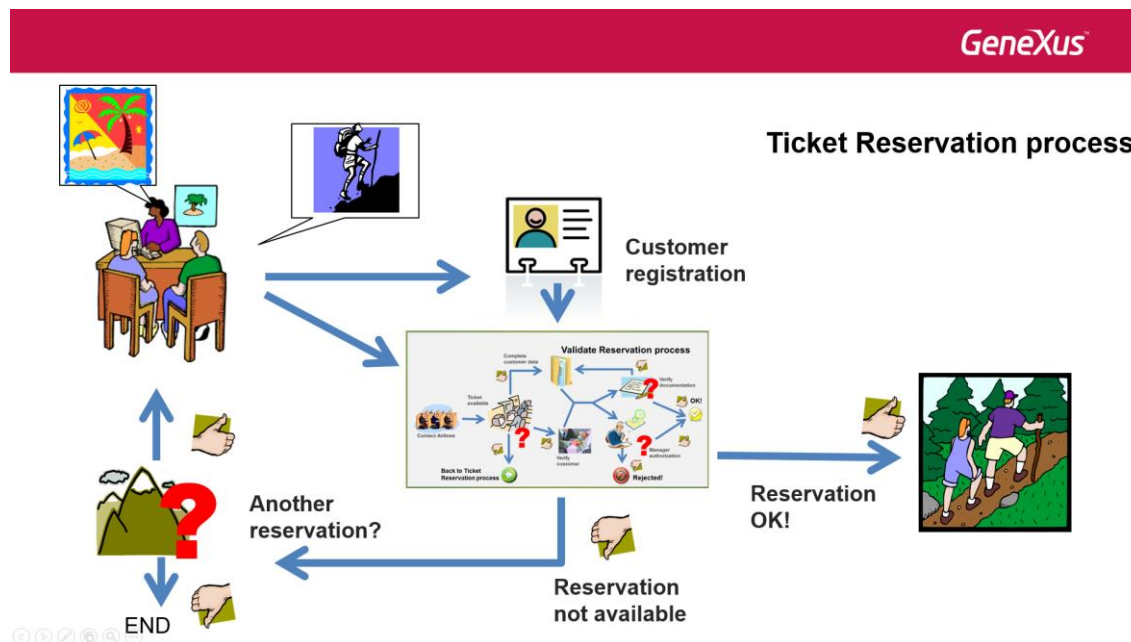
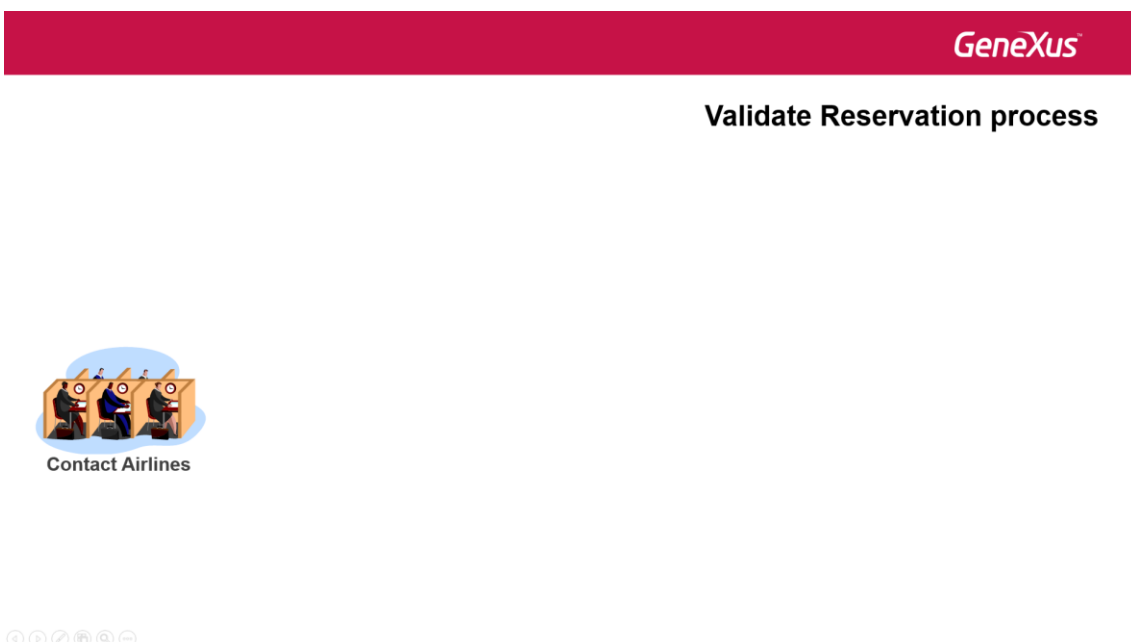


Definição de tarefas simultâneas, detecção e identificação de erros

O processo de reserva de passagens feito pela Agência de viagens inclui um subprocesso que chamamos de Validate Reservation, responsável por validar os dados da reserva registrada.

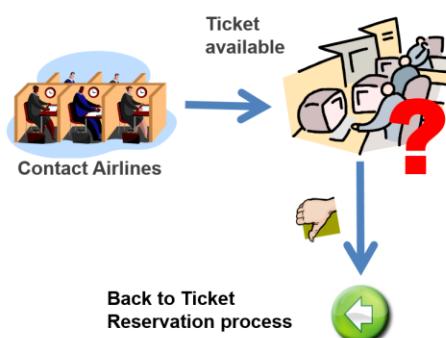


No desenvolvimento deste processo, o primeiro passo é verificar a disponibilidade da reserva entrando em contato com as companhias aéreas para ver se há voos disponíveis nas datas solicitadas.



