

# トランザクションサブプロセス

## 1. – 導入

プロセスには、プロセス フローを進めるために多数のアクティビティを調整する必要があり、そのすべてが正常に実行される必要があり、いずれかのアクティビティが満足に実行されない場合には、すべてのアクティビティを初期状態に戻す必要があります。

このようなアクティビティは、いわゆるロジックワークユニット、つまり分割せずに実行する必要があるアトミックユニットで構成されます。ビジネス プロセス内の論理作業単位は、場合によっては数日、場合によっては数週間続くことがあります。

これらのビジネスシナリオをトランザクション処理としてモデル化するために、すべてのトランザクション処理が、すべて適切に実行および完了する必要があるいくつかの操作で構成されており、そうでない場合の操作も完了しない場合、トランザクションサブプロセスに頼ります。

BPMN では、それらを表すために二重線の境界線を使用します。GeneXus では次のようになります。



データベース内で変更されたデータが一貫したステータスになった場合、トランザクションサブプロセスは正常に終了します。トランザクション処理の実行中に障害が発生した場合、変更を元に戻しデータを変更前のステータスに戻すアクションを開始する必要があります。

ビジネスプロセスモデルでは、トランザクション処理は次の 3 つの結果をもたらす可能性があります。

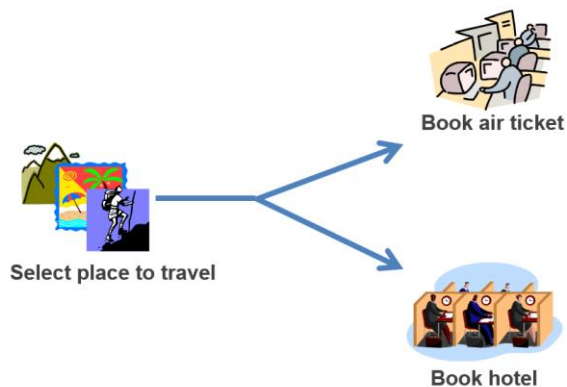
- すべての操作が正常に終了すること。つまり、プロセスが通常のフローで続行されることを意味します。
- 障害が発生し、サブプロセス内ですでに完了したタスクを元に戻さなければなりません。その場合、プロセスをトランザクションサブプロセスの開始前のステータスに戻すいくつかの補正タスクを実行する必要があります。
- 予期しないエラーが発生した場合、サブプロセス内のアクティビティは中断され、補正アクティビティは実行されません。この場合、プロセスは中間エラー イベントの実行を続行します。

したがって、サブプロセスをモデル化するにはこれらの状況に応じたイベント、つまりエラーおよび障害によるプロセスキャンセルイベントをキャプチャする必要があります。

ここで、これまで述べた概念を説明する例を見てみましょう。

## 2.- ユースケース: 旅行のコーディネート

私たちが協力している旅行代理店では、全体を適切に完了するには一連のアクティビティを正常に完了する必要があるため、旅行を調整するタスクがトランザクションサブプロセスの一例となります。そのようなアクティビティには、航空券の予約、ホテルの予約、レンタカー、観光名所のチケットなどが含まれます。



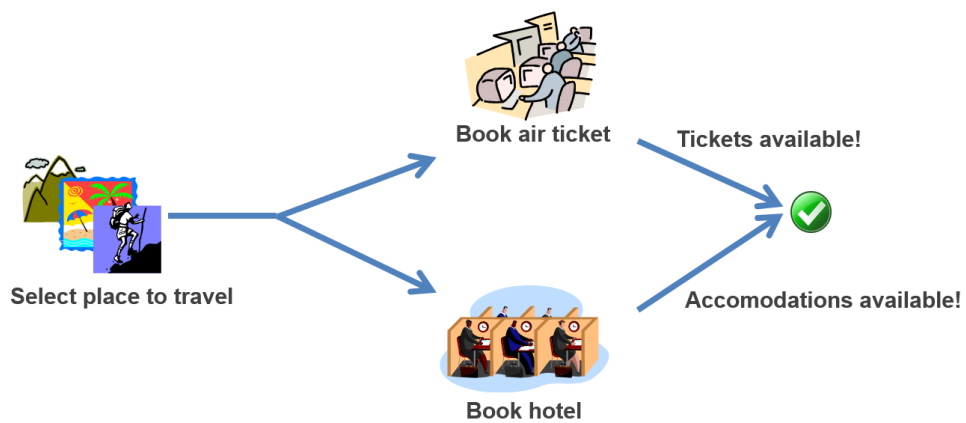
まず、航空券の予約とホテルの部屋の予約という 2 つのアクティビティで構成される予約プロセスについて考えます。

