

時間パターンのモデリング

ビジネスプロセスをモデル化する場合、プロセスに含まれるアクティビティの待機期間、遅延、有効期限や締め切りなど、時間に関連する状況を表現しなければならない場合があります。

これらの状況をモデル化するには、タイマータイプのイベントが使用されます。また、状況をモデル化するために使用するパターンのタイプは、表現する状況のタイプによって異なります。例を挙げてさまざまなケースを見てみましょう。

旅行代理店のプロセスモデリングを使用して、旅行代理店が特定の観光名所を宣伝する方法としてチャーター便の販売プロセスを表すように要求したとします。

プロセス全体の期間は最大 3 か月である必要があり、この期間が経過するとプロセスは完全に終了する必要があります。

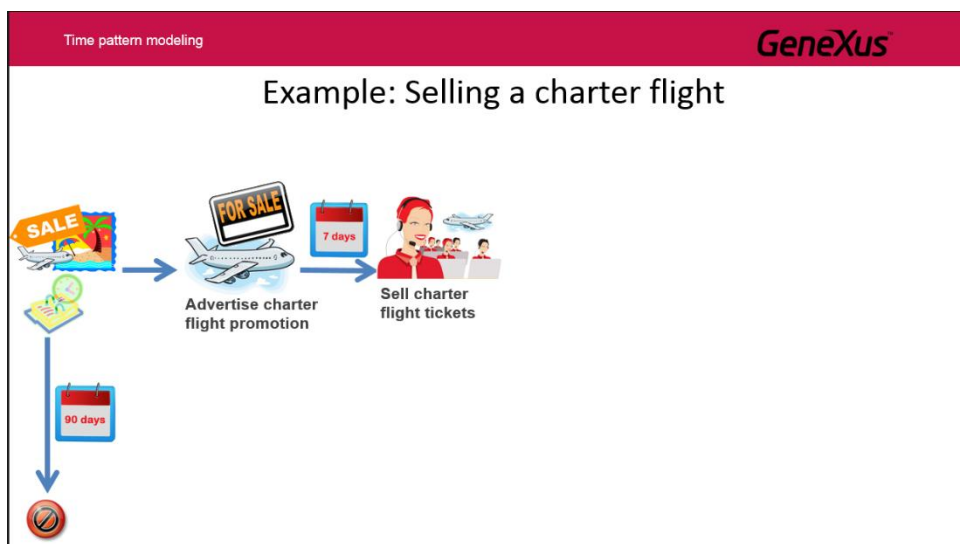


Example: Selling a charter flight



チャーター便の販売プロセスは、プロモーションが複数のメディア、ソーシャル ネットワーク、Web ページに掲載されたときに始まります。目的は、フライトで利用可能なすべての座席を販売して、利益を最大化することです。

