

Integration Designer

バージョン 7.5

バージョン 7 リリース 5

**ビジネス・ステート・マシン・
エディターを使用した
自動販売機の作成**

IBM

注

本書および本書で紹介する製品をご使用になる前に、19ページの『特記事項』に記載されている情報をお読みください。

このエディションは、IBM Integration Designer のバージョン 7.5 に適用されます。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原典： Integration Designer
Version 7.5
Version 7 Release 5
Creating a vending machine using the
business state machine editor

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

© Copyright IBM Corporation 2005, 2011.

目次

第 1 章 概要	1	第 3 章 インポート	13
第 2 章 ユーザーによるビルド	3	既製のサンプルを開く	13
ビジネス成果物の作成	3	第 4 章 サンプルの実行	15
ステート・マシンの作成	4	サンプルのテスト	15
Idle ステートの構成	7	特記事項	19
Depositing ステートの作成	9	ご利用条件	23
2 つのステートをリンク	10		
ステート・マシン・コンポーネントの作成	11		

第 1 章 概要

このサンプルでは、IBM WebSphere Adapter for Flat Files を使用して、アプリケーションでファイル・システム上のファイルの読み取りと書き込みを行う方法の実例を示します。また、着信データを処理し、処理後のデータをファイル・システム上のファイルに書き込むアプリケーションを IBM® Integration Designer で作成する方法についても説明します。このサンプルは、アダプター全般、特にフラット・ファイル・アダプターを使用して作業する開発者向けに書かれています。

学習目標

このサンプルを完成させると、IBM Integration Designer を WebSphere Adapter for Flat Files と共に使用して、ファイル・システム上のファイルからデータを読み取り、そのデータをサービス指向アーキテクチャー (SCA) アプリケーションで処理し、処理済みのデータをファイル・システム上のファイルに書き込むアプリケーションを開発する方法を理解できます。具体的には、以下のタスクの実行方法を学習します。

- フラット・ファイル・アダプターで外部サービス・パターン・ウィザードを使用して、ファイル・システム上のファイルからデータを読み取るインバウンド・サービスを開発する
- 同じウィザードおよびアダプターを使用して、ファイル・システム上のファイルにデータを書き込むアウトバウンド・サービスを開発する
- ファイル・システム上のファイルから読み取ったデータを処理するビジネス・ロジックを実装する
- アプリケーションをテストする

このサンプルは、以下のランタイム環境で使用できます。

- WebSphere® Enterprise Service Bus
- IBM Process Server

このアプリケーションをユーザーがビルドしてテストするために必要な時間は、約 30 分です。既にビルド済みのアプリケーションをインポートしてテストすることもでき、その場合に要する時間は 10 分です。

第 2 章 ユーザーによるビルド

ユーザーは、このサンプルを自分自身でビルドすることができます。

学習目標

IBM Integration Designer でウィザードとエディターを使用してこの実地サンプルをビルドする際は、以下のことを行います。

- ディレクトリーを作成する
- 開発するサービスを格納するためのモジュールを作成する
- 開発時間を節約するために、ビジネス・オブジェクトをインポートする
- アダプター・パターン・ウィザードを使用して、インバウンド・サービスを作成する
- 同じウィザードを使用してアウトバウンド・サービスを作成する
- メディエーション・フローを作成します。
- モジュールのデプロイ
- モジュールのテスト

ビジネス成果物の作成

グラフィカル・エディターでステート・マシンの組み立てを始める前に、ステート・マシンをサポートする成果物を作成する必要があります。

このステップでは、以下の成果物を作成します。

- ビジネス・モジュール
- 2 つのビジネス・オブジェクト
- インターフェース
- 5 つの操作

注: これらの各成果物についての詳細は、「入門」サンプルで説明しています。

1. ビジネス・モジュールを作成するには、以下の手順に従ってください。
 - a. 「ビジネス・インテグレーション」ビューの空の領域を右クリックして、リストから「新規」 > 「モジュール」を選択します。
 - b. 「新規モジュール」ウィンドウで、モジュールに BSM_VendingMachine という名前を付け、デフォルトを受け入れて、「完了」をクリックします。
2. 下の説明に従ってビジネス・オブジェクトを 2 つ作成し、表にリストされた値をそれらに割り当てます。
 - a. 新たに作成したモジュールを右クリックして、リストから「新規」 > 「ビジネス・オブジェクト」を選択します。「新規ビジネス・オブジェクト」ウィンドウが表示されます。
 - b. 下の表を参照して、「ビジネス・オブジェクト名」列から「名前」フィールドに値を入力し、「終了」をクリックします。その他のビジネス・オブジェクトについても同様に繰り返します。
 - c. ビジネス・オブジェクト・エディターで新しいビジネス・オブジェクトを右クリックして、「フィールドの追加」を選択します。

- d. 次の表に示すように、「属性」フィールドと「型」フィールドに値を割り当てます。属性を追加するには「field 1」に上書きします。「string」を受け入れるか、**Tab** キーを押してリストから型を選択してください。

ビジネス・オブジェクト名	属性	タイプ
Coin	id	string
	value	double
Selection	id	string
	item	string

