

ビジネス・プロセス・マネジメントのための Bonita: 第 1 回 単純なワークフローを構成する

ビジネス・プロセスのモデル化および管理に、オープンソースの Java ベースのツールを使用する

Bilal Siddiqui
Consultant

2010年 10月 12日

ビジネス・プロセス・マネジメント (BPM) のための、オープンソースの Java™ ベースのツールがその真価を発揮しつつあります。この2回の連載では、Bilal Siddiqui が BPM の概念を紹介し、BPMN (Business Process Modeling Notation) 標準を実装した BMP エンジンである Bonita オープン・ソリューションの機能を披露します。第1回となる今回は、さまざまな BPMN の要素が機能する仕組みを説明し、実際に Bonita を使ってビジネス・プロセス・ワークフローを構成する手順に取り掛かります。第2回では、残りの構成作業を完了してワークフローを実装します。

[このシリーズの他の記事を見る](#)

ビジネス・プロセスは、人々が共通のビジネス・ゴールを達成するために協調して行動することで生じます。個人から個人へのワークフローのなかで、それぞれの個人は担当するタスクを行い、自分が担っているビジネス・ロールを実行します。例えば、顧客がホテルにチェックインするための単純なワークフローを考えてみてください。フロント係は到着したゲストのために部屋を予約します。部屋の予約情報は、客室整備部門とホテルの会計部門に送られます。客室整備部門はゲストのために部屋を整え、会計部門はゲストの支払いを記録します。

ビジネス・プロセスを管理ということは、すなわちワークフローを管理することです。ビジネス・プロセス・マネジメント (BPM) ツールを使用することで、コンピューター・システムがワークフローを理解して適切に動作するようにワークフローを表現することができます。かつての方法では、何らかのプログラミング言語でコードを作成することによってコンピューターが理解するようにワークフローを表現していましたが、BPM では BPM ソフトウェアを使用し、ワークフローの構成を管理することで行うため、コードを作成する必要はまったくありません。BPM ソフトウェアでワークフローのビジネス要件をグラフィカルに表現することを可能にするのは、BPMN (Business Process Modeling Notation) という標準です。最初に BPMN ワークフロー・エディターを使ってビジネス・プロセスを BPMN で表現し、その BPMN 表現を BPMN ワークフロー・エンジンに収容して、実際のワークフローをこのエンジンに処理させます。

Java をベースとした優れたオープンソースの BPM ソリューションは、いくつか存在します。この 2 回の記事ではそのうちの 1 つ、Bonita オープン・ソリューションを使用して、ホテルの宿泊予約を例にビジネス・プロセスのワークフローを構成する方法を説明します。Bonita には、グラフィカル BPMN エディターとワークフロー・エンジンの両方が組み込まれています。

ワークフローを構成する際のタスク

ワークフローを構成するには、以下の 4 つのタスクを行う必要があります。

1. BPMN を使用してワークフローを作成します。
2. ワークフローの実行中に人間と対話するために使用するフォームを設計します。
3. ワークフローで使用する外部アプリケーションを指定します。ワークフローの要件を満たすには、必ずと言ってよいほど外部アプリケーションが必要になります。
4. 誰がどのワークフローのアクティビティを実行するかを指定するために、ワークフローに関わるすべての個人 (アクター) のビジネス・ロールを定義します。ビジネス・ロールによって、誰がどのアクティビティにアクセスできるのかが決まります。ここで決定するすべてのことが、アクセス制御ポリシーの一部になります。

この第 1 回の記事では 1 番目のタスクのデモンストレーションをするために、Bonita を使用して宿泊予約のワークフローを作成します。第 2 回では、このワークフローで使用する 2 つのフォームを設計する方法を紹介し、ワークフローでよく使用されている 2 つのオープンソース・アプリケーション、PostgreSQL データベースと JasperReports レポート作成エンジンを操作する方法のデモンストレーションを行います。アクセス制御ポリシーを定義する作業についてはこの記事では説明しませんので、「[参考文献](#)」に記載している関連する記事へのリンクを参照してください。

単純なワークフローの BPMN 表現

BPMN について紹介するために、この連載ではホテルのフロント・デスクでの宿泊予約プロセスを例として用います。このプロセスの要件を最も単純な形で表現すると、それは、顧客がフロント・デスクに来てフロント係に部屋の予約を依頼することです。フロント係は部屋の空き状況を調べ、空いている部屋があれば、その部屋を予約します (説明を単純にするために、この記事で取り上げるワークフローは完全なものではないことに注意してください。記事の冒頭で説明したように、対象とするのはホテルのフロント・デスクから部屋整備部門と会計部門までのワークフローです)。

図 1 に、この単純なプロセスの BPMN 表現を示します (この図は、Bonita のワークフロー・エディターを使用して作成したものです)。

図 1. BPMN を使用した、単純なホテル宿泊予約のワークフローの表現

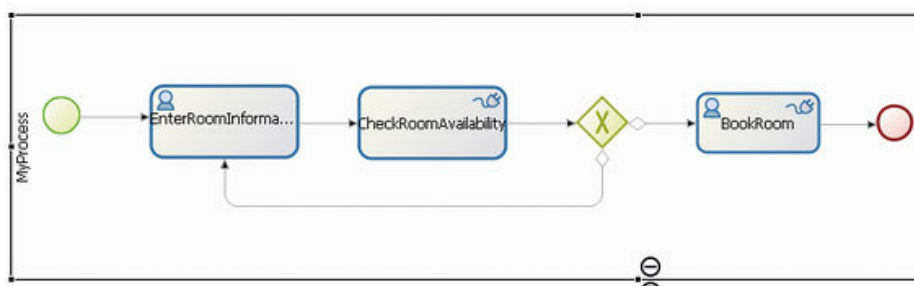


図 1 の左端に示された薄い緑色の円は、開始要素です。これは、BPMN で定義されたワークフローの開始を表します。

開始要素の右側にある矢印は、遷移を表します。この記事を読んでいくとわかりますが、遷移にはさまざまな種類があります。開始要素右側の遷移は、ワークフローの最初のアクティビティを指すだけの単純なものです。

