

15.3.3 OOA of Railway Ticket Reservation System

Assume that domain analysis is complete and DAD is ready for reference. The analyst also has a fair knowledge of the system and the system environment. For the sake of convenience and to reduce complexity at this stage in OOA, we are ignoring certain conditions, constraints and features that the real system may have. For example, we are not considering cancellation of tickets as a requirement. We assume all passengers irrespective of their type (senior citizen, military personnel, special category passengers etc.) are the same, and therefore are charged with the same fare. We begin with the statement of requirement of the system.

✓ **System Requirement**

- The passenger has a prior knowledge of the reservation and ticketing system. The passenger arrives at the railway ticket counter and interacts with the counter clerk first through an enquiry and then follows the process of form filling, tendering, payment and collecting the tickets.
- Passenger accepts the ticket or leaves the counter.
- Passenger seeks information on fare, train timings and availability of tickets.
- Passenger can have single ticket or multiple tickets.
- Journey begins on a day and will be over with one break in between.
- Passenger is identified by name, age, sex and address.
- Trains are identified by name and number.
- No receipt is issued for money transacted.
- Output of the system is ticket(s) with details.
- The process is triggered through a form filled by the passenger detailing the requirements of tickets, train, date, etc.
- A form is used for each train. If the journey requires use of multiple trains, separate forms are used for each train.

✓ **Identify the Actors**

The actors in the system are the passenger, the counter clerk and the reservation system consisting of form processing, reservation, fare computation, ticket processing, ticket printing, collection of fare amount and posting as sub-systems.

The passenger is a passive user-actor who initiates the process and obtains the ticket(s), a goal of measurable value. The counter clerk is an active user-actor, who triggers the system and has the role of issuing the tickets with the responsibility of collecting the correct fare amount from the passenger, which is a measurable value. Predesigned and deployed ticket reservation system at the back end is a system actor-user to ensure that ticket processing is done correctly and different system statuses are updated on issuing of tickets. This actor has an active role and responsibility at the back end.

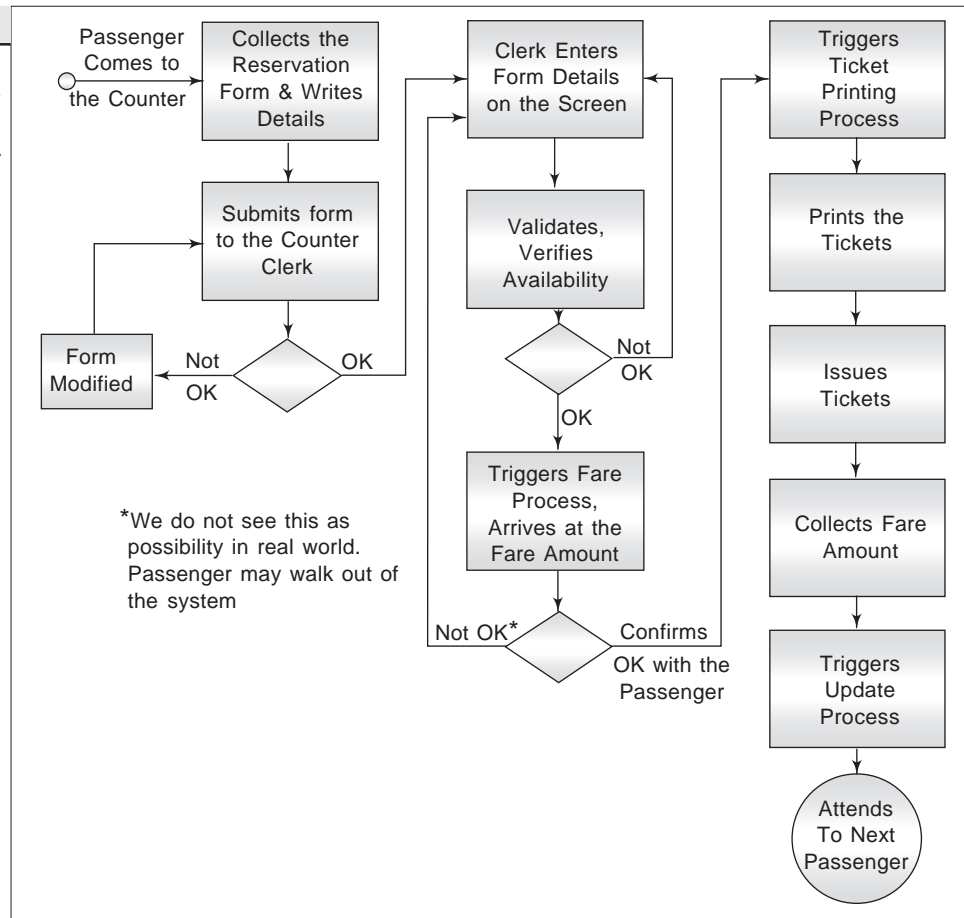
✓ **Develop Business Process Model- Reservation and Issue of Tickets**

