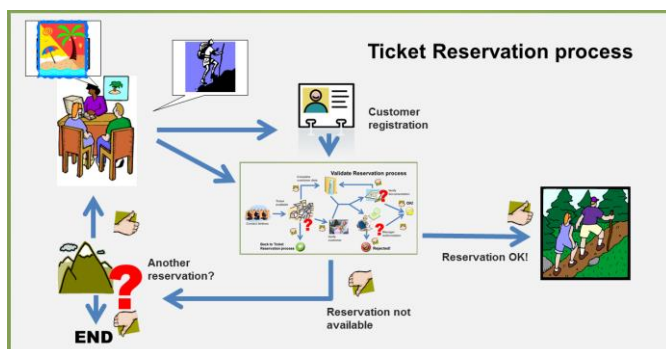


Definición de tareas concurrentes, detección e identificación de errores

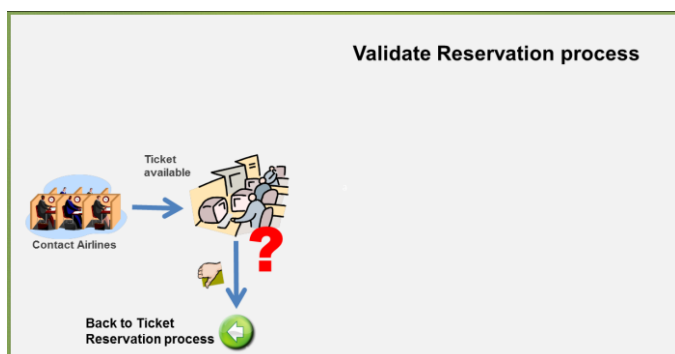
El proceso de reserva de pasajes de la Agencia de Viajes, incluye un subproceso que llamamos Validate Reservation, que se encarga de validar los datos de la reserva registrada.



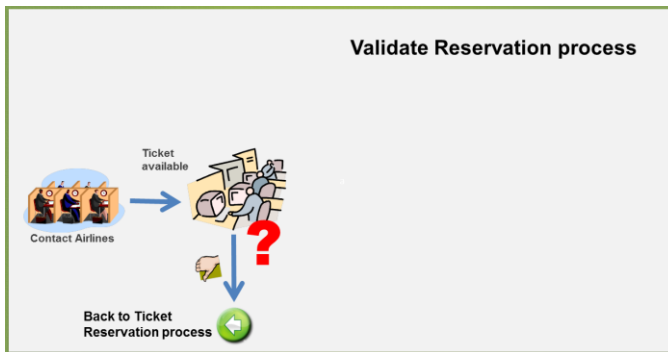
En el desarrollo de este proceso, en primer lugar, se verifica la disponibilidad de la reserva contactando las aerolíneas para ver si hay vuelos disponibles en las fechas solicitadas.



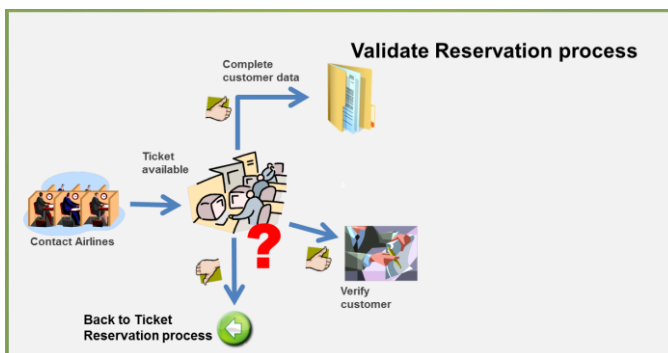
Una vez obtenida la información puede pasar que no haya disponibilidad de tickets, en cuyo caso se deberá volver al proceso de reserva de tickets, para preguntarle al cliente si desea hacer otra reserva.



