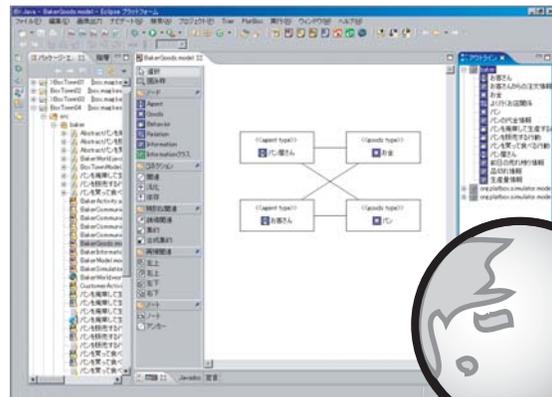


Designers' Guide to Social Simulations, No.2

Component Builder マニュアル

- Chap.1 Component Builder とは
- Chap.2 ソフトウェア更新とモデル作成の準備
- Chap.3 Model Designer
- Chap.4 Activity Designer
- Chap.5 Communication Designer
- Chap.6 Behavior Designer
- Chap.7 Action Designer
- Chap.8 World Composer
- Chap.9 その他の補助ツールを使う
- Chap.10 作成したモデルを公開する

PlatBox Project



目次

第 1 章	Component Builder とは	1
1.1	モデル作成環境「Component Builder」	1
1.2	Component Builder を構成するツール群	1
1.2.1	概念モデリングのためのツール	1
1.2.2	シミュレーションデザインのためのツール	2
1.2.3	その他の補助ツール	2
1.3	Component Builder の画面構成	2
1.3.1	メニューバー	2
1.3.2	ツールバー	3
	Eclipse 基本操作ツールバー	3
	Component Builder ツールバー	3
1.3.3	ビュー	4
	パッケージエクスプローラービュー	4
	エディタビュー	4
	アウトラインビュー	4
1.4	Component Builder のファイル構成	5
1.4.1	各ツールに対応するファイル	5
1.4.2	モデル要素の情報を格納するファイル (modeldb.xml)	5
1.4.3	アクションパーツの情報を格納するファイル (actionparts.xml, .ap_ history)	5
第 2 章	ソフトウェア更新とモデル作成の準備	7
2.1	Component Builder の最新バージョンへの更新	7
2.1.1	初めて更新を行う場合	7
2.1.2	2 回目以降の更新の場合	10
2.2	パースペクティブを設定する	11
2.3	PlatBox プロジェクトを作成する	11
2.4	パッケージを作成する	12
第 3 章	Model Designer	15

3.1	Model Designer とは	15
3.2	Model Designeraaaaaa	15
3.2.1	クラス図を新規作成する	15
3.2.2	すでに作成したクラス図を開く	16
3.3	クラスを配置する	16
3.3.1	クラスを新規作成する	16
	ノード	16
3.3.2	他の図で作成したクラスを図に配置する	16
3.3.3	クラスを編集する	17
	クラス名の編集	17
3.3.4	クラスを削除する	18
3.3.5	クラスを図から削除する	18
3.4	関連・汎化を引く	19
3.4.1	関連・汎化を新規作成する	19
	コネクション	19
3.4.2	関連を編集する	20
	関連のプロパティ ダイアログ	20
	ロール名の編集	21
3.4.3	関連・汎化を削除する	21
3.5	図の配置/表示を調整する	21
3.5.1	移動	21
3.5.2	リサイズ	21
3.5.3	配置の調整	21
3.5.4	コネクションタイプ	22
	コネクションタイプ	22
	通常タイプの線	22
	マンハッタンタイプの線	22
3.6	ソースコードを生成する	22
第 4 章	Activity Designer	25
4.1	Activity Designer とは	25
4.2	Activity Designer を起動する	25
4.2.1	アクティビティ図を新規作成する	25
4.2.2	すでに作成したアクティビティ図を開く	26
4.3	アクティビティ図の要素を配置する	26
4.3.1	アクティビティ図の要素を新規作成する	26
	ノード	26
4.3.2	アクティビティ図の要素を削除する	26

4.4	遷移の線を引く	26
4.4.1	遷移の線を新規作成する	26
	コネクション	27
4.4.2	遷移の線を編集する	27
	遷移のプロパティダイアログ	27
	ガード条件の編集	27
4.4.3	遷移の線を削除する	27
4.5	図の配置/表示を調整する	27
第5章	Communication Designer	29
5.1	Communication Designer とは	29
5.2	Communication Designer を起動する	30
5.2.1	コミュニケーション・シーケンス図を新規作成する	30
5.2.2	すでに作成したコミュニケーション・シーケンス図を開く	30
5.3	コミュニケーション・シーケンス図の要素を配置する	30
5.3.1	コミュニケーション・シーケンス図の要素を新規作成する	30
	ノード	30
5.3.2	コミュニケーション・シーケンス図の要素を削除する	31
5.4	メッセージの線を引く	31
5.4.1	メッセージの線を新規作成する	31
	コネクション	31
5.4.2	メッセージの線を編集する	31
	メッセージのプロパティダイアログ	31
	メッセージの内容の編集	31
	メッセージタイプの編集	32
5.4.3	メッセージの線を削除する	32
5.5	図の配置/表示を調整する	32
5.5.1	メッセージの位置の移動	32
5.5.2	自分へのメッセージの移動	32
5.5.3	リサイズ	32
5.5.4	配置の調整	32
第6章	Behavior Designer	33
6.1	Behavior Designer とは	33
6.2	Behavior Designer を起動する	33
6.2.1	状態遷移図を新規作成する	33
6.2.2	すでに作成した状態遷移図を開く	34
6.3	状態遷移図の要素を配置する	34

6.3.1	状態遷移図の要素を新規作成する	34
6.3.2	状態の編集	34
	特別な状態ダイアログ	34
	状態名の編集	35
	アクションの新規作成	35
	アクションの登録	36
	アクションの登録の解除	36
	アクションの移動	36
	アクションの編集	36
	テンプレートの利用	37
6.3.3	状態遷移図の要素を削除する	37
6.4	遷移を引く	37
6.4.1	遷移を新規作成する	37
6.4.2	遷移を編集する	38
	遷移のプロパティダイアログ	38
	遷移を起こすイベントの設定	39
	ガード条件の設定	39
	アクションの設定	39
	ガード条件の管理	39
6.4.3	遷移を削除する	39
6.5	図の配置/表示を調整する	40
6.6	Behavior の実装	40
6.6.1	Behavior の実装	40
	ソースコードの出力	40
	アクションブロック図の生成	40
	命名規則について	40
第 7 章	Action Designer	43
7.1	Action Designer とは	43
7.2	Action Designer を起動する	44
7.2.1	アクションブロック図を新規作成する	44
7.2.2	すでに作成したアクションブロック図を開く	44
7.3	構造を配置する	44
7.3.1	ブロック	45
7.3.2	条件分岐	45
7.3.3	集合操作	47
7.3.4	回数繰り返し	48
7.3.5	条件付繰り返し	49

7.3.6	メソッド	50
7.4	文を配置する	52
7.4.1	アクションパーツ	52
	(1) アクションパーツを配置したい場所を選択	52
	(2) アクションパーツの選択	52
	(2) アクションパーツの選択	53
7.4.2	操作の呼び出し	53
7.4.3	変数宣言/代入	54
7.4.4	四則演算	55
7.4.5	論理演算	56
7.4.6	文字列	57
7.4.7	return 文	58
7.4.8	continue 文	58
7.4.9	break 文	59
7.5	アウトラインビューで構造を把握する	59
7.6	ソースコードを生成する	59
7.7	型の再読み込み	59
7.8	独自のアクションパーツの作成と管理	59
7.8.1	アクションパーツの作成	60
7.8.2	アクションパーツの管理	62
第 8 章	World Composer	65
8.1	World ファイルを新規作成する	65
8.2	World Composer を起動する	65
8.2.1	World を新規作成する	65
8.2.2	すでに作成した World を開く	66
8.3	World を設定する	66
8.3.1	モデルを読み込む	66
8.3.2	World の基本設定	66
8.3.3	モデル全体が共有するパラメータの追加	67
8.3.4	Agent Group の追加	67
8.3.5	Agent Group 間の関係の追加	69
8.3.6	Relation パターン別の設定	70
	Random パターンで関係を結ぶ	70
	Ring パターンで関係を結ぶ	70
	Small World パターンで関係を結ぶ	71
	Scale Free パターンで関係を結ぶ	71
8.3.7	Agent の数と Goods の量の設定	71

	固定値を入力する	71
	乱数ジェネレータを用いてランダムな値を入力する	71
	世界パラメータの参照する	71
8.4	作成した World を出力する	71
8.5	シミュレーションの実行	72
8.5.1	シミュレーションの起動 (Component Builder 上での操作)	72
8.5.2	シミュレーションの実行 (PlatBox Simulator 上での操作)	72
8.5.3	PlatBox Simulator の終了	72
第 9 章	その他の補助ツールを使う	73
9.1	Boxed Economy プロジェクトからの移行を行う	73
9.2	Type の名前を変更する	74
9.3	Template Action/GuardCondition を作成する	74
9.3.1	Template Action の新規作成	74
9.3.2	Template Guard Condition の新規作成	75
9.3.3	Behavior Designer でテンプレートを呼び出す	75
	テンプレートアクション	75
	テンプレートガード条件	75
第 10 章	作成したモデルを公開する	77
10.1	図を印刷する	77
10.2	図を画像ファイルとして保存する	77
10.3	作成中のモデルを共有する	77
10.3.1	モデルを ZIP 形式でエクスポートする	77
10.3.2	圧縮されているモデルをインポートする	78
10.4	完成したモデルを配布する	78
10.4.1	モデルを JAR 形式でエクスポートする	78
10.4.2	PlatBox Simulator で実行する	78

第1章

Component Builder とは

1.1 モデル作成環境「Component Builder」

Component Builder は、PlatBox Simulator 上で動作するモデルの作成を支援するための環境です。Component Builder は 6 種類のツールで構成されており、概念モデリングフェーズ、およびシミュレーションデザインフェーズにおけるモデル作成を支援する機能を提供しています。

1.2 Component Builder を構成するツール群

1.2.1 概念モデリングのためのツール

Component Builder には、概念モデリングフェーズを支援する 3 つのツールが含まれています。

- Model Designer
- Activity Designer
- Communication Designer

概念モデリングフェーズでは、これらのツールを利用して、反復的にモデル作成を行います。

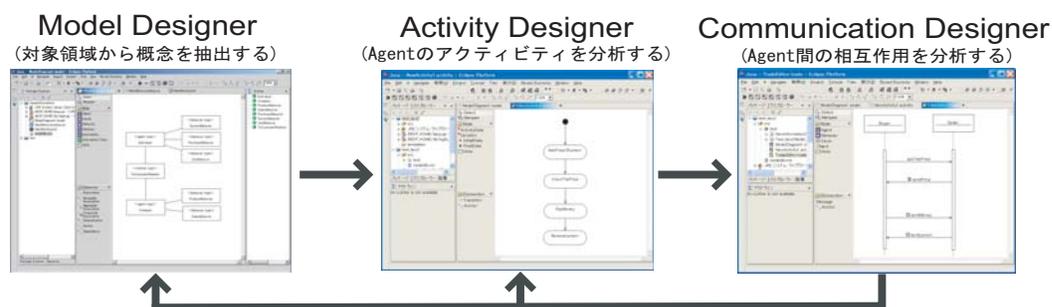


図 1.1: 概念モデリングフェーズを支援するツール

1.2.2 シミュレーションデザインのためのツール

Component Builder には、シミュレーションデザインフェーズを支援する 4 つのツールが含まれています*¹。

- Model Designer
- Behavior Designer
- Action Designer
- World Composer

シミュレーションデザインフェーズでは、これらのツールを利用して、反復的にモデル作成を行います。

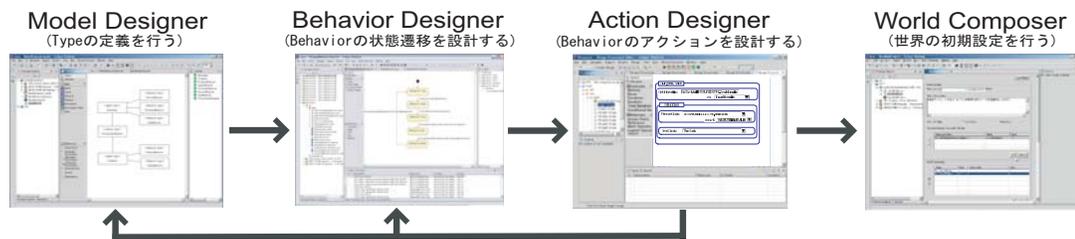


図 1.2: シミュレーションデザインフェーズを支援するツール

1.2.3 その他の補助ツール

Component Builder には、モデル作成を支援するツール以外にも、モデル作成に関連する補助ツールが含まれています。

- Boxed Economy プロジェクトからの移行を行う
- Type の名前を変更する
- Template Action/GuardCondition を作成する