プロモーションの開始から 1 週間後にチケットが販売可能になります。

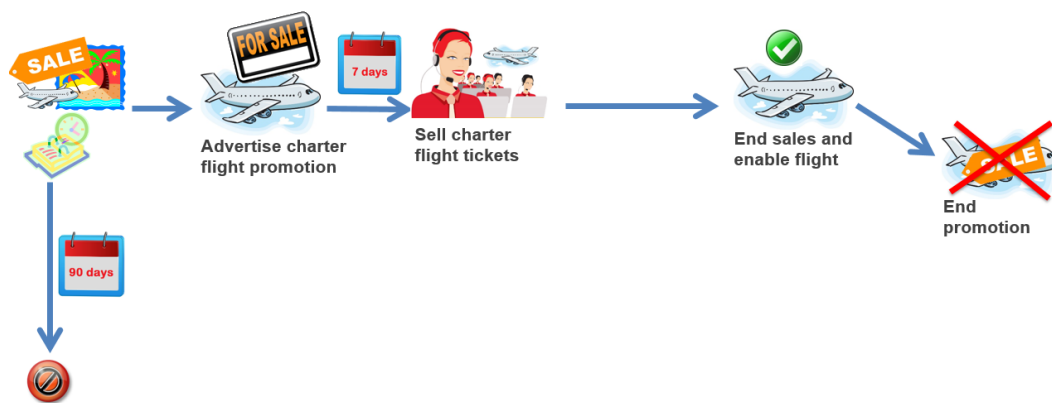


すべての航空券が販売されると、フライトは有効になり（これは、このフライトが実際に運航することを意味します）、宣伝が掲載されたさまざまなメディアから削除されます。

Time pattern modeling

GeneXus™

Example: Selling a charter flight



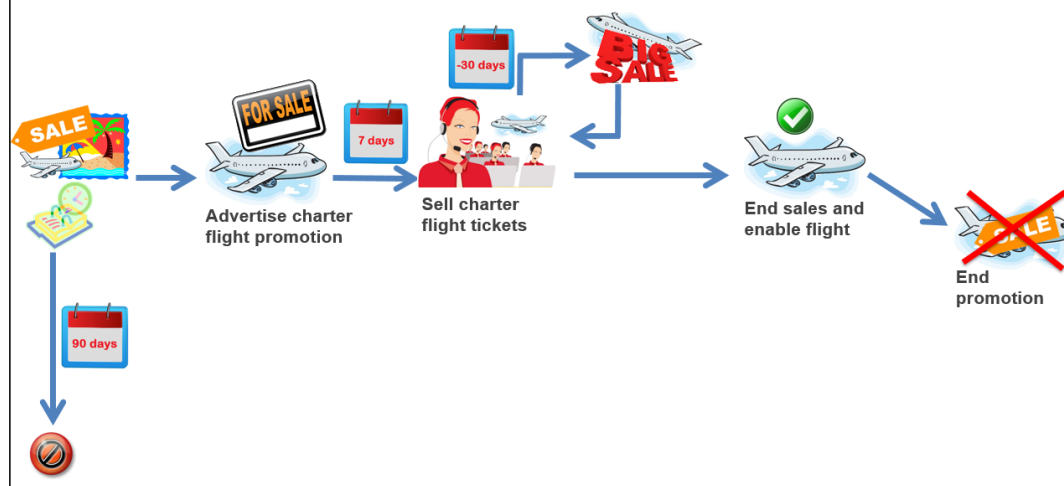
航空券が販売されると、時間の経過とともに特定のアクションが実行されます。

たとえば、販売プロセスが 30 日以内に終了し、すべての座席がまだ販売されていない場合は、さまざまな販売チャネルでプロモーション広告を増やすためのサブプロセスを開始するイベントがトリガーされます。

Time pattern modeling

GeneXus™

Example: Selling a charter flight

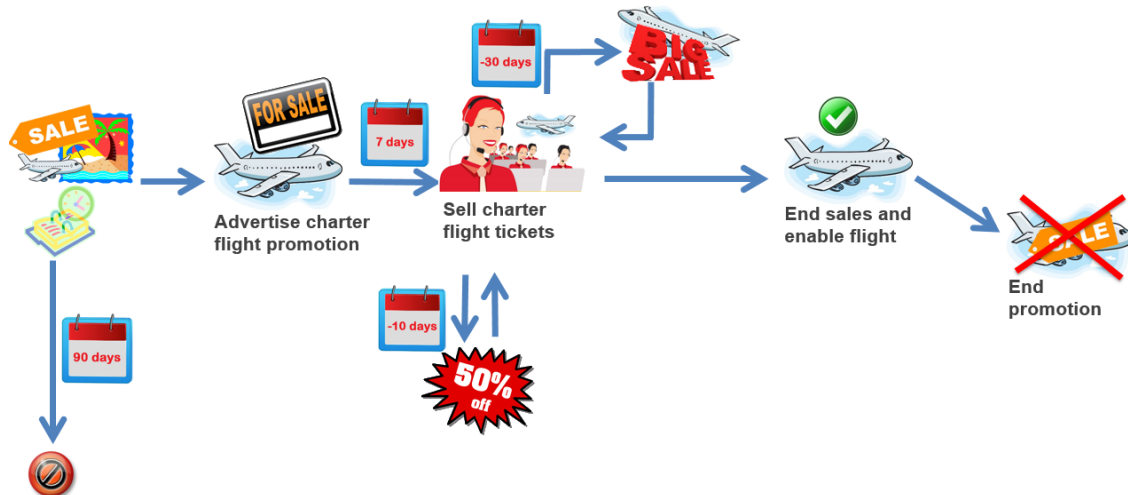


プロセスの終了まで 10 日が残っており、フライトにまだ座席がある場合、イベントがトリガーされます。 このイベントは販売活動を中断するものではありませんが、できるだけ多くのチケットを販売することが目的であるため、より多くの乗客を引き付けるために宣伝価格を下げる活動が開始されます。

