ID2207 - Modern Methods in Software Engineering

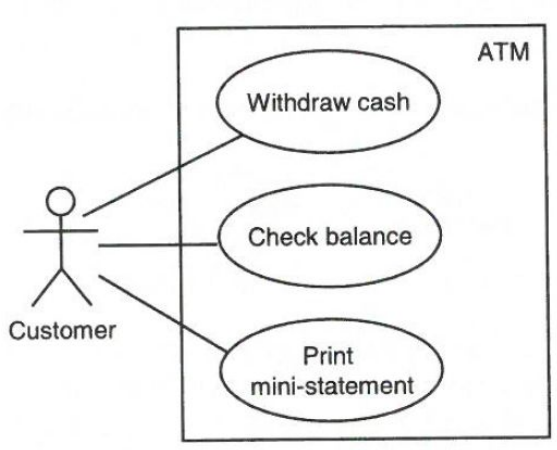
Tutorial 2

September 2020

Exercise 1: Scenarios versus Use Cases

The term scenarios is often used to refer to all the different possible courses that different instances of the same use case might take. The use case ***diagram*** *names only the use cases, but additional documents may show the alternative scenarios that could occur. For example, in use case description, we usually describe the normal course of events and the most likely alternatives (exceptional scenarios).*

Example: In a use case that defines the interaction that takes place between customers and automated teller machines (ATMs), the Customer actor represents the class of all customers who will use the ATM subsystem. When you use the ATM to withdraw cash, you are an instance of Customer using a particular instance of the use case Withdraw cash. Somebody else may use an instance of the use case Check Balance or Print mini-statement. *You may successfully withdraw cash from the machine, but the person behind you may find that he or she does not have enough money deposited, and the use case instance will proceed along a different course from yours, rejecting the request.*



Question: What different scenarios might exist for the use case Withdraw Cash ?

a. The customer's card is not recognized and is rejected
b. The customer enters the wrong PIN and is asked to re-enter it.
c. The customer enters the wrong PIN three times and the card is retained by the ATM.
d. The customer enters an invalid figure for the amount of the cash.
e. The ATM attempts to connect to the bank's system but it is out of action or there is a network failure, so it cannot connect.
f. The ATM doesn't have enough cash to meet the customer's request.
g. The customer's account doesn't have enough funds to meet the request.
h. The customer cancels the transaction part way through.

****

Exercise 2: Identification of actors

Actors are the people or systems that interact with use cases.
A system analyst who is producing use case diagrams and descriptions will normally be working from source documents such as notes and transcripts of interviews. Below is a short excerpt from an interview transcript with one of the directors who is setting up CarMatch (car renting company). Mick is the system analyst and Janet is the director.

- **Mick (SA):** So you´re saying that car sharers will be able to register by telephoning the office and speaking to someone there who will enter their details into the system.
- **Janet (DIR):** Yes. Either the franchisee, or more likely one of the office staff will take the call and enter the details into the computer.
- **Mick (SA):** Who are the office staff?
- **Janet (DIR):** Well, there´s one or two clerks, a receptionist and a supervisor. They all have a role in the administration of the system.
- **Mick (SA):** What will they enter?
- **Janet (DIR):** Oh, the person´s name and address, details of the journeys they want to share, any preferences they have, such as being non-smoker.
- **Mick (SA):** Is that the only way that this information will get into the system?
- **Janet (DIR):** No, it could also be transferred in from the national web-server.
- **Mick (SA):** How will this information be used?
- **Janet (DIR):** Two ways. Firstly, it will be used to match up potential car sharers, and secondly, it will be used to produce a management report for the franchisee showing the number of registrations per week, whether they come from the web-server or by telephone and breaking them down by area.
- **Janet (DIR):** Whether we are entering a new car sharer manually or by transferring data from the web-server, the processing is the same apart from how we deal with membership payments. If we are entering a new car sharer into the system manually, then we need to process their membership payment at the same time. If their data is being transferred from the web-server, then we process it separately later. When we process the payment, the person can pay either by a regular direct debit or using a credit or debit card.
- **Mick (SA):** What about the matching process?
- **Janet (DIR):** That´ll be based on several factors, mostly it´s geographical...

