (or 1 to 11) if desired.

All other cards are worth their face value (2-10).

Playing

The dealer deals a card to each player, and one to the dealer. These cards are face down.

The dealer deals another card to each player, and to the dealer. The dealer's card is face up.

The first player then requests another card (“Hit me”) or sits (“I sit”). If you get a card that pushes your hand over 21, then you go bust, and drop out of the game. A player keeps asking for cards until the player sits (no more cards wanted) or goes bust (value over 21).

Play goes around the table, one player at a time.

The dealer then shows both cards and takes more cards (or not). The dealer must take a card if their hand is worth less than 16.

Winning

The idea is to get as close to 21 as possible, without going bust.

At the end of the hand, a player wins if the value of their hand is higher than the value of the dealer's hand. If the two hands are worth the same value, then the dealer wins.

If the dealer's hand goes over 21, then the dealer busts and all the remaining players win.

“Blackjack” happens when a player or the dealer gets a total of 21 with the first two cards (a value 10 card and an Ace). If the player has Blackjack he/she wins automatically unless the dealer also has Blackjack, in which case it is a Push or a Tie (or a Stand-off).

Exercise 2 – CRC cards

Finding the suitable classes that are part of a system is a very important task in OO analysis. Because it is not an easy task some formal and informal approaches to finding classes have been suggested. One of them is the use of Class Responsibility Collaboration (CRC) cards. This activity involves role-playing scenarios with CRC cards.

Read the Advanced Blackjack rules (see below) to get a better idea of what is involved in a Blackjack game with more rules than those you worked in the previous exercise.

Create CRC card for each class. Review the list of classes you identified in Exercise 1 and add missing classes and/or delete redundant classes. Then create a CRC card for each one of them. Write a brief description of each class on the back of the CRC card. Note that this may need to be updated as the role-playing continues.

Add responsibilities to each class. Add responsibilities that are “obvious” from the name of the class.

Arrange CRC cards. Place the CRC cards in the centre of the table so they are easy to view and arranged so that classes that are likely to communicate with each other are located close to each other.

Pick your class. Each member of the team will nominate to “become” an object of one of the classes involved in the system. You should be aware of the role of this object while the role-play evolves.

Start role-play. Nominate one person in the group to read the scenario aloud step-by-step. The role-play starts here!

One suggestion is to role-play the Simple Blackjack game first and then role-play the Advanced Blackjack.

For each step of the scenario:

> Decide which class should be **responsible** for this action. Ask the questions: what does the class know? what information does it have that makes it unique? (identifying attributes); and what does the class do? (identifying responsibilities).

If it is agreed, then the action is assigned as a responsibility to that card. Write down the new responsibilities and attributes on the respective section of the card.

If none of the available classes is really suitable to carry out a specific action, then you may need to create a new class. Decide who will look after this card.

> Decide if this class needs to **collaborate** with any other classes in order to fulfil its responsibility.

A class collaborates with another class when it needs information that it does not have, or needs to modify information that it does not have, or needs another class to do something. Ask the other card-holders if they think they are the appropriate collaborator.

Write down the names of the collaborating class(es) on the front of the card, next to the associated responsibility.

48024 Object Oriented Design

Self-Study Module: CRC Modelling and Class Diagrams

Rules of Advanced Blackjack

Here are additional rules and descriptions of players' choices and situations that extend the scenario of playing Blackjack under a casino environment.

In a casino

The blackjack table seats about 6 players. Either six or eight decks of cards are used and are shuffled together by the dealer and placed in a card dispensing box called 'Shoe'.

Betting

Once a player is seated at a table, he/she needs to purchase some chips from the dealer to be able to bet. Chip colours are fairly standardized in the casino industry (see table 1). An appropriate buy-in amount ranges from 10 to 20 times the average bet. If a player is a \$5 bettor, this means a buy-in of \$50-\$100 is typical.

CHIP COLOUR	VALUE
Red	\$5
Green	\$25
Black	\$100

Table 1. Value of standard chips according to colour

Before each hand begins (before the dealer deals cards), each player places the desired bet into a specific circle on the table. Chips of different denominations are allowed. Once the cards have been dealt, the players aren't allowed to touch the bet in the circle. If at any stage a player needs to know how much has been bet, the dealer will count down the chips and inform the player. Once the hand is over, the dealer will move around the table to each position in turn, paying winners and collecting the chips from losing hands. After the dealer has paid, a player can remove their chips from the circle, and place the next bet.