3. インターフェースを作成するには、以下の手順に従ってください。
 - a. モジュールを右クリックして、リストから「新規」 > 「インターフェース」を選択します。「新規インターフェース・ウィザード」ウィンドウが表示されます。
 - b. インターフェースに **VendingMachineInterface** という名前を付け、「終了」をクリックします。
4. このインターフェースに 5 つの操作を作成します。各操作は、あるステートから別のステートへの遷移を発生させるアクションを表します。
 - a. 下の表にある最初の列を参照して、作成する操作の型を決定します。次に、インターフェース・エディターを右クリックして、「片方向操作の追加」または「要求/応答操作の追加」を選択します。
 - b. 下の表のフィールドに従って、新しい操作を構成します。

操作	操作タイプ	入力	入力の型	出力	出力の型
on	片方向	id	string	-	-
deposit	要求/応答	coin	Coin	accepted	boolean
select	要求/応答	selection	Selection	processed	boolean
cancel	片方向	id	string		
off	片方向	id	string	-	-

5. 作業を保存します。

これで、必要な成果物の作成が完了しました。ステート・マシンを作成して、必要なコンポーネントの組み立てを始めることができます。

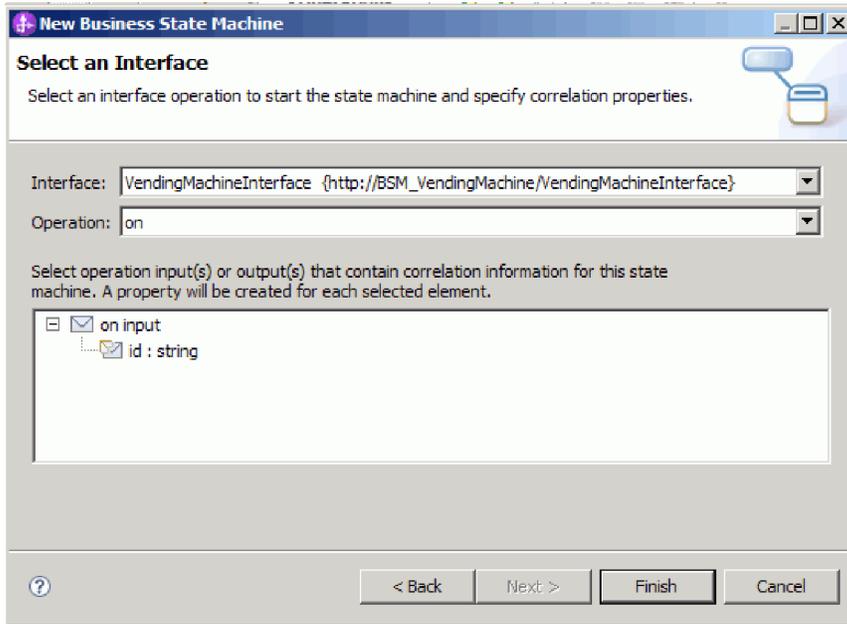
ステート・マシンの作成

必要な成果物の準備ができたので、次はステート・マシンを作成して、適切な構成を行うことができます。

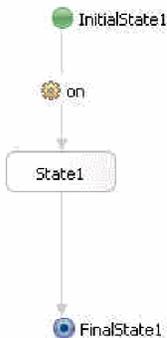
このステップでは以下を行います。

- ステート・マシンの作成
 - 相関の指定
 - 変数の作成
1. 新しいステート・マシンを作成するには、以下の手順に従ってください。
 - a. **BSM_VendingMachine** モジュールを右クリックして、リストから「新規」 > 「ビジネス・ステート・マシン」を選択します。
 - b. 「新規ビジネス・ステート・マシン」ウィザードで、ステート・マシンに **VendingMachine** という名前を付け、「次へ」をクリックします。

- c. 「インターフェースの選択」 ページで、「インターフェース」 ドロップダウン・リストから **VendingMachineInterface** を選択し、**on** 操作を展開して「**id : string**」を選択し、「完了」をクリックします。これは最初の操作で、自動販売機を始動します。