Time pattern modeling

GeneXus™

Example: Selling a charter flight

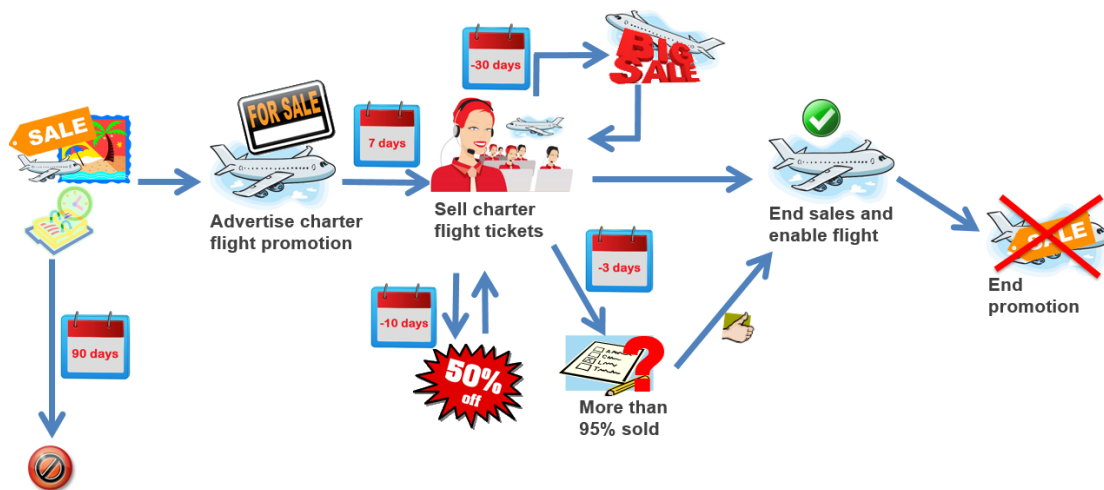


3 か月のプロモーション期間が終了するまで残り 3 日になると、販売タスクが中断され、利用可能な座席の 95%以上が販売されたかどうかチェックされます。この割合に達すると、チケットの販売が停止され、フライトが有効になります。また、宣伝されたプロモーションは削除され、プロセスは終了します。

Time pattern modeling

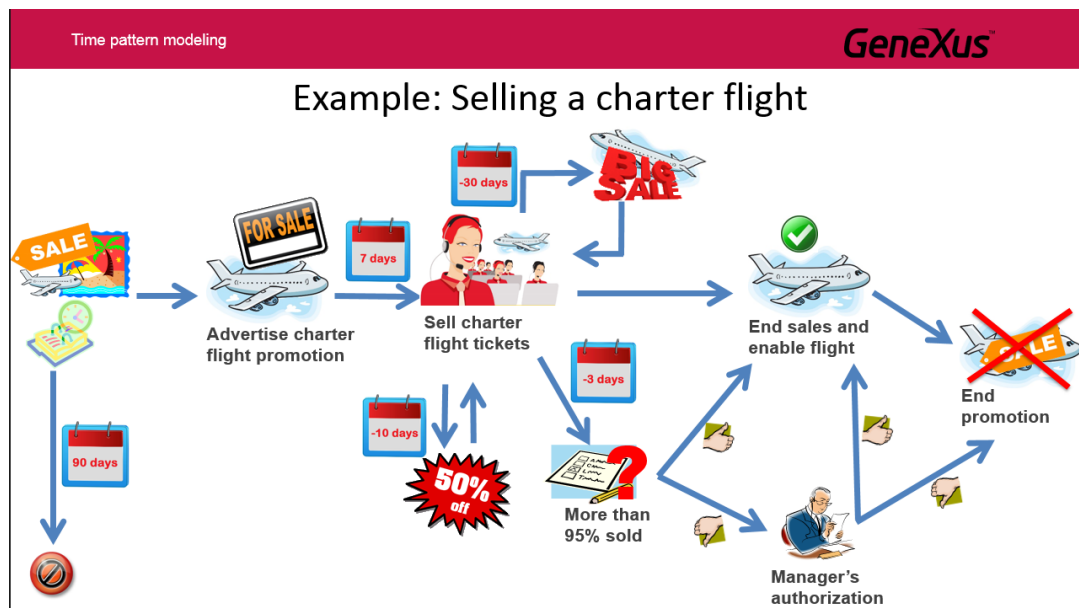
GeneXus™

Example: Selling a charter flight



航空券の販売率が 95%未満の場合は、航空券のほとんどを販売するという目的を達成できないため、フライトが出発するかキャンセルされるかを確認するために旅行代理店の管理者の承認が求められます。

