

1 Tutorial

1.1 The system

We will go through the design and implementation of a small library management system. This system should be able to do the following things:

- Allow for checkout of books, providing a return date to the customer
- Allow for return of books
- Allow for reservation of unavailable books
- Allow for notification of overdue books
- Allow for notification of arrival of reserved books

1.2 System Specification

We have to develop a clear understanding of the system to be developed in this phase. We have to identify how the agent reacts (action) to the inputs from the environment (percept). We also have to identify the goals and sub-goals, appropriate scenarios and roles.

1.2.1 Goals

The goals are identified from the specification based on the functionality required by the system. Here the requirements are converted into goals of the system.

- Checkout books
 - Provide return date
- Return books
- Reserve unavailable books
- Give notification for overdue books
- Give notification of arrival of reserved books

Now we have to ask the question ‘how?’ for each goals and find out sub-goals.

- Checkout books
 - Record book code to the user id checked out list.
 - Provide return date.
- Return books
 - Remove book code from the user id
- Reserve unavailable books
 - Record book code as reserved for user id
 - Show the current due date for the book
- Give notification for overdue books
 - Access book record at the start of the day
 - Send email for overdue books
- Give notification of arrival of reserved books
 - Access the reserved list for user

- Send email notification

Now we will create the goal overview diagram. Select the Goal overview from system specification. Now add the goal by clicking on the goal icon and then clicking on the diagram. Add the sub-goals and connect to the main goals using the edge. Draw a diagram based on the goals and sub-goals we have already identified. The resultant goal overview diagram will be as shown in figure 48.

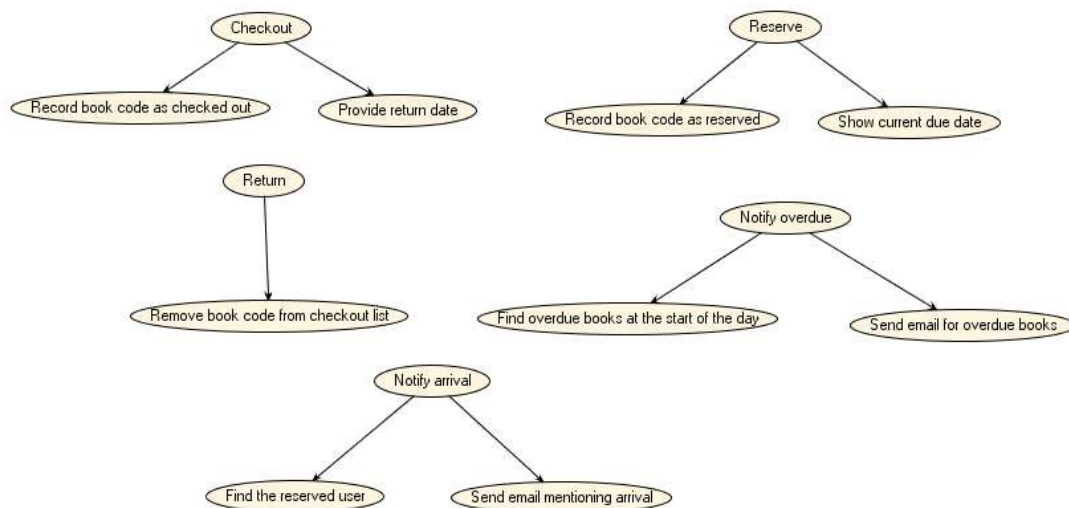


Figure 1. Goal Overview Diagram

1.2.2 Scenarios

Depending on the system we can identify 5 different scenarios.

1. When the user comes to checkout the book.
2. When the user returns the book.
3. When a book becomes overdue.
4. When the user asks to reserve a book.
5. When the reserved book arrives.

We click on the scenario icon and then click on the diagram to add the scenarios. After adding each scenario, we can specify the trigger which fires this scenario. By double clicking on the scenario, we can add, edit or remove the sequence of steps that constitute the scenario.

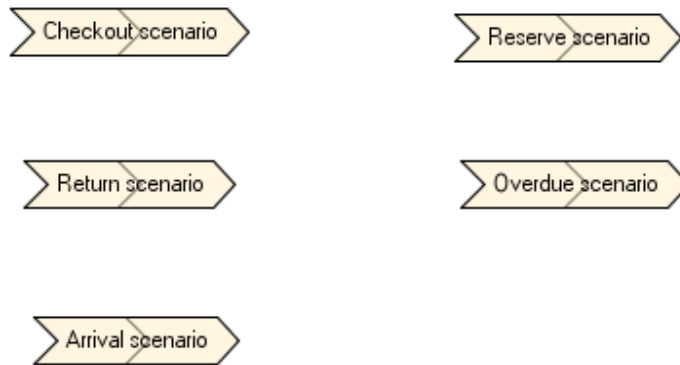


Figure 2. Scenarios Diagram

We can identify the following steps for each of the following scenarios.