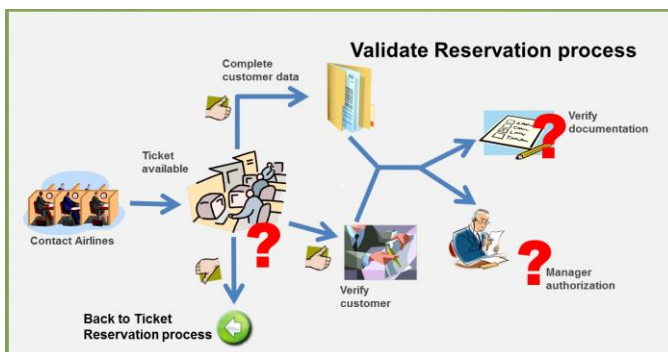
O bien hay tickets disponibles para la fechas solicitadas y se continúa adelante con el proceso de validación, para lo cual se deberá seguir por dos caminos...



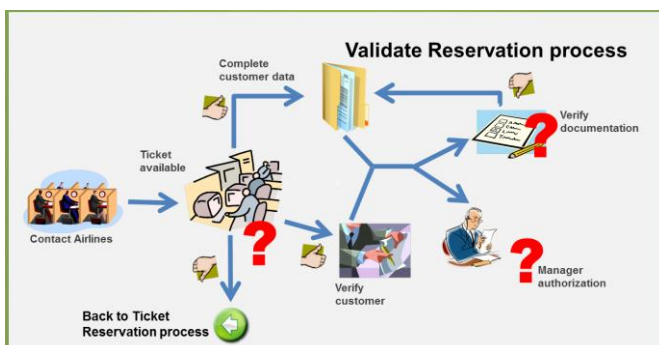
Por un lado se deberá completar los datos del cliente necesarios para el viaje, como visas, pasaportes, vacunas, etc. Y por otro lado, dependiendo del importe del viaje, se deberá obtener información de la situación financiera del cliente.



Una vez finalizada la obtención de la información requerida, se deberá proceder a evaluarla, por una parte verificar toda la documentación requerida por el viaje y por otra parte, el gerente de atención al cliente deberá autorizar la reserva.

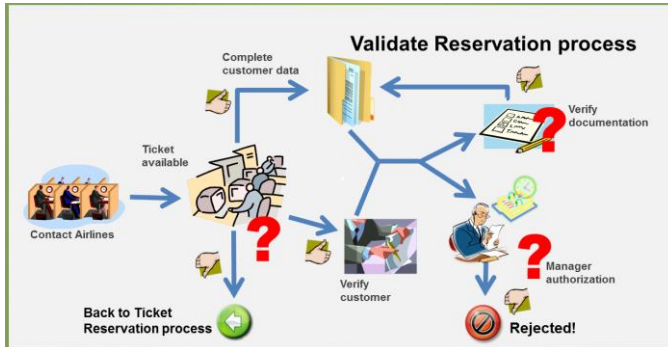


Si falta algo de la documentación, deberá volverse a obtener lo que sea necesario.

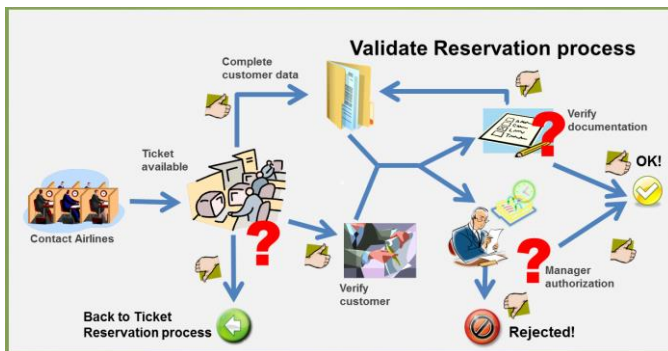


Para estudiar la autorización de la reserva, el gerente de atención al cliente dispondrá de un tiempo determinado y tendrá avisos periódicos que le recuerden esta tarea.