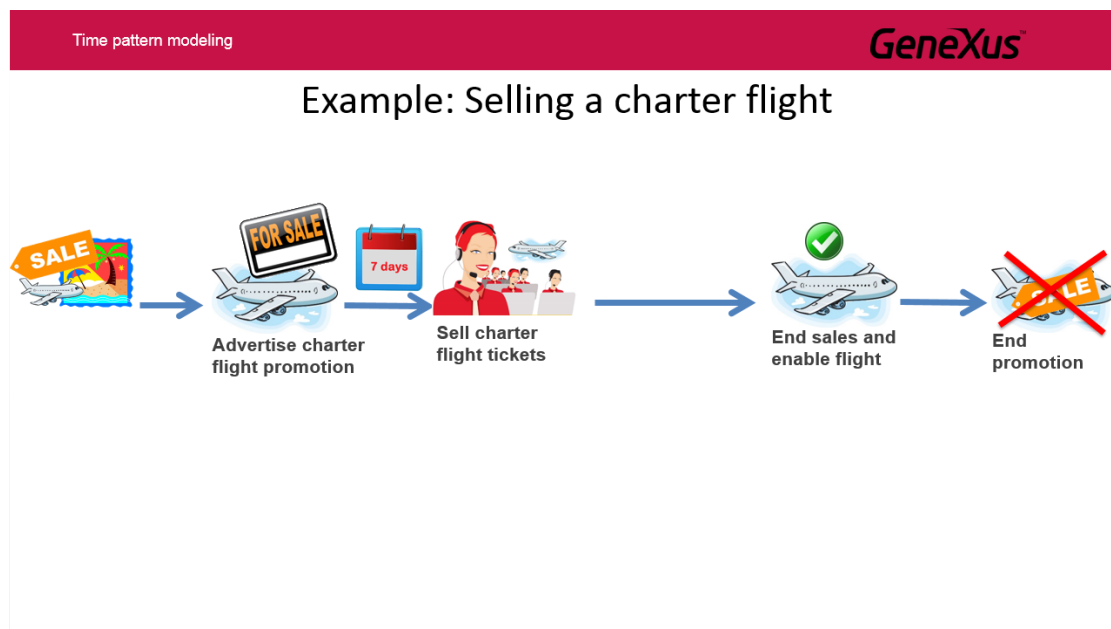
フライトが承認されている場合は有効になります。さらに、プロモーションが削除され、プロセスが終了します。旅行代理店の管理者がフライトを承認しない場合、プロモーションは削除され、プロセスは終了します。



これらの時間関連の状況をそれぞれモデル化する方法を見てみましょう。

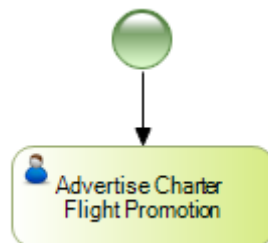
まず、GeneXus で「CharterFlightSale」という名前でビジネスプロセスダイアグラムオブジェクトを作成します。

最初に最も一般的なフローのモデル化に焦点を当てます。つまり、不都合がなければ起こる流れです。この場合、チャーター便のプロモーションの公開、すべてのチケットの販売、フライトの有効化、プロモーションの削除、プロセスの終了が含まれます。

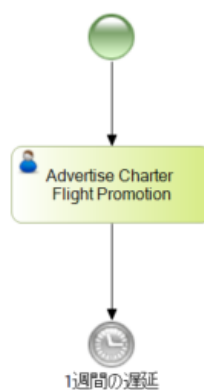


プロセスを開始するには、ツールバーから「開始イベント/トリガーなし」と開始イベントから結合されたユーザータス

クをドラッグします。 このタスクを「Advertise Charter Flight Promotion」とします。



旅行代理店の指示により、プロモーションが掲載されてから航空券の販売を開始するまでに 1 週間待つ必要があります。遅延を実装するには、タイマータイプのイベントを使用するため、「中間イベント/タイマー」をダイアグラムにドラッグし、タスク「Advertise Charter Flight Promotion」から結合します。次に、F2 キーを押して、「1 週間の遅延」という説明を追加します。



「中間イベント/タイマー」のプロパティで、図のように「日」に「7」と入力し、他の値は「0」にします。

▼ 中間イベント: 1週間の遅延

Name	1週間の遅延
Trigger	Timer
Timer definition	Duration
Timer expression type	Rule
Timer duration	P7D

期間を編集

年

月

日

時間

分

秒

OK キャンセル

ここで、ユーザータイプの別のタスクをドラッグし、「Sell charter flight tickets」という名前を付けて、タイマーイベントから接続します。

チャーター便のプロモーションを公開するタスクが完了すると、この方法で使用するタイマーは、航空券の販売タスクを開始する前に 1 週間フローを停止します。



すべてのチケットが販売されたら、フライトを有効にする必要があります。 これを行うには、「End sales and enable flight」という別のユーザータスクをドラッグし、「Sell charter flight tickets」タスクから接続します。