To find actors, ask yourself these questions:

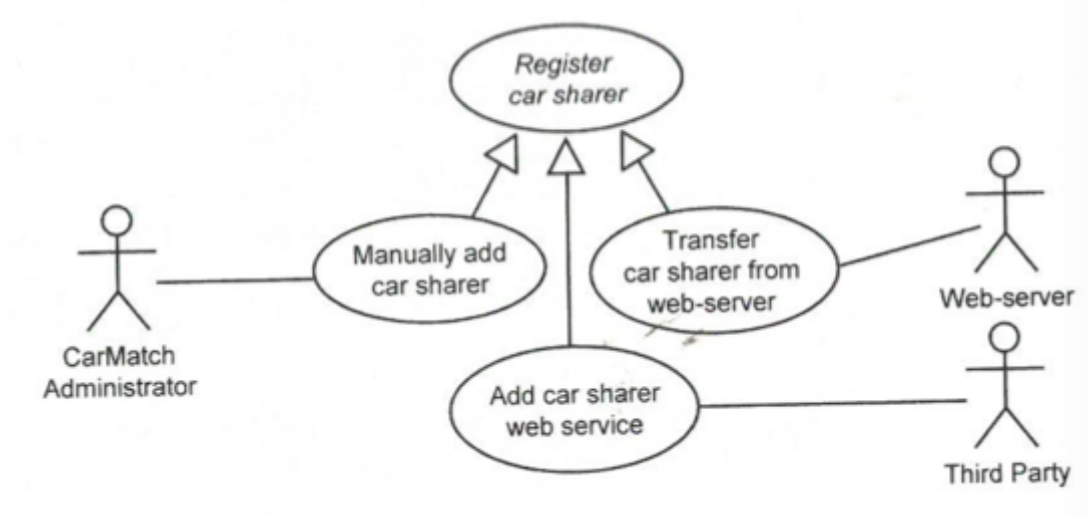
a. ***Who are the people who will use this system to enter information ?***
   Franchisee, Clerks, Receptionist, and the Supervisor.
b. ***Who are the people who will use this system as recipients of information ?***
   Franchisee
c. ***What are the other systems that this system will interact with ?***
   The web server, credit card system

Notice here that the car sharers don't interact with the system directly. So, they are not actors.
There is nothing to distinguish the roles that the clerks, receptionist, and supervisor play in this part of the system, except that they all administer the registration process. So, we can use one actor role to include all of them (CarMatch Administrator).
****

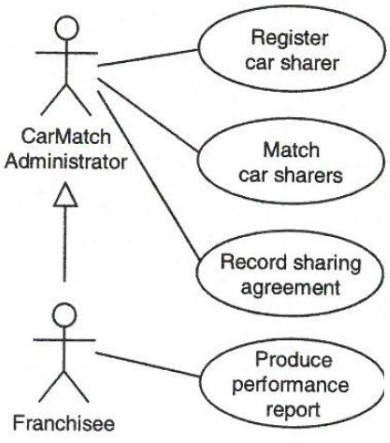
Exercise 3: identification of use cases

Hint [how to name your use cases]: Use case names are normally made up of an active verb and a noun or noun phrase that concisely describe the behaviour of the system that you are modelling. For example: RegisterCar, MatchSharer, etc.

Hint [Generalization between use cases]: Happens when there is more than one version of a use case, and the different versions have some actions in common, and some that are unique to each one. In your question, there are more one way to add new car sharer to the system: manually, one from the web service, and another for transferring them from the web server. These three use cases all serve the same overall function, but they are different in some aspects of how they operate.