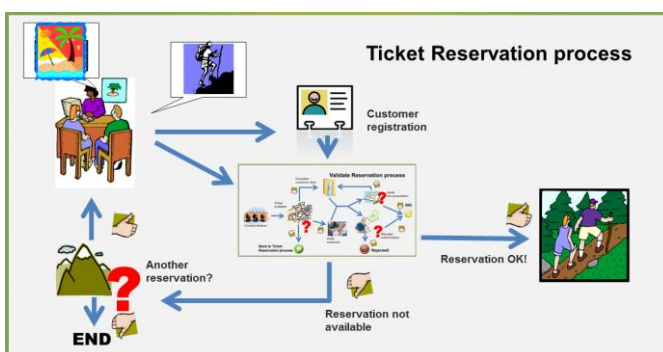
Si al analizar la misma, decide no autorizarla, deberá notificar los motivos al cliente y finalizará el subproceso de validación y también el proceso de reserva de tickets.



En caso de que se otorgue la autorización del gerente y que la documentación esté toda en orden, se autorizará la reserva.



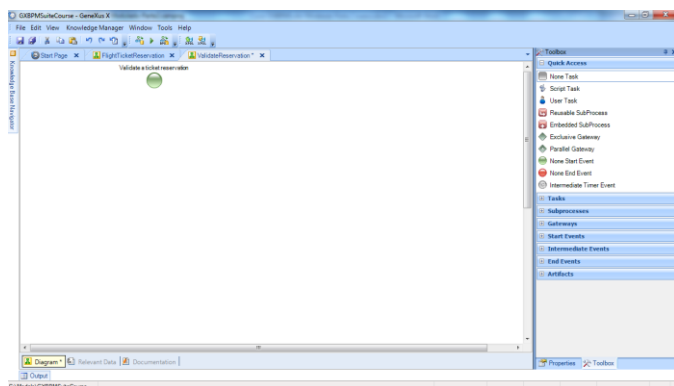
En este caso finalizará el subproceso de validar la reserva y se volverá al proceso principal de reserva de tickets, para notificar al cliente de que la reserva está confirmada.



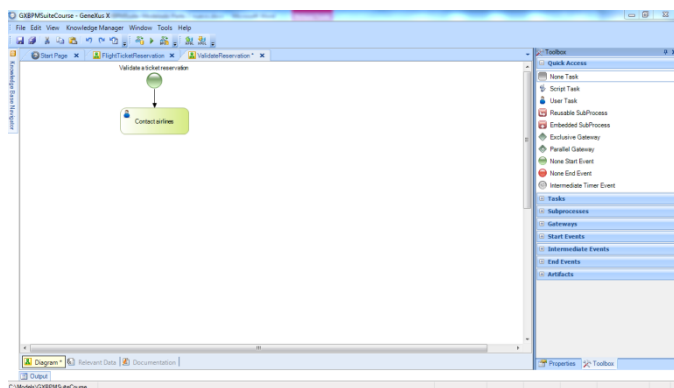
Modelaremos ahora el proceso de validación de la reserva paso a paso utilizando el Business Process Modeler de GeneXus.

Lo primero que hacemos es crear un objeto del tipo Business Process Diagram y lo llamaremos ValidateReservation.

Para indicar el comienzo de nuestro proceso, arrastramos un símbolo de None Start Évén desde la toolbar y le agregamos la descripción “Validate a ticket reservation”.

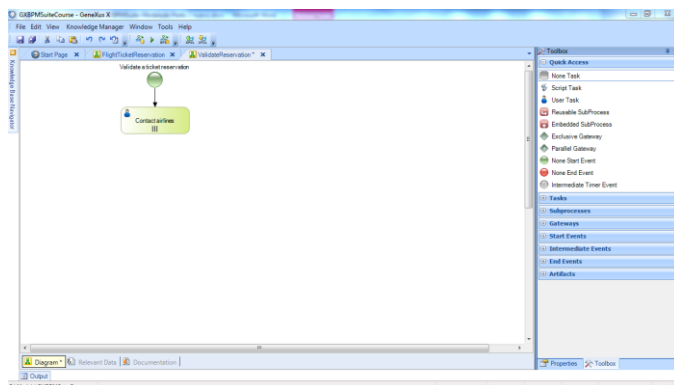


Ahora agregaremos la primer tarea a realizarse que es la de contactar a las aerolíneas. Como es una tarea interactiva arrastramos una tarea User al diagrama, con F2 le ponemos de nombre Contact Airlines y la unimos desde el nodo de comienzo.