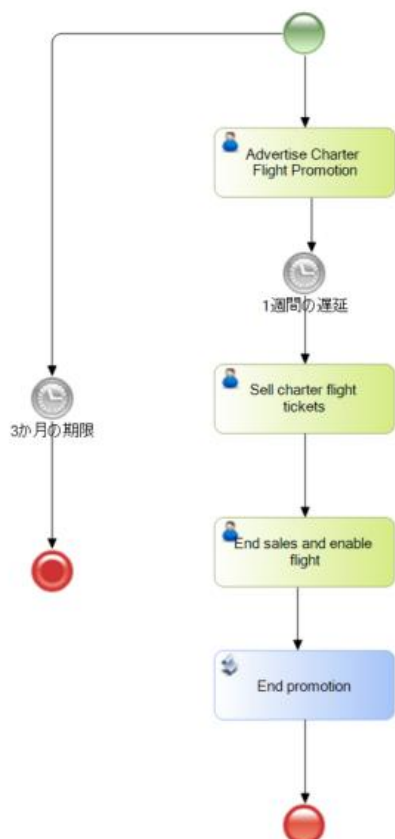
すべてのチャーター航空券が販売され、フライトが運航可能になったら、公開したプロモーションを削除してプロセスを終了する必要があります。 これを行うには、スクリプトタイプのタスクをドラッグし「End promotion」という名前を付けます。前のタスクから接続し、最後に、「終了イベント/トリガーなし」をドラッグしてプロセスを終了します。



これまでモデル化したものは、すべてのチャーター航空券が常に最適なタイミングで販売されることを前提としていますが、代理店が要求したいいくつかの制御と要件を追加する必要があります。

まず、プロセス全体が3 か月を超えないようにする必要があります。この目的のために、「中間イベント/タイマー」をドラッグし、プロセスの最初から（つまり、開始イベント/トリガーなしから）結合し、「3 か月の期限」という説明を付けます。

プロセスがこの期間内に実際に終了すること、つまりすべてのプロセスのフローが終了することを確認するには、ツールバーから終了イベントの終結をドラッグします。



タイマーのプロパティにて、日の時間単位を割り当て、値を 90 に設定します。これは 90 日間続きます。これは、要求された 3 か月の期間に相当します。

▼ 中間イベント: 3か月の期限

Name	3か月の期限
Trigger	Timer
Timer definition	Duration
Timer expression type	Rule
Timer duration	P90D

期間を編集

年: 0

月: 0

日: 90

時間: 0

分: 0

秒: 0

OK キャンセル

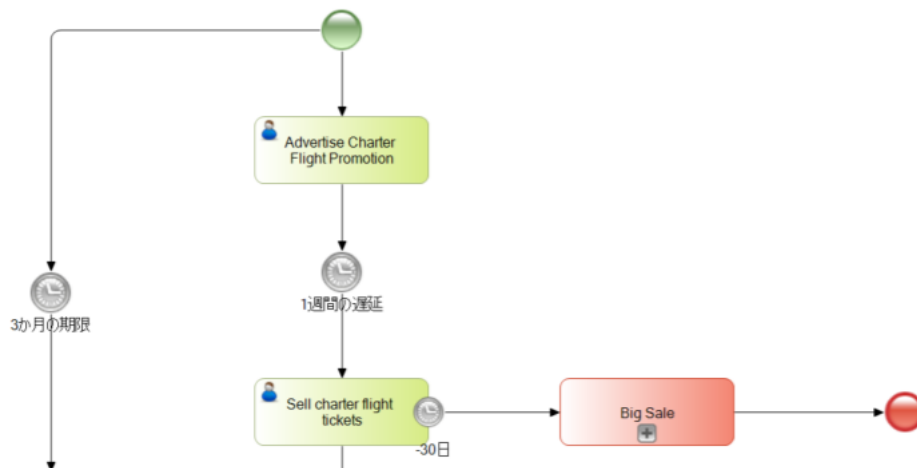
次に、チャーター航空券の販売プロセスに関係するさまざまな通知や期限をモデル化します。

最初のアクションは、販売プロセスに許可されている最大期間に達するまで残り 30 日を切ったときに実行されます。

この時点で、広告と販売の取り組みを強化する必要があります。

これをモデル化するには、タイマータイプの中間イベントをツールバーからドラッグし、タスク「Sell charter flight tickets」にドロップします。「-30 日」という説明を追加します。