Pode ser que não haja mais passagens disponíveis. Caso isso aconteça, é preciso voltar ao processo de reserva de passagens para perguntar ao cliente se ele aceita fazer outra reserva.

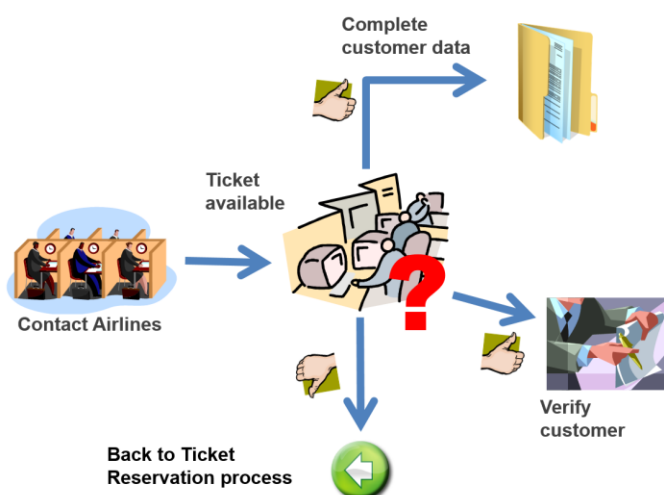
Validate Reservation process



Quando há passagens disponíveis para as datas solicitadas, damos continuidade ao processo de validação e dois caminhos podem ser tomados...

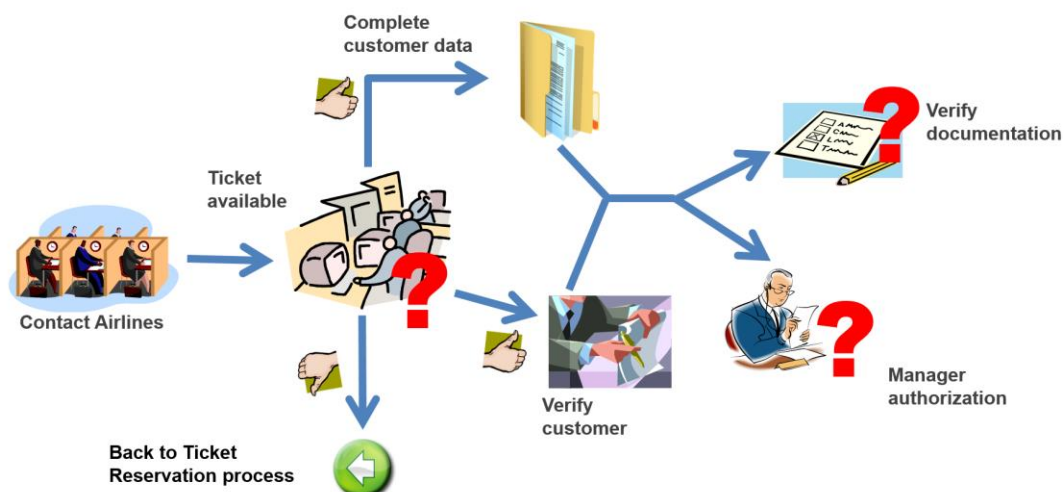
Por um lado, deve-se completar os dados necessários do cliente para a viagem, tais como passaporte, visto, vacinas etc. Por outro, dependendo do porte da viagem, é preciso obter informações sobre a situação financeira do cliente.

Validate Reservation process



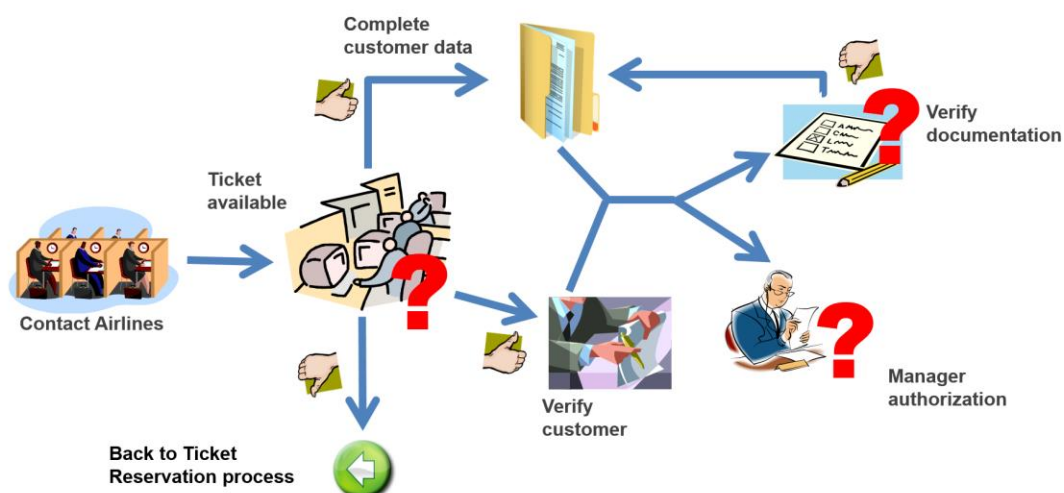
Assim que a informação solicitada for obtida, ela deve ser avaliada. Deve-se conferir toda a documentação necessária para a viagem e o gerente de atendimento ao cliente deverá decidir se autoriza a reserva.