Player's choices

A casino might or might not accept to implement some rules about what players are allowed to do when betting/playing in Blackjack. The standards options are:

- **Insurance:** side bet up to half the initial bet against the dealer having a Blackjack - allowed only when the dealer's showing card is an Ace. If the dealer has a 10 face down and makes a blackjack, insurance pays at 2-1 odds, but loses if the dealer does not.
- **Surrender:** giving up the hand and lose only half the bet.
- **Early Surrender:** surrender allowed before the dealer checks for blackjack.
- **Late Surrender:** the dealer first checks to see if he has blackjack. If he does, surrender is not permitted.
- **Double Down:** double the initial bet following the initial two-card deal, but only one additional card can be handed. Players are allowed to double down for any amount up to the original bet amount, so you could double down "for less" if you wanted.
- **Split Hand:** split the initial two-card hand into two and play them separately - allowed only when the two first cards are of equal value. Each card is used as the start to a separate hand and a second bet equal to the first must be placed.

Exercise 3 - Construct a class diagram from CRC cards

Arrange the classes identified in the CRC card activity into a class diagram.

Use the following set of guidelines to construct a class diagram from the set of CRC cards:

- Draw a rectangular box for each class and label with the *name* of the *class*.
- In the middle compartment of the box, list the name, visibility and type of the *attributes* of the class after evaluating the attributes defined in the CRC card.
- Decide whether the responsibilities listed for each class are suitable *operations*. If so, write the name of the operation in the bottom compartment of the class box.
- Draw *association* lines between the classes that are linked - indicated by the *collaborating* classes.
- Determine the *multiplicity* of each association.

Exercise 4 - Construct a class diagram from use cases

Construct a class diagram from given requirements in the form of use case descriptions and diagram.

In Table 1 you'll find the list of requirements extracted from the Agate Ltd case study used in the guide textbook (see Bennett's A1 for a full introduction to the case study, page 84). Table 2 presents a description of each use case. Finally, the use case diagram shown in Figure 1 gives you a broader view of how users will interact with the system according to its expected functionalities.