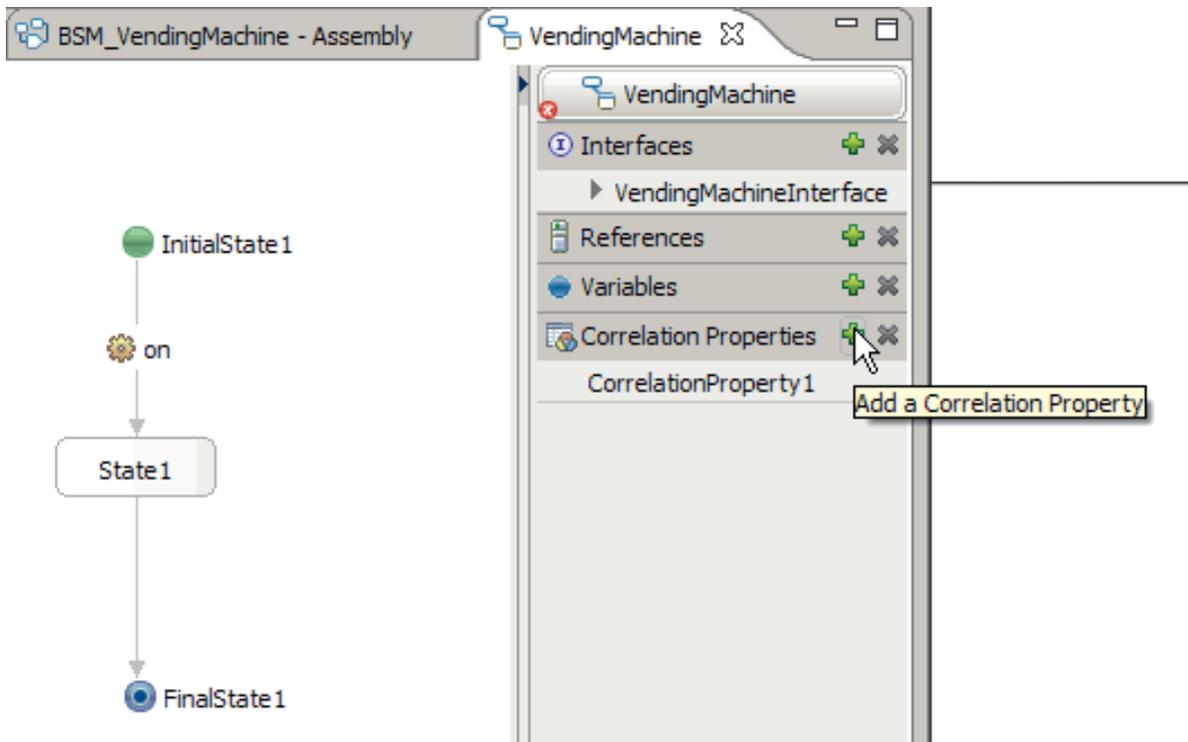


新しいステート・マシンが作成され、次の図のようにエディターに表示されます。

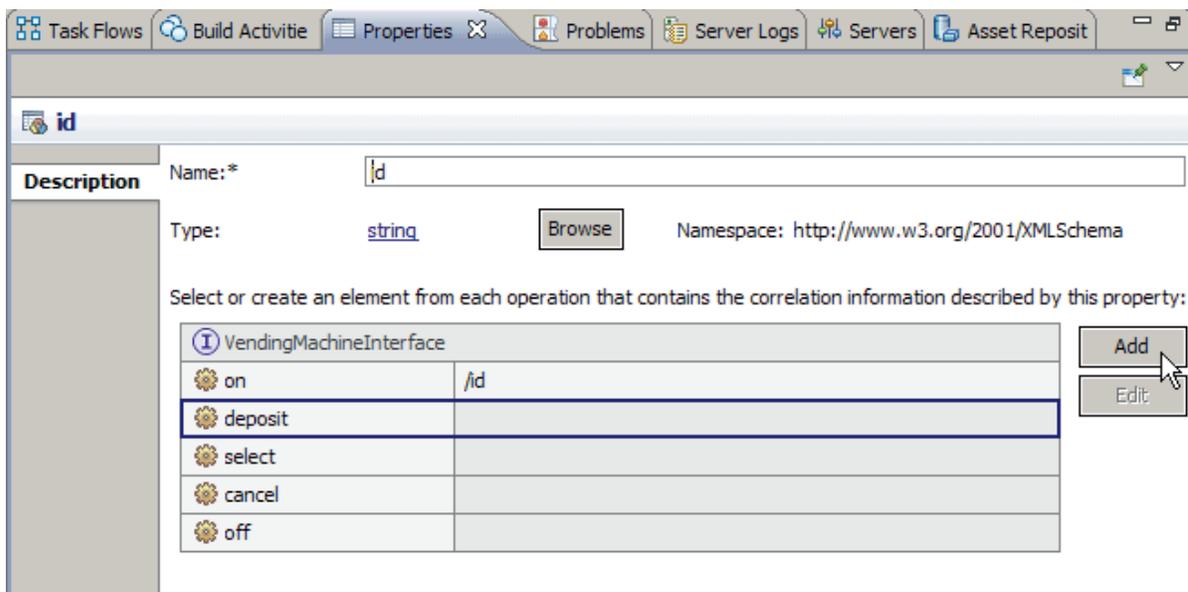


ワークスペース内のエラーの数が赤い「X」で表示されます。実装にまだ組み込んでいない (それまでは **on** のみを使用) インターフェース内のすべての操作にエラーが存在し、すべての操作に相関セットを参照するエラーが 1 つ存在します。次に、相関セットを修正します。

2. 相関を指定します。相関は、ランタイム環境内で、ステート・マシンの特定のインスタンスを別のインスタンスから識別するのに使用されるプロパティを定義します。ステート・マシンが応答する各操作 (イベント) に対して、プロパティの別名は、定義された各相関プロパティに対応する入力を見つけます。
 - a. 「相関プロパティ」 カテゴリーにあるプラス・アイコンをクリックします。



- b. 「相関プロパティの追加」ダイアログで、名前を id に変更します。
- c. 「string」を選択して、「OK」をクリックします。
- d. id 相関プロパティを選択した状態で、「プロパティ」タブをクリックして「プロパティ」ビューを開きます。以下の表に示すように、「プロパティ」ビューの「説明」タブで各操作を順番に選択し、「追加」をクリックしてプロパティ別名を指定します。