Validate Reservation process



Se faltar algum documento, deve-se pedir o que for necessário.

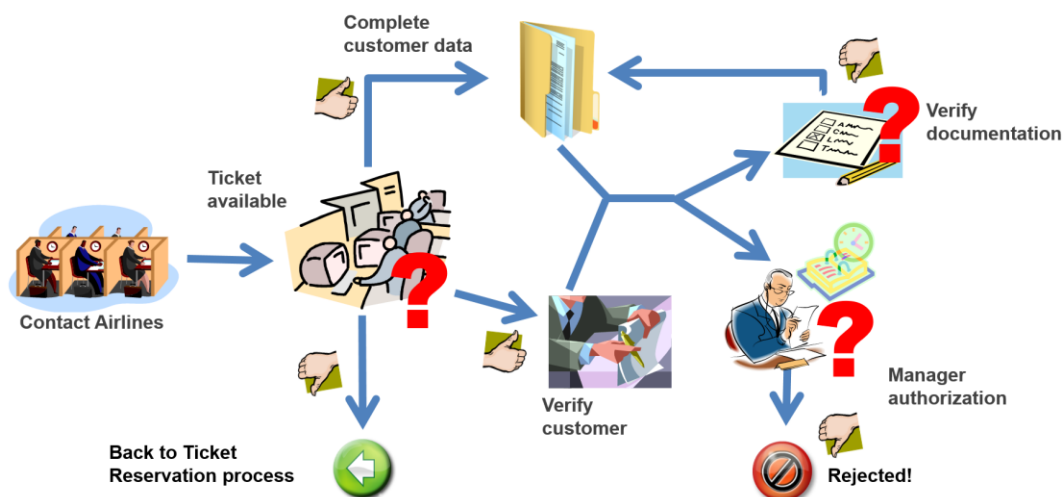
Validate Reservation process



Para estudar a autorização da reserva, o gerente de atendimento ao cliente terá um tempo determinado e receberá avisos periódicos que o lembrem desta tarefa.

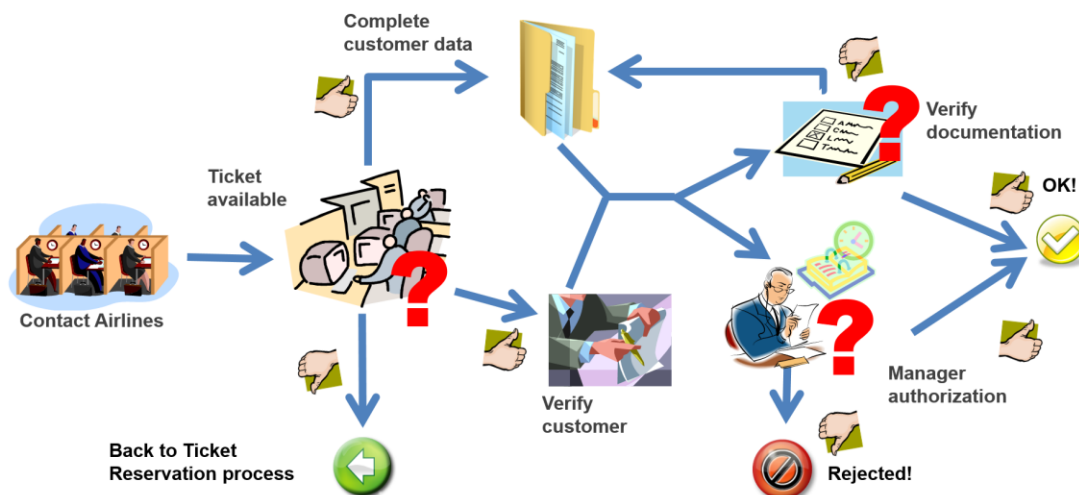
Se ao analisar o reserva, o gerente decide não a autorizar, deverá comunicar os motivos ao cliente, finalizar o processo de validação e também o processo de reserva de passagens.