ワークフローを構成するアクティビティ

図 1 の最初の遷移の右側には、角丸の四角形があります。図 1 には、このような四角形が 3 つ示されています。これらの四角形はワークフローのアクティビティまたはタスクを表し、自動化されることもあれば、ユーザーが実行することもあります。左端にある四角形 (「EnterRoomInformation」という名前) の内側には、これがユーザー・アクティビティであることを示す人間のシンボルが含まれています。このアクティビティでは、フロント係が、顧客が希望する部屋のタイプに関する情報を入力するためのフォームを使用します。

BPM と ERP との違い

ERP (Enterprise Resource Planning) ツール (「[参考文献](#)」を参照) は、ビジネス・ワークフローを管理するもう 1 つの手段です。ERP 実装を構成する場合も、BPM でのタスクと同じようなタスクを実行します (ワークフローの作成、フォームの設計、外部アプリケーションへのリンク、アクセス制御ポリシーの定義)。ERP ツールは進化するにつれ、互いに競い合って各ツールがより充実したワークフロー機能を提供するようになりました。いずれの ERP ツールも事前構成されたワークフローを提供しますが、なかにはワークフローのタイプ (ビジネス・プロセスを管理する際に一般に必要なワークフローのカテゴリ) を選択できるようになっているツールもあります。ワークフローのタイプを使用することで、ビジネス要件に応じたアプリケーションを迅速に構成することができます。けれども ERP ツールはその進化の結果、標準化という点で BPM に遥かに及ばなくなっています。BPM は、ERP ツールがこれまで解決しようとしてきた問題に対しての、より組織的かつ標準化されたソリューションとなります。

中央の四角形には、「CheckRoomAvailability」という名前が付けられています。この四角形の内側にある電源プラグにコードが付いたシンボルは、自動アクティビティを示します。つまり、このアクティビティはバックエンドでその操作を行うということです。実行できる自動アクティビティには複数のタイプがあります。例えば、顧客が希望している部屋のタイプが空いているか否かをチェックするためのデータベース操作などです。

右端にある 3 番目の四角形 (「BookRoom」という名前) には、人間のシンボルと電源プラグのシンボルの両方が示されています。これは、このアクティビティの一部をユーザーが実行し、残りはバックエンドの何らかの自動プロセスによって実行されることを意味します。「BookRoom」アクティビティで人間が実行する部分となるのは、フロント係による宿泊予約情報の提供です。それ以外の部分は自動的に行われます (データベース・トランザクションによって、宿泊予約情報を保存します)。

以上の説明をまとめると、1 番目のアクティビティで部屋に関する必要なデータを集め、2 番目のアクティビティで顧客が希望している部屋のタイプが空いているか否かを調べ、3 番目のアクティビティで該当する部屋の予約を完了します。

単純な条件付き遷移

図 1 に示されている、「EnterRoomInformation」アクティビティから

「CheckRoomAvailability」アクティビティにつながっている単純な遷移 (矢印) に注目してください。この単純な遷移は、フロント係が部屋の空き状況を問い合わせた結果のデータを収集した後に、顧客が希望している部屋のタイプが空いているか否かを調べるというステップが必ず行われることを意味しています。

次の遷移 (「CheckRoomAvailability」アクティビティから「BookRoom」アクティビティへの矢印) は、単純な遷移ではありません。この遷移は、「CheckRoomAvailability」アクティビティの結果に依存します。つまり、顧客が希望している部屋のタイプが空いているか否かによって、次のアクティビティが変わってくるということです。部屋が空いている場合は、フロント係が顧客のためにその部屋を予約するので、「BookRoom」アクティビティが開始されます。一方、要望されているタイプの部屋が空いていなければ、フロント係は別のタイプの部屋が空いているか否かをチェックしなければならないため、制御は「EnterRoomInformation」アクティビティに戻されます。

したがってここでは、どの方向に進むかについての決定が必要になります。このような決定を行わなければならない場合には必ず、実行のパスが分岐します。この分岐を表すのが、中にクロス記号のあるひし形 (ゲート要素、あるいは BPMN で言うゲートウェイ) です。図 1 に示されているとおり、「CheckRoomAvailability」アクティビティの出力に接続しているゲート要素には、2 つの実行可能なパスがあります。

ゲートには、矢印付きの線が 3 本接続されています。この 3 本の線は、それぞれ以下の遷移を表します。

- 1 番目の遷移は、ゲートへの入力です。この遷移は「CheckRoomAvailability」アクティビティから始まり、ゲートで終了します。これは、単純な無条件の遷移です。
- 2 番目の遷移は、ゲートから「EnterRoomInformation」アクティビティにつながっています。これは、指定された部屋のタイプが空いていない場合の遷移を表します。つまり、条件付きの遷移です。
- 3 番目の遷移は、ゲートから「BookRoom」アクティビティにつながっている遷移で、これも同じく条件付きで、顧客が希望している部屋のタイプが空いている場合の遷移を表します。

注意する点として、ゲート要素は遷移が到達するポイントしか示しません。ゲートに入る遷移、またはゲートから出る遷移のそれぞれに、独自のプロパティがあります (つまり、単純な遷移であるか、条件付きの遷移であるかというプロパティです)。

この単純な宿泊予約のプロセスを BPMN で表現する方法を説明したところで、次のセクションではこのワークフローを Bonita で構成する実際の手順を紹介します。この記事の実践演習に従うには、次のセクションに移る前に Bonita をダウンロードしてインストールし、起動してください (「参考文献」を参照)。

Bonita での単純なワークフローの構成

Bonita のグラフィカル・ツールは簡単に使えます。このツールでは、コンポーネントをパレットからホワイトボードと呼ばれるエディター・ウィンドウにドラッグ・アンド・ドロップした後、各コンポーネントをビジネス・プロセスの要件に応じて個別に構成することができます。

図 2 に、Bonita のウェルカム・ページを示します。このページには、新しいプロセスの作成や既存のプロセスのオープンなどといった共通タスクに直接アクセスするためのリンクが組み込まれています。

図 2. Bonita のウェルカム・ページ



宿泊予約ワークフローを構成する作業を開始するには、「New (新規)」ボタンをクリックして、新規プロセスを表示します。新規プロセスには、「MyProcessDiagram」(バージョン 1.0) という名前が付いています (図 3 を参照)。