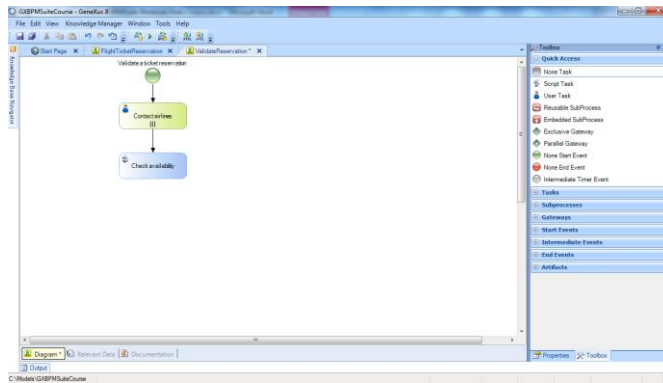


Esta tarea tiene la particularidad de que va a ser ejecutada un cierto número de veces, ya que es necesario contactar a varias aerolíneas y esto incluso podría realizarse simultáneamente.

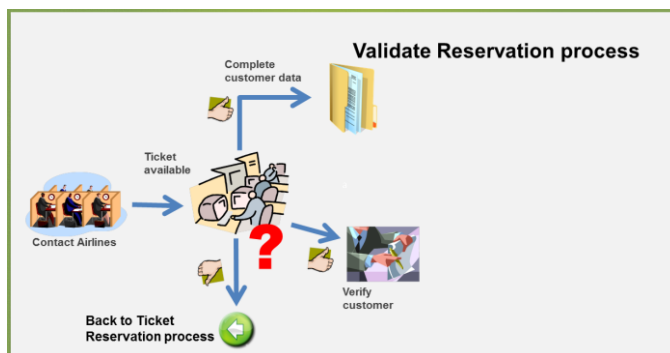
Para definir esto, vamos a la propiedad **Loop type** y asignamos el valor **Multi-Instance**.



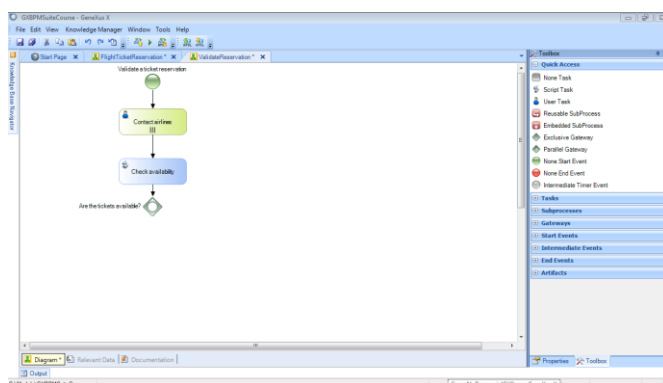
Una vez contactadas todas las aerolíneas debemos analizar la información obtenida y chequear si hay vuelos disponibles en las fechas solicitadas. Para modelar esto insertamos una tarea Script con el nombre Check availability y la conectamos desde la tarea Contact Airlines.



Ahora debemos evaluar el resultado del chequeo. Si no hay disponibilidad de tickets debemos volver al proceso principal de reservaciones para que se le pregunte al cliente si desea hacer otra reserva, pero si hay disponibilidad debemos seguir por 2 caminos a la vez: completar los datos para el viaje y verificar la situación financiera del cliente.