Validate Reservation process

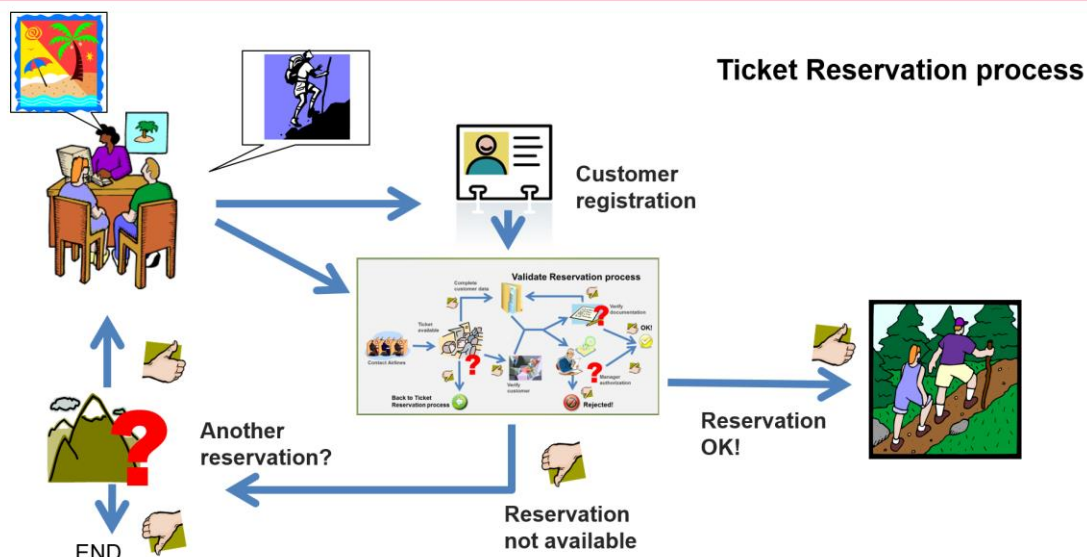


Se o gerente autorizar e se a documentação estiver em ordem, a reserva é autorizada.

Validate Reservation process



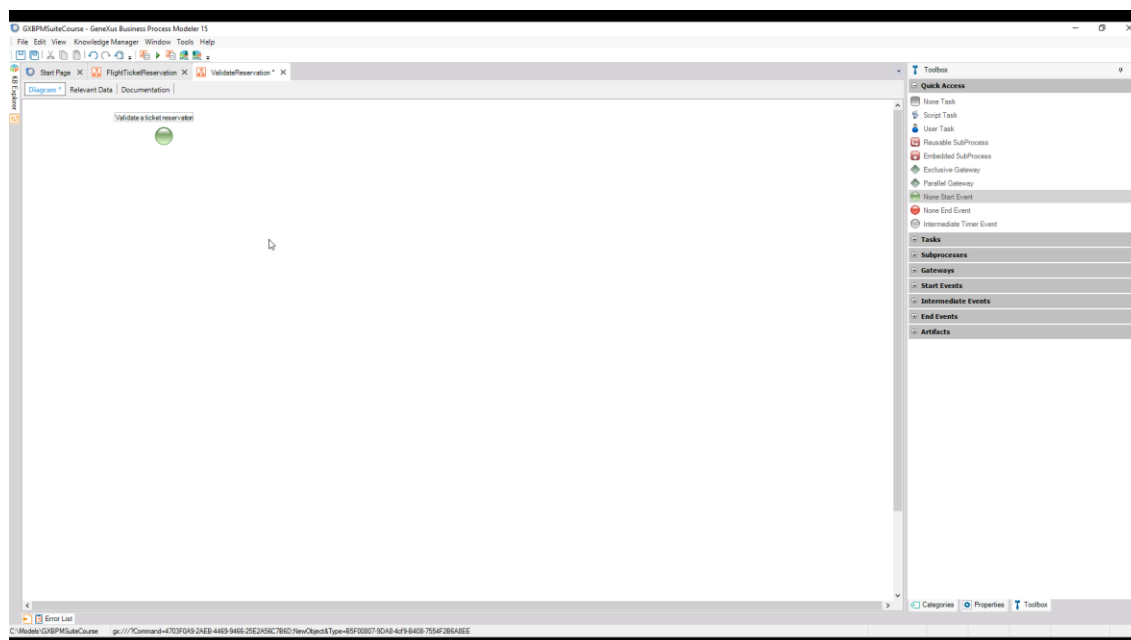
Neste caso, o subprocesso de validação da reserva é finalizado e é preciso voltar ao processo principal de reserva de passagens para notificar ao cliente que a reserva foi confirmada.



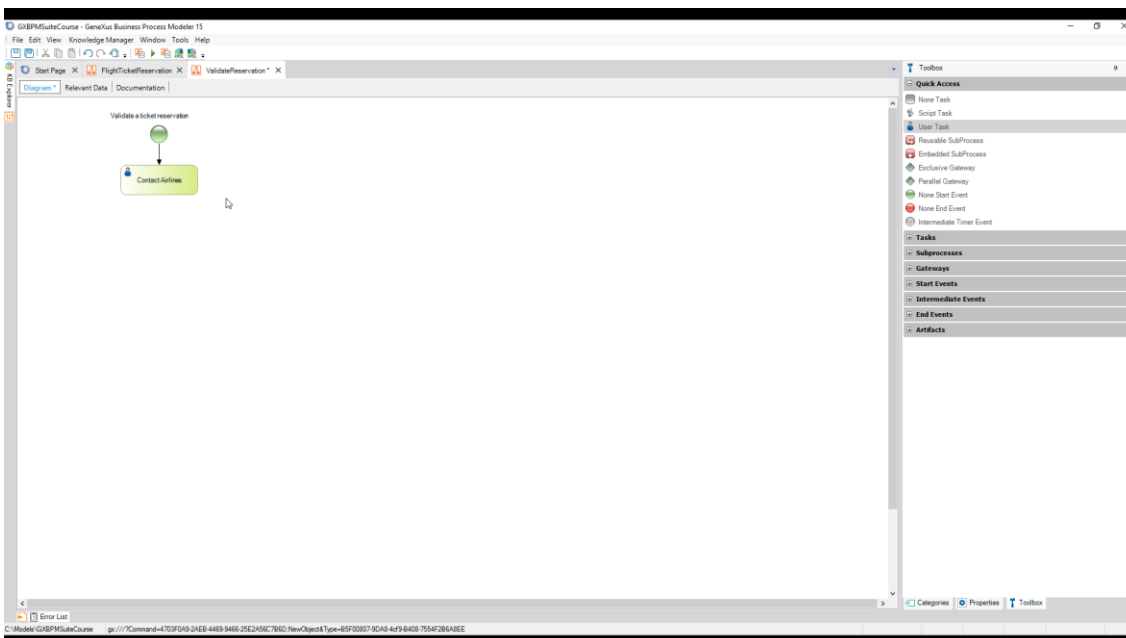
Modelaremos agora o processo de validação da reserva passo a passo utilizando o Business Process Modeler de GeneXus.

