14.5 BEHAVIOUR DIAGRAMS

The object diagrams that we discussed so far are models showing the state of the object or entity from the system. In other words it is a simple frontal view of the object. To understand the object as a system, we must understand its behaviour in association with other objects in the system. The object diagram, along with its behaviour diagram, explains the system. When a system performs in real time, many objects come into play. They send messages to each other and activate the methods in the object. When a method is chosen and performed on the data, the object status changes. The static object assumes a dynamic state. A state of the object is the result of its behaviour, which changes dynamically.

For example, a car, when stationary, is a static object, but when the owner drives it (messaging), it demonstrates behaviour that changes with every action of the owner. Further, the owner's driving actions also respond to the external traffic environment and car's feedback on various actions taken by the owner. Any qualitative comment on the car is better based on the behaviour of the car in driving mode.

Business system objects are similar to a car. We have to understand them first in then static state and then also in the dynamic state, where it interacts other objects. Booch and other specialists in UA, recommend four different diagrams to explain the dynamic state of the object. The diagrams explaining the behaviour are called Behaviour Diagrams. Behaviour diagrams put the object model as a dynamic model. The behaviour diagrams are

- Interaction diagrams
  - Sequence diagrams
  - Collaboration diagrams
- State chart diagrams
- Activity diagrams

14.5.1 Interaction Diagrams

The purpose of an interaction diagram is to understand the role of other objects that are in collaboration with the object in question, the objective being to complete the job. When a car key is turned, the driver object, the engine object and the starter object collaborate to start the car and put car in a dynamic state. This has happened through interaction between the concerned objects. The interaction has two dimensions, the manner (logical time sequence) in which interaction takes place, and style (collaboration) in which it is executed. Object behaviour is better understood, if the sequence of interaction between objects, and collaboration between objects, is analysed in detail.

This is achieved through specific diagrams covered under interaction diagrams. The two diagrams are

- Sequence diagram
- Collaboration diagram

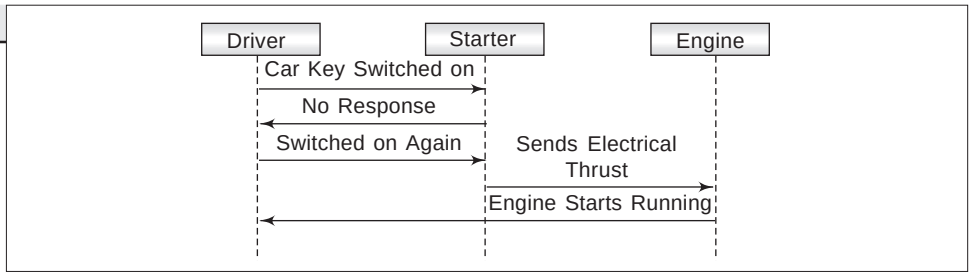
The basis for drawing these diagrams is use case models of the system. These diagrams are explained using the action scripts given in use case models.

A sequence diagram shows an interaction arranged in a time sequence in its logical order. It also shows objects participating in the behaviour, and the messages that they exchange in order to perform.

Figure 14.15 shows a sequence diagram of a use case 'starting the car' by starting the engine.

Fig. 14.15

Sequence Diagrams



A sequence diagram uses the following notations. A rectangle shows objects that participate in the behaviour. Existence of the object in the behaviour is shown by a vertical dotted line. The horizontal arrowhead line shows message 'from - to'. The horizontal lines are arranged in sequence of their occurrence.

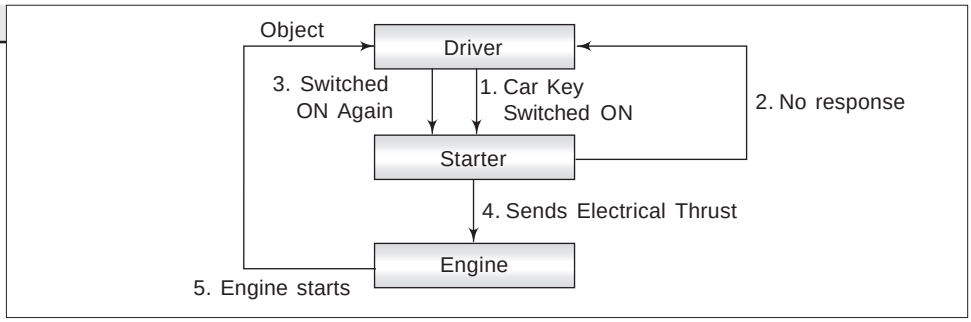
The sequence diagram is simple to understand, and it follows the use case transaction sequence. The sequence diagram also is an alternative to understand the program that executes the object behaviour. The sequence diagrams show the task or activity sequence but do not show the relationship between objects through the roles that they play in the interaction. In sequence diagrams, objects are shown with no connection. But in collaboration diagrams, those objects that collaborate to perform the role based on the message received are connected. .

Both sequence and collaboration diagrams are alternatives to each other. The sequence diagram is easier to understand but does not show the flow of the behaviour and collaboration of objects. The collaboration diagram shows both collaboration and sequence.

Figure 14.16 shows the collaboration diagram for starting the car engine.

Fig. 14.16

Collaboration Diagram



Outwardly, there is no difference between two diagrams. A sequence diagram is easy to read, and the collaboration diagram shows the interaction between objects and sequence of activities denoted by numbers. The disadvantage of the

interaction diagram is that in complex interactions, with couple of objects, the diagrams are difficult to draw and difficult to read. The remedy is split the use cases into smaller activities or tasks with specific goals, and to then use interaction diagrams.

The interaction diagrams are not good enough if the behaviour of the object is conditional and gets into loop. It also loses its clarity with more complex conditional behaviour. In such cases, the best designing tool is the Activity Diagram, which we will discuss here later in the chapter.

Amongst the two, the sequence diagram is better when we want to see the sequence of message occurrence, to directly employ use case models to show interactions, and also want to show details of messages in terms of parameters, values, results, etc. Otherwise, a collaboration diagram is better to show interaction.