図 3. 新規プロセス

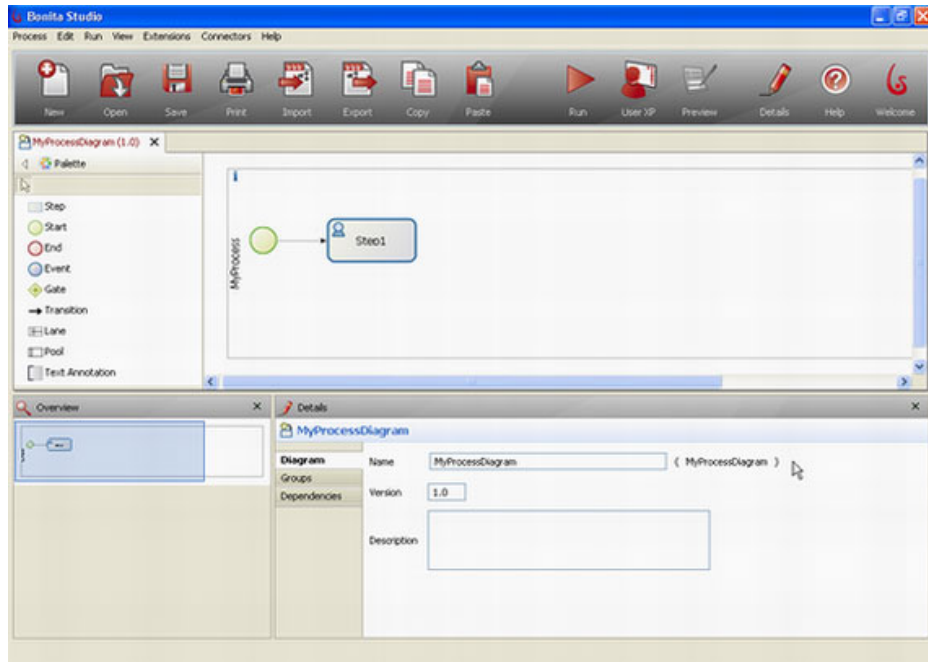


図 3 を見るとわかるように、Bonita の画面は上下半分に分割され、上下それぞれの部分がさらに縦に分割されています。

左上のセクションには、ワークフローを構成するために使用できる BPMN 要素のパレットが表示されます。

右上の部分はプロセス・ホワイトボードで、ここに、パレットの BPMN 要素をドラッグ・アンド・ドロップします。ホワイトボードは、プロセスの実際の設計ビューです。図 3 に示されているように、Bonita は新しいプロセスを作成する際に、自動的に 3 つの BPMN 要素をホワイトボードに追加します。その 3 つの要素とは、「Start (開始)」、「Transition (遷移)」、そして「Step1」という名前が付いたアクティビティです。

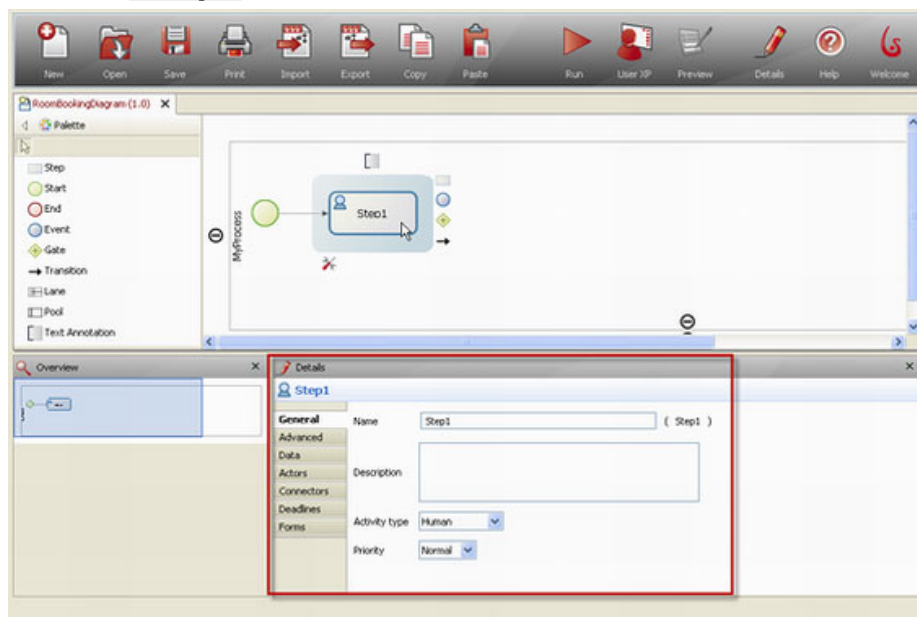
左下の部分は、開発中のワークフロー全体の概要を示す「Overview (概要)」ウィンドウです。このウィンドウでは、ワークフローの特定の部分に素早くジャンプすることができるため、ワークフロー定義の規模が大きく、スクロール・バーなしではホワイトボードに表示しきれない場合には特に重宝します。

右下の部分は、個々の BPMN 要素を構成するために使用する「Details (詳細)」ウィンドウです。このウィンドウには、構成作業時に設定する BPMN 属性が表示されます。図 3 では選択されている BPMN 要素がないため、「Details (詳細)」ウィンドウにはワークフローの属性が表示されています。例えば属性の 1 つとして、このウィンドウの「Diagram (ダイアグラム)」タブには、「Name (名前)」フィールドに「MyProcessDiagram」という値が入力されています。この「MyProcessDiagram」を「RoomBookingDiagram」という値に変更して、Bonita のメイン・ツールバーにある「Save (保存)」ボタンをクリックしてください。

Bonita でユーザー・アクティビティを構成する

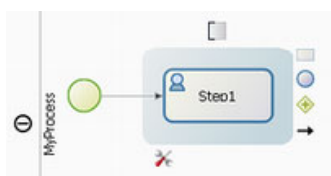
ここからは早速、宿泊予約ワークフローの構成に取り掛かります。「Step1」アクティビティをクリックしてください。すると、このアクティビティの属性が「Details (詳細)」ウィンドウに表示されます (図 4 を参照)。

図 4. 「Step1」アクティビティの属性詳細



「Step1」アクティビティをクリックすると、コンテキスト・パレット (BPMN 要素アイコンのサブセット) がアクティビティの周りに表示されます (図 5 を参照)。

図 5. コンテキスト・パレットが表示された「Step1」アクティビティ



コンテキスト・パレットを使用することで、ワークフローの次のアクティビティや遷移を簡単に追加することができます。この記事の演習では、コンテキスト・パレットを頻繁に使用します。