XPath Expression Builder で、「Simple XPath の挿入 (Insert Simple XPath)」をクリックし、必要なパスを参照します。選択する項目をダブルクリックし、「OK」をクリックします。

操作	別名
on	/id
deposit	/coin/id
select	/selection/id
取り消し	/id
off	/id

3. 預金されている現在高を保持するための変数を以下のようにして作成します。
 - a. 「変数」カテゴリーにあるプラス・アイコン (+) をクリックします。
 - b. 「変数の追加」ダイアログで、新しい変数に `total` という名前を付けます。
 - c. 型として「倍精度」を選択し、「OK」をクリックします。
4. 「問題」ビューに、ビジネス・ステート・マシンの作成時に作成されたデフォルトの相関プロパティーに関連するエラーがあります。このプロパティーはサンプルで使用されていないため、削除できます。
「相関プロパティー」カテゴリーで、「CorrelationProperty1」をクリックします。削除アイコン (✖) をクリックします。

作業を保存します。この時点で、「問題」ビューで報告されているエラーは 4 つのみです。

Idle ステートの構成

Idle ステートにあるとき、ステート・マシンはただ単に、イベントの到着を (コインの形状で) 待っています。

ステートは、ビジネス・トランザクションでの個別のステージです。ステージは、いずれかの定義されたエントリー・アクションを実行して始まります。次に、ステートはイベントの発生を `listen` し、発生したイベントに適したパスを選択します。エグジット・アクションが存在する場合は、ステート・マシンが次のステートへ遷移する前に、それが実行されます。

このアプリケーションには、4 つのステートがあります。すべてのステート・マシンの開始点である**初期**ステートと、**Idle** および **Depositing** の 2 つのシンプル・ステート、およびステート・マシンが正常に終了する**最終**ステートです。

遷移は、1 つのステートから別のステートに移動するために使用されます。遷移では、その条件を評価して、制御がその遷移を通過すべきかどうかを判別します。制御が遷移を通過する場合、定義されたアクションがあれば、それも実行されます。

Idle ステートを構成するには、以下の手順を実行します。

- 自動生成されたステートの名前変更
 - Idle ステートのエントリーとエグジットの追加
 - アクションを Idle ステートへの遷移に追加
 - Idle ステートを出る遷移に操作を追加
1. **InitialState1** を `Ready` に名前変更します。
 2. 同様に、**FinalState1** を `Off` に、**State1** を `Idle` に名前変更します。
 3. このステートのエントリーを追加します。このエントリー・アクションは、このステートに入った瞬間に実行され、現在ステート・マシンにある変更を返して合計を `nil` にリセットします。

- a. 「Idle」をクリックして、ポップアップしたアクション・メニューから「エントリーの追加」アイコンを選択します。
- b. この新しいエントリーをクリックして、resetTotal に名前変更します。
- c. 「プロパティ」ビューで「詳細」タブをクリックし、次に「Java」ラジオ・ボタンをクリックします。
- d. 次のコードを Java エディターに貼り付けます。

```
System.out.println("Entering the Idle State");
if (total.doubleValue() > 0) {
    System.out.println("VendingMachine returning change..." + total.toString() +
        " has been returned."); total = new Double(0.0d);
}
```

注: これと同じ機能は、Visual Snippet を使用しても実行できます。

4. 同様に、Idle ステートに次のコードを持つエグジット・ステート exitIdle を作成します。

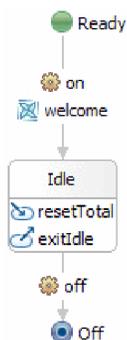
```
System.out.println("Exiting the Idle state");
```

5. 最初の遷移で、入手可能な品目とそのコストをリストするウェルカム・メッセージをユーザーに表示するアクションを作成します。
 - a. on ステートと Idle ステートの間の遷移をクリックします。
 - b. アクション・メニューで「アクションの追加」アイコンをクリックして、Welcome という名前を付けます。
 - c. 次のコードを Java エディターに追加します。

```
System.out.println("VendingMachine is turned on...Prices:");
System.out.println("VENDINGMACHINE: pop: $0.5");
System.out.println("VENDINGMACHINE: chips: $0.75");
System.out.println("VENDINGMACHINE: candy: $1.0");
total = new Double(0.0d);
```

6. 右側のトレイにあるインターフェース **VendingMachineInterface** を展開して、Idle ステートと Off ステートの間の遷移に **off** 操作をドラッグ・アンド・ドロップします。 **off** 操作により、自動販売機がシャットダウンされます。

ステートと遷移は、次の図のようになります。



作業内容を保存します。「問題」ビューで報告されるエラーは 3 つだけであることが分かります。それらのエラーは、まだ使用されていない cancel、deposit、selection の各操作に対応しています。4 番目のエラーは、**off** 操作を正しく追加したので、解決されています。

Depositing ステートの作成

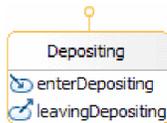
Depositing ステートとは、ユーザーがコインの投入を開始し、まだ選択や取り消しを行っていないステートです。このステートでは、自動販売機に投入された金額を追跡します。