次に、「Big Sale」という組み込みサブプロセスを挿入し、タイマーから接続します。また、「終了イベント/トリガーなし」を挿入し、サブプロセスから接続します。

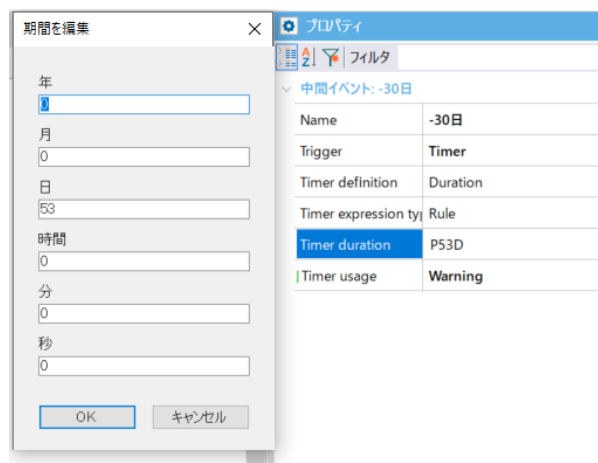


サブプロセスの実行開始時に警告を表示したいため、タスクに配置したタイマーのプロパティで、タイマーの「Timer usage」プロパティを「Warning」に設定します。「Warning」を選択すると、「Interrupts activity」プロパティが消えることに注意してください。理由は、警告を表示したいだけなので意味がないためです。

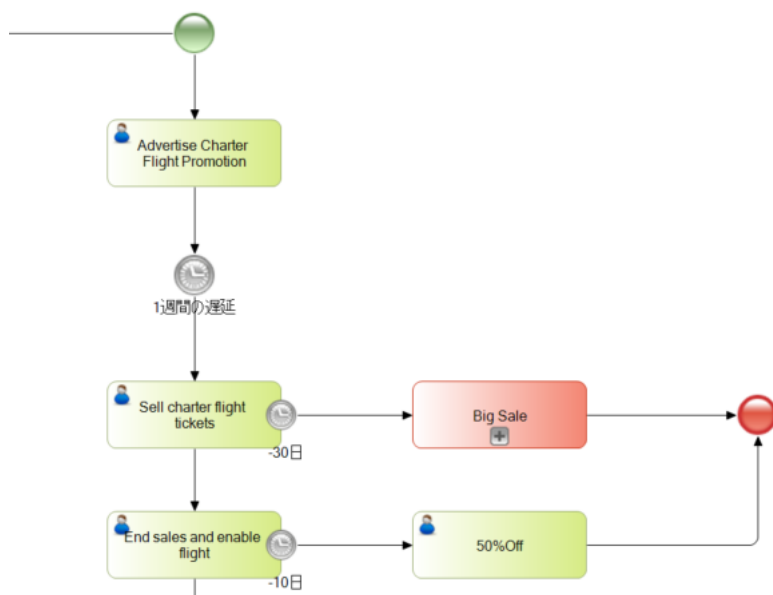
タイマーイベントに設定された時間は、挿入されたイベントを含むタスクが実行されるとカウントされ始めることに注意してください。広告を増やすサブプロセスは、プロセス全体で許可されている最大期間である 90 日に達する 30 日前に開始する必要があるため、いくつかの計算を行う必要があります。

プロセスの実行時に 90 日がカウントされ始めることがわかります。「Advertise Charter Flight Promotion」タスクに時間がかからないと仮定すると、「Sell charter flight tickets」プロセスはその 7 日後に開始されることがわかります。つまり、プロセス期限の 83 日前です。したがって、サブプロセス「Big sale」は、「Sell charter flight tickets」タスクの実行後 $83 - 30 = 53$ 日後に開始する必要があります。

中間イベント/タイマーのプロパティにて Days の値に「53」を設定します。



私たちがモデル化したものは、プロセスの完了まで残り 30 日になったときにタイマーイベントによって警告が発行されることを保証します。この警告は、ワークフロークライアントの受信箱でタスクを実行するときに表示されます。チャーター便の販売終了まであと 10 日となった時点で、すべての航空券を確実に販売するために価格を引き下げたいと考えています。これを行うには、中間イベント/タイマーを再度タスク「Sell charter flight tickets」にドラッグし、「-10 日」という説明を割り当てます。次に、ユーザータスクを挿入し、航空券を初期価格の半額に値下げするプロセスを表す「50%Off」という名前を割り当てます。このタスクの出力をサブプロセスの「終了イベント/トリガーなし」に接続します。



タイマーイベントのプロパティに移動し、「Interrupts activity」プロパティを「False」に設定します。警告を出さず、チケット販売プロセスを中断せずに値下げプロセスを開始するため、「Timer usage」プロパティで値「None」を選択します。

このイベントはプロセス終了の 10 日前に行う必要があります。つまり、前にモデル化した警告から 20 日後です。そのため、日に「73」と設定します。

期間を編集

年: 0
月: 0
日: 73
時間: 0
分: 0
秒: 0

OK キャンセル

プロパティ

フィルター

中間イベント: -10日

Name	-10日
Trigger	Timer
Interrupts activity	False
Timer definition	Duration
Timer expression type	Rule
Timer duration	P73D
Timer usage	None