両方のアクティビティを 1 つのトランザクション処理で実行する場合、これはホテルを予約したときに航空券を取得できなかった場合はホテルの予約を取り消す必要があり、逆にホテルが見つからない場合は航空券の予約を取り消す必要があることを意味します。

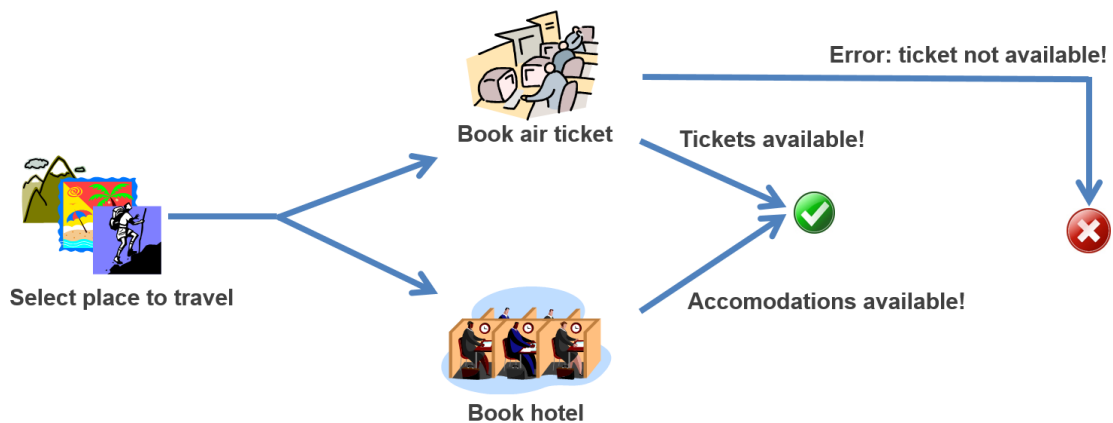
これは、前述の 3 つのケースを考慮する必要があることを意味します。

- プロセスの正常な終了を可能にする両方の保留を取得した場合
- 予約の 1 つが失敗してプロセスがキャンセルされた場合、取得したもう 1 つの予約は取り消さなければなりません。これは両方を持っているか、まったく持っていないかのどちらかになるためです。
- 予期せぬエラーが発生し、処理が終了した場合

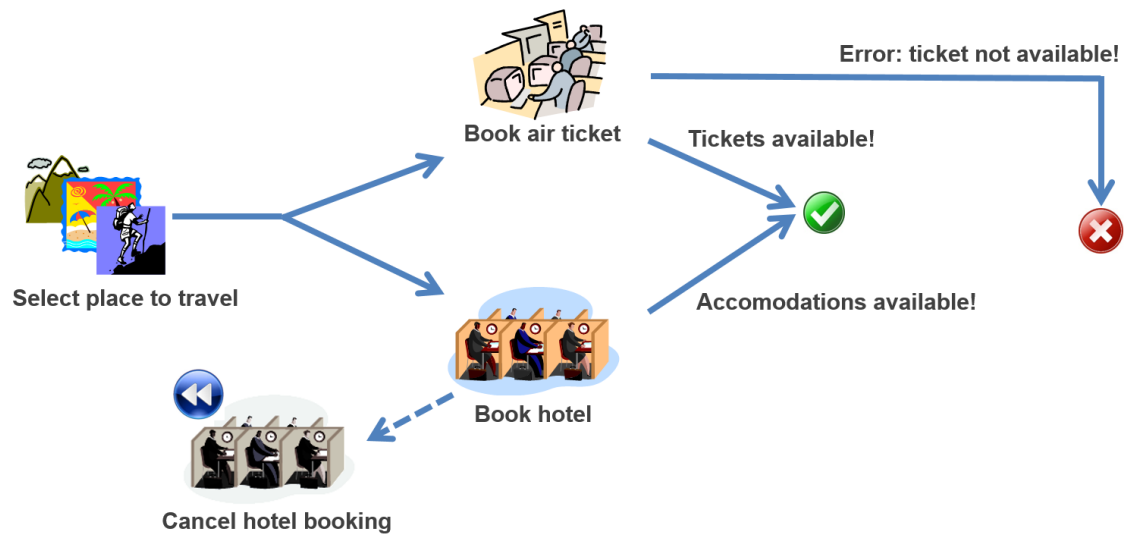
両方の予約を満足に取得した場合、プロセスは完了します:



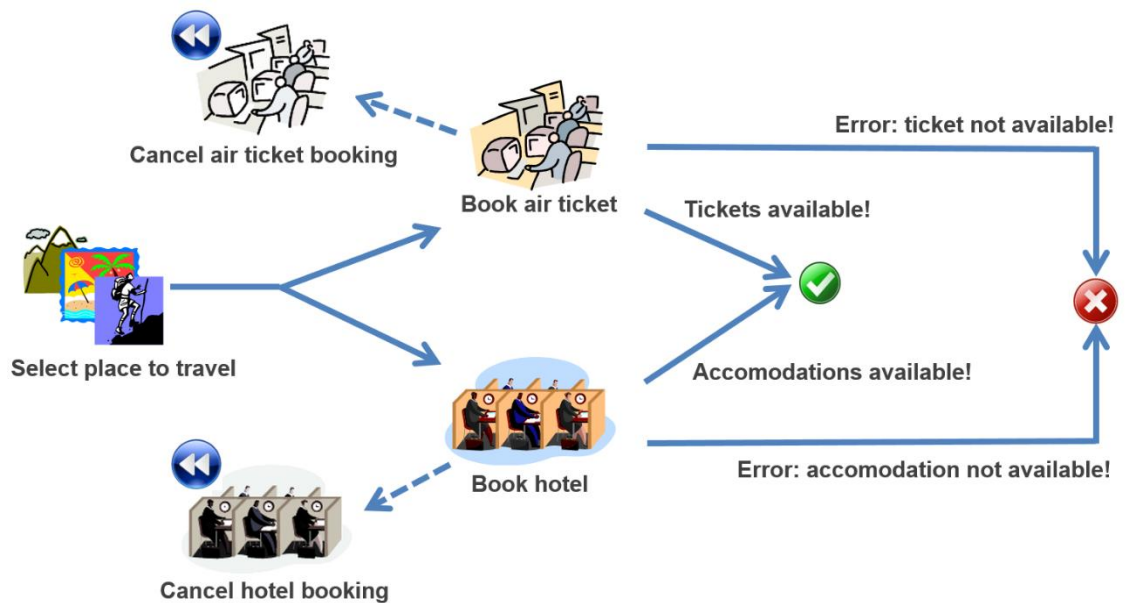
ただし、航空券の予約が失敗し、すでにホテルの予約がある場合はトランザクション処理内の 1 つのアクティビティが完了していないためにプロセスを完了できません。そのため、実際に完了したアクティビティを元に戻す必要があります。



このような場合、旅行調整プロセスがまったく実行されないように、ホテルの予約を取り消し、補償プロセスを実行する必要があります。



別のケースとして、航空券の予約が正常に完了してホテルの宿泊を取得できない場合は、航空券の予約を取り消さなければなりません。



ホテルや航空券の予約のキャンセルを意味するタスクは補償アクティビティとして知られており、多くの場合、外部システムによって実行されます。

これまで、トランザクションプロセス内のすべてのアクティビティが正常に終了したためにトランザクションプロセスが正常に終了するケース、またはトランザクションプロセスのアクティビティの 1 つで障害が発生し、その障害によりプロセスがキャンセルされるケースを見てきました。 プロセスを開始前の状態に戻すために、完了したアクティビティを元に戻す必要があります。