このステップでは以下を行います。

- Depositing ステートの作成
 - Depositing ステートの入り口と出口の追加
 - Depositing ステートでの自己遷移の作成
 - 操作、条件、アクションをこの自己遷移に追加
1. パレットで、「シンプル」アイコンをクリックして Idle ステートの右側のキャンバス上にドロップし、これを Depositing に名前変更します。
 2. Depositing ステートに次のコードを持つ入り口 enterDepositing を作成します。

```
System.out.println("Entering the Depositing state");
```
 3. 同様に、Depositing ステートに次のコードを持つ出口 leavingDepositing を作成します。

```
System.out.println("Leaving the Depositing state");
```
 4. Depositing ステートに自己遷移を作成し、構成します。自己遷移とは、ソース・ステートとターゲット・ステートが同じ遷移です。この場合、Depositing ステートから出る遷移は、後続のコインが投入されるたびに実行されます。このときコインの妥当性が検査され、コインの金額が現在高に加算され、Depositing ステートに戻ります。このステートで、ステート・マシンは次のイベントの発生を待機します。
 - a. 黄色のグラバーが次の図のように表示されるまで、Depositing ステートの上にカーソルを置きます。



- b. マウスを左クリックして遷移の始まりを作成し、次にカーソルを Depositing ステートにドラッグし

戻してクリックします。自己遷移が次の図のように表示されます。



青い線の角でブラック・ボックスをグラブし、遷移が大きく表示されるようにそれをドラッグアウトします。ブラック・ボックスにたくさんの情報を追加します。

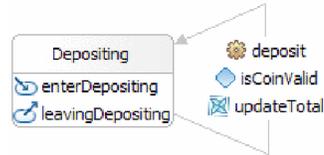
- c. **deposit** 操作をこの遷移にドラッグします。
- d. 次のコードを持つ条件 isCoinValid をこの遷移に追加します。

```
System.out.println("Money is being deposited in VendingMachine. Checking to see if the coin is valid");
double coin = deposit_input_coin.getDouble("value");
if (coin == 1.0d || coin == 2.0d || coin == 0.25d || coin == 0.1d || coin == 0.05d)
    return true;
return false;
```

条件は遷移をガードし、「true」に評価された場合にのみ処理を許可します。それ以外の場合は、現在のステートが維持されます。

- e. 次のコードを持つアクション updateTotal をこの遷移に追加します。

```
double coin = deposit_input_coin.getDouble("value");
double newTotal = total.doubleValue() + coin;
total = new Double(newTotal);
System.out.println("Successfully deposited "+coin+" in the VendingMachine");
```



Depositing ステートとその自己遷移は次の図のようになります。
 ます。次のステップでエラーを解決します。

作業を保存し

2 つのステートをリンク

2 つのステートをリンクする 4 つの遷移があります。これらはそれぞれ異なる機能を提供します。最初の遷移はコイン・イベントを登録し、2 番目は販売を管理し、3 番目はタイムアウトをモニターし、4 番目は取り消しオプションを提供します。

このステップでは、これらの遷移を作成し構成します。

注: このタスクでは、4 つの遷移を作成します。これらの方向はすべて異なります。指示に十分注意して、ターゲット・ステートとソース・ステートを混同しないようにしてください。

1. Idle ステートから Depositing ステートへの遷移を作成します。この遷移は、いつコインが到着したかを認識し、コインの妥当性を検査し、合計を更新し、ステート・マシンを Depositing ステートに移行します。

- a. **deposit** 操作をトレイからこの遷移にドラッグします。
- b. 次のコードを持つ条件 `isCoinValid` をこの遷移に追加します。

```
double coin = deposit_Input_coin.getDouble("value");
if (coin == 1.0d || coin == 2.0d || coin == 0.25d || coin == 0.1d || coin == 0.05d)
    return true;
System.out.println("The coin deposited is not a valid coin.");
return false;
```

- c. 次のコードを持つアクション `updateTotal` をこの遷移に追加します。

```
double coin = deposit_Input_coin.getDouble("value");
double newTotal = total.doubleValue() + coin;
total = new Double(newTotal);
System.out.println("Successfully deposited "+coin+" in the VendingMachine");
```

2. Depositing ステートから Idle ステートへの遷移を作成します。この遷移は、ユーザーが選択を行ったときに実行されます。合計金額が選択された品目を販売するために十分である場合は、該当する販売メッセージを送信し、ステート・マシンを Idle ステートに移行するアクションが起動されます。

- a. **select** 操作をこの遷移にドラッグします。
- b. 次のコードを持つ条件 `enoughCoin` をこの遷移に追加します。

```
String item = select_Input_selection.getString("item");
double totalTemp = total.doubleValue();
if (item.equals("pop")) return totalTemp >= 0.5d;
else if (item.equals("chips")) return totalTemp >= 0.75d;
else if (item.equals("candy")) return totalTemp >= 1.0d;
System.out.println("Incorrect item selected. Please try again.");
return false;
```

