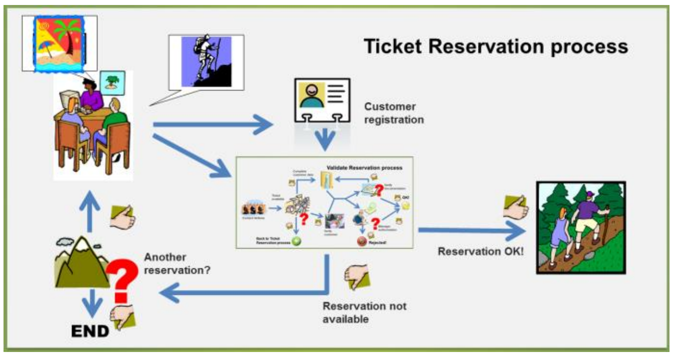
Definition of concurrent tasks, detection and identification of errors.

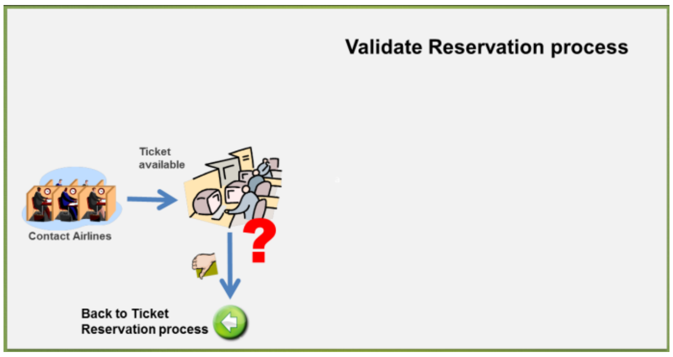
The ticket reservation process used by the Travel Agency includes a sub-process that we've called Validate Reservation. This sub-process validates the details of the reservation that has been entered.



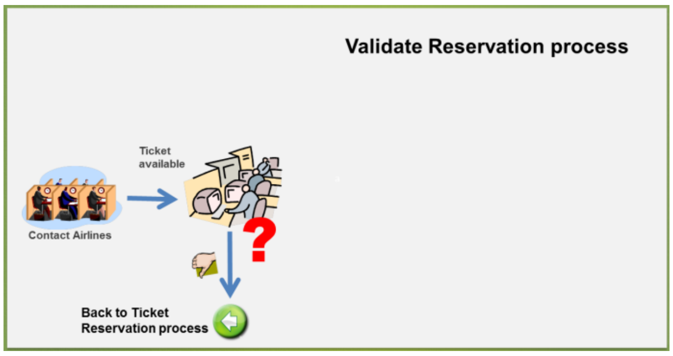
During this process, the reservation availability is checked in the first place by contacting the corresponding airlines to confirm if there are flights available on the dates requested.

Validate Reservation process
Contact Airlines

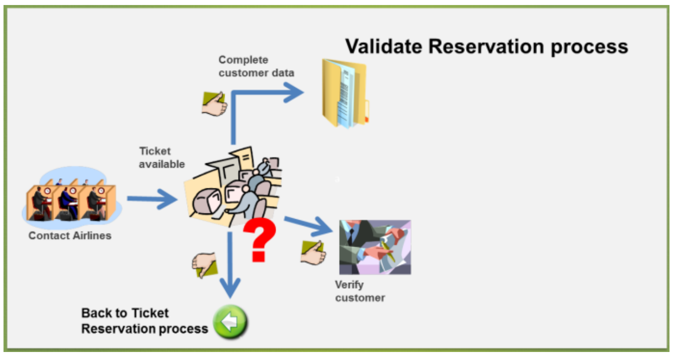
The information obtained may indicate that there are no tickets available. In this case, we should go back to the ticket reservation process to ask the customer if he wants to make another reservation.