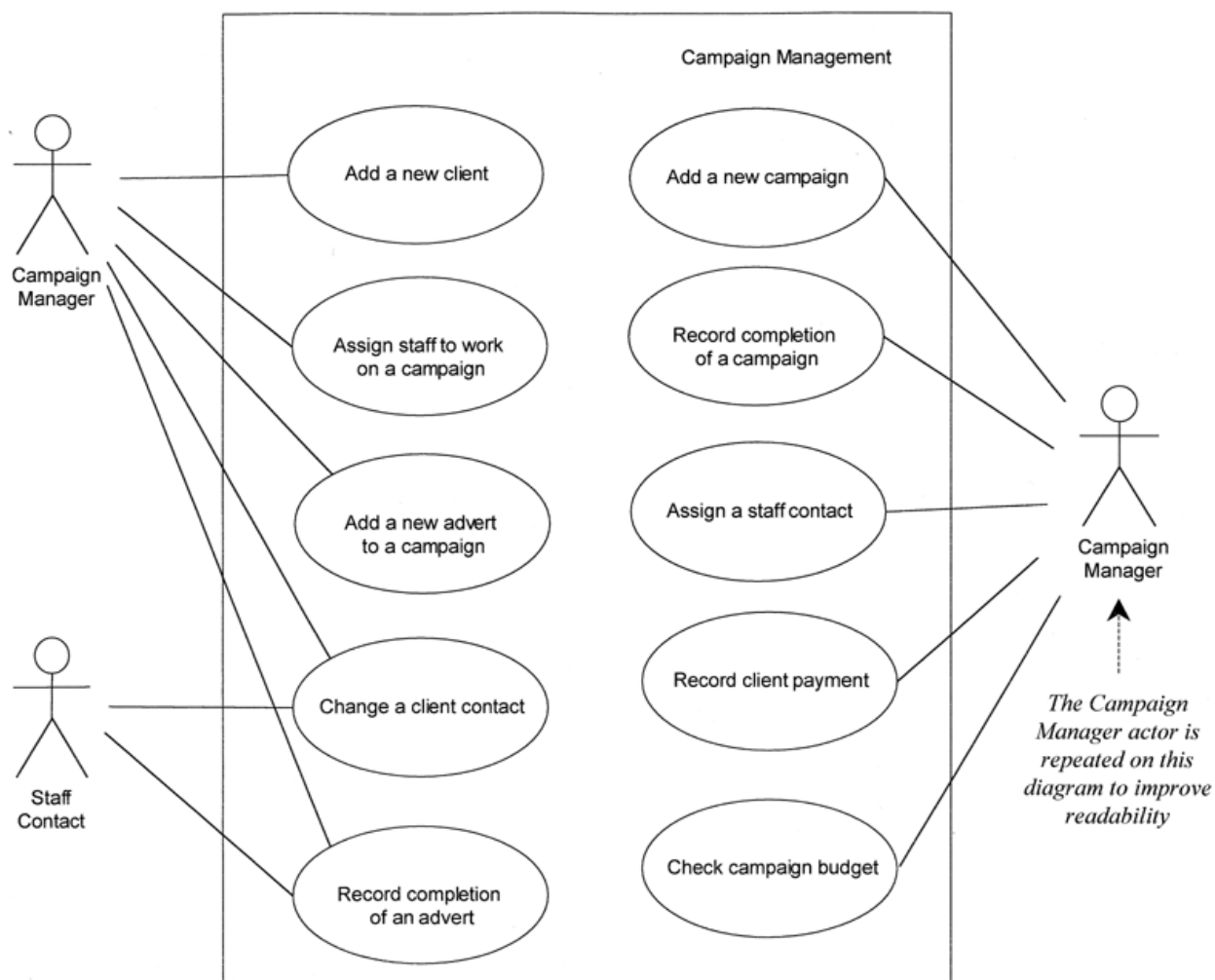


Figure 1. Use case diagram

Similar to the CRC cards exercise, the first step is to identify possible classes in the system and then work on allocating responsibilities among them. However, this time you will draw a class diagram immediately using UML notation.

Your class diagram won't contain all the details that a final version of the class diagram would contain, but if the elements of your class diagram (class, attribute, operation) support the requirements, then your diagram should be a good attempt. Later on you'll learn how to produce good class diagrams based on OO design principles.

48024 Object Oriented Design

Self-Study Module: CRC Modelling and Class Diagrams

For more guidelines, refer to the questions and suggestions described in sections 7.5.3 - 7.5.9 of Bennett's book.

No.	Requirement	Use Case(s)
1	To record names, address and contact details for each client.	Add a new client
2	To record the details of each campaign for each client. This will include the title of the campaign, planned start and finish dates, estimated costs, budgets, actual costs and dates, and the current state of completion.	Add a new campaign
3	To provide information that can be used in the separate accounts system for invoicing clients for campaigns.	Record completion of a campaign
4	To record payments for campaigns that are also recorded in the separate accounts system.	Record client payment
5	To record which staff are working on which campaigns, including the campaign manager for each campaign.	Assign a staff to work on a campaign
6	To record which staff are assigned as staff contacts to clients.	Assign a staff contact
7	To check on the status of campaigns and whether they are within budget.	Check campaign budget
8	To record details of adverts, including the progress on their production.	Add a new advert to a campaign; Record completion of an advert

Table 1. List of requirements

Use Case	Description
Add a new client	When Agate obtains a new client, the full details of the client are entered. Typically this will be because of the new campaign, and therefore the new campaign will be added straight away.
Add a new campaign	When Agate gets the business for a new campaign, the details of the campaign are entered, including the intended finish date and the estimated cost. The manager for the campaign is the person who enters it.
Record completion of a campaign	When a campaign is completed, the actual completion date and cost are entered. A record of completion form is printed out for the Accountant as the basis for invoicing the client.
Record client payment	When a client pays for a campaign, the payment amount is checked against the actual cost and the date paid is entered.
Assign a staff to work on a campaign	The campaign manager selects a particular campaign. A list of staff not already working on that campaign is displayed, and he or she selects those to be assigned to this campaign.
Assign a staff contact	Clients have a member of staff assigned to them as their particular contact person.
Check campaign budget	The campaign budget may be checked to ensure that it has not been exceeded. The current campaign cost is determined by the total cost of all the adverts and the campaign overheads.
Add a new advert to a campaign	A campaign can consist of many adverts. Details of each advert are entered into the system with a target completion date.
Record completion of an advert	The actor selects the relevant client, campaign and advert. The selected advert is then completed by setting its completion date.

Table 2. Use case descriptions