Primeiro, criamos um objeto do tipo Business Process Diagram e lhe damos o nome ValidateReservation.

Para indicar o início do nosso processo, arrastamos um símbolo de None Start Event da barra de ferramentas e adicionamos a descrição “Validate a ticket reservation”.



Agora, adicionamos a primeira tarefa que deve ser cumprida: entrar em contato com as companhias aéreas. Como é uma tarefa interativa, arrastamos uma tarefa User para o diagrama, com F2 lhe damos nome Contact Airlines e o unimos partindo do nó de começo.



Essa tarefa tem a particularidade de ser executada certo número de vezes, já que é necessário entrar em contato com várias companhias aéreas, o que inclusive pode ser feito simultaneamente.

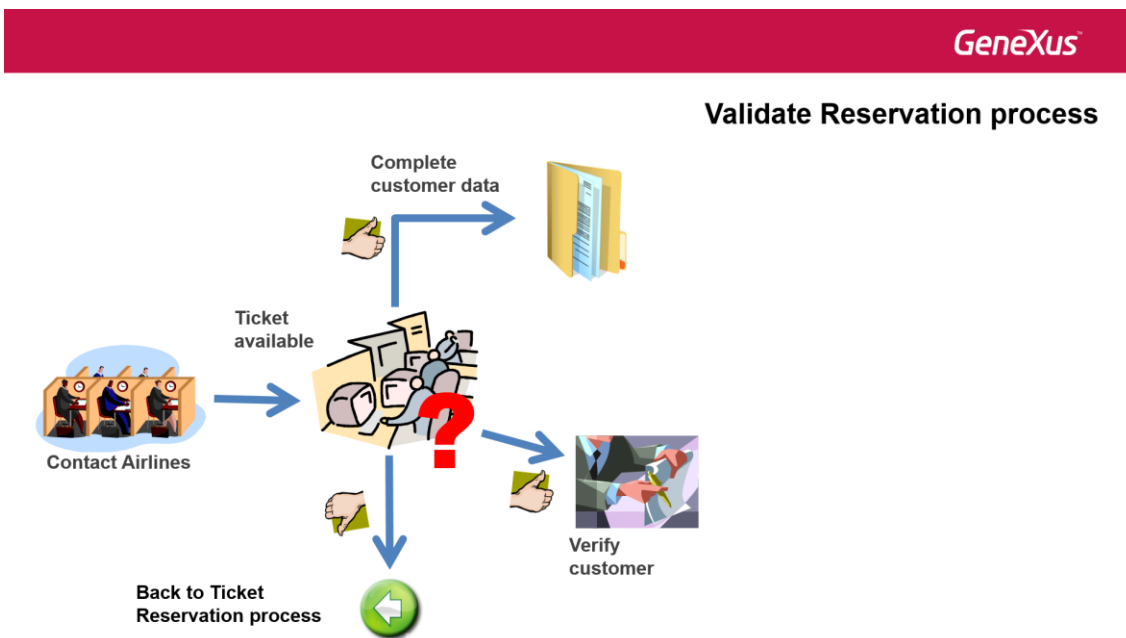
Para definir isso, vamos à propriedade **Loop Type** e lhe atribuímos o valor **Multi-Instance**.



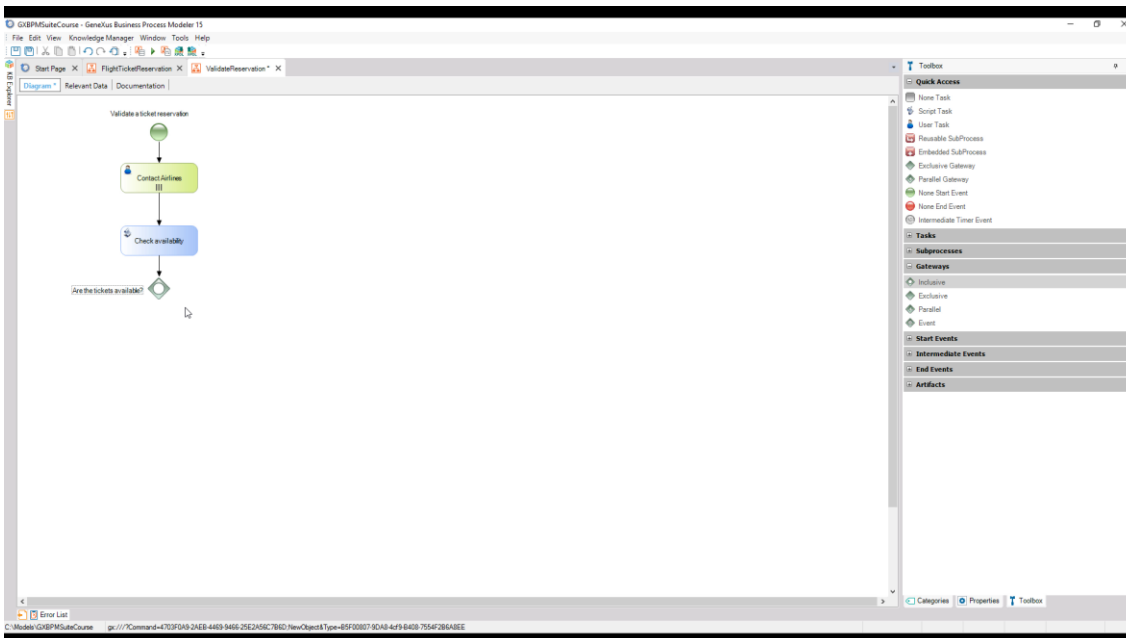
Após o contato com todas as companhias aéreas, devemos analisar a informação obtida e checar se há voos disponíveis nas datas solicitadas. Para modelar isso, inserimos uma tarefa Script com o nome Check availability e a conectamos partindo da tarefa Contact Airlines.