1.3 Component Builder の画面構成

1.3.1 メニューバー

Component Builder の操作は全て、メニューバーから行うことができます。ここでは、主に使用するメニューを説明します。

*¹ Model Designer は、概念モデリングフェーズとシミュレーションデザインフェーズの両方で使用します。

メニュー項目	説明
[ファイル]	プロジェクトやファイル、フォルダの新規作成、移動、印刷等が行えます。
[編集]	ファイルやフォルダの編集が行えます。
[PlatBox]	PlatBox プロジェクトファイルや作画ファイルを新規に作成します。
[ウインドウ]	表示するビューを変更したり、Eclipse の詳細設定を行うことができます。

1.3.2 ツールバー

よく使用する操作については、ツールバーにも、対応するボタンが配置されています。ここでは、主なツールバーを説明します。

Eclipse 基本操作ツールバー

Eclipse 基本操作ツールバーは、Eclipse 本体で提供されているもので、ファイルの保存や印刷などの基本操作を行うことができます。Eclipse 基本操作ツールバーには、以下のようなボタンがあります。

アイコン	ボタン名
	プロジェクト作成ボタン
	保存ボタン
	名前をつけて保存ボタン
	印刷ボタン

Component Builder ツールバー

Component Builder ツールバーは、Component Builder が提供するもので、プロジェクトやファイルの作成等を行うことができます。Component Builder ツールバーには、以下のようなボタンがあります。

アイコン	ボタン名	解説
	PlatBox プロジェクト作成ボタン	p. 11
	Model Designer ファイル作成ボタン	p. 15
	Activity Designer ファイル作成ボタン	p. 25
	Communication Designer ファイル作成ボタン	p. 29
	Behavior Designer ファイル作成ボタン	p. 33
	World Composer ファイル作成ボタン	p. 65
	テンプレートアクション作成ボタン	p. 74
	テンプレートガード作成ボタン	p. 74

1.3.3 ビュー

Component Builder を利用する場合、主に使用するビューは、「パッケージエクスプローラービュー」、「エディタビュー」、「アウトラインビュー」の3つがあります。

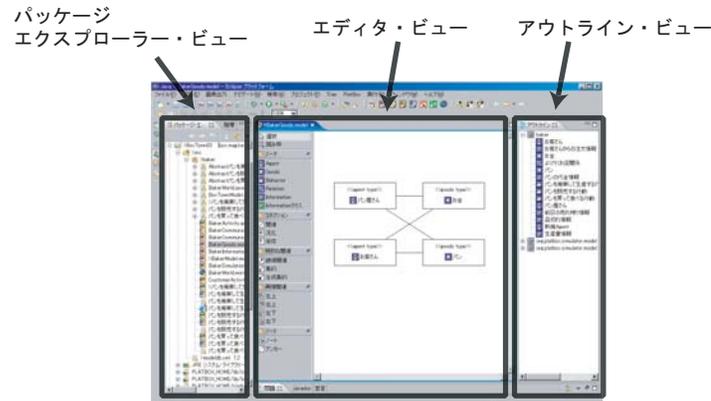


図 1.3: 3つのビュー

パッケージエクスプローラービュー

Eclipse 上でファイルやディレクトリを編集するためには、プロジェクトを作成しなければなりません。パッケージエクスプローラービューでは、作成した各プロジェクトのファイルやディレクトリ構成を表示します。パッケージエクスプローラービューから、ファイルやディレクトリの新規作成や削除、名前の変更、ファイルの移動などが行えます。

エディタビュー

エディタビューでは、ファイルの内容を閲覧したり編集するためのビューです。Component Builder では、クラス図やアクティビティ図が表示され、エディタビューで編集できます。

アウトラインビュー

アウトラインビューでは、エディタビューで開かれているファイルに対する要素を全体的に閲覧するためのものです。開かれているファイルによってはアウトラインビューは使用しません。Component Builder では、Model Designer , Behavior Designer , Action Designer がアウトラインを使用します。

1.4 Component Builder のファイル構成

1.4.1 各ツールに対応するファイル

表示情報は、作画するエディタに対応して以下の拡張子を持ったファイルに保存されます。これらのファイルを削除すると表示情報は削除されますが、図のモデルが削除されることはありません。

エディタ名	ファイル名 (拡張子)
Model Designer	~.model
Communication Designer	~.communication
Activity Designer	~.activity
Behavior Designer	~.behavior
Action Designer	~.action

1.4.2 モデル要素の情報を格納するファイル (modeldb.xml)

作成された全てのモデルの情報は、「modeldb.xml」というファイルに保存されます。modeldb.xml は、図を初めて保存するときに作成されます。同じプロジェクト内で作成した図のモデルの情報は全て、modeldb.xml に保存されるので、このファイルは決して削除しないで下さい。

1.4.3 アクションパーツの情報を格納するファイル (actionparts.xml, .ap_history)

自分で作成したアクションパーツは、「actionparts.xml」として保存されます。他人の作成したアクションパーツを共有したいときは、actionparts.xml を “actionparts” を含む別のファイル名にし、自分の actionparts.xml のあるフォルダにいます。そうすると、自動的にアクションパーツを読み込み、使用することができます。

アクションパーツの使用履歴は、「.ap_history」に保存されます。このファイルは、使用履歴を保存してあるだけなので、万が一消えてしまっても問題ありません。

第2章

ソフトウェア更新とモデル作成の準備

2.1 Component Builder の最新バージョンへの更新

Component Builder は、メニューから、最新バージョンへの更新を行うことができます。以下の手順に従って、こまめに更新するようにしましょう。

2.1.1 初めて更新を行う場合

1. Component Builder を起動して、メニューの「ヘルプ」から、「ソフトウェア更新」>「検索とインストール」を選択します。

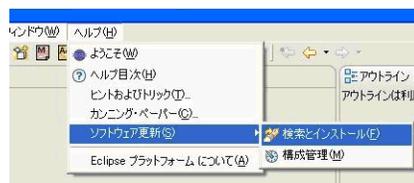


図 2.1: 「ソフトウェア更新」>「検索とインストール」の選択

2. ウィンドウが開いたら、「インストールする新しいフィーチャーを検索」を選択し、「次へ」を押します。

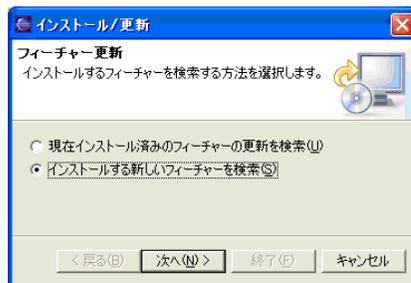


図 2.2: 「インストールする新しいフィーチャーを検索」の選択

3. ウィンドウ右にある「新規リモート・サイト (T)」を押します。

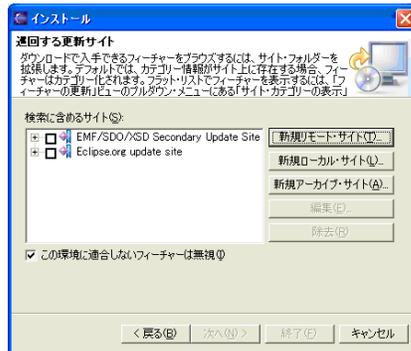


図 2.3: 「新規リモート・サイト」の選択

4. 情報更新サイトウィンドウが開いたら、「名前」欄に「ComponentBuilder」、「URL」欄に「http://box.mag.keio.ac.jp/CBUpdateSite/」と入力し、「OK」を押します。

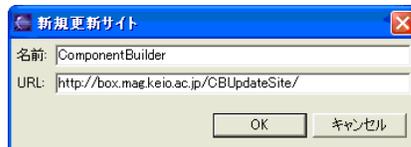


図 2.4: サイト名と URL の入力

5. 「ComponentBuilder」が選択肢に現れるので、チェックボックスにチェックを入れ「次へ」を押します。

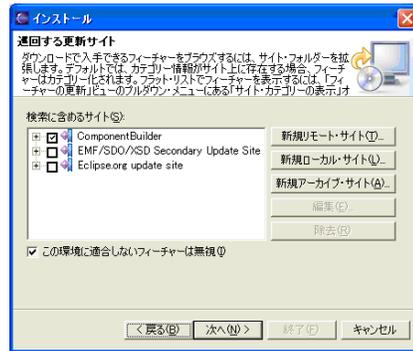


図 2.5: 検索に含めるサイト「ComponentBuilder」の選択

6. インストールすべき最新のバージョンがある場合には、リストに「ComponentBuilder」と表示されるので、それをチェックして、「次へ」を押します。初回は必ず出てくるはずですが。

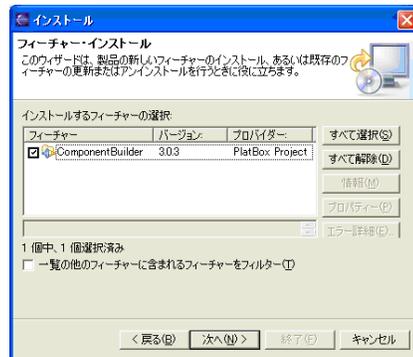


図 2.6: フィーチャー「ComponentBuilder」の選択

7. ライセンスについて表示されるので、「使用条件の条項に同意します」をチェックし、「次へ」を押します。使用条件の条項に同意しない場合には、インストールできません。

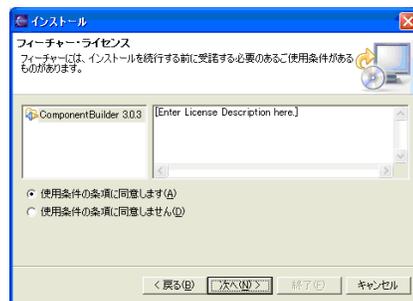


図 2.7: 「使用条件の条項に同意します」の選択

8. インストール・ロケーションのウィンドウが開いたら、「終了」を選択してください。
9. 「警告: 署名なしのフィーチャーをインストールしようとしています。」と警告が出ます。今回に関しては心配ありませんので、「インストール」を選択してください。
10. インストールに成功すると、以下のような質問がされます。「はい」を選択し、ComponentBuilder を再起動させます。