Checkout Scenario

1. Request for checkout
2. Provide return date
3. Record book code as checked out
4. Provide book

Return Scenario

1. Book returned
2. Remove book code from checkout list

Reserve Scenario

1. Request for reservation
2. Record book code as reserved
3. Show current due date
4. Provide current due date

Arrival Scenario

1. Reserved book arrives
2. Find the reserved user
3. Send arrival email

Overdue Scenario

1. Start of the day
2. Find overdue books at the start of the day
3. Send overdue email

Now based on the different functionalities of the system we group it into different roles.

1. Checkout books
2. Return books
3. Overdue books
4. Reserve books
5. Send Arrival Notification

Now link the percepts, goals and actions to the roles as shown in the diagram.

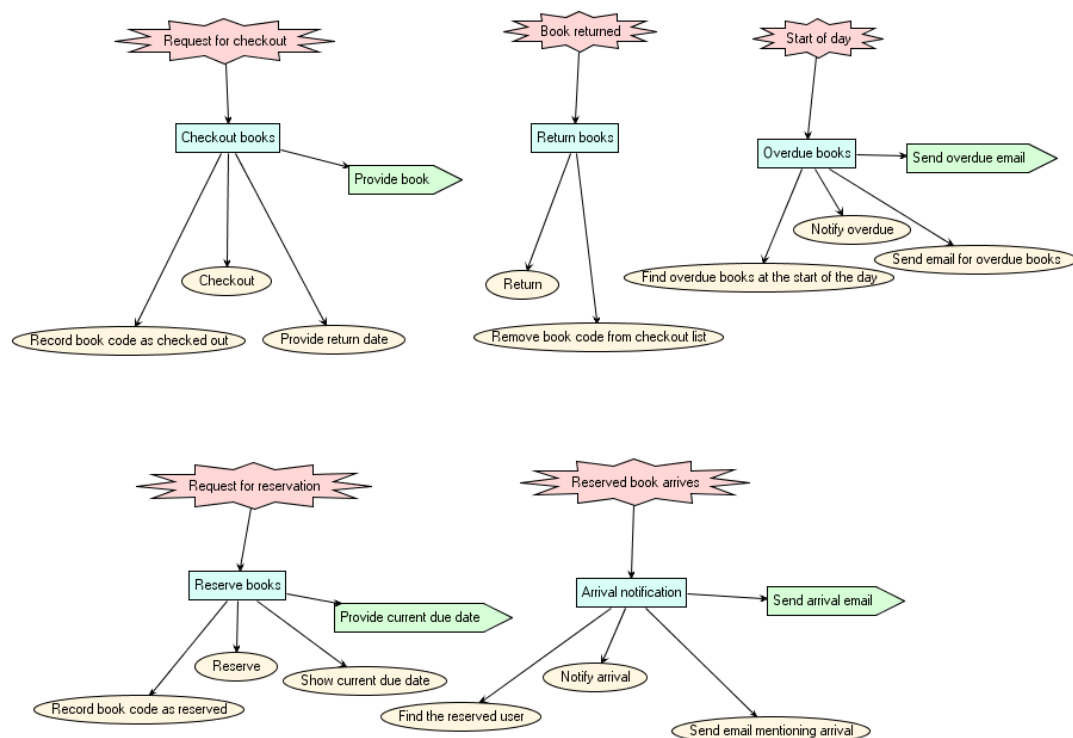


Figure 3. System Roles Diagram

1.3 Architectural Design

Now we need to identify what type of data need to be stored in the system as beliefs. By carefully checking the functionalities, we can see that we need to keep the information of all books that has been checked out and also the books that has been reserved. The checkout books role and return books role update the checkout belief and the overdue books role uses this to send overdue email. The reserve books role updates the reserve belief and the arrival notification uses it.