Agora, devemos avaliar o resultado da checagem. Se não houver disponibilidade de passagens, devemos voltar ao processo de reservas para perguntar ao cliente se ele aceita fazer outra reserva. No entanto, havendo disponibilidade, devemos continuar por 2 caminhos simultâneos: completar os dados para viagem e verificar a situação financeira do cliente.



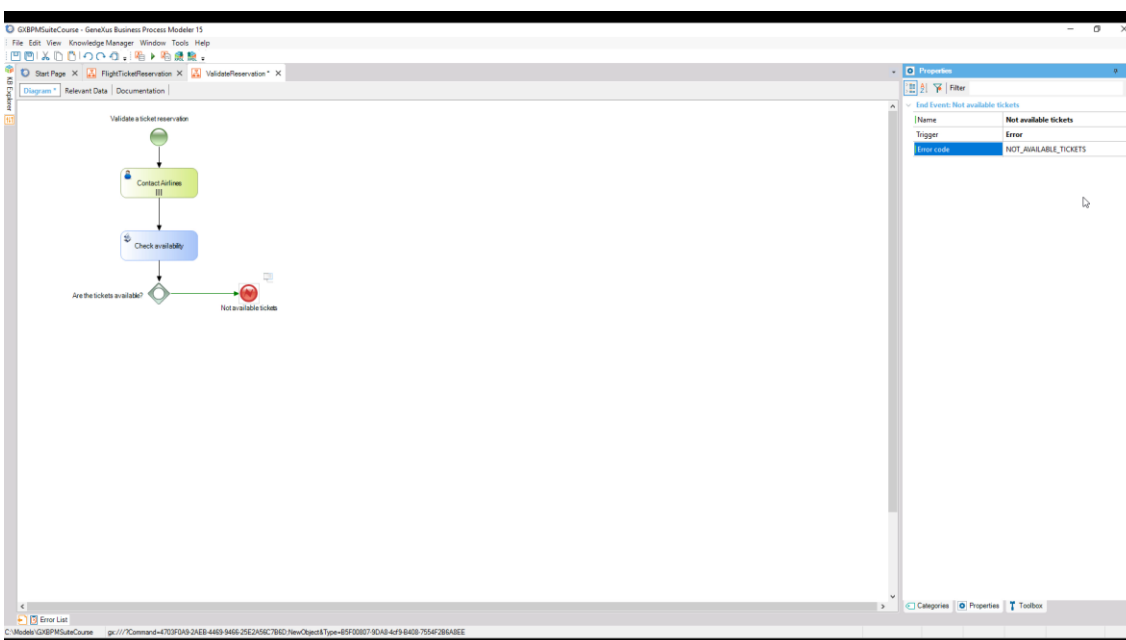
Para modelar isso, não podemos utilizar um exclusive gateway, já que o fluxo deverá continuar por mais de um caminho. Por isto arrastamos da Toolbox um símbolo de **Inclusive Gateway**, o conectamos partindo da tarefa Check Availability e adicionamos a descrição “Are the tickets available?”.



Neste tipo de nó, é definida uma condição para cada caminho que sai dele e o fluxo seguirá por todos os caminhos que cumprirem as condições.