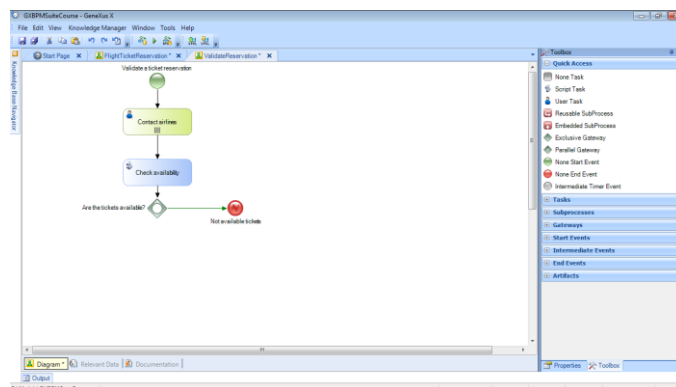
Para modelar esto, no podemos utilizar un exclusive gateway ya que el flujo deberá continuar por más de un camino, por lo que arrastramos desde la toolbar un símbolo de **Inclusive Gateway**, lo conectamos desde la tarea Check Availability y le agregamos la descripción “Are the tickets available?”



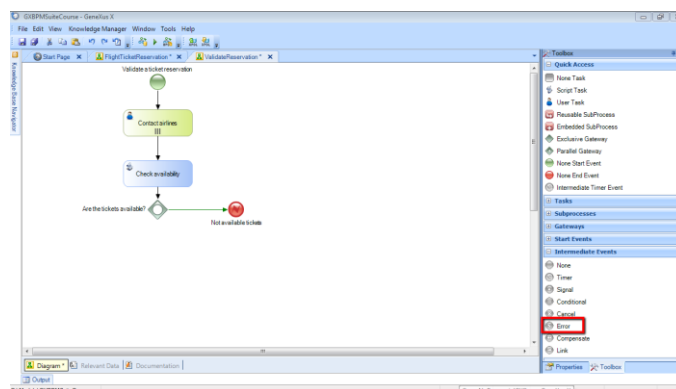
En este tipo de nodo se define una condición por cada camino que sale del mismo y el flujo seguirá por todos los caminos que cumplan las condiciones.

Un camino posible es cuando no hay disponibilidad de tickets. Si pasa esto, debemos indicar que debe cancelarse el proceso de validación de tickets ya que no hay tickets para validar y debe volverse al proceso principal para notificar esto al cliente y ofrecerle la posibilidad de solicitar nuevas fechas para la reserva.

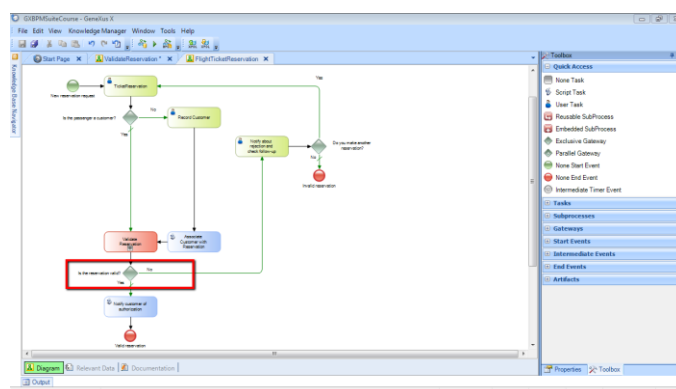
Podemos modelar esta situación de “finalización por error”, insertando un Error End Event, al que unimos desde el inclusive Gateway y agregamos la descripción “Not available tickets”. En la propiedad Error Code escribimos “NOT_AVAILABLE_TICKETS”.



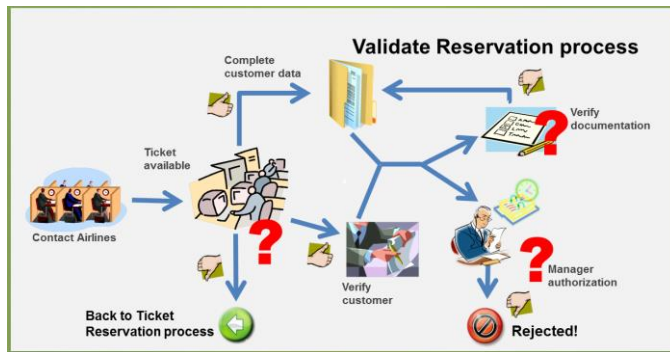
Este tipo de evento de fin, dispara un error que permite que el flujo continúe en otro proceso, al que agregaremos un evento intermedio de error, con el fin de detectar dicho error.