- c. 次のコードを持つアクション `dispenseItem` をこの遷移に追加します。

```
String item = select_Input_selection.getString("item");
double totalTemp = total.doubleValue();
if (item.equals("pop"))
{
    totalTemp = totalTemp - 0.5d;
    System.out.println("A pop has been dispensed.");
}
```

```

else if (item.equals("chips"))
{
    totalTemp = totalTemp - 0.75d;
    System.out.println("A bag of chips has been dispensed.");
}
else if (item.equals("candy"))
{
    totalTemp = totalTemp - 1.0d;
    System.out.println("A candy bar has been dispensed.");
}
}
total = new Double(totalTemp);

```

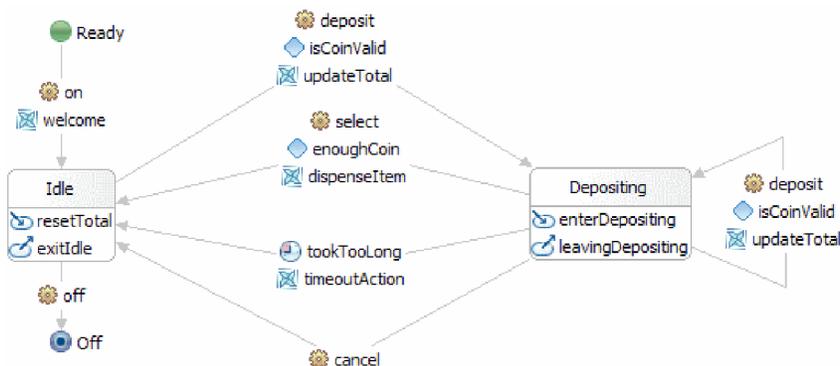
3. Depositing ステートから Idle ステートへの 2 番目の遷移を作成します。この遷移には、経過時間が長すぎた場合にステート・マシンを Idle ステートに移行するタイムアウトがあります。ステート・マシンが Idle ステートに入ると、マシン内のお金は自動的に返され、合計が nil にリセットされます。
 - a. tookTooLong というタイムアウトをこの遷移に追加します。「詳細」で「期間」および「リテラル」ラジオ・ボタンをクリックして、30 を「秒」フィールドに入力します。
 - b. 次のコードを持つアクション timeoutAction をこの遷移に追加します。

```

System.out.println("Took too long: moving to Idle State");

```
4. Depositing ステートから Idle ステートへの遷移を作成します。この遷移は、ユーザーがトランザクションを取り消したときに実行され、マシンを Idle ステートに戻します。前述のように、ステート・マシンが Idle ステートに入ると、マシン内のお金は自動的に返され、合計が nil にリセットされます。
 - a. **cancel** 操作をこの遷移にドラッグします。

完了したステート・マシンは、以下の画面のようになります。



作業を保存します。「問題」ビューは何も表示しなくなり、エラーのない状態になります。

ステート・マシン・コンポーネントの作成

このステップでは、アセンブリ・ダイアグラムにステート・マシン・コンポーネントを作成して、そのテストができるようにします。

以下のようにして、ステート・マシン・コンポーネントを作成します。

1. 「ビジネス・インテグレーション」ビューで「BSM_VendingMachine」 > 「アセンブリ・ダイアグラム」をダブルクリックすることにより、「BSM_VendingMachine」アセンブリ・ダイアグラムを開きます。
2. 「ビジネス・インテグレーション」ビューから、「BSM_VendingMachine」 > 「統合ロジック」 > 「ステート・マシン」 > 「VendingMachine」を選択し、アセンブリ・ダイアグラムのキャンバスまでドラッグします。モジュールを保存します。

第 3 章 インポート

このサンプルの完全な既製バージョンをインポートすることができます。

既製のサンプルをインポートするには、以下を実行します。

1. IBM Integration Designer で、「ヘルプ」>「サンプルおよびチュートリアル」>「**IBM Integration Designer 7.5**」を選択します。「**サンプルおよびチュートリアル**」ページが開きます。
2. 「**ファイルの読み書き**」セクションの下で「**インポート**」リンクをクリックします。「**サンプルのインポート**」ページが開きます。このページを使用して、単にそのサンプルだけをインポートすることも（ワークスペース内にフラット・ファイル・コネクタ・プロジェクト **CWYFF_FlatFile** が既に存在する場合）、フラット・ファイル・アダプター付きのサンプルをインポートすることもできます。該当するオプションを選択し、「**OK**」を選択します。
3. 既製のサンプルがワークベンチにインポートされます。

サンプルを実行する手順は、トピック『**サンプルの実行**』に書かれています。その説明には、インポートしたアプリケーションにアダプターを追加するための別のインポートが含まれています。アダプターのインポートを参照してください。

既製のサンプルを開く

既製のステート・マシン・サンプルのワークベンチへのインポートを完了後、サンプルをステート・マシン・エディターで開いて、その内容を参照することができます。

インポートした既製のサンプルを表示するには、以下の指示に従ってください。

1. 「**ビジネス・インテグレーション**」ビューで、「**BSM_VendingMachine**」>「**統合ロジック**」を展開します。
2. このサンプル用に事前に作成されたステート・マシンを開くには、「**ステート・マシン**」を展開して、「**自動販売機**」をダブルクリックします。 **ビジネス・ステート・マシン・エディター**が起動します。