このようにすることで、プロセス終了まで残り 10 日になると値下げタスクが自動的にトリガーされ、チケット販売タスクが続行されます。

ここで、チャーター航空券の販売期限に達する前に最後のアクションを実行します。プロセス終了の 3 日前にチケットの 95%以上が販売された場合、販売プロセスは終了しチャーター便は自動的にキャンセルされません。

この最後のコントロールでは、別のタイマー イベントをタスク「Sell charter flight tickets」にドラッグし、「-3 日」という名前を付けます。「Timer usage」プロパティではデフォルト値(Deadline)のままにし、「Interrupts activity」プロパティもデフォルト値(True)のままにします。最後の期限から 7 日後にトリガーする必要があるため、値に 80 を割り当てます。

期間を編集

年: 0
月: 0
日: 80
時間: 0
分: 0
秒: 0

OK キャンセル

プロパティ

フィルター

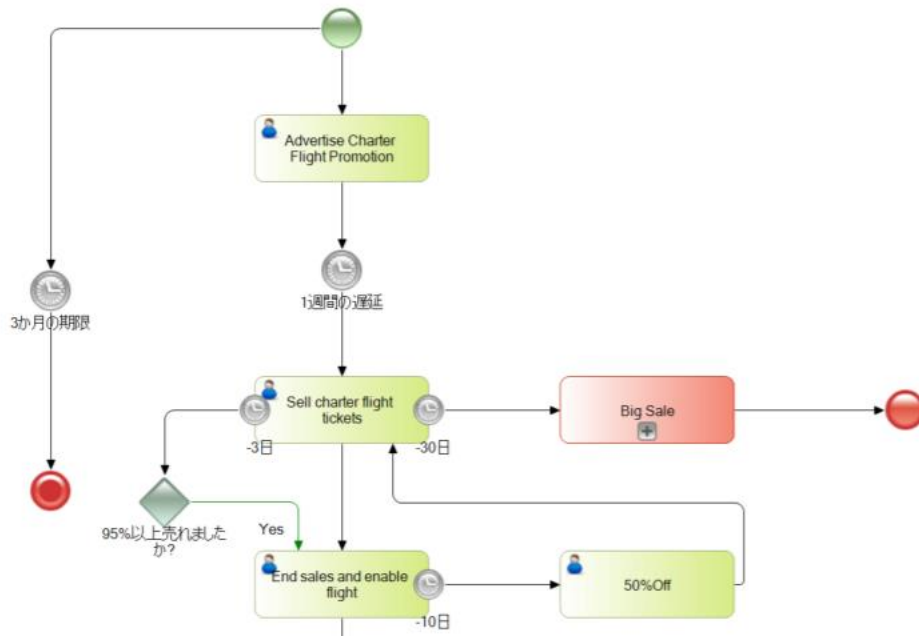
中間イベント: -3日

Name	-3日
Trigger	Timer
Interrupts activity	True
Timer definition	Duration
Timer expression type	Rule
Timer duration	P80D
Timer usage	Deadline

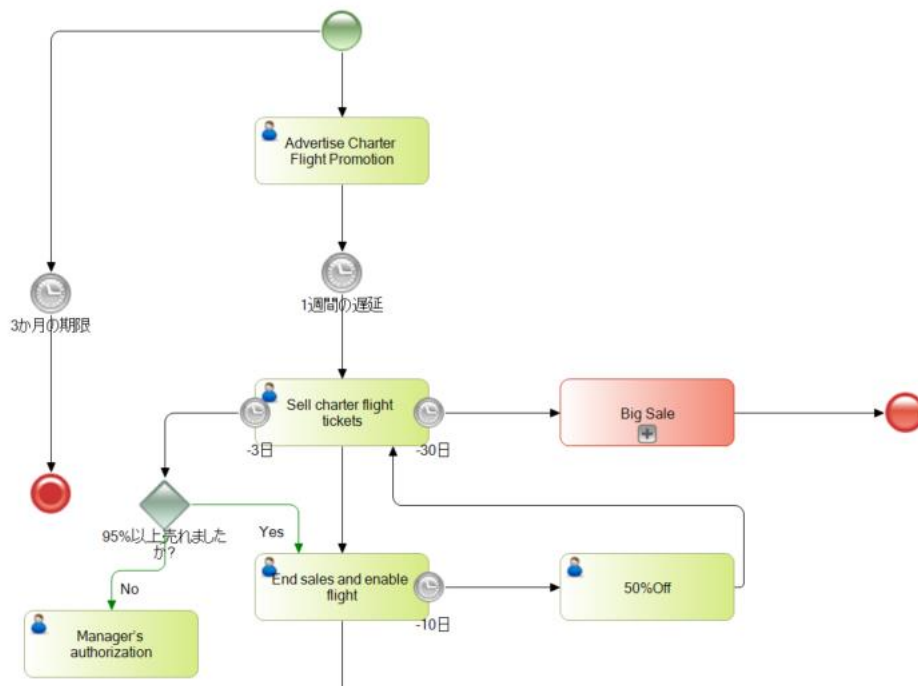
この 80 日間に、チケットの販売開始まで待機した 7 日間を加えた合計 87 日間になります。つまり、この期限はプロセス全体に設定されている 90 日間の制限の 3 日前にトリガーされることになります。