図 2.8: 再起動をするかの質問

2.1.2 2 回目以降の更新の場合

1. ComponentBuilder を起動して、メニューの「ヘルプ」から「ソフトウェア更新」>「検索とインストール」を選択します。

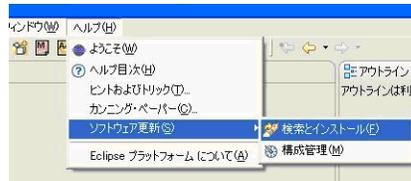


図 2.9: 「ソフトウェア更新」>「検索とインストール」

2. ウィンドウが開いたら、「現在インストール済みのフィーチャーの更新を検索」>「次へ」。

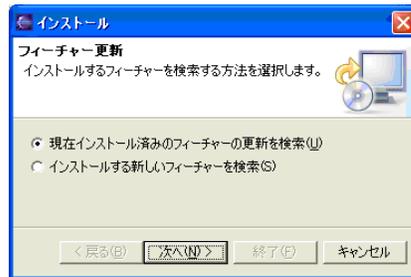


図 2.10: 「現在インストール済みのフィーチャーの更新を検索」の選択

3. これ以降は、初めて更新を行う場合と同様、表示にしたがって更新を行ってください。

2.2 パースペクティブを設定する

作業を始めるにあたり、パッケージエクスプローラーでどのような情報を表示するのかが設定しておきます。パースペクティブの設定は、画面左側のパースペクティブ設定のボタンをクリックして行います。「Java パースペクティブ」を選択します。

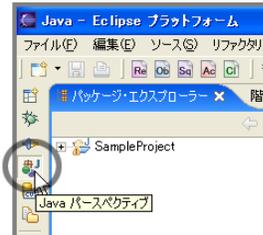


図 2.11: Java パースペクティブボタンによる設定

「Java パースペクティブ」ボタンが隠れていて、見当たらないことがあります。その場合には、「パースペクティブを開く」ボタンをクリックし、そのなかから「Java」を選択します。もしくは、メニューバーから「ウィンドウ」>「ビューの表示」>「その他」を選び、「Java」を選択して [OK] を押します。



図 2.12: Java パースペクティブボタンがない場合の設定方法

2.3 PlatBox プロジェクトを作成する

モデルを作成するための作業領域を、「プロジェクト」といいます*1。PlatBox プロジェクトの作成方法は以下のとおりです。

1. ツールバーの [PlatBox プロジェクト作成] ボタンをクリックします。
2. PlatBox プロジェクト作成ウィザードが開くので、プロジェクト名(例: BoxTown)を入力し、[OK] を押します。

*1 PlatBox プロジェクトを作成すると、PLATBOX_HOME(PlatBox Simulator へのパス) の設定を元に、Component Builder を動かすために必要なライブラリへのクラスパスが自動で設定されます。そのため、通常の Java プロジェクトではなく、PlatBox プロジェクトのボタンで作成するようにしてください。



図 2.13: PlatBox プロジェクト作成ボタン

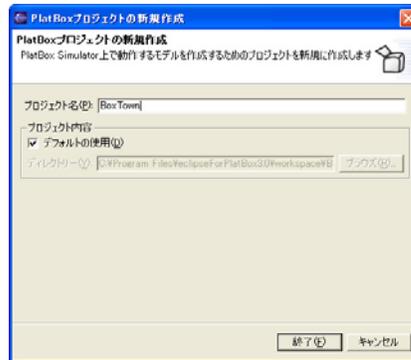


図 2.14: PlatBox プロジェクト作成ウィザード

もしくは、[ファイル] メニューから [新規]>[プロジェクト]>[PlatBox Project] を選択して作成することもできます。プロジェクトが正しく作成されたかどうか、パッケージエクスプローラー上で確認しましょう。



図 2.15: 作成されたプロジェクト

プロジェクトの内容は、プロジェクトの左にある **[+]** をクリックすると、表示することができます。



図 2.16: プロジェクトの内容の表示

2.4 パッケージを作成する

プロジェクトが作成できたら、次に「パッケージ」を作成する必要があります。パッケージはフォルダに相当するもので、この中に描いたクラス図等を保存したり、ソース

コードを生成したりします。パッケージの作成は、以下の手順で行います。

1. 左側にあるパッケージ・エクスプローラーの中にある、対象となるプロジェクトをクリックします。
2. 「新規 Java パッケージ」ボタンをクリックします。

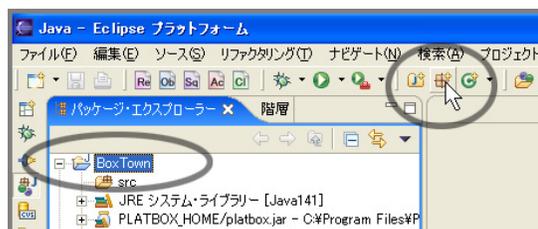


図 2.17: プロジェクトの選択と「新規 Java パッケージ」ボタン

3. 新しいパッケージを作成するためのウィザードが開くので、「名前:」のところにパッケージ名(例: baker)を入力し、[終了]ボタンを押します。

パッケージ・エクスプローラー上で、パッケージが作成されたか確認しましょう。



図 2.18: 作成されたパッケージ

パッケージの作成は、メニューバーの [ファイル]>[新規]>[パッケージ] から行うこともできます。

第3章

Model Designer

3.1 Model Designer とは

記述したモデルから、PlatBox Simulator 上で動作するコンポーネントのソースコードを生成することができます。

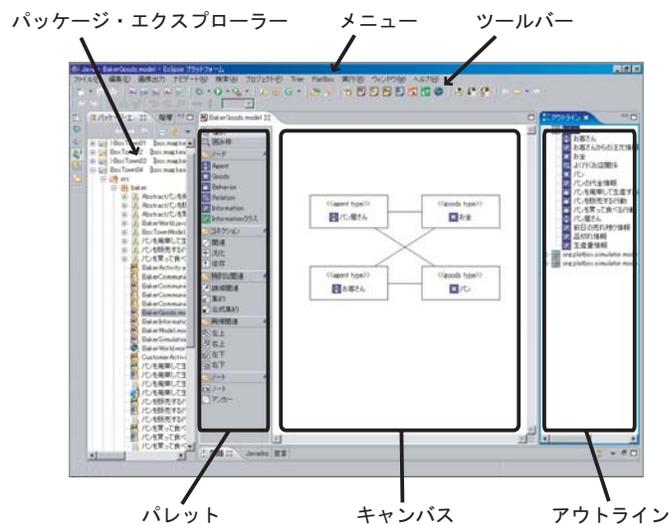


図 3.1: Model Designer

3.2 Model Designeraaaaaa

3.2.1 クラス図を新規作成する

クラス図ファイルの作成は、次の手順で行います。

1. パッケージ・エクスプローラーから、ファイルを新規作成したいプロジェクトの src フォルダにある パッケージ を選択します。
2. ツールバーから、クラス図ファイル作成ボタン  をクリックします。
3. ファイルを保存する場所が、先ほど選択した場所であることを確認して、「ファイル名」欄にファイル名を入力し、[終了] を押します。

もしくは、メニューバーの [PlatBox]>[Create Model Diagram] を選択して作成することもできます。

3.2.2 すでに作成したクラス図を開く

すでに作成したクラス図を開きたい場合は、パッケージ・エクスプローラー上で、モデルファイルをダブルクリックします。もしくは、右クリックして [アプリケーションから開く]-[PlatBox Model Designer] を選択します。モデルファイルを読み込んだ Model Designer が起動して、クラス図が表示されます。

3.3 クラスを配置する

3.3.1 クラスを新規作成する

クラスを新規作成するには、Model Designer のパレットにある「クラス」のボタンを選択し、続けてキャンバスをクリックします。Model Designer で作画できるクラスは、以下のとおりです。

ノード

要素のアイコン	要素の説明
	AgentType (クラス)
	GoodsType (クラス)
	BehaviorType (クラス)
	RelationType (クラス)
	InformationType (クラス)
	Information Class Type (クラス)

3.3.2 他の図で作成したクラスを図に配置する

モデルのクラス図は、1つの図に収める必要はなく、通常、複数の図に分けて記述していきます。例えば、「Agent とそれらをつなぐ Relation の関連を表すクラス図」や、「Agent と所有する Goods の関連を表すクラス図」というように、ある特定の視点からモデルを捉え、表現するわけです。このとき、すでに定義している Agent やその他のクラスは、再度定義する必要はありません。

作成したクラスの一覧は、アウトラインに表示されます。すでに作成してあるクラスを他の図で使いたい場合は、このアウトラインからキャンバスまで、そのクラスをドラッグ&ドロップして配置します。

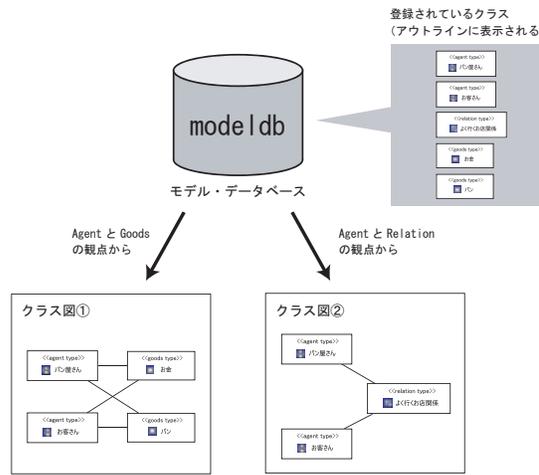


図 3.2: クラス図とモデルデータベースとの関係

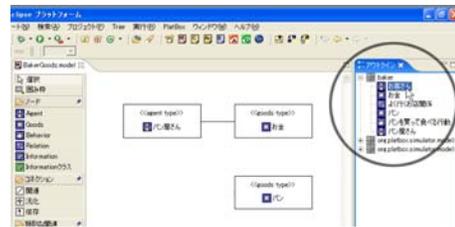


図 3.3: アウトラインビューに表示されたクラス

3.3.3 クラスを編集する

推奨プロセスでは、使用しません。

クラス名の編集

クラス名は、キャンバス上でクラスをダブルクリックすることで編集できます。これから説明する“クラスのプロパティ”ダイアログでは、クラス名も含めて、クラスのいろいろな編集が可能です。

クラスの編集は、“クラスのプロパティ”ダイアログから行います。ダイアログを開くには、編集したいクラスを右クリックして、表示されるメニューから [プロパティ] を選択します。

“クラスのプロパティ”ダイアログでは、以下のような編集が出来ます。

- クラス名の編集
- クラスの可視性の編集
- クラスの修飾子の編集
- クラスの属性の編集
- クラスのメソッドの編集

ダイアログで編集すると、リアルタイムにキャンバスに編集内容が反映されます。[OK]



図 3.4: クラスのプロパティダイアログ

を押すと、確定しダイアログを閉じます。[キャンセル]を押すと、ダイアログを聞く以前の状態に戻ります。

3.3.4 クラスを削除する

クラスを削除するには、削除したい要素を右クリックして、表示されるメニューから [モデル要素の削除] を選択します。この [モデル要素の削除] を行うと、クラス図だけでなく、モデル (modeldb.xml) からそのクラスが削除されます。

3.3.5 クラスを図から削除する

クラスを 図 から削除するには、削除したい要素を右クリックして表示されるメニューから [削除] を選択します。また、削除したい要素を選択して、キーボードの **Delete** キーを押しても削除できます。

ただし、これはキャンバスの図から削除しただけであり、モデル (modeldb.xml) から削除されていません。クラスを完全に削除したい場合には、上述の「クラスを削除する」(モデル要素の削除)を参照してください。