第 4 章 サンプルの実行

サンプルのビルドまたはインポートが完了したら、サンプルを実行できます。

既製のサンプルをインポートした場合は、ディレクトリーおよび入力ファイルの作成に示すように、ディレクトリーを作成し、入力ファイルを追加する必要があります。また、アダプターのインポートに示すように、WebSphere Adapter for Flat Files もインポートする必要があります。

サンプルを実行するには、以下の手順に従います。

- 必要であれば、このサンプルの PDF 版を印刷します。既にサンプルをインポートしてあり、それを実行したい場合もあります。PDF 版を見ると、サンプルがどのようにして作成されたかが分かります。
- サンプルを自分自身でビルドするのではなく、インポートした場合は、WebSphere Adapter for Flat Files をインポートします。
- モジュールをサーバーにデプロイします。
- モジュールを実行し、テストします。

サンプルのテスト

必要なリソースをインポートした後、Business Process Choreographer (BPC) エクスプローラー Web クライアントまたはテスト・クライアントを使用して、アプリケーションの実行、テスト、およびデバッグができます。以下の手順では、BPC エクスプローラーを使用します。

自動販売機を実行およびテストするには、以下の手順に従ってください。

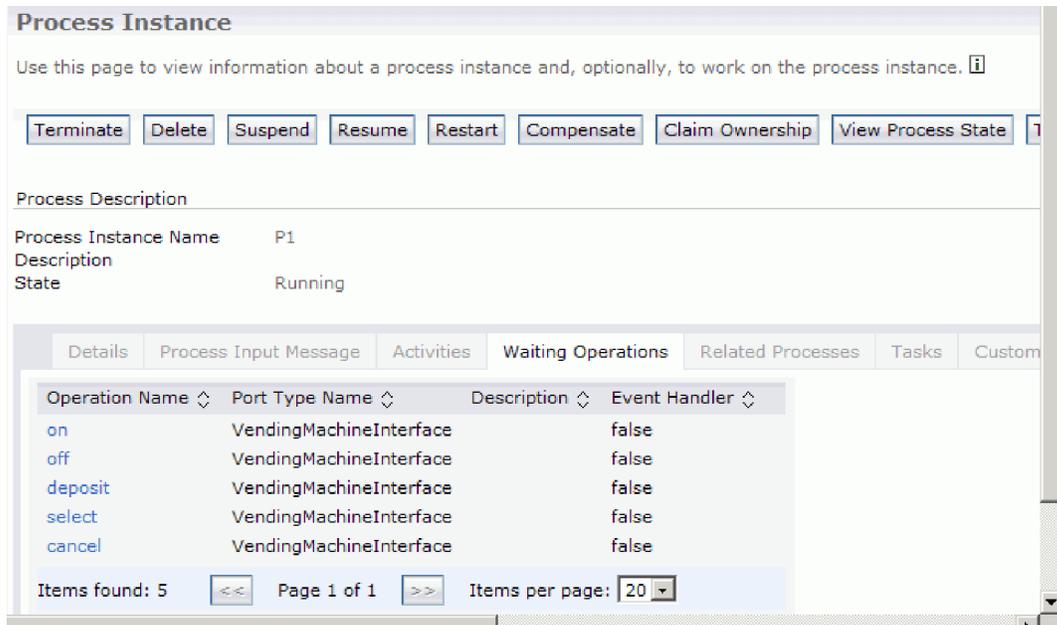
1. 「サーバー」ビューで、「**IBM Process Server v7.5**」を右クリックし、「**開始**」を選択します。サーバーの始動が完了するまで待ってください。
2. アプリケーションをサーバーに追加します。
 - a. ビジネス・インテグレーション・パースペクティブで、「サーバー」ビューに進みます。
 - b. 「IBM Process Server」を右クリックし、リストから「**プロジェクトの追加と削除**」を選択します。「プロジェクトの追加と削除」ウィンドウが開きます。
 - c. ナビゲーション・ツリーで、「**BSM_VendingMachine**」をクリックします。
 - d. 「**追加**」をクリックし、「**完了**」をクリックします。サーバーが始動して自動販売機を公開するのに数分かかります。「サーバー・ログ」ウィンドウに「**アプリケーションが開始されました**」と表示されるまで待ちます。
3. 次のようにして Business Process Choreographer エクスプローラーを起動します。
 - a. 「サーバー」ビューで「**IBM Process Server v7.5**」を右クリックし、「**起動**」 > 「**Business Process Choreographer エクスプローラー**」を選択します。
 - b. ログインのページでは、デフォルトの資格情報は、**UserID: admin** および **password: admin** です。
Business Process Choreographer エクスプローラーが開き、ユーザーに割り当てられているすべてのタスクを表示します。現在使用可能なタスクはないので、「**項目が見つかりません**」というメッセージが表示されます。
4. アプリケーションを呼び出します。

- サーバーへ公開されたすべてのプロセス・テンプレートのリストを表示するには、「ビュー」タブのナビゲーション・ペインで「プロセス・テンプレート」をクリックします。
- 「VendingMachine」テンプレートのチェック・ボックスを選択し、「インスタンスの開始」をクリックして自動販売機のプロセスを開始します。