Validate Reservation process
Ticket available
Contact Airlines
Back to Ticket Reservation process

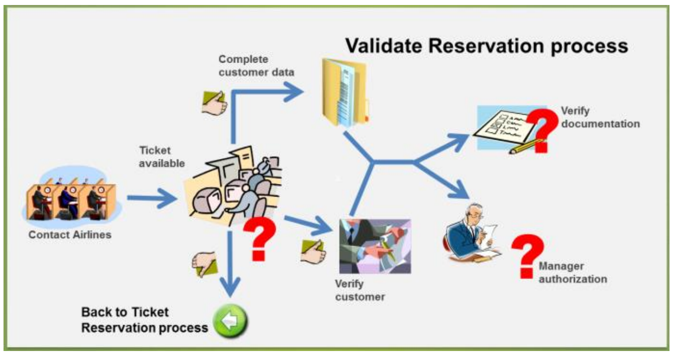
Or it can also happen that there are tickets available for the dates requested and the validation process continues. To do so, two paths must be followed.


Validate Reservation process
Ticket available
Contact Airlines
Back to Ticket Reservation process

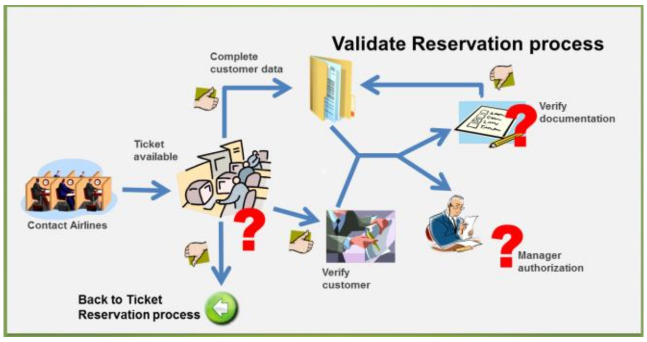
On one hand the customer's details necessary for the trip need to be completed, such as visas, passports, immunizations, etc. In addition, depending on the cost of the trip, information on the customer's financial situation must be obtained.



After obtaining all the required information, it has to be evaluated. First, the documents required for the trip have to be verified, and then the customer care manager will have to authorize the reservation.

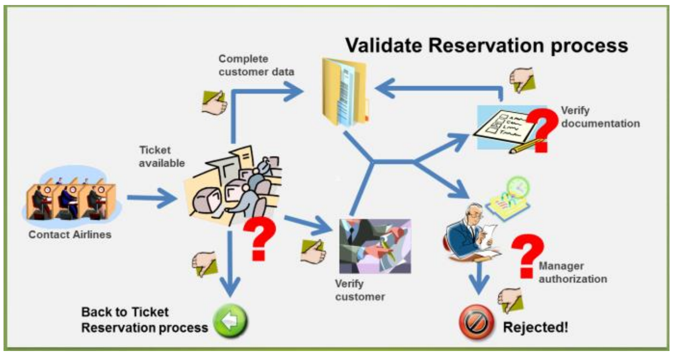


If a document is missing, we must go back to obtain what is needed.

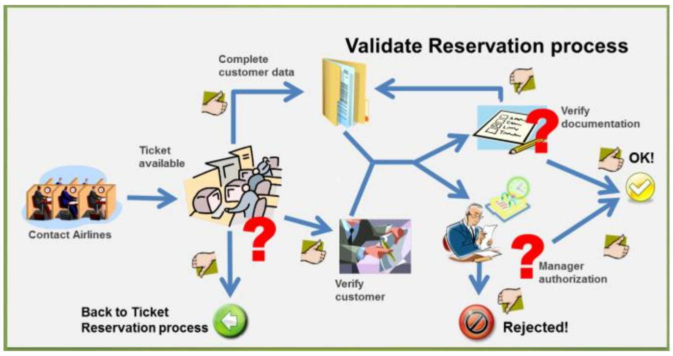


The customer care manager will have to examine the reservation and decide whether to authorize it within a certain time frame. For this reason, regular notifications will be displayed to remind him about this task.

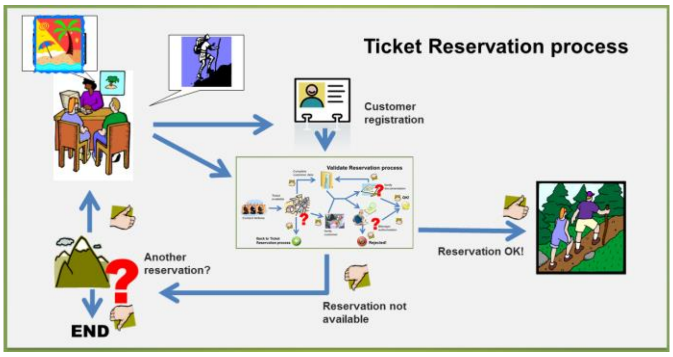
If, after examining it, he decides not to authorize it, he will have to inform the reason to the customer. The validation sub-process will end and so will the ticket reservation process.