Based on the system observation by the analyst, a high-level activity diagram is drawn modeling the process of reservation and issue of tickets to the passenger. The activity diagram brings everybody concerned with the system on the ground to a common understanding of the system as it functions.

We use this activity diagram of the process to develop use cases, which together achieves the goal of issuing the ticket.

Fig. 15.3

Activity Diagram of Reservation and Issue of Tickets



✓ Identify and Develop Use Cases

In the ticket reservation systems, users are the passenger, the counter clerk and the Reservation and Ticketing System (R&T System). Take each user and identify the roles played, which would lead us to identify the roles played, which, in turn, would lead us to an identification of use case. Table 15.1 shows the result of the process of identifying the use cases.

The system has three users, eight roles and eleven use cases. To illustrate the process of identifying the use cases, let us take the passenger (a user of the system). A passenger as a user may play one or more of three roles. The roles are

1. Enquiring about the availability of tickets on particular dates to a destination and the fare per ticket.

The role is enquiring.

Table 15.1

<i>Users–Roles–Use Cases.</i>	<i>User</i>	<i>Role</i>	<i>Use case</i>
	<ul style="list-style-type: none"> • Passenger 	<ul style="list-style-type: none"> • Enquiry • Reservation and ticketing • Cancellation 	<ul style="list-style-type: none"> • Enquire ticket availability and other details. • Reserve seats and berths, tickets • Cancel tickets
	<ul style="list-style-type: none"> • Counter clerk 	<ul style="list-style-type: none"> • Form data entry • Requisition processor • Ticket processor • Data manager 	<ul style="list-style-type: none"> • Enter Reservation Requisition Form • Process requisition for booking • Process ticket to print • Submits ticket data for updation
	<ul style="list-style-type: none"> • Reservation and ticketing system 	<ul style="list-style-type: none"> • System server 	<ul style="list-style-type: none"> • Process reservation data, process ticketing process cancellation • Update the status by date, train, etc.

2. Reserving the ticket(s) on a particular train on particular date for a destination by requisitioning through a reservation form

The role is reserving and booking tickets.

3. Cancelling the tickets after issuing and payment

The role is cancelling.

As explained in the case of passenger, the roles are use cases. Similarly, one can probe into the roles and use cases for counter clerk and reservation and ticketing system.