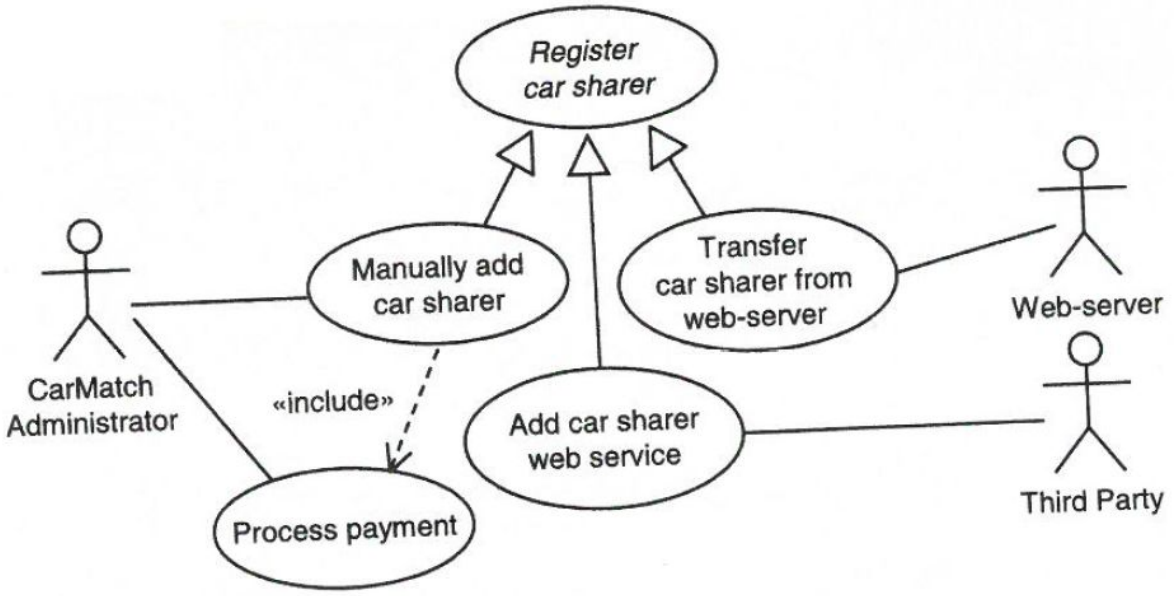


Hint [Generalization between actors]: In our example, the Franchisee can handle registration of new car sharers, but can also be the recipient of management reports. So, it is possible to show the Franchisee as a specialisation of the CarMatch Administrator. This means that the Franchisee can do everything that the CarMatch Administrator can do, and some more.

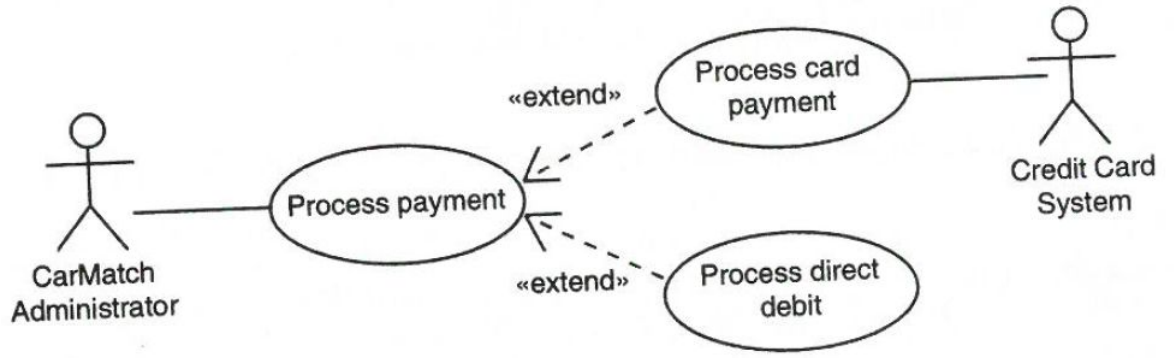


Hint [Include Relationships between use cases]: Sometimes one use case includes the functionality of another use case. In our example, when they manually add a car sharer, they have to do the payment at the same time. However, the ProcessPayment use case can be done also in the other registration cases as a separate process. So, it is both a use case on its own and it is included in the use case of Manually add car sharer.