If the manager authorizes the reservation and all the documentation has been provided, the reservation will be authorized.



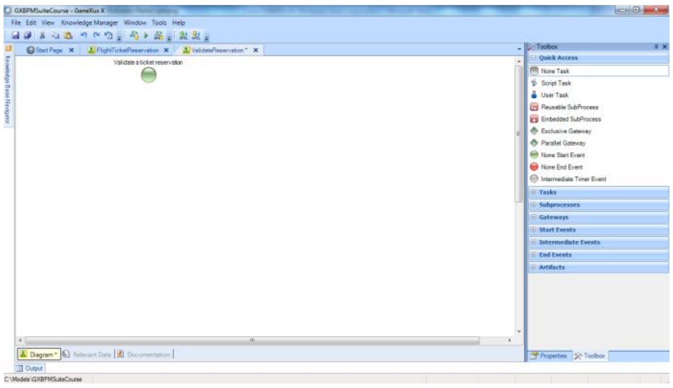
In this case, the reservation validation sub-process will end. The system will return to the main ticket reservation process to notify the customer that the reservation has been confirmed.



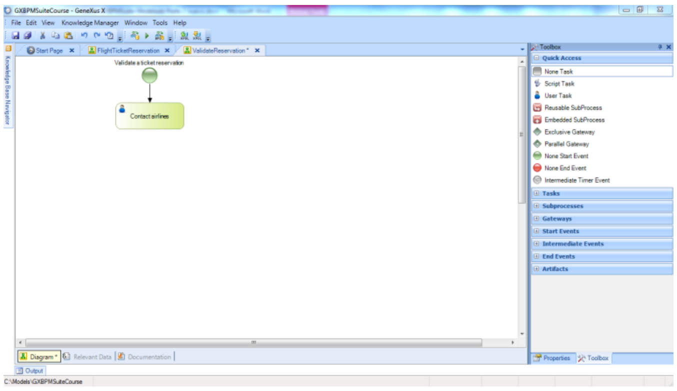
Now we will model the reservation validation process step by step using GeneXus' Business Process Modeler.

First, we create an object of Business Process Diagram type and name it ValidateReservation.

To indicate the start of the process, we drag a None Start Event symbol from the toolbar and add the description “Validate a ticket reservation”.

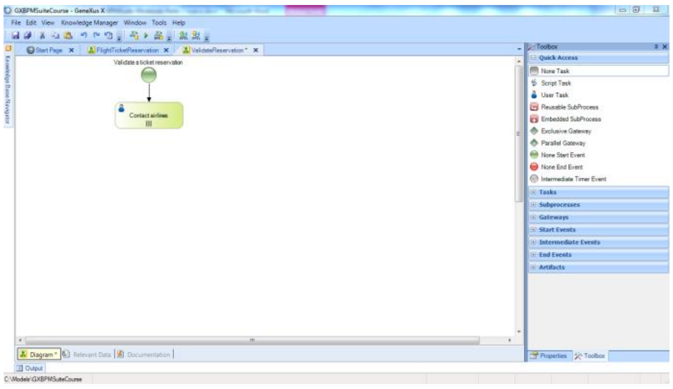


Now we’ll add the first task, which is to contact airlines. Because it is an interactive task, we drag a User task to the diagram, press F2 and name it Contact Airlines. Lastly, we join it from the start node.