図 4 を見ると、「Details (詳細)」ウィンドウには複数のタブがあります。これらのタブを使用して、BPMN 要素 (例えば、「Step1」アクティビティ) を Bonita のホワイトボードにドラッグ・アンド・ドロップしながら、その要素の振る舞いを制御することができます。各タブには、ビジネス・プロセスを管理する際に考えられる要件をほぼすべて網羅する豊富な機能が提供されています。この記事では、これらの BPMN 機能のすべてを取り上げることはしません。この単純なホテル予約のケースでは、「Details (詳細)」ウィンドウのタブのうち、「General (一般)」、「Data (データ)」、「Connectors (コネクタ)」、「Forms (フォーム)」を使うだけで用が足ります。

図 4 に示されている「General (一般)」タブでは、アクティビティの名前、説明、タイプ、優先度を指定することができます。「Name (名前)」テキスト・フィールドに

「EnterRoomInformation」と入力して、「Step1」となっているアクティビティの名前を変更してください。ホワイトボードには、アクティビティの名前として「EnterRoomInformation」と表示されるようになります。

「Description (説明)」フィールドには、BPMN 要素に関する説明を入力することができます。この「Description (説明)」フィールドに、「Show a form to fetch the information about the room required by the customer (顧客が希望する部屋に関する情報を取得するためのフォームを示します)」と入力してください。

次に、このアクティビティのタイプを指定します。「EnterRoomInformation」アクティビティでは、顧客が希望している部屋のタイプを指定するために、データ入力フォームに入力する必要があることから、このアクティビティは宿泊予約ワークフロー・アプリケーションを使用するフロント係が行うユーザー・アクティビティということになります。「Activity type (アクティビティ・タイプ)」フィールドはドロップダウン・リストです。このリストにデフォルトで選択されている「Human (人間)」を、「EnterRoomInformation」アクティビティのタイプとして指定します。

「EnterRoomInformation」アクティビティはユーザー・アクティビティなので、そのためのデータ入力フォームを作成しなければなりません。このフォームは連載の第2回で「Details (詳細)」ウィンドウの「Forms (フォーム)」タブを使って設計して実装してから、アクティビティに追加します。第2回ではまた、「Data (データ)」タブを使用して、顧客が希望している部屋に関するデータを格納する変数も定義します。

自動アクティビティをワークフローに追加する

続いて、ワークフローに遷移を追加するとともに、次のアクティビティとして

「CheckRoomAvailability」という名前のアクティビティもワークフローに追加します。

「CheckRoomAvailability」アクティビティは「EnterRoomInformation」フォームからデータを受け取り、ホテルのデータベース (このデータベースは第2回で構成します) で部屋の空き状況をチェックした後、部屋が空いているか否かを示すフラグを設定します。

「EnterRoomInformation」アクティビティをクリックして、コンテキスト・パレットを起動します。コンテキスト・パレットの右上隅にあるステップ要素アイコン (小さなボックス) をドラッグし、「EnterRoomInformation」アクティビティの隣にドロップします。図6に示すように、この操作によって遷移と新しい「Step1」要素がワークフローに追加されます。

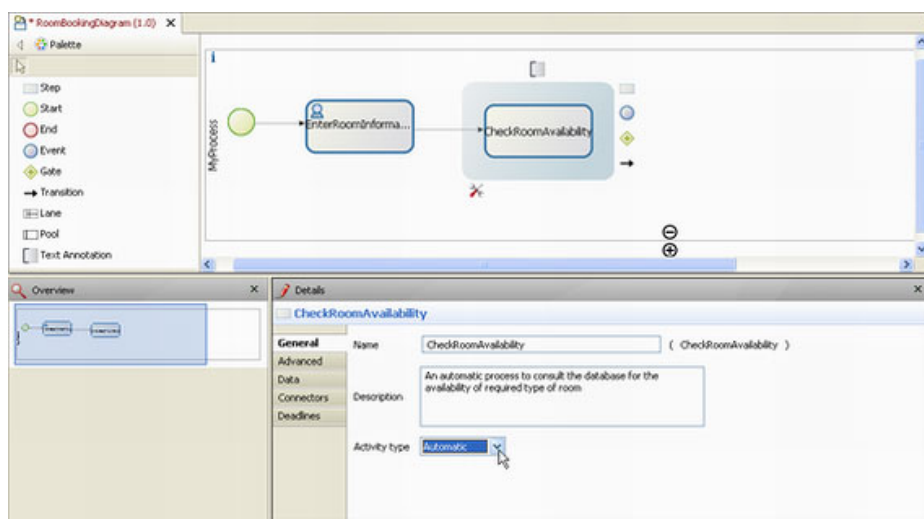
図6. ワークフローに新しく追加された遷移と「Step1」要素



新しく作成された「Step1」要素の「Details (詳細)」ウィンドウには、アクティビティの名前、説明、タイプの各フィールドが表示されます。このアクティビティの名前として、「CheckRoomAvailability」と入力してください。「Description (説明)」フィールドには、

「An automatic process to consult the database for the availability of required type of room (顧客が希望している部屋のタイプの空き状況をデータベースに問い合わせるための自動プロセス)」と入力します。「Activity type (アクティビティー・タイプ)」ドロップダウン・リストからは、「Automatic (自動)」を選択します。図 7 に、以上のように構成した「CheckRoomAvailability」アクティビティーを示します。