3 番目のケースは、エラーが発生し、サブプロセスで処理できず、処理の進行が妨げられる場合です。たとえば、サーバーが応答しない場合やシステムがダウンした場合などです。このような場合、サブプロセス内のアクティビティは補償なしで中断され、データベースはサブプロセスの開始前の最後のコミットにロールバックされます。

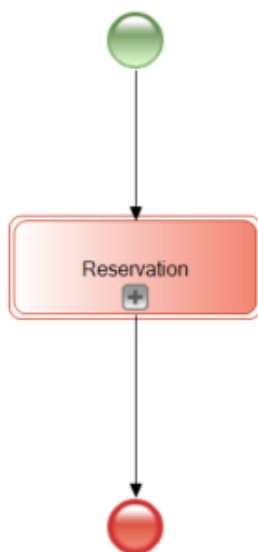
この例では、予約プロセスが正常に完了しなかった場合、その状況が検出され、予約ができないことが顧客に通知されます。

### 3.- GeneXus での実装: 旅行の調整

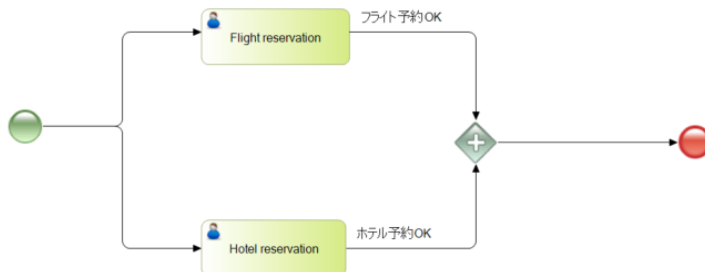
次に、ビジネスプロセスダイアグラムを使用してこのプロセスをモデル化する方法を見てみましょう。

まず GeneXus IDE から、ナレッジ ベースにビジネスプロセスダイアグラムオブジェクトを作成します。

ツールバーから、「開始イベント/トリガーなし」、開始イベントから接続するサブプロセスシンボル、およびサブプロセスから接続する「終了イベント/トリガーなし」をドラッグします。サブプロセスのプロパティで、「Is transaction」プロパティに「True」を設定します。これにより、サブプロセス シンボルが変更され、二重線の境界線が表示されます。これは、BPMN でトランザクションサブプロセスを識別する方法です。

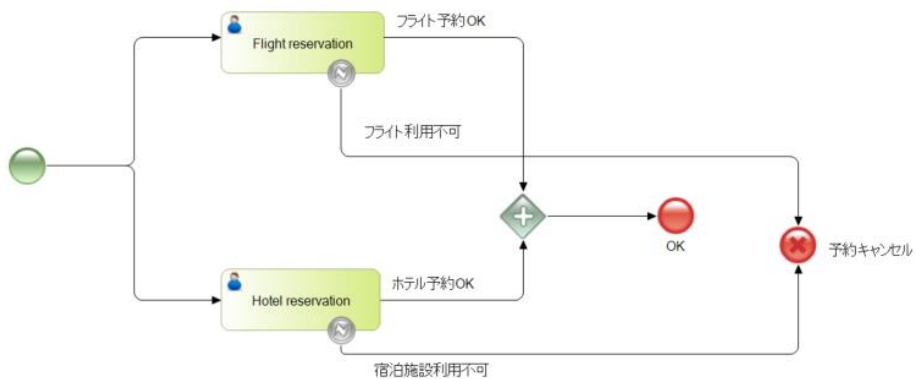


次に、サブプロセスを開き、「開始イベント/トリガーなし」を追加してから、2 つのユーザータスク（1 つはフライトの予約用、もう 1 つはホテルの予約用）を挿入し、開始イベントからそれらを接続します。並列ゲートウェイで 2 つの出力を結合し、ゲートウェイの出力を「終了イベント/トリガーなし」に接続します。