- 「プロセス入力メッセージ」ページで、**P1** を「プロセス名」として、**1111** を「ID」としてそれぞれ入力し、「サブミット」をクリックします。

- 「VendingMachine」テンプレートのチェック・ボックスをクリックし、「インスタンス」をクリックします。

- 「P1」プロセス・インスタンス名をクリックし、「プロセス・インスタンス」ページに進みます。
- 「待機中の操作」タブをクリックすると、次の画面に示すように、すべての操作がリストされます。



- g. **on** 操作名をクリックして自動販売機を開始します。
 - h. 「**id**」フィールドに「**1111**」が取り込まれていることを確認し「送信」をクリックします。投入と選択のステップに時間がかかりすぎると、自動販売機は投入金額を返却し、Idle ステートに戻ります。
 - i. 「**P1**」プロセスをクリックし、「待機中の操作」タブを選択して「**deposit**」操作名をクリックします。
 - j. **1111** を「**coin id**」として、**1.0** を「**coin value**」として入力し、「**サブミット**」をクリックします。「コンソール」ビューで、**1.0** が自動販売機に投入されたことが分かります。
 - k. 「**P1**」プロセスをクリックし、「待機中の操作」タブを選択して「**select**」操作名をクリックします。
 - l. **1111** を「**selection id**」として、**pop** を「**selection item**」として、それぞれ入力します。「コンソール」ビューで、**pop** が販売され、**0.5** ドルのおつりが返却されたことが分かります。
 - m. 自動販売機を停止するために、「**P1**」プロセスをクリックし、「待機中の操作」タブを選択して「**off**」操作名をクリックし、「**id**」として「**1111**」を入力します。
5. 終了後、エクスプローラーをログアウトし、プロジェクトを除去してサーバーを停止します。

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒242-8502

神奈川県大和市下鶴間1623番14号

日本アイ・ビー・エム株式会社

法務・知的財産

知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

Intellectual Property Dept. for WebSphere Software

IBM Corporation

3600 Steeles Ave. East

Markham, Ontario

Canada L3R 9Z7

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

表示されている IBM の価格は IBM が小売り価格として提示しているもので、現行価格であり、通知なしに変更されるものです。卸価格は、異なる場合があります。

本書はプランニング目的としてのみ記述されています。記述内容は製品が使用可能になる前に変更になる場合があります。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほめかしたり、保証することはできません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。 © Copyright IBM Corp. _年を入れる_. All rights reserved.

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

プログラミング・インターフェース情報

プログラミング・インターフェース情報は (提供されている場合)、プログラムを使用してアプリケーション・ソフトウェアを作成する際に役立ちます。

一般使用プログラミング・インターフェースにより、お客様はこのプログラム・ツール・サービスを含むアプリケーション・ソフトウェアを書くことができます。

ただし、この情報には、診断、修正、および調整情報が含まれている場合があります。診断、修正、調整情報は、お客様のアプリケーション・ソフトウェアのデバッグ支援のために提供されています。

警告: 診断、修正、調整情報は、変更される場合がありますので、プログラミング・インターフェースとしては使用しないでください。

商標

IBM、IBM ロゴおよび ibm.com は、世界の多くの国で登録された International Business Machines Corp. の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、www.ibm.com/legal/copytrade.shtml をご覧ください。

Adobe、Adobe ロゴ、PostScript、PostScript ロゴは、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Intel Centrino、Intel Centrino ロゴ、Celeron、Intel Xeon、Intel SpeedStep、Itanium、Pentium は Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

UNIX は The Open Group の米国およびその他の国における登録商標です。

ご利用条件

これらの資料は、以下の条件に同意していただける場合に限りご使用いただけます。

個人使用: これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、非商業的な個人による使用目的に限り複製することができます。ただし、IBM の明示的な承諾をえずに、これらの資料またはその一部について、二次的著作物を作成したり、配布（頒布、送信を含む）または表示（上映を含む）することはできません。

商業的使用: これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、お客様の企業内に限り、複製、配布、および表示することができます。ただし、IBM の明示的な承諾をえずにこれらの資料の二次的著作物を作成したり、お客様の企業外で資料またはその一部を複製、配布、または表示することはできません。

ここで明示的に許可されているもの以外に、資料や資料内に含まれる情報、データ、ソフトウェア、またはその他の知的所有権に対するいかなる許可、ライセンス、または権利を明示的にも黙示的にも付与するものではありません。

資料の使用が IBM の利益を損なうと判断された場合や、上記の条件が適切に守られていないと判断された場合、IBM はいつでも自らの判断により、ここで与えた許可を撤回できるものとさせていただきます。

お客様がこの情報をダウンロード、輸出、または再輸出する際には、米国のすべての輸出入関連法規を含む、すべての関連法規を遵守するものとします。

IBM は、これらの資料の内容についていかなる保証もしません。これらの資料は、特定物として現存するままの状態を提供され、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任なしで提供されます。

© Copyright IBM Corporation 2005, 2011. All Rights Reserved.