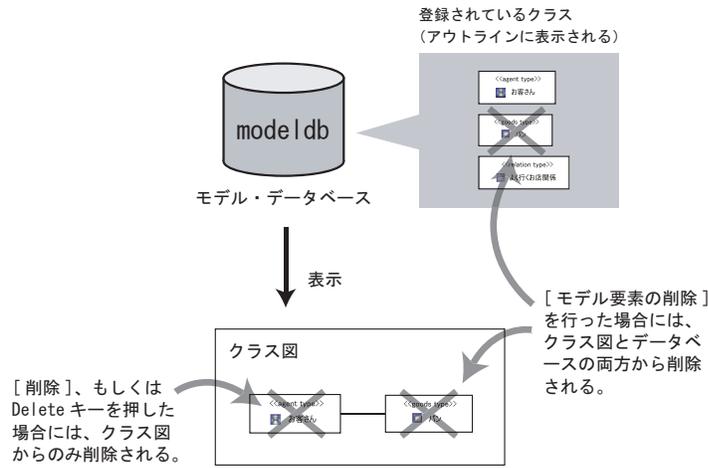


図 3.5: [削除] と [モデル要素の削除] の違い

3.4 関連・汎化を引く

3.4.1 関連・汎化を新規作成する

作成したクラス間に、関連等のコネクションを引きたい場合には、以下の手順で行います。

1. Model Designer パレットから [関連] を選択する。
2. 関連の線を引きたいノードをクリックする。
3. もう一方のノードをクリックする。

コネクションが 1 つ引かれた後も、コネクションを続けて引くことができます。コネクションを引いている時にキャンセルしたい場合には、**[Esc]** キーを押すか、ノードやコネクションでないキャンバス上をクリックします。Model Designer で作画できるコネクションは、以下のものです。

コネクション

要素のアイコン	要素の説明
	関連
	継承
	依存

このほかの装飾付きの関連も用意されています。これらの関連は、次に説明する「関連を編集する」ことによって作成できます。

推奨プロセスでは、使用しません。

要素のアイコン	要素の説明
	誘導関連
	集約関連
	合成集約関連
	左上に付ける再帰関連
	右上に付ける再帰関連
	左下に付ける再帰関連
	右下に付ける再帰関連

なお、汎化・特化関係のコネクションを結んだ場合、ソースコード生成を行うと、これらのクラス間には、継承関係が設定されます。

3.4.2 関連を編集する

推奨プロセスでは、使用しません。

関連のプロパティ ダイアログ

関連の編集は、“関連のプロパティ”ダイアログから行います。ダイアログを開くには、関連をダブルクリックします。もしくは、編集したい関連を右クリックして、表示されるメニューから [プロパティ] を選択しても、ダイアログを開くことができます。

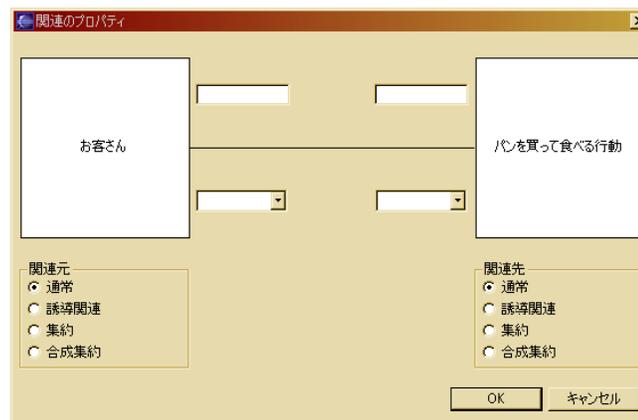


図 3.6: 関連のプロパティ ダイアログ

“関連のプロパティ”ダイアログでは、以下の編集が行えます。

- ロール名の編集
- 多重度の編集
- 関連の種類の変更

[OK] を押すと、確定しダイアログを閉じます。[キャンセル] を押すと、ダイアログを開く以前の状態に戻ります。

ロール名の編集

ロール名の編集は、ダイアログのロール名入力欄に変更したいロール名を入力することで行います。

3.4.3 関連・汎化を削除する

関連・汎化の削除は、削除したい関連・汎化を選択して Delete キーを押すか、右クリックメニューからから [削除] を選択します。

3.5 図の配置/表示を調整する

3.5.1 移動

パレットの [選択] ボタンを選択した後、移動させたいノードの中心をドラッグ&ドロップします。また、十字キーで移動もできます。複数選択すれば、一度に複数のノードを移動することができます。

3.5.2 リサイズ

パレットの [選択] ボタンを選択した後、リサイズしたいノードを選択し、ノードの縁の黒点をドラッグ&ドロップすることで、ノードの大きさを調整できます。

3.5.3 配置の調整

エディタツールバーの整列ボタンをクリックすることで、選択中の複数のノードを整列できます。以下の調整機能が提供されています。

アイコン	説明
	選択中の全てのノードを、黒枠で囲まれたノードの左辺にそろえる
	選択中の全てのノードを、黒枠で囲まれたノードの垂直方向の中心にそろえる
	選択中の全てのノードを、黒枠で囲まれたノードの右辺にそろえる
	選択中の全てのノードを、黒枠で囲まれたノードの上辺にそろえる
	選択中の全てのノードを、黒枠で囲まれたノードの水平方向の中心にそろえる
	選択中の全てのノードを、黒枠で囲まれたノードの下辺にそろえる
	選択中の全てのノードを、横の等間隔にそろえる
	選択中の全てのノードを、縦の等間隔にそろえる

3.5.4 コネクションタイプ

コネクションタイプ

コネクションタイプは、「通常」と「マンハッタン」の2種類があります。コネクションタイプを変更するには、キャンバスから変更したいコネクションを選択し、右クリックメニューから [線の表示形式]>[通常]、もしくは [線の表示形式]>[マンハッタン] を選択します。

通常タイプの線

「通常」タイプは、線の経由点を指定するためのタイプです。初めて線を引いたときには、ノード同士の最短経路で線が引かれます。折れ線にしたい場合は、キャンバスから折れ線にしたいコネクションを選択し、線の中心に表示された小さな点を、任意の場所までドラッグ&ドロップします。

マンハッタンタイプの線

「マンハッタン」タイプは縦と横の線のみでコネクションを引くためのタイプです。マンハッタンタイプでは、線形を自由に編集することはできません。

3.6 ソースコードを生成する

Model Designer で作成したタイプと Information クラスは、Java のソースコードに変換できます。キャンバス上で右クリックして表示されるメニューから、[モデルのソースコード生成] を選択すると、“生成する Type の選択” ダイアログが表示されます。

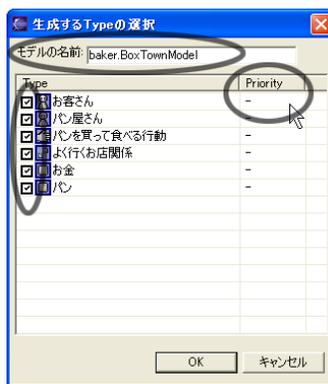


図 3.7: 生成する Type の選択 ダイアログ

まず、ダイアログの「モデルの名前」入力欄にモデルの名前を入力します。そして、ソースを生成したいモデル要素にチェックを入れます。[OK] ボタンを押すと Java のソースコードが生成されます。

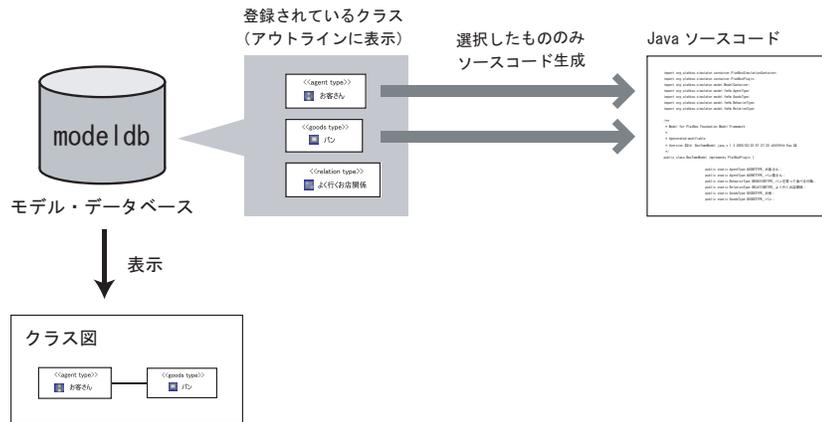


図 3.8: モデルデータベースからソースコードの生成

第 4 章

Activity Designer

4.1 Activity Designer とは

Activity Designer は、エージェントの振る舞いについての概念モデルを記述するためのエディタです。モデル作成プロセスにおいては、概念モデリングフェーズで使用します。概念モデル・クラス図で定義した Agent(もしくは Behavior) が、どのような流れで活動するのかを、UML(統一モデリング言語) のアクティビティ図で記述していきます。

この章では、Activity Designer を用いてアクティビティ図を描く方法について説明します。

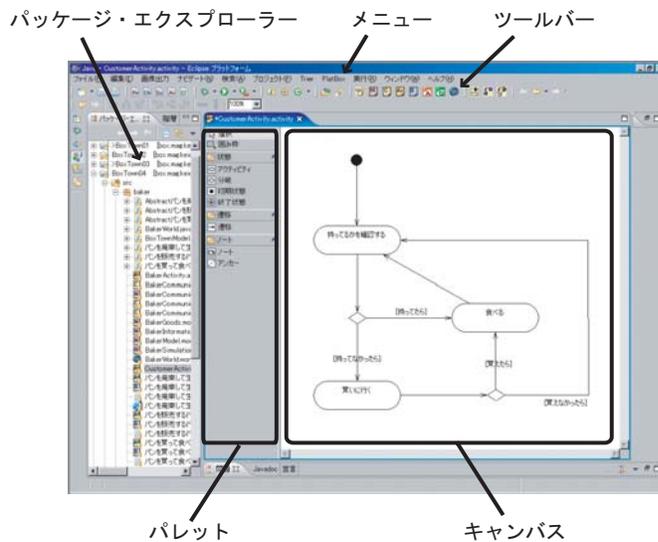


図 4.1: Activity Designer

4.2 Activity Designer を起動する

4.2.1 アクティビティ図を新規作成する

アクティビティ図ファイルの作成は、次の手順で行います。

1. パッケージ・エクスプローラーから、ファイルを新規作成したいプロジェクトの src フォルダにある パッケージ を選択する。
2. ツールバーから、アクティビティ図ファイル作成ボタン  をクリックする。
3. ファイルを保存する場所が、先ほど選択した場所であることを確認して、「ファイル名」欄にファイル名を入力し、[終了] を押す。

もしくは、メニューバーの [PlatBox]>[Create Activity] を選択して作成することもできます。

4.2.2 すでに作成したアクティビティ図を開く

すでに作成したアクティビティ図を開きたい場合は、パッケージ・エクスプローラー上で、アクティビティファイルをダブルクリックします。もしくは、右クリックして [アプリケーションから開く]-[PlatBox Activity Designer] を選択します。アクティビティファイルを読み込んだ Activity Designer が起動して、アクティビティ図が表示されます。

4.3 アクティビティ図の要素を配置する

4.3.1 アクティビティ図の要素を新規作成する

Activity Designer で作画できるノードは、以下のとおりです。

ノード

要素のアイコン	要素の説明
	アクティビティ
	分岐
	開始状態
	終了状態

4.3.2 アクティビティ図の要素を削除する

削除したい要素を選択して `delete` キーを押すか、右クリックして出てくるメニューから [削除] を選択します。

4.4 遷移の線を引く

4.4.1 遷移の線を新規作成する

作成した要素間に遷移の線を引くには、以下の手順で行います。

1. Activity Designer パレットから [遷移] を選択します。

2. 遷移の線を引きたいノードをクリックします。
3. もう一方のノードをクリックします。

コネクションを引いた後でも、さらに続けて引くことができます。コネクションを引いている時にキャンセルしたい場合には、Esc キーを押すか、ノードやコネクションでないキャンバス上をクリックします。