図 7. 「CheckRoomAvailability」アクティビティーの構成



部屋の空き状況フラグをワークフローに追加する

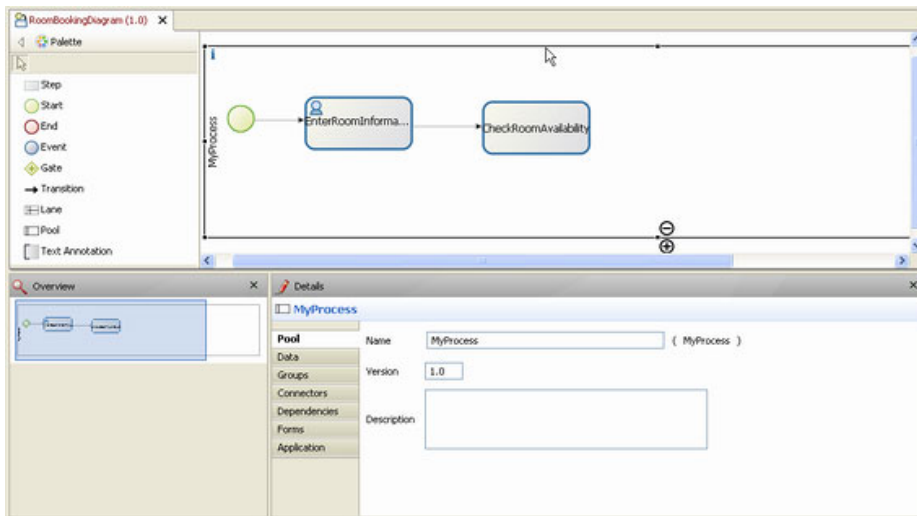
「CheckRoomAvailability」アクティビティーは、顧客が希望している部屋のタイプが空いている場合には「isRequiredRoomTypeAvailable」という名前のフラグを設定します。ワークフローの次の遷移は、この「isRequiredRoomTypeAvailable」フラグが設定されているか否かに依存するため、Bonita で、「isRequiredRoomTypeAvailable」フラグを変数として定義する必要があります。

Bonita では、個別のアクティビティーごとの変数も、ワークフロー全体の変数も定義することができます。特定のアクティビティー内でのみ変数が必要な場合には、アクティビティー・レベルの変数を定義します。変数を複数のアクティビティーにまたがって使用する場合には、ワークフロー・レベルで変数を定義し、すべての要素がその変数にアクセスできるようにします。

宿泊予約ワークフローの場合、「isRequiredRoomTypeAvailable」フラグはワークフロー・レベルの変数にしなければなりません。このフラグには複数の要素がアクセスするためです。具体的には、CheckRoomAvailability アクティビティーがフラグを設定し、次の遷移がフラグのステータスをチェックします。

変数をワークフロー・レベルで定義するには、ワークフローの定義全体を囲んでいる四角形をクリックします (図 8 では、この四角形が強調表示されています)。

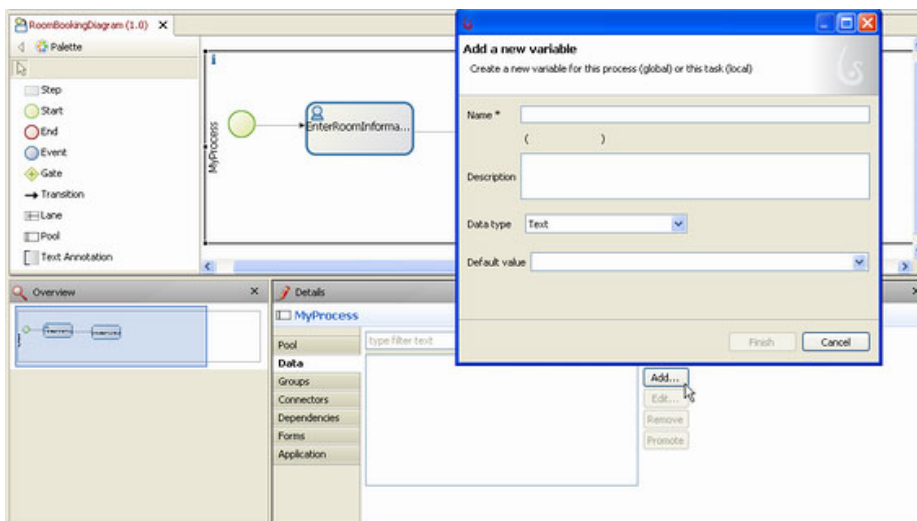
図 8. 宿泊予約ワークフローの「Details (詳細)」ウィンドウ



「Details (詳細)」ウィンドウには宿泊予約ワークフロー全体の属性が表示されていることに注目してください。

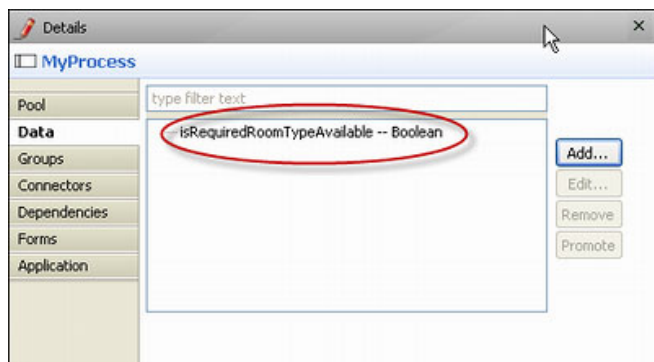
「Details (詳細)」ウィンドウの「Data (データ)」タブをクリックし、このタブに表示されている「Add (追加)」ボタンをクリックして「Add a new variable (新規変数の追加)」ダイアログを開きます (図 9 を参照)。

図 9. ワークフローへの新規変数の追加



「Name (名前)」テキスト・フィールドに「isRequiredRoomTypeAvailable」と入力し、「Description (説明)」フィールドには「A workflow-level flag to show whether the required type of room is available or not (顧客が希望している部屋のタイプが空いているか否かを示すワークフロー・レベルのフラグ)」と入力します。「Data type (データ型)」ドロップダウン・リストから「Boolean (ブール値)」を選択して、フラグのデフォルト値として「false」を選択します。以上の設定が完了したら、「Finish (完了)」ボタンをクリックします。これで、「Data (データ)」タブに isRequiredRoomTypeAvailable 変数が追加されます (図 10 を参照)。

図 10. 「Data (データ)」 タブのテキスト・ボックスに追加された **isRequiredRoomTypeAvailable** 変数



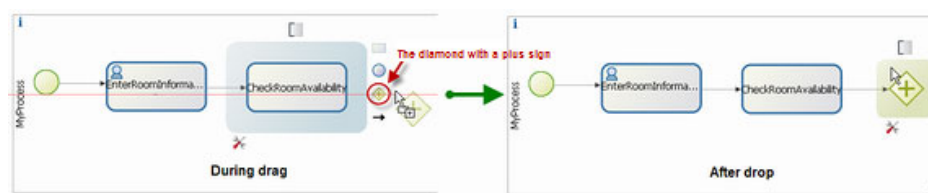
ここまでの手順で、「CheckRoomAvailability」アクティビティと併せて