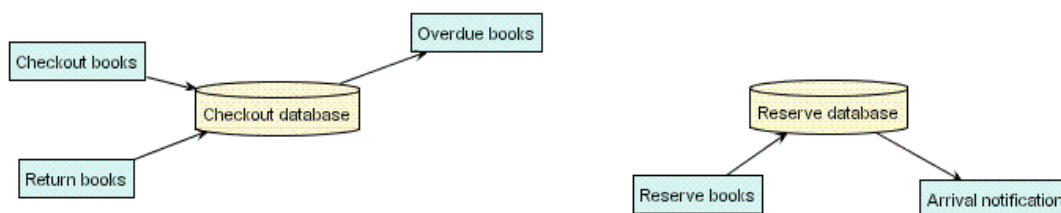


Figure 4. Data Coupling Diagram

Now we group the roles and identify 3 agents to carry out this roles in the system. The agents identified are:

1. Checkout Agent
2. Reservation Agent
3. Overdue Agent

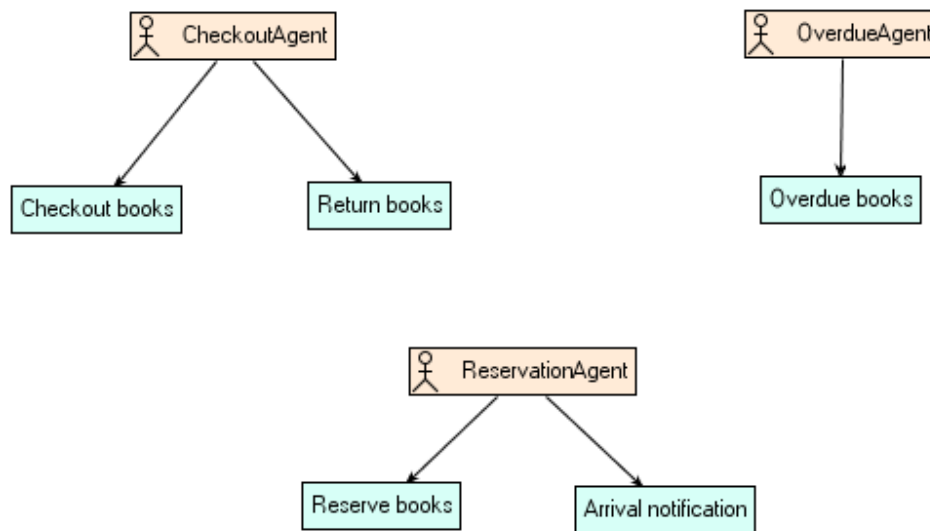


Figure 5. Agent Role Coupling Diagram

The agents interact with each other. This is shown in the agent acquaintance diagram. The links between the agents are automatically established when a message is sent from one agent to another as shown in the system overview diagram.