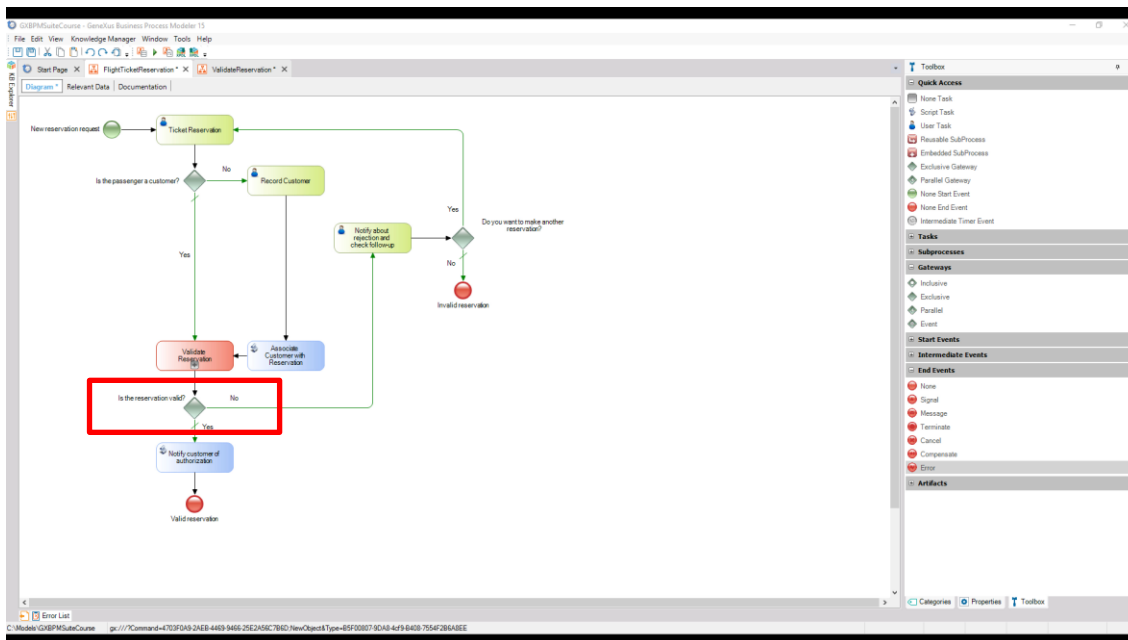
Um caminho possível é quando não há disponibilidade de passagens. Se isso acontecer, devemos indicar que o processo de validação de passagens deve ser cancelado, já que não há passagem para validar e devemos voltar ao processo para notificar isso ao cliente e oferecer a possibilidade de solicitar novas datas para a reserva.

Podemos modelar essa situação de “finalização por erro” inserindo um Error End Event que unimos partindo do inclusive Gateway e adicionamos a descrição “Not available tickets”. Na propriedade Error Code, escrevemos “NOT_AVAILABLE_TICKETS”.

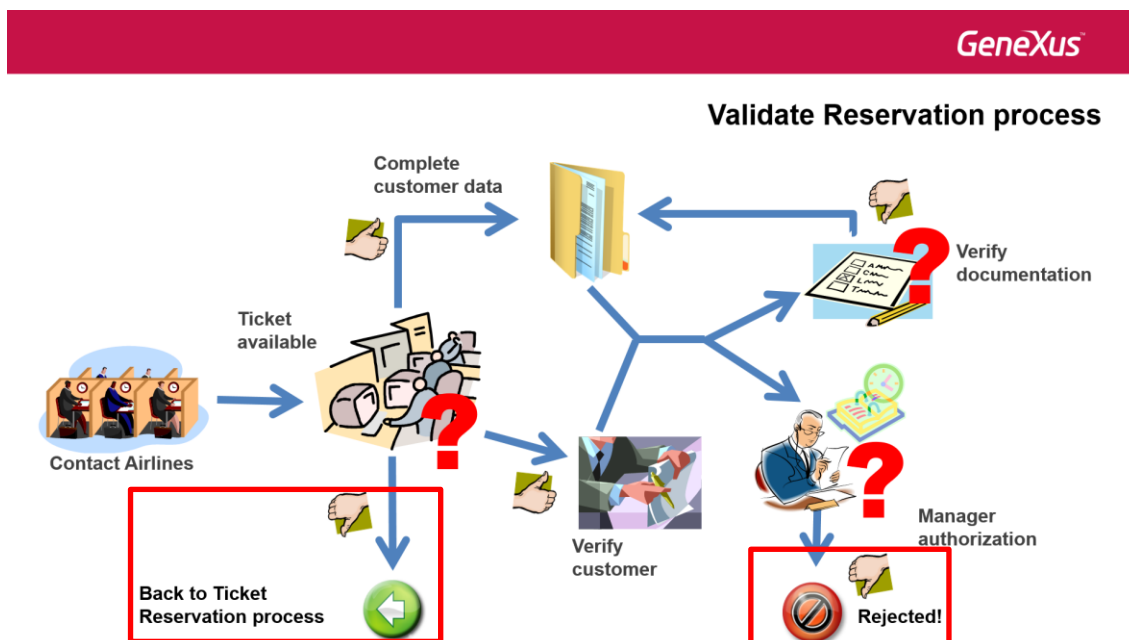


Este tipo de evento de fim dispara um erro que permite que o fluxo continue em outro processo, ao qual adicionaremos um evento intermediário de erro para detectar esse erro.

Se voltarmos ao processo principal de reserva de passagens, vemos que nele era preciso utilizar um exclusive gateway para avaliar se a reserva era válida.