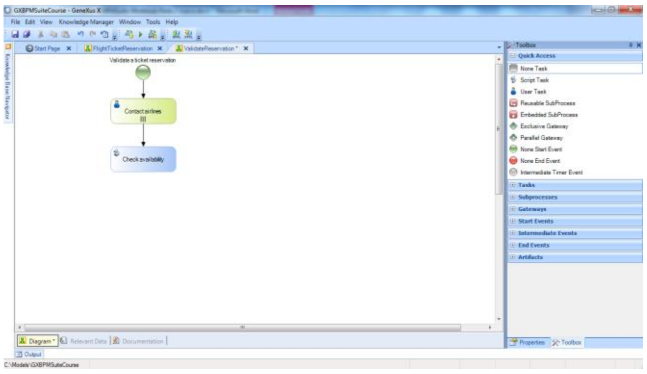


A feature of this task is that it will be executed a certain number of times because several airlines have to be contacted. This may also be performed simultaneously.

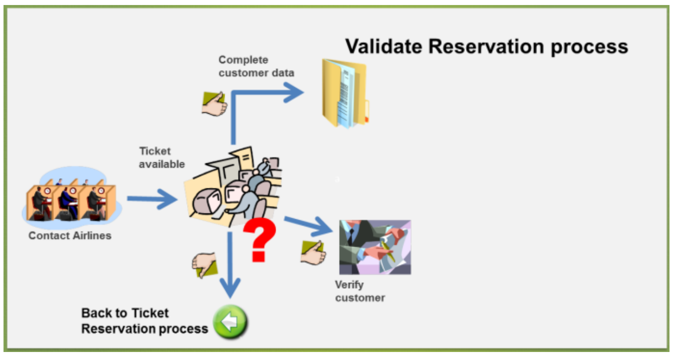
To define it, we open the **Loop type** property and set it to **Multi-Instance**.



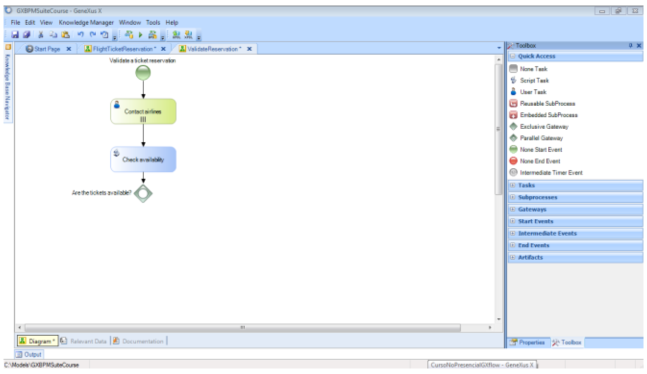
Once all airlines have been contacted, we need to examine the information obtained and check if there are flights available on the dates requested. To model this, we insert a Script task called Check availability and connect it from the Contact Airlines task.



Now we need to evaluate the result of these enquiries. If there are no tickets available, we need to return to the main reservation process and ask the customer if he wants to make another reservation. If there are flights available, we need to follow 2 paths at the same time: complete the trip's details and check the customer's financial situation.



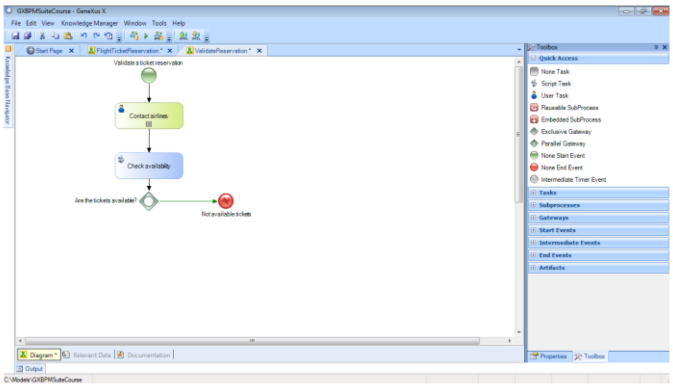
We can't use an exclusive gateway to model this because the flow will have to follow more than one path. Therefore, we drag an **Inclusive Gateway** symbol from the toolbar, connect it from the Check Availability task and add the description "Are the tickets available?"