Si volvemos al proceso principal de reserva de tickets, vemos que en el mismo era necesario utilizar un exclusive gateway para evaluar si la reserva era válida.

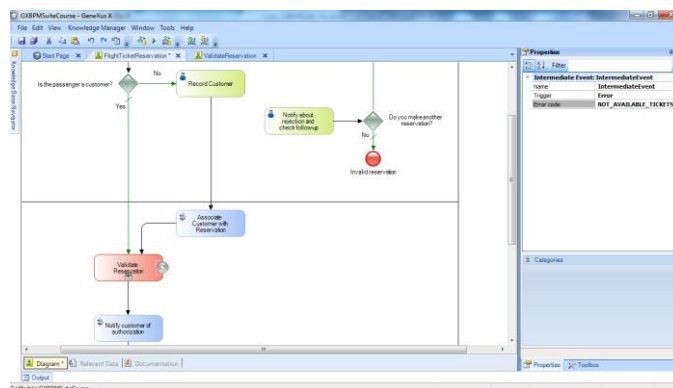


Sin embargo, este modelo no nos permite discernir por qué razón la reserva no es válida, ya que como vimos, podía pasar que no pudiera continuarse con la reserva porque no había tickets disponibles o porque el gerente de atención al cliente no autorizaba la reserva.

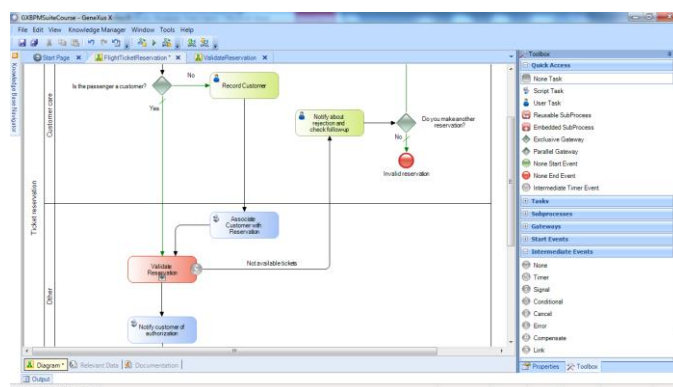


El modelo utilizando un “error end event” y un “error intermediate event” nos permite detectar qué situación provocó el error en el subproceso y tomar distintas acciones en el proceso principal.

Para modelar esto, modificamos el proceso FlightTicketReservation, eliminando el Gateway que evaluaba la validez de la reserva y lo sustituimos por un “error intermediate event” que colocamos sobre el símbolo del subproceso. Editamos las propiedades de este evento y en Error Code escribimos “NOT_AVAILABLE_TICKETS”. Este código de error debe coincidir exactamente con el definido en el “error end event” del subproceso ValidateReservation.



Conectamos este evento con la tarea “Notify about rejection and check follow up”, a esta conexión le agregamos la descripción “Not available tickets” y completamos el diagrama conectando el subproceso ValidateReservation a la tarea Notify customer of authorization.



Con esto modelamos el caso de que no pudiera continuarse con la reserva porque no había tickets disponibles.

Para modelar el caso de que el gerente de atención al cliente no autoriza la reserva, agregamos otro “error intermediate event” al subproceso. Luego arrastramos una tarea script desde la toolbar, le ponemos la descripción “Notify rejection to customer” y la conectamos desde el error event. A esta conexión le ponemos la descripción “Rejected by Customer Care Manager”.

Por último agregamos un None end event con la descripción “Invalid reservation” y lo conectamos desde la tarea de la notificación.