達成された売上の割合については、この情報を保存する関連データがあり、チケット販売タスクから継続的に更新されると仮定します。売上の95%に達したかどうかを確認するには、タイマー イベント「-3 日」から排他的なゲートウェイを挿入し、「95%以上売れましたか?」という説明を追加します。

右側で、これをタスク「End sales and enable flight」に接続し、フローに「Yes」タグを追加します。



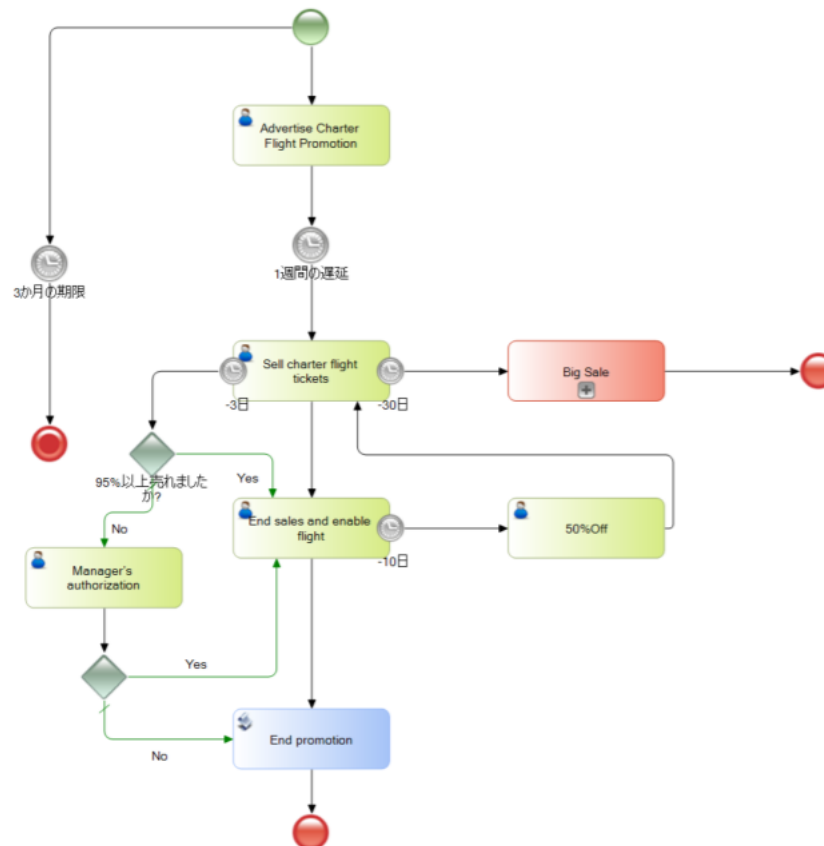
「Manager's authorization」というユーザータスクを挿入し、排他的なゲートウェイから参加します。 結合に「No」タグを追加し、デフォルトのパスになるため、「Condition Type」プロパティを「Default」に設定します。



このタスクは、チャーター便を承認またはキャンセルするマネージャーのアクションを表します。 行われた決定を照会するには、別の排他的なゲートウェイを挿入し Manager's authorization タスクから接続します。

ゲートウェイの右側で、「End sales and enable flight」タスクに接続し、このフローに「Yes」タグを追加して、フライトの出発に対するマネージャーの承認を表します。次に、販売を終了し、フライトを有効にするタスクが実行されます。最後にプロモーションが削除され、プロセスは終了します。

ゲートウェイを下位に結合し、「End promotion」タスクに接続します。 このフローに「No」タグを追加し、「Connection Type」プロパティを「Default」に設定します。したがって、マネージャーがチャーター便を承認しない場合、プロモーションは削除され、航空券の販売プロセスは終了します。



このプロセスでは、監視対象のタスクの中断の有無にかかわらず、遅延をモデル化し、プロセスの最大期間を制御し、警告を発行して期限を管理する方法を確認する機会がありました。

これには、タイマーイベントを使用して時間のパターンをモデル化するほとんどのケースが含まれます。