「isRequiredRoomTypeAvailable」という名前のワークフロー・レベルのフラグを定義しました。けれどもまだ、実際のデータベースも構成されていなければ、部屋の空き状況データを読み取って isRequiredRoomTypeAvailable フラグをセット (またはリセット) する SQL 文も構成されていません。これらの構成方法については第 2 回で説明することとして、ここではとりあえず、次の遷移を構成する方法の説明に移ります。この遷移は動的遷移であり、部屋が空いているか否かに基づいて遷移が決定されなければなりません。

決定による動的遷移

「CheckRoomAvailability」アクティビティをクリックして、コンテキスト・パレットを起動してください。コンテキスト・パレットのプラス記号が付いたひし形を「CheckRoomAvailability」アクティビティの隣にドラッグ・アンド・ドロップすることで、この位置にゲート要素を配置します (図 11 を参照)。

図 11. BPM プロセスでの決定を表すゲート要素

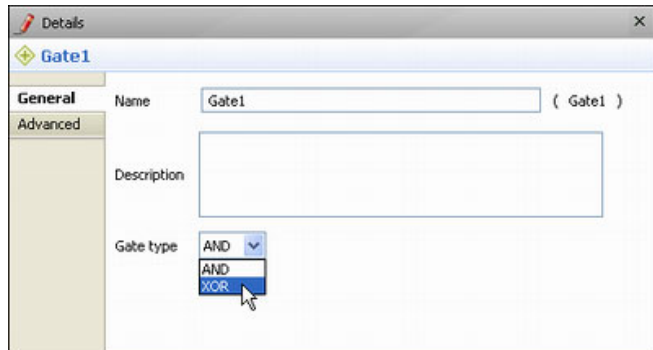


ゲート要素には、以下の 2 つのタイプのいずれかを設定することができます。

- XOR ゲート: ゲートの内側にクロス記号がある場合、このゲートは 1 つのパスを複数のパスに分岐するために使用されます。
- AND ゲート: ゲートの内側にプラス記号がある場合、このゲートは 2 つのパスを結合します。

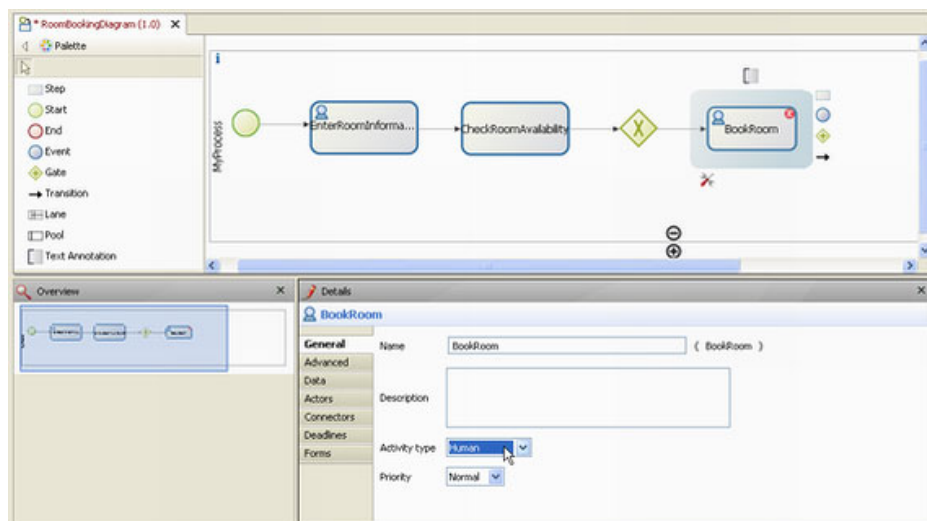
「**単純な条件付き遷移**」セクションで説明したように、ここで必要となるのは 1 つのパスを複数のパスに分岐するゲートです。正しいタイプのゲートを設定するためには、ホワイトボードでゲート要素をクリックしてから、「General (一般)」タブで「Gate type (ゲート・タイプ)」プロパティとして「XOR」を選択してください (図 12 を参照)。

図 12.1 つのパスを分岐する XOR ゲートの選択



正しいゲートのタイプを設定したところで、ゲート要素のコンテキスト・パレットでステップ要素アイコンをクリックし、ゲートの右側にドラッグします。これにより、ゲートの右側に「Step1」という新しいアクティビティが配置されるので、「Details (詳細)」ウィンドウの「General (一般)」タブで、この新規アクティビティの名前を「BookRoom」に変更し、タイプを「Human (人間)」に設定します。このようにゲートを構成すると、プロセスは図 13 のようになっているはずです。

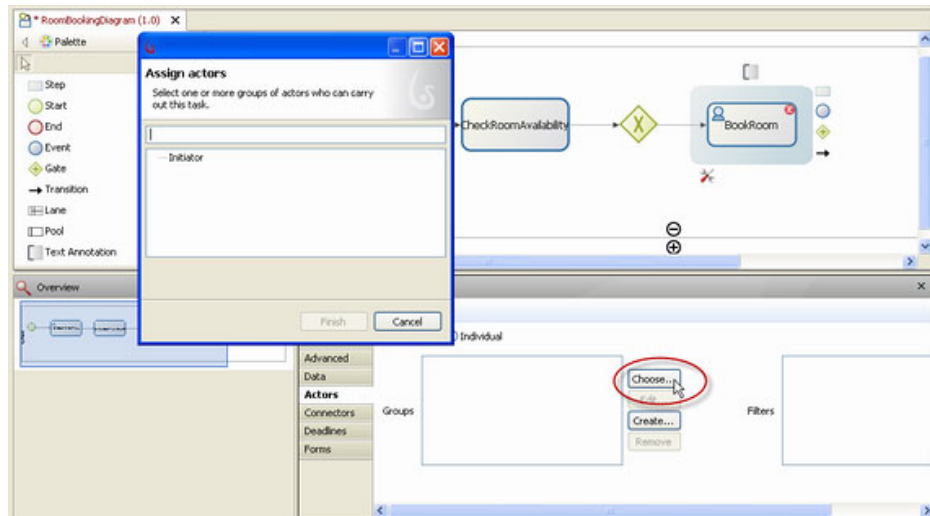
図 13. ゲートの隣に配置された「BookRoom」アクティビティ



Bonita でのビジネス・ロール

「BookRoom」要素の左隅には人間を表すシンボルがあり、右隅にはバツ印があります (両方とも図 13 に示されています)。このバツ印は、誰がこのアクティビティを行うかが、まだ指定されていないことを意味するので、「Actors (アクター)」タブをクリックし、続いて「Choose (選択)」ボタンをクリックしてください。この操作によって「Assign actors (アクターの割り当て)」ダイアログが開きます (図 14 を参照)。