これは、両方のタスクが正しく終了するケースをモデル化しています。この場合、並列ゲートウェイは両方のタスクの終了を同期でき、同期した結果、終了イベントに進みます。ただし、2 つのタスクのいずれかが失敗すると、並列ゲートウェイによりフローの進行が許可されなくなるため、他のケースも考慮する必要があります。

予約の失敗の可能性をすべて考慮するために、各タスクに中間エラーイベントを追加し、その出力を終了キャンセルイベントに接続します。



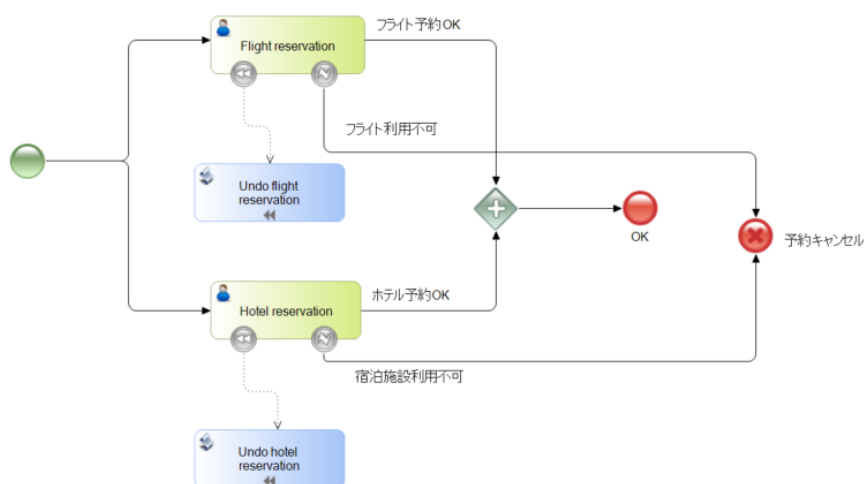
このキャンセルイベントは、ユーザータスクのいずれかが失敗するとトリガーされます。キャンセルイベントは、サブプロセスの終了時ではなくサブプロセスの実行中にトリガーされる、トランザクションプロセスで使用される特別なイベントです。

トランザクション処理がキャンセルされ、親プロセスに制御が渡される前に、補正タスクがトリガーされます。これは、その時点までに正常に終了したすべてのアクティビティを、それぞれに定義された補正タスクを実行することによって元に戻す必要があることを意味します。

この場合、ホテルや航空券の予約を取り消すタスクを定義する必要があります。

そのためにフライト予約タスクに中間キャンセルイベントを追加し、「Undo flight reservation」という名前のスクリプトタスクを挿入し、中間キャンセルイベントから接続します。

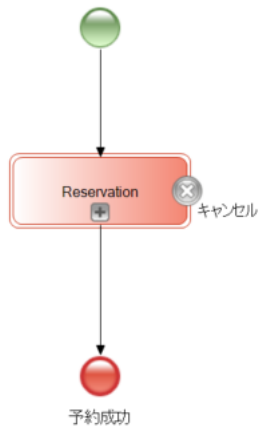
ホテル予約タスクに対しても同じことを行い、別の中間キャンセルイベントと、「Undo hotel reservation」という名前でスクリプトタスクを挿入し、中間キャンセルイベントから接続します。