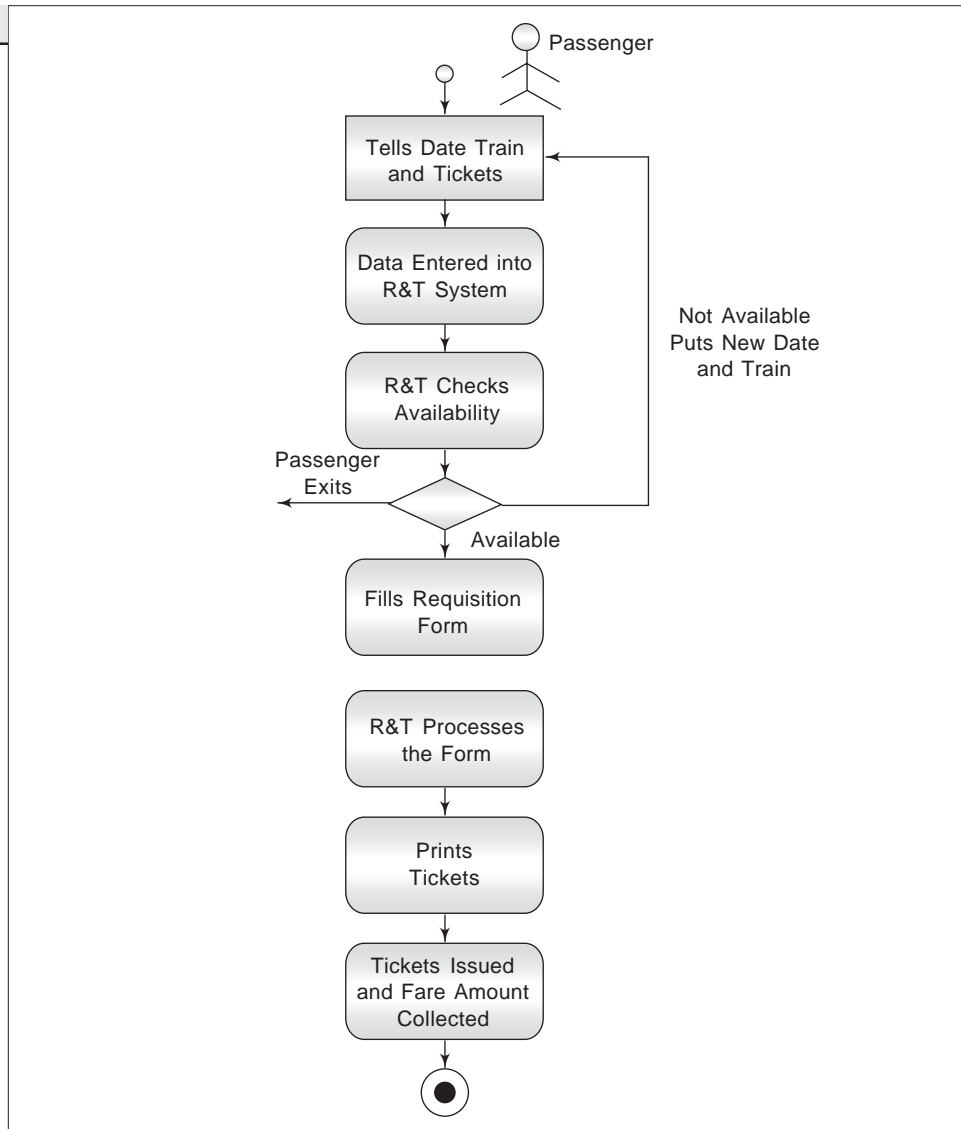
✓ **Draw Interaction Diagrams**

Interaction diagrams are used to show the interactions between user/actor and the system. Use case is a scenario that develops through interaction. Let us model different scenarios through interaction diagrams:

- Use Case: enquiring on ticket availability
 1. Passenger submits information about the date and train, and requirement of tickets to the counter clerk
 2. Clerk checks the availability of tickets on the date and train.
 3. Communicates the availability status to the passenger.
 4. If OK, the passenger proceeds to book the ticket through a requisition form.
 5. If not OK, the passenger changes the date or train and requests availability.
 6. *Steps 2 to 4 are repeated.*

Figure 15.4 shows the steps in the activity diagrams of use case enquiry to issue of Tickets.