コネクション

Activity Designer で作画できるコネクションは、以下の通りです。

要素のアイコン	要素の説明
	遷移

4.4.2 遷移の線を編集する

遷移のプロパティダイアログ

ガード条件の編集は、“遷移のプロパティ”ダイアログで行います。“遷移のプロパティ”ダイアログを表示するには、編集したい遷移を右クリックして、表示されるメニューから“プロパティ”を選択します。編集後、“遷移のプロパティ”ダイアログの [OK] ボタンを選択すると変更が反映され、[キャンセル] ボタンを選択すると操作が取り消されます。

ガード条件の編集

ガード条件の編集は、“ガード条件”欄にガード条件を入力して行います。

4.4.3 遷移の線を削除する

削除したい遷移の線を選択して Delete キーを押すか、右クリックメニューから [削除] を選択します。

4.5 図の配置/表示を調整する

Model Designer と同じ操作で、配置/表示を調整できます。p.21 を参照してください。

第 5 章

Communication Designer

5.1 Communication Designer とは

Communication Designer は、Agent (もしくは Behavior) 間のやり取りを分析して、そのコミュニケーションの流れを記述するためのエディタです。モデル作成プロセスにおいては、概念モデリングフェーズで使用しますが、必要に応じて、シミュレーションデザインフェーズでも使用します。概念モデル・クラス図で定義した Agent(もしくは Behavior) のオブジェクトが、どのような流れでコミュニケーションや処理を行うのかを、コミュニケーション・シーケンス図で記述していきます。コミュニケーション・シーケンス図は、UML(統一モデリング言語) のシーケンス図を単純化・拡張したものです。

この章では、Communication Designer を用いてコミュニケーション・シーケンス図を描く方法について説明します。

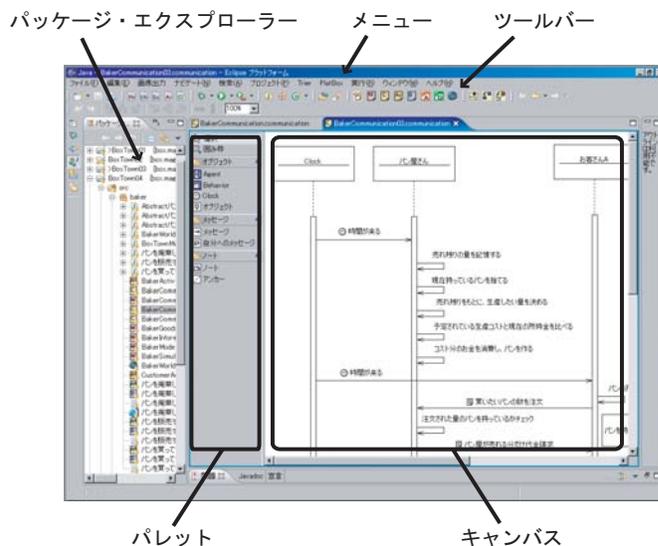


図 5.1: Communication Designer

5.2 Communication Designer を起動する

5.2.1 コミュニケーション・シーケンス図を新規作成する

コミュニケーション・シーケンス図ファイルの作成は、次の手順で行います。

1. パッケージ・エクスプローラーから、ファイルを新規作成したいプロジェクトの src フォルダにある パッケージ を選択します。
2. ツールバーから、コミュニケーション・シーケンス図ファイル作成ボタン  をクリックします。
3. ファイルを保存する場所が、先ほど選択した場所であることを確認して、「ファイル名」欄にファイル名を入力し [終了] を押します。

もしくは、メニューバーの [PlatBox]>[Create Communication] を選択して作成することもできます。

5.2.2 すでに作成したコミュニケーション・シーケンス図を開く

すでに作成したコミュニケーション・シーケンス図を開きたい場合は、パッケージ・エクスプローラー上で、コミュニケーションファイルをダブルクリックします。もしくは、右クリックして [アプリケーションから開く]-[PlatBox Communication Designer] を選択します。コミュニケーションファイルを読み込んだ Communication Designer が起動して、コミュニケーション・シーケンス図が表示されます。

5.3 コミュニケーション・シーケンス図の要素を配置する

5.3.1 コミュニケーション・シーケンス図の要素を新規作成する

コミュニケーション・シーケンス図の要素は、Communication Designer のパレットにあるクラスのボタンを選択し、続けてキャンバスをクリックすることでクラスを配置します。Communication Designer で作画できるノードは、以下のとおりです。

ノード

要素のアイコン	要素の説明
	AgentType(オブジェクト)
	BehaviorType(オブジェクト)
	Clock オブジェクト
	オブジェクト (クラス指定なし)

5.3.2 コミュニケーション・シーケンス図の要素を削除する

図からノードを削除するには、削除したい要素を右クリックして表示されるメニューから [削除] を選択します。もしくは、削除したい要素を選択して、キーボードの Delete キーを押しても削除できます。

5.4 メッセージの線を引く

5.4.1 メッセージの線を新規作成する

作成したノード間にメッセージの線を引くには、以下の手順で行います。

1. Communication Designer パレットから、[メッセージ] を選択します。
2. メッセージの線を引きたいノードをクリックします。
3. もう一方のノードをクリックします。

[~の自己遷移] を引く場合は、対象となるノードを一度だけクリックします。

コネクションを引いた後も、さらに続けて引くことができます。コネクションを引いている時にキャンセルしたい場合には、Esc キーを押すか、ノードやコネクションでないキャンバス上をクリックします。

コネクション

Communication Designer で作画できるコネクションは、以下の通りです。

要素のアイコン	要素の説明
	メッセージ
	自分へのメッセージ

5.4.2 メッセージの線を編集する

メッセージのプロパティダイアログ

メッセージの編集は、“メッセージのプロパティ”ダイアログで行います。“メッセージのプロパティ”ダイアログを表示するには、編集したいメッセージ線を右クリックして表示されるメニューから、[プロパティ] を選択します。“メッセージのプロパティ”ダイアログの [OK] ボタンを選択すると変更が反映され、[キャンセル] ボタンを選択すると操作が取り消されます。

メッセージの内容の編集

メッセージの内容を編集するには、「メッセージ」欄に内容を入力します。

メッセージタイプの編集

メッセージタイプを編集するには、「コミュニケーションの種類」欄をクリックして表示されるプルダウンメニューから、仕意のタイプを選択します。

5.4.3 メッセージの線を削除する

メッセージの線を削除するには、削除したいメッセージの線を選択して キーを押すか、右クリックメニューから [削除] を選択します。

5.5 図の配置/表示を調整する

5.5.1 メッセージの位置の移動

線を選択すると表示される線の始点か中心を、上下にドラッグします。

5.5.2 自分へのメッセージの移動

線を選択すると表示される線の始点を、移動先へドラッグします。

5.5.3 リサイズ

リサイズしたいノードを選択し、ノードの縁の黒点をドラッグ&ドロップすることで、ノードの大きさを調整できます。

5.5.4 配置の調整

エディタツールバーの整列ボタンをクリックすることで、選択中の複数のノードを整列できます。p.21 を参照してください。

第 6 章

Behavior Designer

6.1 Behavior Designer とは

Behavior Designer は、Behavior の状態遷移をモデル化するためのエディタです。モデル作成プロセスにおいては、シミュレーションデザインフェーズで使用します。Behavior の状態遷移を、UML(統一モデリング言語) の状態遷移図 (ステートチャート図) によって記述していきます。記述したモデルから、PlatBox Simulator 上で動作するコンポーネントのソースコードを生成することができます。

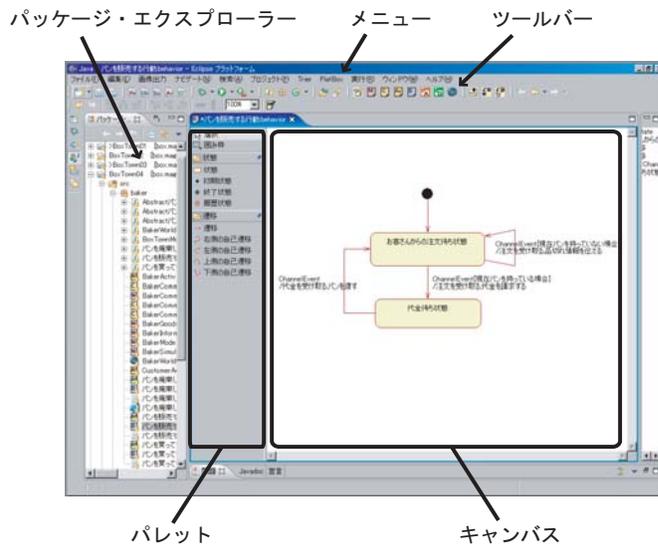


図 6.1: Behavior Designer

6.2 Behavior Designer を起動する

6.2.1 状態遷移図を新規作成する

状態遷移図ファイルの作成は、次の手順で行います。

1. パッケージ・エクスプローラーから、ファイルを新規作成したいプロジェクトの

- src フォルダにある パッケージ を選択します。
2. ツールバーから、状態遷移図ファイル作成ボタン  をクリックします。
 3. ファイルを保存する場所が、先ほど選択した場所であることを確認して、「ファイル名」欄にファイル名を入力し [終了] を押します。

もしくは、メニューバーの [PlatBox]>[Create Behavior] を選択して作成することもできます。

6.2.2 すでに作成した状態遷移図を開く

すでに作成した状態遷移図を開きたい場合は、パッケージ・エクスプローラー上で、ビヘイビアファイルをダブルクリックします。もしくは、右クリックして [アプリケーションから開く]-[PlatBox Behavior Designer] を選択します。ビヘイビアファイルを読み込んだ Behavior Designer が起動して、状態遷移図が表示されます。

6.3 状態遷移図の要素を配置する

6.3.1 状態遷移図の要素を新規作成する

状態遷移図の要素は、Behavior Designer のパレットにあるボタンを選択し、続けてキャンバスをクリックすることで配置することができます。Behavior Designer で作画できる要素は、以下のとおりです。

要素のアイコン	要素の説明
	状態
	開始状態
	終了状態
	履歴状態

なお、Behavior Designer では、状態の中に状態と遷移を入れるという「入れ子状態」をつくることができます。入れ子状態をつくるには、親になる状態の中に新しく状態を配置するか、親の状態の外で配置した状態を、親の状態の内側に移動することで作成できます。入れ子状態になった状態の遷移がどのように行われるのかについては、UML の文献等を参照してください。

6.3.2 状態の編集

特別な状態ダイアログ

状態プロパティの編集は、“特別な状態”ダイアログから行います。“状態の設定”ダイアログを開くには、編集したい状態を右クリックして、表示されるメニューから [プロパティ] を選択します。

推奨プロセスでは、使用しません。

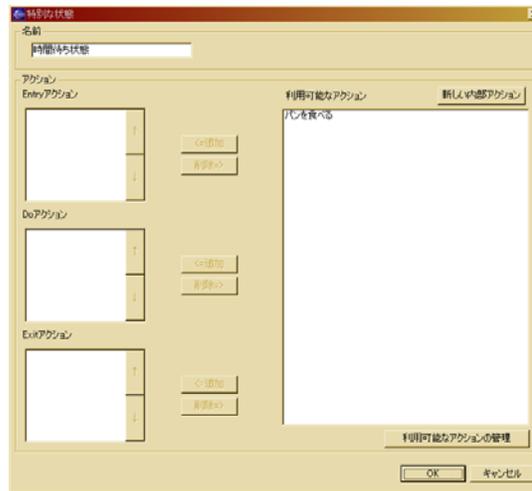


図 6.2: 特別な状態ダイアログ

“特別な状態” ダイアログでは、以下の操作を行うことができます。

- 状態名の編集
- アクションの新規作成
- アクションの登録
- アクションの登録の解除
- アクションの移動
- アクションの編集

アクションは、Entry アクション、Do アクション、Exit アクションの 3 種類のアクションがありますが、操作方法はどれも同じです。

[OK] を押すと確定し、ダイアログを閉じます。[キャンセル] を押すと、ダイアログを閉く以前の状態に戻ります。

状態名の編集

状態名の編集は、“名前” 欄に状態名を入力することで行います。状態名は、状態をダブルクリックすることでも変更できます。

アクションの新規作成

アクションの新規作成は、以下の手順で行います。

1. [新しい内部アクション] ボタンを押して、“内部アクション” ダイアログを開きます。
2. 新規作成するアクション名を入力すると、「利用可能なアクション」欄にそのアクションが追加されます。