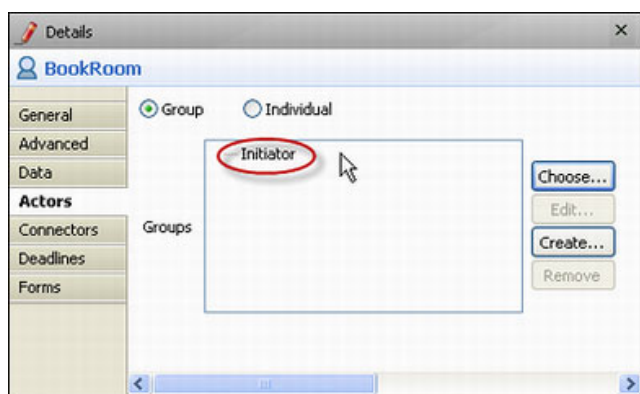
図 14. 「Assign actors (アクターの割り当て)」 ダイアログ



「Assign actors (アクターの割り当て)」ダイアログでは、「BookRoom」アクティビティを実行できるアクターのグループを選択します。Bonita ではワークフローに必要なビジネス・ロールに応じて、アクターのグループを定義できるようになっています。ビジネス・ロールについての説明は、この記事では行わないので、ここでは単純に、Bonita に事前定義されている「Initiator」という名前のデフォルト・アクター・グループを使用します。「Initiator」グループは、ワークフローを開始する任意のユーザーで構成されます。

テキスト・ボックスに表示されている「Initiator」を選択し、ウィンドウの下部にある「Finish (完了)」ボタンをクリックします。これで、「Initiator」が「Actors (アクター)」タブのリストに追加されます (図 15 を参照)。

図 15. 「Actors (アクター)」タブのリストに追加されたデフォルトの「Initiator」アクター



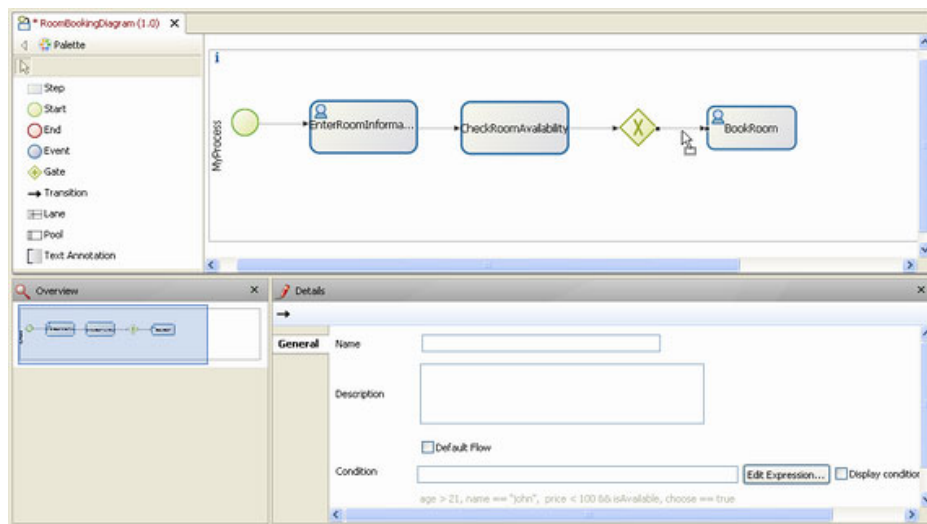
「BookRoom」アクティビティには、ホテルのフロント係に顧客情報を入力するよう求めるユーザーとの対話用フォームも必要です。このフォームを含め、ワークフローに必要なすべてのフォームは第 2 回で設計します。また、今のところ「BookRoom」はユーザー・アクティビティとなっていますが、第 2 回ではこのアクティビティの自動で行われる部分を構成します。

遷移に条件を追加する

今度はゲートから「BookRoom」アクティビティへの遷移に、部屋の空き状況の条件を追加します。このワークフローは、顧客が希望している部屋のタイプが空いている場合にのみ、「BookRoom」アクティビティが開始されるように構成しなければなりません。

ゲートから「BookRoom」アクティビティへの遷移要素をクリックして、この遷移要素の詳細の属性を「Details (詳細)」ウィンドウに表示します (図 16 を参照)。