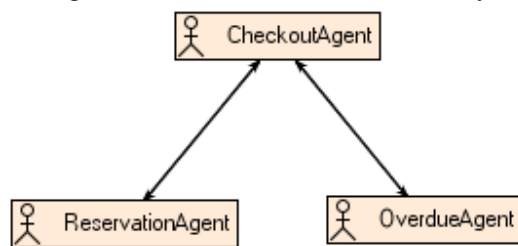


Figure 6. Agent Acquaintance Diagram

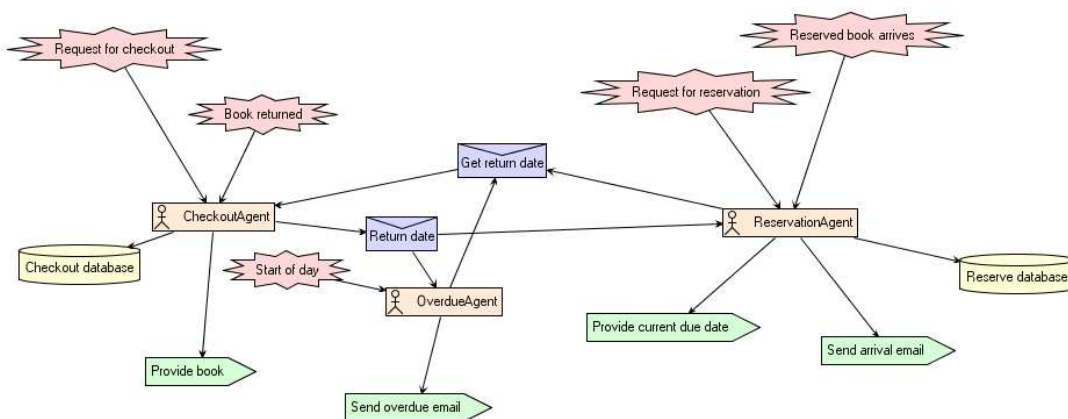


Figure 7. System Overview Diagram

1.4 Detailed Design

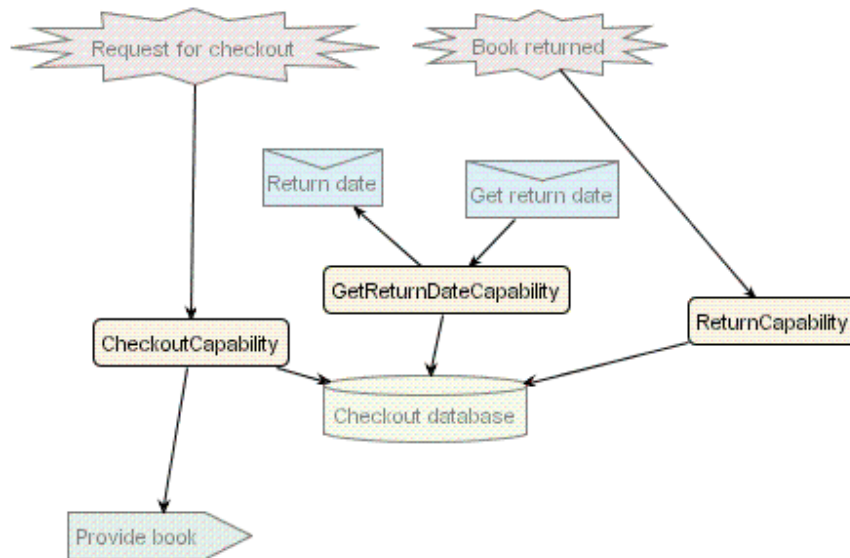
We will go into details of agents and their capabilities.

1.4.1 Checkout Agent

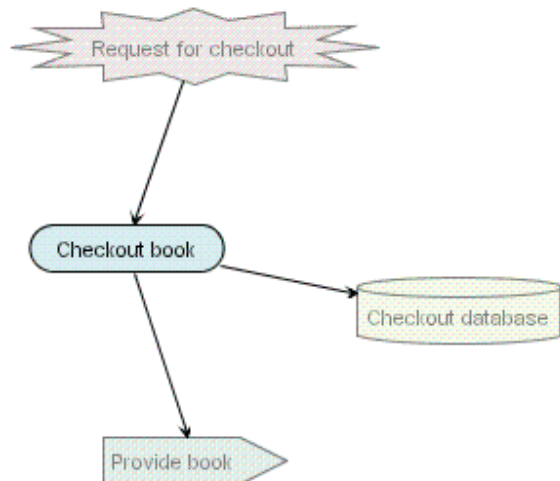
For the checkout agent we can identify 3 capabilities:

1. Checkout Capability
2. Return Capability
3. Get Return Date Capability

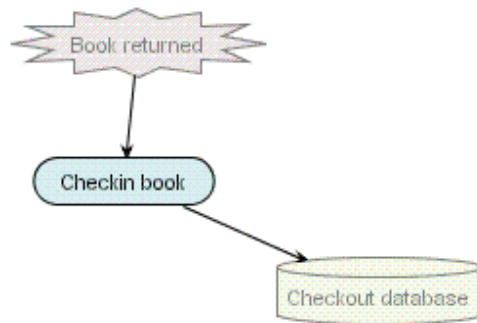
The agent overview diagram of the checkout agent is as shown in the figure.