アクションテンプレートには、その先頭に「(テンプレート)」と表示されています。

アクションの登録

アクションの登録は、以下の手順で行います。

1. 「利用可能なアクション」欄のアクションリストから、追加したいアクションを選択します。
2. 追加したい Entry/Do/Exit アクションの [\leftarrow =追加] ボタンを押します。

アクションの登録の解除

アクションの登録の解除は、以下の手順で行います。

1. Entry/Do/Exit アクションから、解除したいアクションを選択します。
2. 選択したアクションの右の [削除= \rightarrow] ボタンを押します。

アクションの移動

Entry/Do/Exit アクションのリストにあるアクションは、その順番に処理されます。アクションの順番を変えることで、処理順番を変えることができます。

1. Entry/Do/Exit アクションから、移動したいアクションを選択します。
2. 選択したアクションの右の [] もしくは [] ボタンを押します。

アクションの編集

アクションの編集は、“利用可能なアクションの管理”ダイアログで行います。“利用可能なアクションの管理”ダイアログを表示するには、[利用可能なアクションの管理] ボタンをクリックします。“利用可能なアクションの管理”ダイアログでは、次の操作を行うことができます。

操作	説明
アクションの新規作成	状態および遷移の設定ダイアログから、アクションの新規作成を行った場合と同様です。
アクションの名前変更	アクションの名前を変更します。変更したアクションを使用する状態または遷移があった場合は、それら全てが変更されます。
アクションの削除	アクションの削除を行います。ここで削除したアクションは今後、状態や遷移のアクションに設定することはできません。しかし、削除したアクションをすでに利用している状態または遷移があった場合は、それらに関連付けられたアクションは削除されません。
テンプレートアクションの読み込み	[再読み込み] ボタンを押すと、テンプレートアクションの再読み込みを行います。現在編集集中の Behavior がある PlatBox プロジェクトのクラスパス上に、新たなテンプレートアクションを追加した場合には、それらを利用するためにこの処理を行う必要があります。

テンプレートの利用

状態遷移図で新たに定義したアクションやガード条件以外にも、クラスパス上にあるテンプレートアクションやテンプレートガード条件を、利用することができます。テンプレートは、状態、遷移の設定ダイアログが持つ選択可能なアクション/ガード条件のリストにおいて、名前の前に (Template) という表示がされています。

6.3.3 状態遷移図の要素を削除する

ノードを図から削除するには、削除したい要素を右クリックして表示されるメニューから、[削除] を選択します。また、削除したい要素を選択して、キーボードの Delete キーを押しても削除できます。

6.4 遷移を引く

6.4.1 遷移を新規作成する

状態から状態への遷移は、以下の手順で記述することができます。

1. Behavior Designer パレットから [遷移] を選択します。
2. 遷移を引きたいノードをクリックします。
3. もう1つのノードをクリックします。

遷移の線を引いた後も、さらに続けて引くことができます。遷移の線を引いている時にキャンセルしたい場合には、Esc キーを押すか、ノードや遷移以外のキャンパス上をクリックします。[自分へのメッセージ] を引きたい場合には、対象となるノードを一度だけクリックします。

Behavior Designer で作画できる遷移は、以下の通りです*1。

要素のアイコン	要素の説明
	状態遷移
	右に付ける自己遷移
	左に付ける自己遷移
	上に付ける自己遷移
	下に付ける自己遷移

6.4.2 遷移を編集する

遷移のプロパティダイアログ

状態遷移の編集は、“遷移のプロパティ”ダイアログから行います。“遷移のプロパティ”ダイアログを開くには、編集したい状態を右クリックして、表示されるメニューから [プロパティ] を選択します。

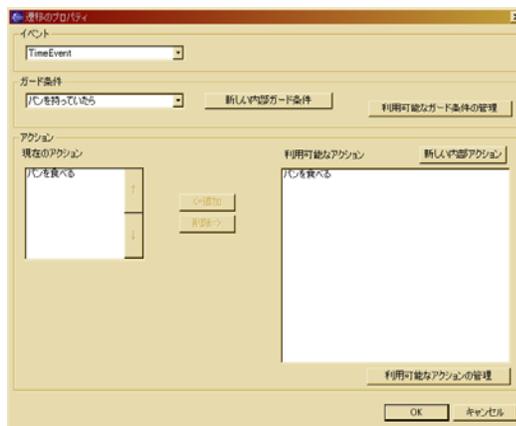


図 6.3: 遷移のプロパティダイアログ

“遷移のプロパティ”ダイアログでは、以下の操作を行うことができます。

- 遷移を起こすイベントの設定
- ガード条件の設定
- アクションの設定

[OK] を押すと、確定してダイアログが閉じます。[キャンセル] を押すと、ダイアログを聞く以前の状態に戻ります。

*1 自己遷移は、上下左右の 4 種類が提供されていますが、これらは描画上の表現の違いであって、意味的に異なるわけではありません。

遷移を起こすイベントの設定

設定可能なイベントのリストの中から、編集中の遷移に対応させたいイベントを設定します。

ガード条件の設定

設定可能なガード条件のリストの中から、編集中の遷移に対応させたいガード条件を設定します。ガード条件の新規作成は、[新しい内部ガード条件] ボタンを押して表示されるダイアログに、ガード条件の名前を入力して行います。ガード条件の編集は、[利用可能なガード条件の管理] ボタンを押して表示されるダイアログで行います。

アクションの設定

アクションの設定については、p.34 のアクションの操作方法を参照して下さい。

ガード条件の管理

遷移の設定ダイアログから、[利用可能なガード条件の管理] というボタンをクリックすると、ガード条件の管理ダイアログが開きます。ガード条件の管理ダイアログには、以下の4つの機能があります。

機能	機能の説明
ガード条件の新規作成	遷移の設定ダイアログから、ガード条件の新規作成を行った場合と同様です。
ガード条件の名前変更	ガード条件の名前を変更します。変更したガード条件を使用する遷移があった場合は、それら全てが変更されます。
ガード条件の削除	ガード条件の削除を行います。ここで削除したガード条件は、今後、遷移のガード条件に設定することはできません。しかし、削除したガード条件をすでに利用している遷移があった場合は、それらに関連付けられたガード条件は削除されません。
テンプレートガード条件の読み込み	[再読み込み] ボタンをクリックすると、テンプレートガード条件の再読み込みを行います。Behavior の編集集中に Plat-Box プロジェクトのクラスパス上に、新たなテンプレートガード条件が追加された場合、それらを遷移のガード条件として設定するためには、この処理を行う必要があります。

6.4.3 遷移を削除する

遷移を削除するには、削除したいメッセージの線を選択して Delete キーを押すか、右クリックメニューから [削除] を選択します。

6.5 図の配置/表示を調整する

Model Designer と同じ操作で、配置/表示を調整することができます。p.21 を参照してください。

6.6 Behavior の実装

6.6.1 Behavior の実装

ソースコードの出力

[プログラムの生成] ボタンをクリックすると、現在選択している Behavior のソースコードの雛形 (スケルトン) が生成できます。これらは、変更する必要はありません。

ファイル名	ファイルの種類	ファイルの説明
~ Behavior.behavior	.behavior ファイル	エディットした状態遷移図のファイルです。
Abstract ~ Behavior.java	Behavior の雛形	Behavior の雛形です。絶対に編集しないでください。
~ Behavior.java	Behavior ファイル	実際に編集する Behavior の java ファイルです。手動で実装する際には、このファイルの空になっているメソッドを実装してください。

アクションブロック図の生成

Behavior のアクションを定義するには、Action Designer を用います。Action Designer は、右クリックメニューから [Action Designer で開く] を選択することで起動し、アクションブロック図の基本型が生成されます。

生成されたアクションブロック図には、次のメソッドが定義されています。必要に応じて実装してください。それぞれのメソッドが持つ意味は、以下のとおりです。

メソッド	メソッドの説明
アクション名と同名のメソッド	状態遷移図で定義されたアクションを表します。このアクションが呼ばれたときに行いたい処理を実装してください。
ガード条件と同名のメソッド	状態遷移図で定義されたガード条件を表します。このガード条件によって判定したい条件を判定し、返り値として真偽値を返すよう実装してください。

命名規則について

.behavior ファイルのファイル名と、状態遷移図で設定される状態、遷移、アクション、ガード条件の各要素の名前は、ソースコードの生成の際にクラス名、メソッド名、変数名

となります。命名規則に違反すると、不正確でわかりにくいソースコードが生成されたり、エラーが発生したりするので、以下の規則に従うようにしてください。

状態遷移図の要素名	Java の要素名	命名規則
.behavior ファイルのファイル名	クラス	単語の先頭のみ大文字のアルファベット、または全角日本語の名前 (例:SendGoodsBehavior、財送信行動)。数字や記号を頭文字にすることはできません。
アクションの名前、ガード条件の名前	メソッド	名前の先頭は小文字で、以降は単語の先頭のみ大文字のアルファベット、または全角日本語の名前 (例:receiveGoods、財を受け取る)。数字や記号を頭文字にすることはできません。
状態の名前	変数	単語の先頭のみ大文字のアルファベット、または全角日本語の名前 (例:waitSentGoodsState、財送信待ち状態)。数字や記号を頭文字にすることはできません。

第7章

Action Designer

7.1 Action Designer とは

Action Designer は、Behavior のアクションを記述するためのエディタです。モデル作成プロセスにおいては、シミュレーションデザインフェーズで使用します。Behavior のアクションとガード条件を、独自に定義された「アクション記述言語」(Action Block Language: ABL) を用いて記述していきます。記述したモデルから、PlatBox Simulator 上で動作するコンポーネントのソースコードを生成することができます。

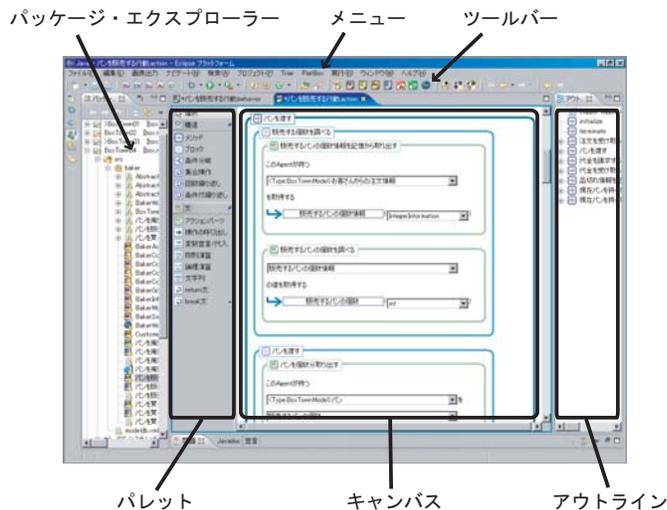


図 7.1: Action Designer

アクション記述言語には、その語彙として、社会シミュレーションのモデル作成において頻出する処理の集まりが定義されています。モデル作成者は、Action Designer を用いることによって、作成したい処理の目的からアクションのブロックを選択・設定することで、エージェントのアクションの設計・実装を行うことができます。