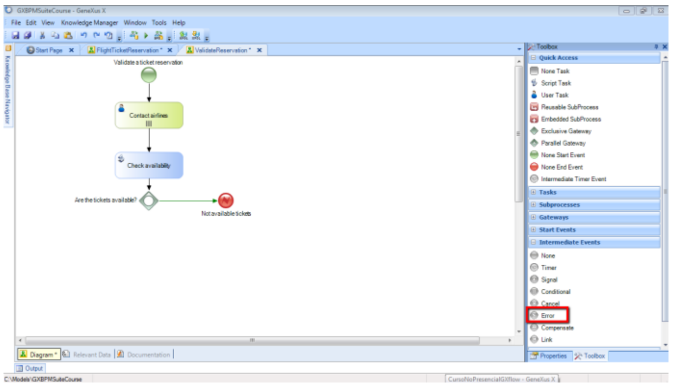
In this type of node, a condition is defined for each path coming out of it, and the flow will follow all the paths that meet the conditions.

A possible path is when there are no tickets available. In this case, we need to indicate that the ticket validation process must be cancelled because there are no tickets to validate. In addition, we must return to the main process to notify the customer and offer the possibility to request new dates for the reservation.

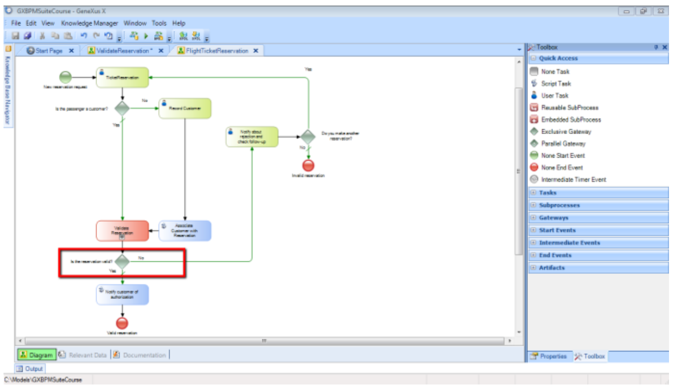
We can model this situation of “termination by error” by inserting an Error End Event. Also, we join it from the Inclusive Gateway and add the description “No tickets available”. In the Error Code property we type “NO_TICKETS_AVAILABLE”.



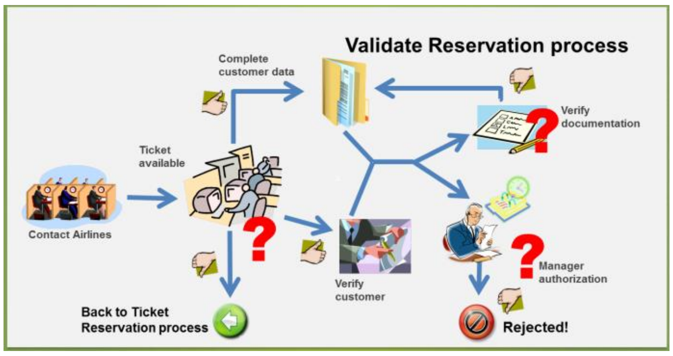
This type of end event throws an error that allows the flow to continue in another process, to which we will add an intermediate error event in order to find this error.



If we return to the main ticket reservation process, we can see that it required an Exclusive Gateway to evaluate if the reservation was valid.

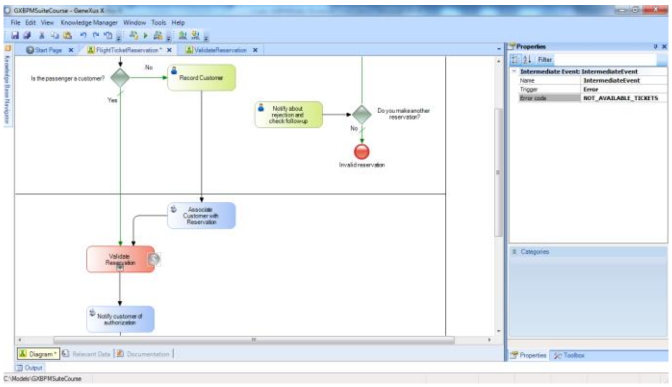


However, this model doesn’t tell us why the reservation is not valid. As we’ve said before, the reservation may not be completed when there are no tickets available or the customer care manager doesn't authorize it.



Using an “error end event” and an “error intermediate event”, this model allows us to find the situation that caused the error in the sub-process and take various actions in the main process.