Fig. 15.4

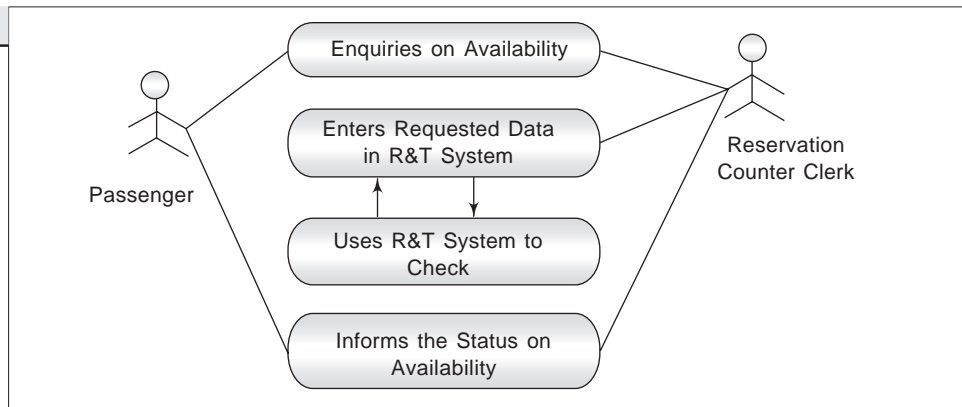
Enquiry to Issue of Tickets

Let us draw use case scenarios in use case diagrams for actor passenger.

- Use Case enquiry: Fig 15.5.

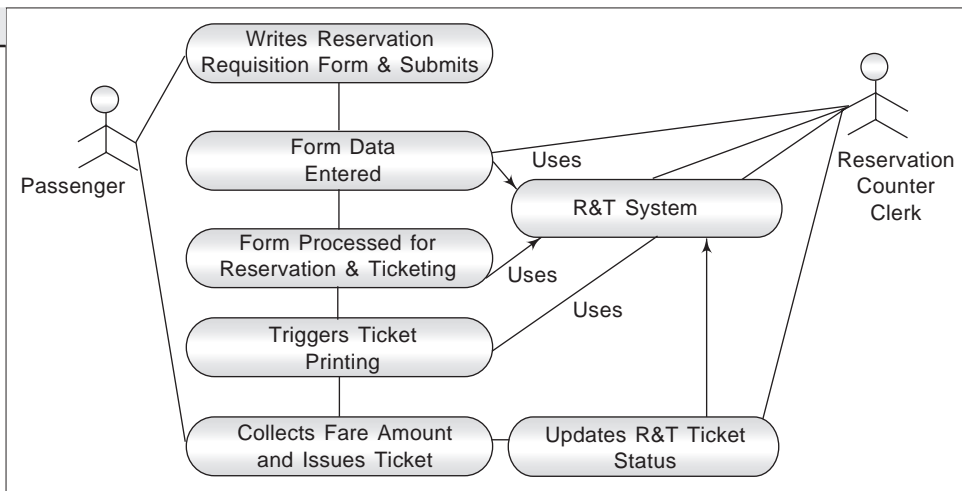
Here use case goal is to convey the ticket availability status to the passenger on the requested date and train. If the status is 'available', the passenger proceeds to book the ticket or may leave the counter. If the status is 'not available', the passenger may leave the counter or seek availability for a new date or new train.

Fig. 15.5

Use Case Enquiry

- Use Case: reservation and ticketing: Fig. 15.6.

Fig. 15.6

Use Case: Reservation and Ticketing

Steps involved in this use case are:

1. Passenger writes reservation requisition form.
2. Submits to the counter clerk.
3. Counter clerk calls 'Form' screen.
4. Counter clerk enters form data.
5. Triggers R and T processing.
6. On OK processing, triggers ticket printing.
7. Issues ticket to passenger.
8. Update the system status.

So far we have completed two of the most common and frequently used use cases, i.e. enquiry and reservation and ticketing. The following use cases can be modeled on similar lines.