7.2 Action Designer を起動する

7.2.1 アクションブロック図を新規作成する

新しくアクションブロック図を作成するには、Behavior Designer のキャンバスで右クリックし、表示されたメニューから「Action Designer で開く」を選択します。プログレッシブが表示され、アクションファイルが生成されます。パッケージ・エクスプローラー上に、新しく作成したアクションファイルがあることを確認してください。作成されたアクションファイルの名前は、Behavior Designer で開いていた Behavior ファイルと同じ名前となっています (ただし、拡張子は.action)。

自動的に Action Designer が起動し、アクションファイルが読み込まれた状態になります。キャンバスには、アクションブロック図が表示されます。

7.2.2 すでに作成したアクションブロック図を開く

すでに作成したアクションブロック図を開きたい場合は、パッケージ・エクスプローラー上で、アクションファイルをダブルクリックします。もしくは、右クリックして [アプリケーションから開く]-[PlatBox Action Designer] を選択します。アクションファイルを読み込んだ Action Designer が起動して、アクションブロック図が表示されます。

7.3 構造を配置する

Action Designer では、単に処理を並べて書いていくというのではなく、制御構造や論理的な構造によって構造化してつくっていく仕組みになっています。構造の配置は、Action Designer の左側のパレットから構造を選択し、配置したいメソッド上でクリックします。すると、クリックした場所に、選択された構造が配置されます。配置できる構造には、以下のものがあります。

1. ブロック

処理の論理的なまとまりを構造としてまとめます。

2. 条件分岐

条件に応じて処理を切り替えます。

3. 集合操作

集合の各要素に対して何らかの処理を行います。

4. 回数繰り返し

処理を指定回数行います。

5. 条件付繰り返し

処理を一定の条件を満たすまで繰り返し行います。

6. メソッド

複雑なアクションを構造化したり、繰り返し現れる記述をサブルーチンとしてひとまとめにします。

7.3.1 ブロック

ブロックは、処理の論理的なまとまりを構造として記述するために用います。ブロックには名前をつけることができます。名前には、そのブロックの中に定義された下位の要素の 目的 を記述します。

ブロックの配置は、パレットから「ブロック」を選択し、配置したいところをクリックします。クリックした場所にブロックが配置されます。



図 7.2: ブロックの配置

新しく配置したブロックの名前は「新規ブロック」となっていますが、名前の部分をダブルクリックして、新しい名前を入力・変更できます。処理の目的に応じた名前をつけましょう。

7.3.2 条件分岐

条件分岐は、条件に応じて処理を切り替えるために用います。条件分岐には、条件式として真偽値の変数を設定することができます。

パレットから「条件分岐」を選択し、配置したいメソッド上でクリックします。クリックした場所に条件分岐が配置されます。条件分岐を用いるときには、事前に条件式として真偽値の変数を作成しておく必要があります。真偽値の変数を作成する方法は、p.56の「論理演算」を参照してください。

(1) if 文と else 文の作成

ある条件式と、それ以外の分岐を作成したい場合は、条件分岐で if 文と else 文を作成します。if 文と else 文の条件分岐を作成したい場合は、間に何も挟まずに隣り合わせで条件分岐を 2 つ配置し、下の条件分岐の条件を「true」に設定します。これで、上の条件分岐は if 文となり、下の条件分岐は else 文となります。

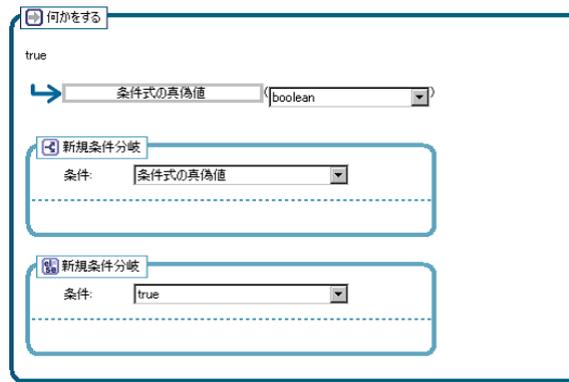


図 7.3: 条件分岐の配置:if 文と else 文

(2) if 文と else if 文の作成

排他的な条件で、2 つ以上の分岐を作成したい場合は、条件分岐で if 文と else if 文を作成します。if 文と else if 文の条件分岐を作成したい場合は、(1) の場合と同様に、間に何も挟まずに隣り合わせで条件分岐を 2 つ以上配置します。そして、二番目以降の条件分岐の条件に、独自に宣言した真偽値の変数を設定します。一番上の条件分岐は if 文となり、二番目以降の条件分岐は else if 文となります。

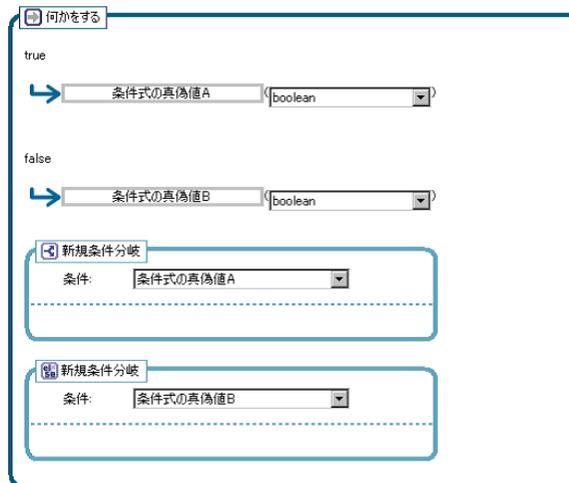


図 7.4: 条件分岐の配置: if 文と else if 文

条件分岐を追加することによって、else if 文はいくつも増やすことができます。

(3) それぞれ独立した if 文の作成

それぞれが独立した条件の条件分岐を作成したい場合は、異なるブロックを作成して、そのなかに条件分岐を配置します。このようにブロックを利用するのは、条件分岐を単に 2 つ並べるだけでは、2 つ目は else if 文になってしまうからです。配置されているブロッ

クが異なる条件分岐は、独立した条件分岐となり、それぞれが if 文となります。

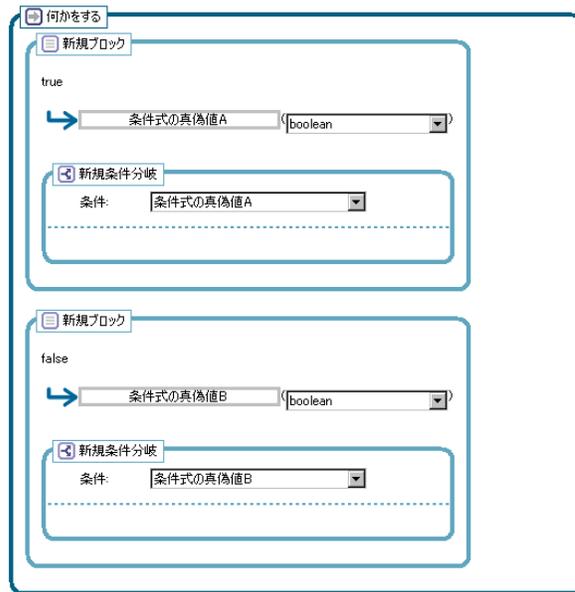


図 7.5: 条件分岐の配置: 異なるブロックに配置された独立した if 文

7.3.3 集合操作

集合操作は、集合の各要素に対して何らかの処理を行いたいときに用います。集合操作では、操作を行いたい対象の集合と、集合の要素をどのような型のオブジェクトとして扱うかを指定する (キャストする) ことができます*1。

目的

集合の各要素に対して何らかの処理を行います。

使用にあたって必要なもの

処理を行いたい要素の集合

設定する項目

集合: 処理を行いたい要素の集合を指定します。

要素の種類: 取得される各要素の型を指定します。

集合操作のなかで利用できるもの

「集合の要素」

パレットから「集合操作」を選択し、配置したいメソッド上でクリックします。クリックした場所に、集合操作が配置されます。配置された集合操作の「集合:」のところでは、

*1 集合操作の概念については、データ構造やデザイン・パターンの文献に掲載されている「Iterator」の説明を参照して下さい。

操作を行いたい要素の集合を選択します。「要素の種類:」のところでは、操作を行いたい要素の型を選択します。集合操作のなかに配置される構造と文では、取得された「集合の要素」を、指定された型で使うことができます。

例えば、アクションパーツ「Worldに存在する全てのAgentを取得する」を用いて、「取得したAgentの集合」という集合を配置したとします。この集合の中のそれぞれのAgentに対して、何らかの処理を行う場合には、集合操作を配置して、「集合:」から「取得したAgentの集合」を選択して、「要素の種類:」から「Agent」を選択します。そうすると、集合操作のなかで、集合の各要素がAgentという型の「集合の要素」という名前で取得され、使用できるようになります。

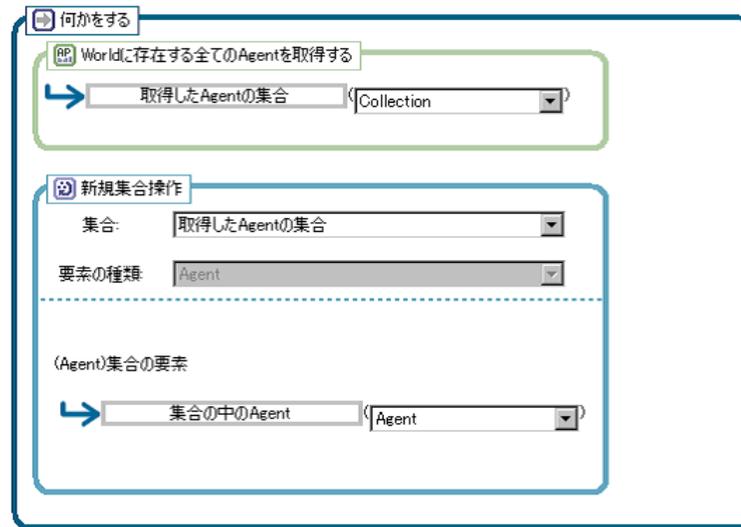


図 7.6: 集合操作の配置

7.3.4 回数繰り返し

回数繰り返しは、何らかの操作を指定回数繰り返し行いたいときに用います。回数繰り返しでは、処理を何回繰り返すのか、を変数や値で設定することができます*2。

目的

何らかの処理を指定回数繰り返し行います

使用にあたって必要なもの

繰り返したい回数

設定する項目

回数: 繰り返したい回数を指定します

*2 回数繰り返しは、for 文と同じものです。

回数繰り返しのなかで利用できるもの

「回数」(現在までで繰り返された回数)

パレットから「回数繰り返し」を選択し、配置したいメソッド上でクリックします。クリックした場所に、回数繰り返しが配置されます。配置された回数繰り返しの「回数:」から繰り返したい回数を変数や値で選択します。回数繰り返しのなかでは、現在繰り返された回数を取得して利用することができます。

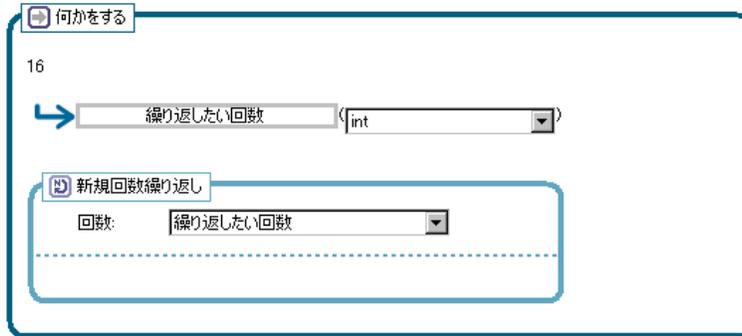


図 7.7: 回数繰り返しの配置

7.3.5 条件付繰り返し

条件付繰り返しは、何らかの処理を一定の条件を満たすまで繰り返し行いたいときに使います。条件付繰り返しでは、繰り返しの継続条件を設定することができます。なお、繰り返しの継続条件の判定が、条件付繰り返しのなかの処理によって変化しない場合は、処理がずっと繰り返されるので、注意が必要です*3。