The include relationship can also be used when a particular use case is included in other use cases because it encapsulates some functionality that is used at several points in the system. This avoids having to define the same sequence of actions in multiple use cases. The including use case will continue up to the point where it includes the included use case, the full sequence of activities in the included use case will be carried out, and then the including use case will carry on at the point where it left off.



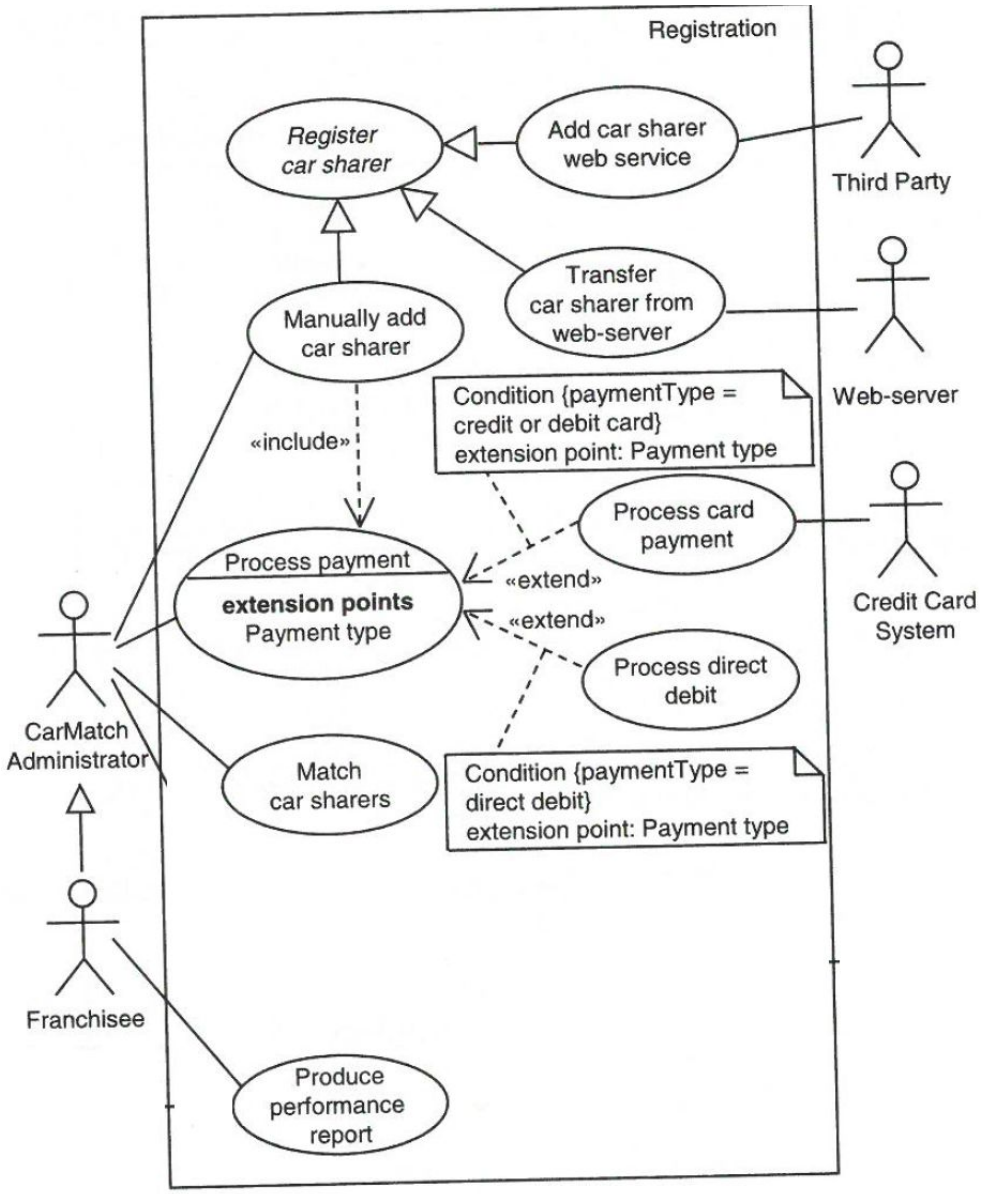
Hint [Extend Relationships between use cases]: While an include relationship means that one use case always includes another, there are occasions where one use case may optionally be extended by the functionality in another use case. Please remember that only on of the extending use cases will be used at a time.