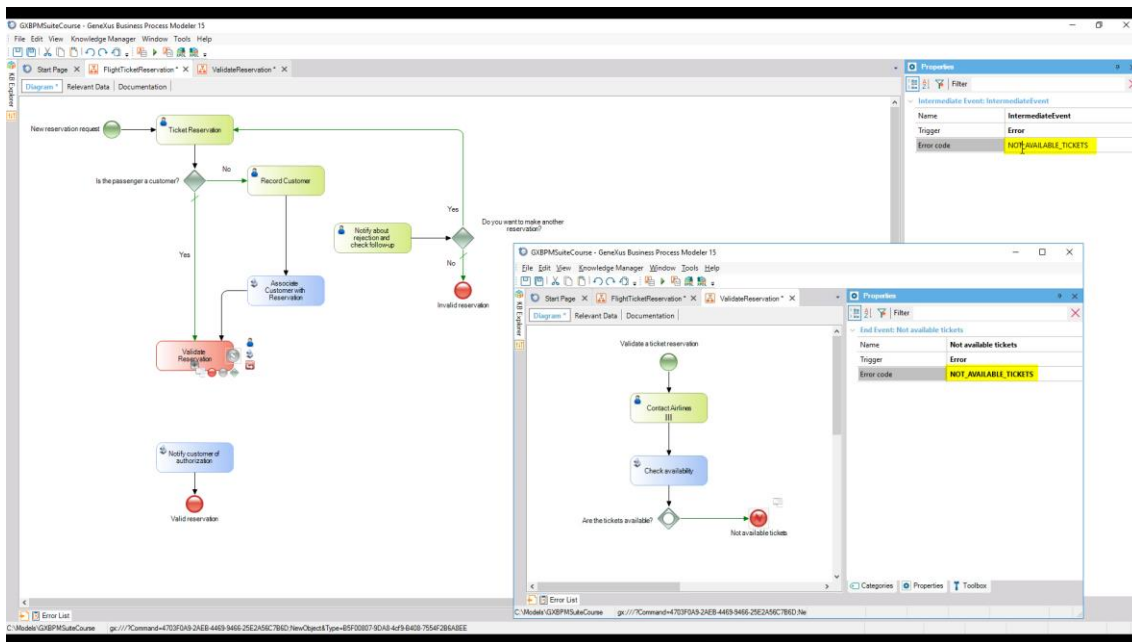
No entanto, este modelo não nos permite discernir por que razão a reserva não é válida, já que, como vimos, a reserva poderia não se confirmar porque não havia passagens disponíveis ou porque o gerente de atendimento ao cliente não autorizou a reserva.



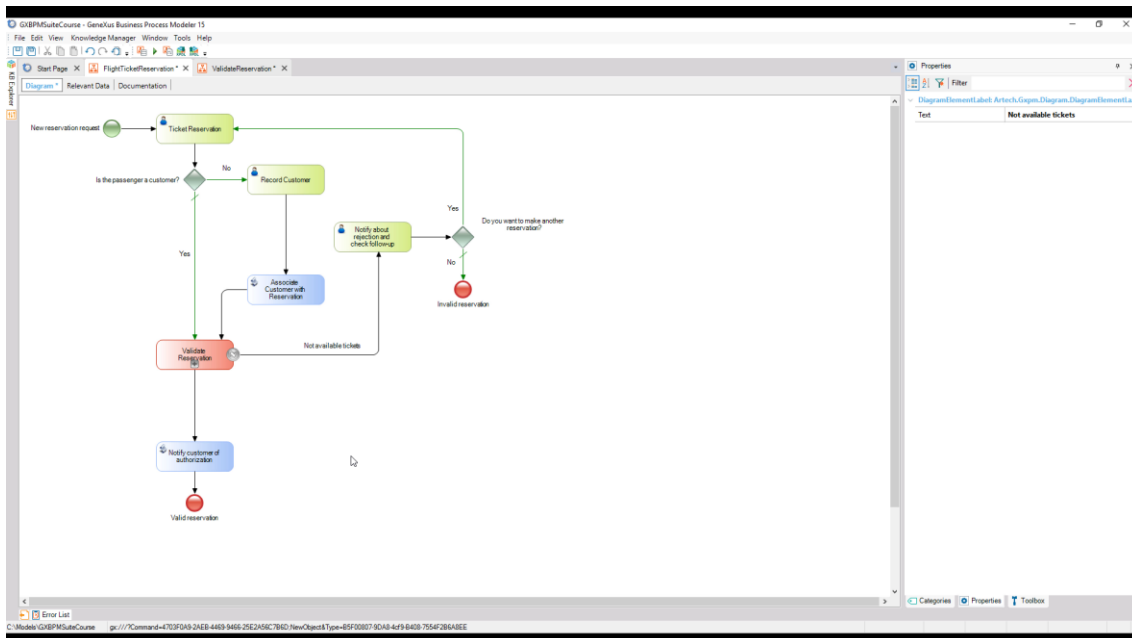
O modelo utilizando um “error end event” e um “error intermediate event” nos permite detectar que situação provocou o erro no subprocesso e adotar ações diferentes no processo principal.

Para modelar isso, modificamos o processo FlightTicketReservation, excluindo o Gateway que avaliava a validade da reserva e o substituímos por um “error intermediate event” que colocamos sobre o símbolo do subprocesso. Editamos as propriedades deste evento e no Error Code, escrevemos

“NOT_AVAILABLE_TICKETS”. Esse código de erro deve coincidir exatamente com o definido no “error end event” do subprocesso ValidateReservation.



Conectamos este evento com a tarefa “Notify about rejection and check follow up”, adicionamos a essa conexão e a descrição “Not available tickets” e completamos o diagrama conectando o subprocesso ValidateReservation à tarefa Notify customer of authorization.



Com isso, modelamos o caso em que não é possível continuar com a reserva porque não há passagens disponíveis.

Para modelar o caso em que o gerente de atendimento ao cliente não autoriza a reserva, adicionamos outro “error intermediate event” ao subprocesso. Depois, arrastamos uma tarefa script da barra de ferramentas,

adicionamos a descrição “Notify rejection to customer” e a conectamos partindo do error event. Nesta conexão, colocamos a descrição “Rejected by Customer Care Manager”.

Por último, adicionamos um None end event com a descrição “Invalid reservation” e o conectamos partindo da tarefa da notificação.