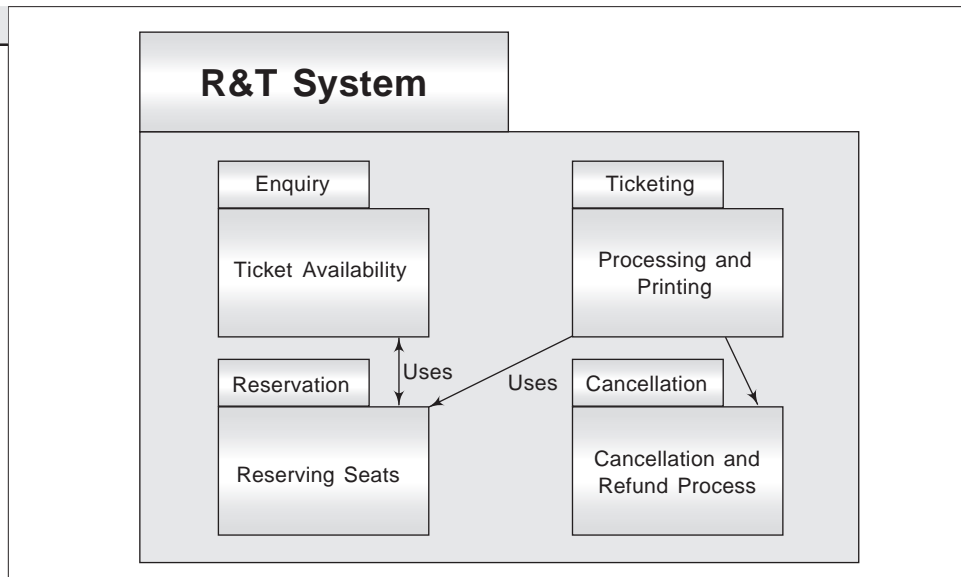
- Cancellation
- Process reservation data

- Form data entry
- Form processing
- Ticket printing
- Process ticketing
- Process cancellation
- Status updation

When the analyst has completed all use cases in the system, s/he has described and modeled the requirement of reservation and ticketing system. It is possible that in first go s/he may not be able to identify actors and hence use cases. But s/he will come across their presence in the modeling exercise, and will then go back and analyse further to introduce more actors and use cases. OOA is an intuitive process. Use case-driven OOA up till now has given us broad system requirements in terms of use cases. The OOA model using use cases is to be packaged to model the system. Figure 15.7 shows the packaging of use cases considered in the R and T system.

Fig. 15.7

*R and T
System
Packaging*



Realise that though there are eleven use cases, we have grouped them under four major use case groups, namely

- Enquiry
- Ticketing
- Reservation
- Cancellation

The remaining use cases are sub-use cases, or, in other words, these four use cases are further decomposed to bring clarity to main use case scenario.

How many case scenarios are necessary to represent the system and then to spell out the requirement? There are no set rules or guidelines on this point. More use cases may not necessarily bring better understanding or more clarity. A lot depends on users' and developers' level of comfort. What is definitely required is a use case for each major scenario: that is, for enquiry, reservation, ticketing and cancellation. Use cases for different scenarios arising out of smaller input variations need not be modeled. For example, cancellation could be part of a journey,

reduced number of tickets and so on. Amongst five recommended processes of analysis, use case-driven analysis for ascertaining system requirement is considered best as it considers users' perspective of the system. When the system is modeled into different case scenarios it not only goes down to the level of function and features, but also reveals relationships and behaviours amongst different system components.

Each use case scenario is an instance in the system that has clarity regarding goals and how they are to be achieved. The system can be decomposed from major use case at higher level going down to lowest level. This means that lower level use case scenarios together build the major scenario.

In use case-driven analysis, so far we have only understood in addition to domain knowledge.

- System scope (enquiry, reservation, ticketing and cancellation)
- System players (users, actors)
- Major functions and processes based as use case scenarios.
- Some idea on packaging system components for deployment.
- System requirements at functional and process level.

In short, so far in OOA, we have reached to some extent a situation in which the requirement analysis made so far can be put into a formal RDD document. We still have not reached the specifications level. This is possible when we go further, identifying classes, their relationship, attributes and methods. Use case-driven analysis, displayed in use case models, is a basis for moving into the step of identifying classes and designing classes.