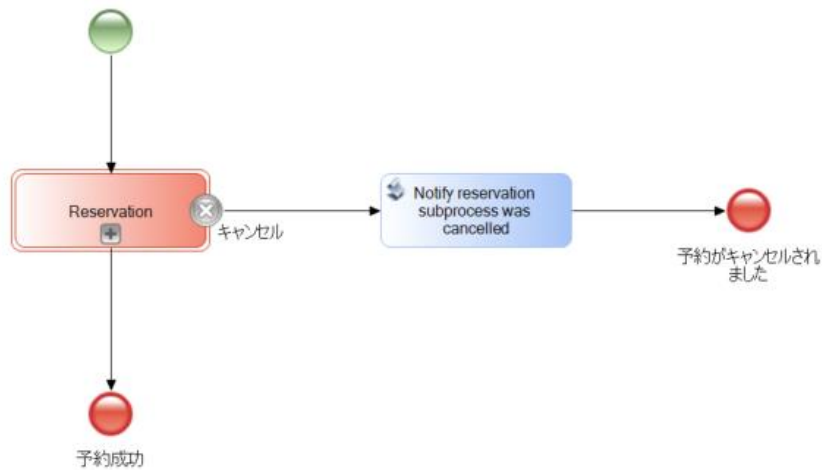
スクリプトタスクを中間キャンセルイベントに接続した時点から、接続の外観が変化していることがわかります。これは、コネクタが関連付けタイプになり、タスク上に「<<」記号が表示されるためです。点線で表されるこのタイプのコネクタを使用すると、関連するアクティビティをフロー シーケンスの一部にすることができないため、シーケンス コネクタとは異なる関係を確立できます。これは、入力または出力シーケンス コネクタを持つことができないことを意味します。

トランザクション補正が完了すると、サブプロセスの実行が中断され、サブプロセスから親プロセスに対してキャンセル イベントがトリガーされます。

親プロセスからこのイベントを取得するには、サブプロセスのシンボルに中間キャンセルイベントをアタッチする必要があります。



親プロセスがキャンセルを受け取ると、ユーザーに予約プロセスのキャンセルを通知し、予約プロセスを終了します。



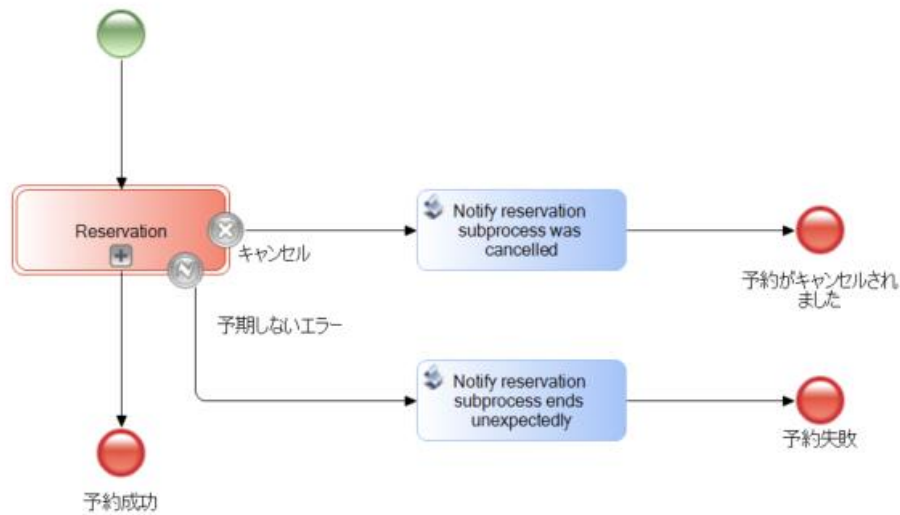
エラーイベントの場合で前に見たように、キャンセルイベントは、サブプロセスの 終了キャンセルイベントから、親ダイアグラム内のサブプロセスにアタッチされた 中間キャンセルイベントに至るスローキヤッチメカニズムを使用します。

このキャンセルイベントは、サブプロセスで定義された補正タスクを自動的にトリガーし、トランザクション処理の整合性を保証します。

エラーイベントなど、サブプロセスで発生する他のタイプの中断は、補償なしで実行が中止されることを意味します。



このエラーの管理形式を定義するために、サブプロセスシンボルに付加された中間エラーイベントを挿入し、スクリプトタスクを通じて予約プロセスが予期しない方法で終了したことを通知する通知を送信します。



以上、トランザクションサブプロセスがどのように機能するか、そして GeneXus でそれらをモデル化する方法について見てきました。

詳細については、次のリンクをご覧ください。:

<http://wiki.genexus.com/commwiki/servlet/hwikibypageid?24922>