

シミュレーションモデルの作成に関する補足

「企業と市場のシミュレーション」

2003年12月10日

井庭 崇

1 各種ファイルの説明

コンポーネントビルダー (on Eclipse) の Package Explorer 内には、~.behavior、~.world、~.java など、いろいろな拡張子のファイルがあります。ここでは、それぞれのファイルがどのようなものかを説明します。

~.world

World Editor 用の世界 (World) 設定ファイルです。World Editor 用の設定が保存されているだけなので、シミュレーションを実行したい場合には、World Editor でソースコードを生成する必要があります。

このファイルをダブルクリックすることで、World Editor が起動します。シミュレーション作成者は、ここで World の設定を行います。

~.behavior

Behavior Editor 用の行動 (Behavior) の状態遷移図ファイルです。あくまで Behavior Editor 用の設定が保存されているだけなので、シミュレーションに組み込むには、Behavior Editor でソースコードを生成する必要があります。

このファイルをダブルクリックすることで、Behavior Editor が起動します。シミュレーション作成者は、ここに Behavior の状態遷移図を描きます。

~.java

Java プログラムのソースコードです。シミュレーションの作成では、いくつかの種類の Java プログラムを書きます。以下では、主なものを説明します。

~ World.java

World Editor から自動生成されるソースコードです。世界 (World) の初期化のプログラムが書かれています。このファイルは触る必要はありません。

シミュレーションを実行するには、この ~ World.java ファイルを選択して、ツールバーの実行ボタン  の右側の  部分を押し、「Run As...」 「Java Application」を選びます。

~ Model.java

Type Editor から自動生成されるソースコードです。モデル要素の定義が書かれています。このファイルは通常は触る必要はありません (カスタマイズする必要がある場合には、『社会シミュレーション デザイナーズガイド』の p.185 を参照してください)。

モデル要素の定義を設定する場合には、このファイルを選択し、Type ボタン  を押すと、Type Editor が起動します。シミュレーション作成者は、この Type Editor 上でモデル要素の定義を行い、ソースコードを生成します。

Abstract ~ Behavior.java

Behavior Editor から自動生成されるソースコードで、Behavior の状態遷移のメカニズムの部分が書かれています。このファイルは触らないでください。

~ Behavior.java

Behavior Editor から自動生成されるソースコードです。シミュレーション作成者は、ここに Behavior の具体的な action などを記述します。

~ .wml

World Editor 用から生成されるデータファイルで、今のところ使用しません (今後、世界設定を表示するツール等が開発されれば、このデータファイルを読み込むことになるでしょう)。

~ .class

Java プログラムを実行するためのバイナリコード (つまり実行用ファイル) です。通常、Eclipse の Package Explorer では、隠れていて見えません。背後では、~.java ファイルを保存するたびに自動的にコンパイルされ、この~.class ファイルが作成されています。

一歩先へのメモ (1) 送られてきたものによって遷移を変える

このメモは、余裕のある人だけ読んでください。

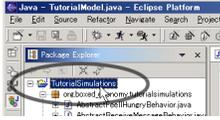
Behavior の状態遷移では、送られてきたものを判別して遷移させたりすることができます。そのとき、遷移ごとに条件分けを記述します。UML での正式名称では「ガード条件」(guard condition) といいます。その条件が満たされないと、遷移がなされないで、その条件がガード (監視・防護) しているわけです。やり方は、『社会シミュレーション デザイナーズガイド』の p.296 のサンプルを参照してください。

なお、第 21 章「モデル・パターンカタログ」のサンプルコードでは、World を自分で作成していますが、World Editor が開発された現在では、自分でソースコードを書く必要がなくなりました。図やソースコードのコメントを参考に、World Editor で設定してください。

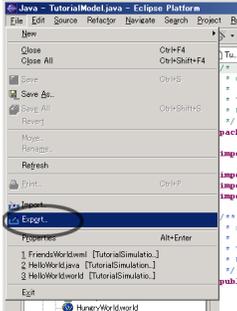
2 プロジェクトをエクスポートする (宿題の提出用)

ここでは、TutorialSimulations プロジェクトを、圧縮ファイル (Zip file) として出力する方法について説明します。

1. まず最初に、Package Explorer 内で、出力したいプロジェクト (TutorialSimulation プロジェクト) を選択します。



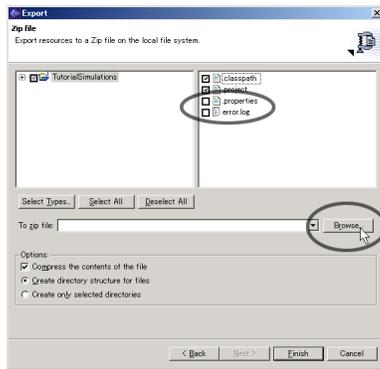
2. 次に、「File」メニューの中から、「Export」を選択します。



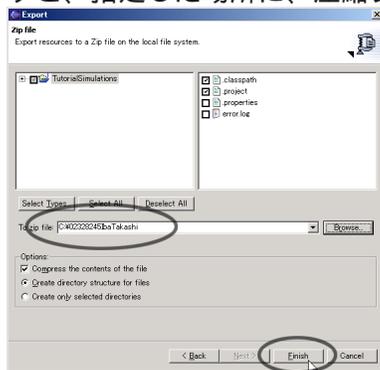
3. 出てきた Export ウィンドウで、「Zip file」を選択し、「Next >」ボタンを押します。



4. 右上にある「. properties」と「error.log」のチェックボックスのチェックを外す
5. 「Browse」ボタンを押して、出てきたウィンドウで、フォルダを選択し (ここでは、C ドライブを選択)、ファイル名を入力する。ファイル名は、自分の学籍番号と名前を半角英数字でつけ、「保存」ボタンを押す
フォルダ名などに日本語が入っていると失敗するようです。



6. 「To Zip file:」の後に先ほどの名前が入っていることを確認し、「Finish」ボタンを押すと、指定した場所に、圧縮ファイルが作成される。



今回の宿題では、以下のものを提出してもらいます。

- 圧縮ファイル (添付)
- シミュレーションの簡単な説明
- フィードバックコメント (FC09)

一歩先へのメモ (2) 不確実性を導入する

このメモは、余裕のある人だけ読んでください。

これまで作ってきたモデルでは、毎回実行するたびに、決まりきったことしか起こりません。もちろん、シミュレーションに不確実性(ランダムな要素)を入れることもできます。ランダムな値を生成するには、乱数ジェネレーターを使います。『社会シミュレーション デザイナーズガイド』の p.278 「乱数ジェネレータの追加/参照」に説明が載っているので、読んでみてください。RandomNumberGenerator の名前は、World Editor で設定します (p.192 参照)。実行時にランダム系列を変えるには、BESP の WorldInitializer でランダムシード (乱数の種) の値を変えます。

サンプルは、『社会シミュレーション デザイナーズガイド』の p.88 の BuyBreadBehavior の「//送るお金の額をランダム (1 ~ 10) に決定します」の部分を参考にしてみてください。例えば、このようなランダムな値を生成することで、if 文で分岐させることができます。