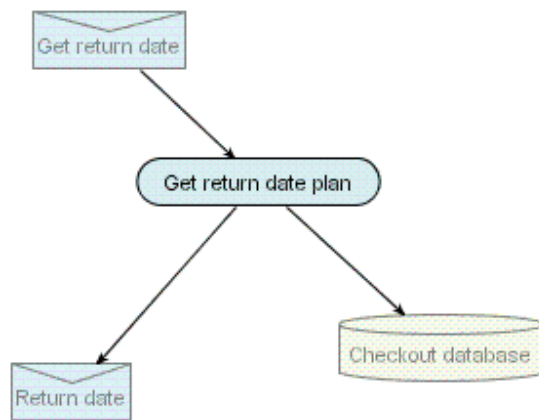
1.4.1.1 Checkout Capability



1.4.1.2 Return Capability



1.4.1.3 Get Return Date Capability

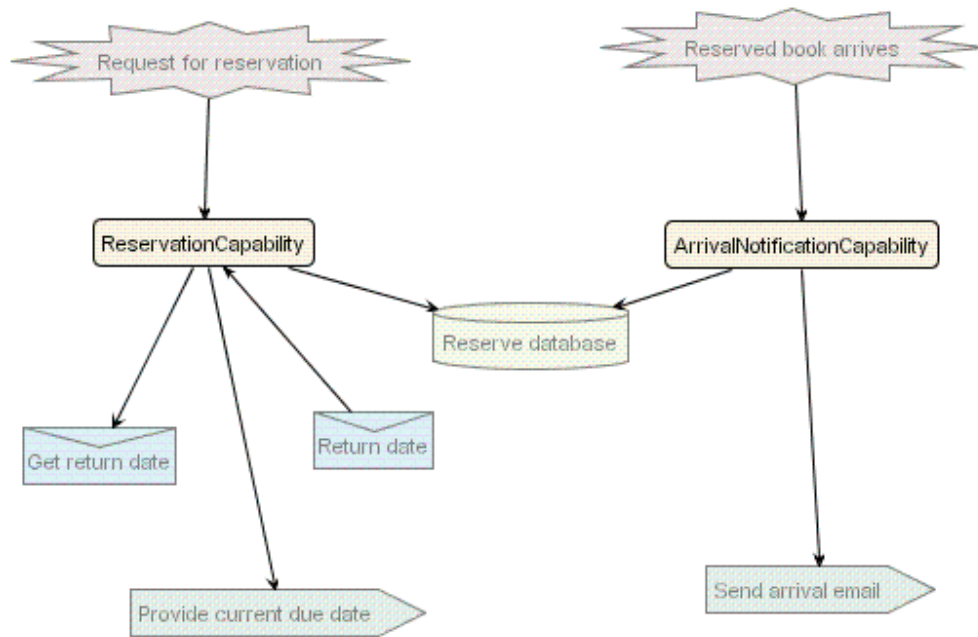


1.4.2 Reservation Agent

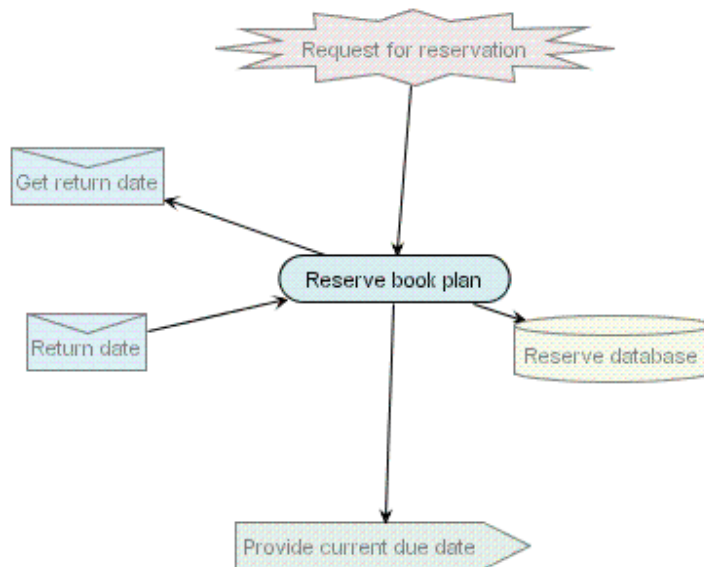
For the reservation agent, we can identify 2 capabilities:

1. Reservation Capability
2. Arrival Notification Capability

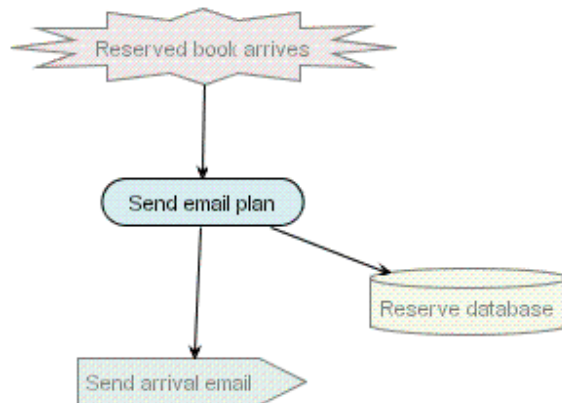
The agent overview diagram of the reservation agent is as shown in the figure.



1.4.2.1 Reservation Capability



1.4.2.2 Arrival Notification Capability



1.4.3 Overdue Agent

The agent overview diagram of the overdue agent is as shown in the figure.