図 16. 遷移の属性が表示された「Details (詳細)」ウィンドウ

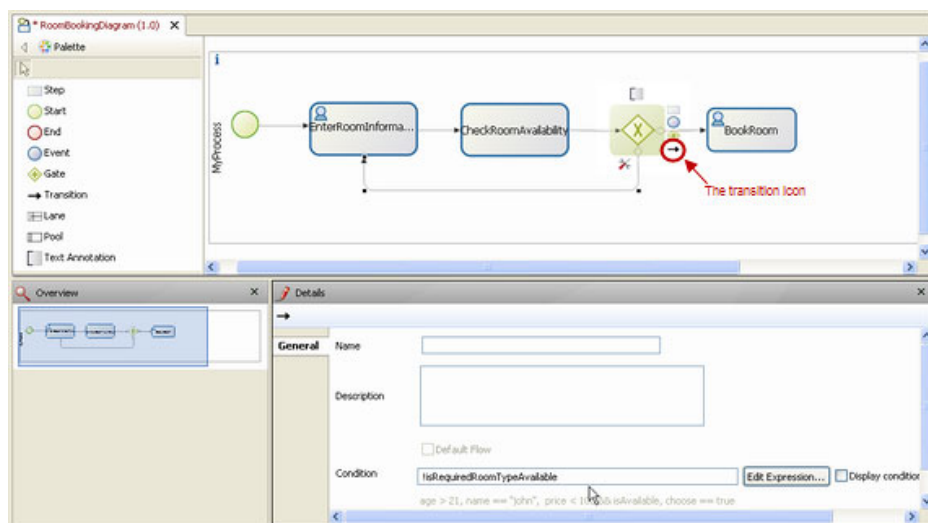


遷移要素の「Details (詳細)」ウィンドウには、「General (一般)」というタブしかありません (図 16 を参照)。「General (一般)」タブには「Condition (条件)」という名前のフィールドがあり、ここに、遷移を発生させる条件を指定します。「Condition (条件)」テキスト・フィールドには「isRequiredRoomTypeAvailable」と入力します。「[部屋の空き状況フラグをワークフローに追加する](#)」セクションで、isRequiredRoomTypeAvailable をワークフロー・レベルの変数として定義したことを思い出してください。部屋が空いている場合、「CheckRoomAvailability」アクティビティは isRequiredRoomTypeAvailable フラグを true に設定します。この条件は単に、isRequiredRoomTypeAvailable 変数が true であれば (つまり、フラグがセットされている場合)、「BookRoom」アクティビティへの遷移を有効にするというだけにすぎません。

遷移の条件を入力すると、「Transition (遷移)」要素のゲート側に小さなひし形が表示されることに注目してください。このひし形は、遷移が条件付きであることを示します。

ご想像のとおり、もう一方の遷移にも条件を設定する必要があります。プロセスのホワイトボードでゲート要素をクリックして、コンテキスト・パレットを起動してください。コンテキスト・パレットの一番下にある矢印 (遷移アイコン) を「EnterRoomInformation」アクティビティまでドラッグします。これにより、ゲートから「EnterRoomInformation」アクティビティにつながる「Transition (遷移)」要素が描画されます (図 17 を参照)。

図 17. 顧客が希望している部屋のタイプが空いていない場合のゲートから「EnterRoomInformation」アクティビティへの遷移



あとは、図 17 に示されているように、この遷移の条件として `isRequiredRoomTypeAvailable` を設定するだけです。これで、顧客が希望している部屋のタイプが空いていない場合には、ワークフローが「EnterRoomInformation」アクティビティに戻ることになります。

ワークフローの終了を構成する

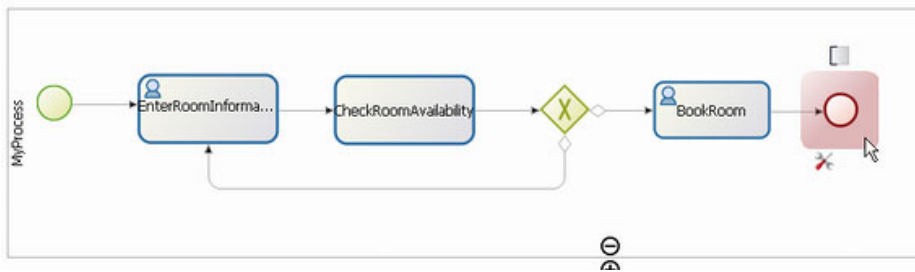
ワークフローを構成するための最終ステップは、プロセスに「End (終了)」要素を追加することです。それには、「BookRoom」のコンテキスト・パレットから円の項目を選択して右方向にドラッグします。すると、イベント・セクター・パレットが表示されます (図 18 を参照)。

図 18. イベント・セクター・パレット



イベント・セクター・パレットには、BPMN で定義されている各種のイベントを表す記号が含まれています。ここで必要なのは「End (終了)」要素で、この要素はイベント・セクター・パレットの右上隅にあります (図 18 に赤い丸で示している赤い円)。イベント・パレットからこの「End (終了)」要素を選択すれば、ワークフローに「End (終了)」要素が追加されます (図 19 を参照)。

図 19. 「End (終了)」要素が追加された宿泊予約ワークフロー



これで宿泊予約ワークフローの構成は完了しましたが、このワークフローにはまだフォームもデータベースも用意されていません。これまで行った作業を保存するために、Bonita のメイン・メニューで「Save (保存)」をクリックして、ワークフロー情報を RoomBookingDiagram.proc ファイルに保存してください。ワークフロー情報がすでに保存されている RoomBookingDiagram.proc ファイルは[ダウンロード](#)することもできます。

まとめ

事前定義されたワークフローやよく使用されるワークフロー・タイプなど、ERP に用意されている機能は便利ですが、それを使用するには犠牲も伴います。ERP の機能は、それぞれの製品に固有のものだからです。私の経験では、IT の専門家に製品固有の機能を使って日常のワークフローの問題を解決する方法を教えることが、どの ERP プロジェクトでも最も難しい部分となります。

こうした理由から、BPM は新しい時代の波になると思います。標準化はまだ完全ではないと言え、着々と進行中です。私は数年のうちに、BPM が一般的なワークフロー機能にも、一般的ではないワークフロー機能にも対応してすべてのワークフロー機能をカバーするようになると確信しています。そうなれば、標準化された BPM トレーニングが IT カリキュラムの一部となり、製品固有の ERP トレーニングは、もはや問題にならなくなることでしょう。

この記事では、BPMN について簡単に説明し、Bonita で BPMN を使用して単純なワークフローを構成する方法を紹介しました。第 2 回では、この第 1 回でやり残した作業をすべて終えるため、フォームの設計方法、変数の使用方法、PostgreSQL データベースに接続する方法、そして JasperReports でレポートを生成する方法を説明し、宿泊予約ワークフローを実際に起動します。

ダウンロード

内容	ファイル名	サイズ
BPMN file for the example	j-bpm1.zip	4KB

著者について

Bilal Siddiqui

Bilal Siddiqui は、電気工学エンジニア、XML コンサルタント、そして e-business の簡易化を専門とする会社、WaxSys の共同設立者でもあります。1995年にパキスタンのラホールにある University of Engineering and Technology を卒業して電子工学技術の学位を取得した後、産業用制御システムを対象としたソフトウェア・ソリューションの設計を始めました。その後、XML に転向し、C++ のプログラミング経験を生かして Web ベースや Wap ベースの XML 処理ツール、サーバー・サイドの構文解析ソリューション、そしてサービス・アプリケーションを作成しました。テクノロジー・エバンジェリストである彼が書いた[技術書](#)は、頻繁に発行されています。

© Copyright IBM Corporation 2010

(www.ibm.com/legal/copytrade.shtml)

商標

(www.ibm.com/developerworks/jp/ibm/trademarks/)