Main use cases in this business:

a. Manually add car sharer
b. Transfer car sharer from web server
c. Match car sharers
d. Produce management reports

Notice that we didn't break down the use cases into the level of details that is described in the transcript. We are trying to group sequences of actions together into units that are meaningful in the context of the business. So, we don't create a use case such as: Enter details.



****

Exercise 4: Non-Functional requirements

A company hires out tools and equipment (for example drills, power saws, cement mixers, ladders, scaffolding) to customers and requires a computerized system to record details of bookings. Equipment may be booked in advance, or customers may appear at the reception desk and ask if there is an item available for immediate hire. When dealing with a booking or allocating an available item to a customer, the receptionist has to check whether the customer has previously hired equipment from the company or is a new customer. For a new customer, the receptionist has to enter the customer's details. Otherwise, the receptionist has to retrieve the existing customer's record and update any details if necessary. The minimum period of hire is one day and all hires are made for a number of complete days. The return of an item at the end of the hire period is recorded by the receptionist, or by a technician if the reception desk is closed. The manager of the company requires a summary of the status of all equipment at the beginning of each day, giving details of: items out on hire, items booked and items that will be available for hire that day.

1. From the description, identify functional and non-functional requirements for the system.
2. Are there any functional or non-functional requirements that you think are likely to be relevant but that have not been explicitly stated in the problem?

Functional Requirements

Retrieve customer's details
Update customer's details
Add new customer
Check availability of tools
Make an advance booking
Record an immediate hire
Book out a tool
Record return of a tool
Produce management summary

Non-functional requirements

The fact that a hire is for a whole number of days could be seen as a non-functional requirement; alternatively this could lead to a new functional requirement to allow the minimum hire period to be specified. But probably the restriction to whole numbers of days would permeate the system when we get on to more detailed design, so it makes most sense as a non-functional requirement.

Missing requirements

It is possible to think of many functional and non-functional requirements that are likely to be important in a system of this kind.
Functional requirements could include recording damage to tools, allowing regular repeated bookings, recording late returns and penalties, changing or cancelling existing bookings, and no doubt many others. Of course there is the whole area of handling payments, but the

question explicitly said that this is not included.

Obvious **non-functional requirements** would be compatibility with an existing computer system and the kind of user interface required.

Security could be important. If user names and passwords are required specifically for this system, then this would become a functional requirement. Alternatively it could be a nonfunctional requirement that the system must run on a computer that is adequately protected by passwords.

Backups of data could be important. This is probably considered a non-functional requirement, for example to say that the system must have a redundant disk system or make automatic backups at a certain frequency.

It would be useful to interview the client and raise these and other points, in order to be sure that all requirements are discovered. (Note that "client" means the tool hire company, not a customer of the tool hire company).

Summary:

- Use cases are created during the early stages of the project (analysis technique rather than a design technique).
- Use cases are represented graphically in a use case diagram to allow the analyst to visualize each use case in the context of the other use cases in the system, and to show its relationships with actors and other use cases.
- Use cases can also be used later in the development process, for example to specify test cases.
- Use cases provide a high-level view of what the system does and who uses it.
- They use simple diagrammatic notation that is comprehensible to end users.