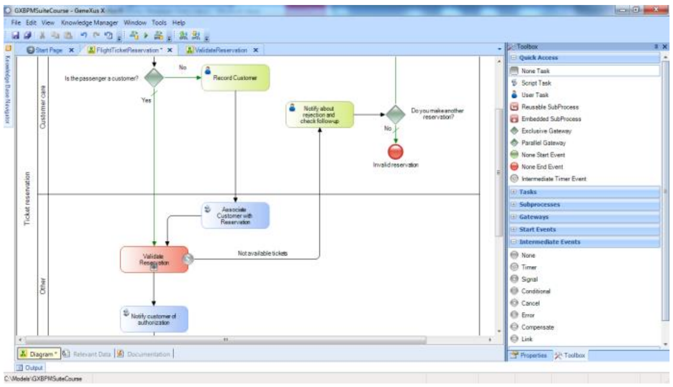
To model this, we modify the FlightTicketReservation process by deleting the Gateway that evaluated the reservation’s validity and replacing it with an “error intermediate event” that we add on the sub-process symbol. We edit this event’s properties and in the Error Code we type “NO_TICKETS_AVAILABLE”. This error code must match the one defined in the “error end event” of the ValidateReservation sub-process.


GXBPMSuiteCourse - GeneXus X
File Edit View Knowledge Manager Window Tools Help
Start Page
FlightTicketReservation*
ValidateReservation
Is the passenger a customer?
No
Yes
Record Customer
Notify about rejection and check followup
Do you make another reservation?
No
Invalid reservation
Associate Customer with Reservation
Validate Reservation
Notify customer of authorization
Properties
Filter
Categories
Diagram*
Relevant Data
Documentation
Output
Toolbox
C:\Models\GXBPMSuiteCourse

Intermediate Event: IntermediateEvent

Name	IntermediateEvent
Trigger	Error
Error code	NOT_AVAILABLE_TICKETS

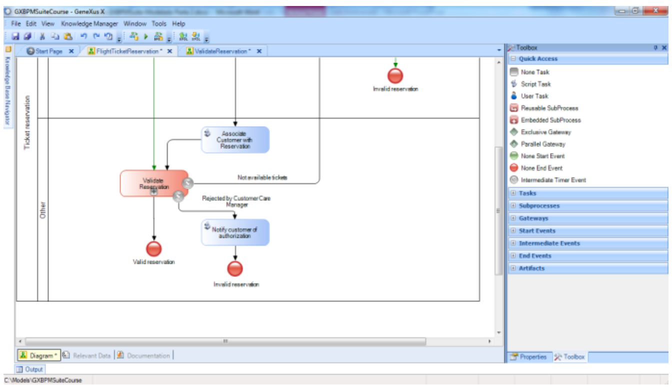
We connect this event to the task “Notify about rejection and check follow up”; to this connection we add the description “No tickets available” and complete the diagram by connecting the ValidateReservation sub-process to the task “Notify customer of authorization”.



In this way we have modeled the case when the reservation cannot be completed because there are no tickets available.

To model the case when the customer care manager doesn't authorize the reservation, we add another "error intermediate event" to the sub-process. Next, we drag a script task from the toolbar, add the description "Notify rejection to customer" and connect it from the error event. To this connection we add the description "Rejected by Customer Care Manager".

Lastly, we add a None End Event with the description "Invalid reservation" and connect it from the notification task.


GXBPMSuiteCourse - GeneXus X
File Edit View Knowledge Manager Window Tools Help
Start Page
FlightTicketReservation *
ValidateReservation *
Knowledge Base Navigator
Ticket reservation
Other
Invalid reservation
Associate Customer with Reservation
Validate Reservation
Not available tickets
Rejected by Customer Care Manager
Notify customer of authorization
Valid reservation
Invalid reservation
Toolbox
Quick Access
None Task
Script Task
User Task
Reusable SubProcess
Embedded SubProcess
Exclusive Gateway
Parallel Gateway
None Start Event
None End Event
Intermediate Timer Event
Tasks
Subprocesses
Gateways
Start Events
Intermediate Events
End Events
Artifacts
Diagram *
Relevant Data
Documentation
Properties
Toolbox
Output
C:\Models\GXBPMSuiteCourse