目的

何らかの処理を一定の条件を満たすまで繰り返し行います

使用にあたって必要なもの

繰り返しが継続するための条件の真偽値

繰り返しから抜けるための終了条件の真偽値

設定する項目

条件 (新規条件付繰り返し): 継続のための条件を指定します

条件 (継続条件の判定をする): 終了のための条件を指定します

パレットから「条件付繰り返し」を選択し、配置したいメソッド上でクリックします。クリックした場所に、条件付繰り返しが配置されます。配置された条件付繰り返しの「新

*3 条件付繰り返しは、if 文による break が付加された while 文と同じものです。

「規条件付繰り返し」の「条件:」のところで、継続するための条件式の真偽値を指定します。

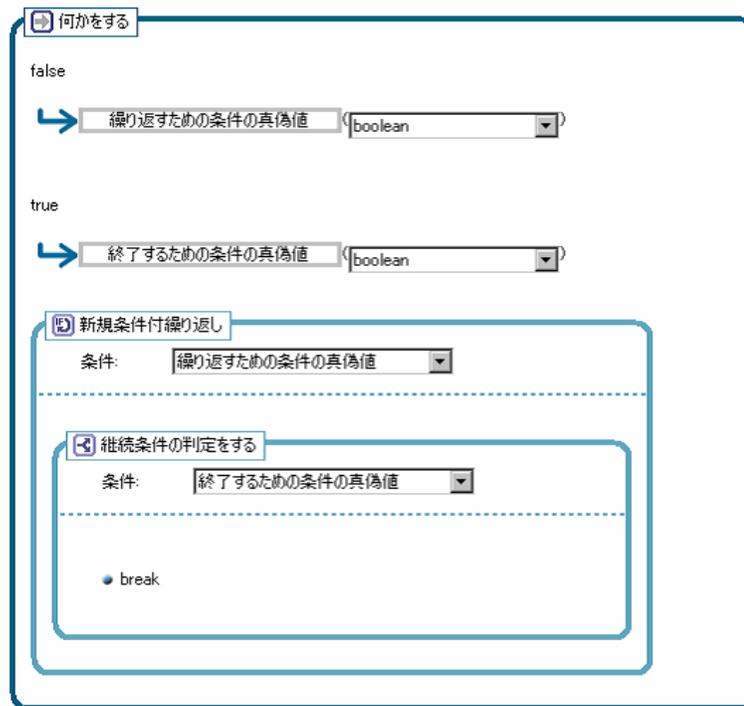


図 7.8: 条件付繰り返しの配置

「継続条件の判定をする」の「条件:」のところで、終了するための条件式の真偽値を指定します。とくに必要がないのであれば、この条件分岐を削除しても構いません。

7.3.6 メソッド

メソッドは、複雑なアクションを構造化したり、アクションの中で繰り返し現れる記述を、サブルーチンとしてひとまとめにしたいときに用います。メソッドでは、そのメソッドの中で利用したい変数を引数として与えたり、メソッドの呼び出し元に対して、処理の結果を「戻り値」として返すように設定することができます。

パレットから「メソッド」を選択し、キャンバス上の配置したい場所でクリックします。クリックした場所にメソッドが配置されます。



図 7.9: メソッドの配置

作成したメソッドの可視性、引数、戻り値を新たに設定するためには、メソッドを選択

した状態で右クリックして、表示されたウィンドウから [プロパティ] を選択して、メソッド編集ダイアログを開きます。

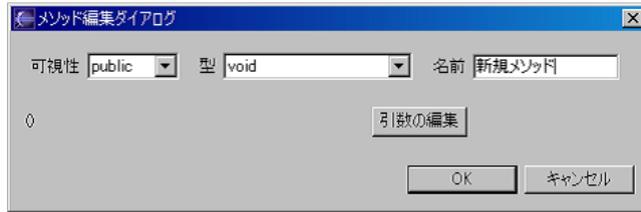


図 7.10: メソッド編集ダイアログ

(1) 戻り値の設定

メソッドの呼び出し元に対して、メソッドの処理の結果を返したい場合は、メソッド編集ダイアログの「型」プルダウンメニューで、「戻り値」の型を選択します。初期値では「void」(何も返さない)が選ばれています。

(2) 引数の設定

メソッドの処理のなかで利用する変数を与えたい場合は、メソッド編集ダイアログの [引数の編集] ボタンを選択します。引数の編集ダイアログが表示されるので、与えたい引数の型と名前を設定します。

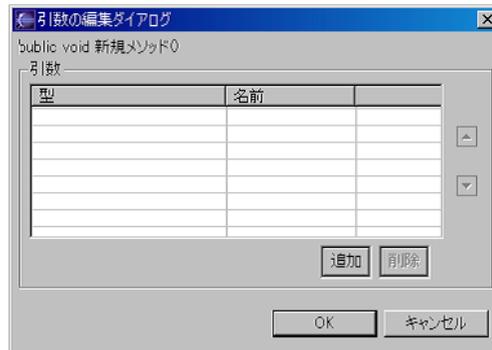


図 7.11: 引数の編集ダイアログ

(3) 可視性の設定

メソッドがどのクラスから呼び出せるのか、という可視性を設定したい場合は、メソッド編集ダイアログの「可視性」プルダウンメニューで、可視性を選択します。メソッドに対して設定できる可視性の一覧は、以下のとおりです。

可視性	範囲
public	すべてのクラスから呼び出せます。
protected	このクラスと同じパッケージ内のクラスから呼び出せます。
private	このクラスだけが呼び出せます。メソッドが内部的な処理を担うのであれば、private にしてください。
package	このメソッドをもつクラスの可視性と同じになります (Action Designer で生成されるクラスの可視性は public なので、package を設定したメソッドは public と同じ可視性となります)。

7.4 文を配置する

文の配置は、Action Designer の左側のパレットから文を選択して、配置したい場所 (メソッドの内部) をクリックします。設定のためのダイアログが表示されるので、文に応じた設定を行い、実際に配置します。

7.4.1 アクションパーツ

アクションパーツは、シミュレーションのアクション記述において頻出する処理が文として定義されたものです。アクションパーツには、「この Agent が持つ Information を取り出す」というような名前がつけられており、モデル作成者は自分の作成したい処理の目的から、実際の処理を探して選択することができるようになっています。

(1) アクションパーツを配置したい場所を選択

パレットから「アクションパーツ」を選択し、配置したい場所でクリックします。アクションパーツの選択ダイアログが開きます。

(2) アクションパーツの選択

アクションパーツの選択ダイアログから、配置したいアクションパーツを選択します。アクションパーツの選択には、以下の3種類の方法があります。

1. カテゴリから選択

新規作成したいアクションパーツをカテゴリから選択したい場合は、ウィンドウの上部にあるイメージか、その下にある「カテゴリ」プルダウンメニューから選択します。

2. キーワードで検索して選択

新規作成したいアクションパーツをキーワードで検索したい場合は、ダイアログの中央右にある [検索] ボタンをクリックします。アクションパーツの検索ダイアログが表示されるので、選択したいアクションパーツに関するキーワードを入力し、検索します。

3. 最近利用したアクションパーツから選択

以前利用したアクションパーツを配置したい場合は、ウィンドウの下部の「最近利



図 7.12: アクションパーツの選択ダイアログ

用したアクションパーツ」リストから選択します。

(2) アクションパーツの選択

どのようなアクションパーツやカテゴリがあるかということについては、『PlatBox リファレンスガイド』を参照してください。

7.4.2 操作の呼び出し

操作の呼び出しは、アクションパーツとして用意されていない操作を行いたいときに用います。呼び出したいメソッドやオブジェクトを選択することができます。アクション記述のなかでは、モデル作成者が自ら設計したクラスやメソッドを利用するために用います。

パレットから「操作の呼び出し」を選択して、配置したいメソッド上でクリックすると、式を編集するダイアログが表示されます。

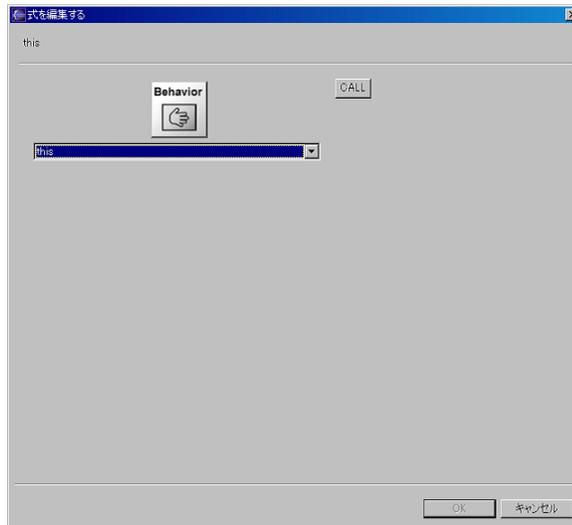


図 7.13: 操作の呼び出しにおける式を編集するダイアログ

プルダウンメニューから利用したいメソッドを選択します。あるいは、利用したいメソッドをもつオブジェクトを選択して、[Call] ボタンをクリックすると、選択されたオブジェクトのもつメソッドとフィールドを選択するプルダウンメニューが表示されます。このようにして、オブジェクトを呼び出していき、利用したいメソッドを選択します。

選択されたメソッドに引数が必要な場合は、ダイアログの右側に [実引数を選択して下さい] ボタンが表示されます。[実引数を選択して下さい] ボタンをクリックすると、引数を編集するダイアログが表示されます。そのダイアログのプルダウンメニューから、指定したい引数を選択し、型を変更したい場合には「型を変更する」プルダウンメニューから [OK] ボタンを押します。全ての設定が終わったら、[OK] ボタンを押すと、設定した操作が配置されます。

7.4.3 変数宣言/代入

変数宣言/代入は、アクションの中で一時的に利用したいデータを変数として格納したいときに用います。変数宣言/代入の配置設定は、前述の「操作の呼び出し」に変数宣言が加わったかたちとなります。

プルダウンメニューから変数宣言に用いたいメソッドを選択します。「変数を作成する:」のところには、変数名を入力します。型を変更して変数を宣言したい場合は、「型を変更する:」チェックボックスをオンにして、プルダウンメニューから型を選択します。全ての設定が終わったら、[OK] ボタンを押すと、設定した変数宣言が配置されます。

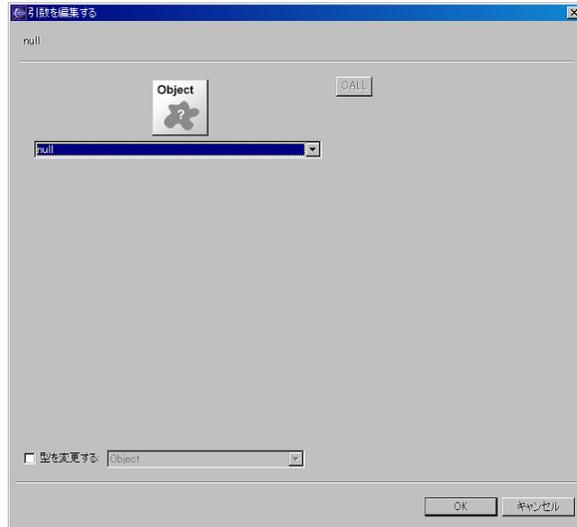


図 7.14: 引数を編集するダイアログ

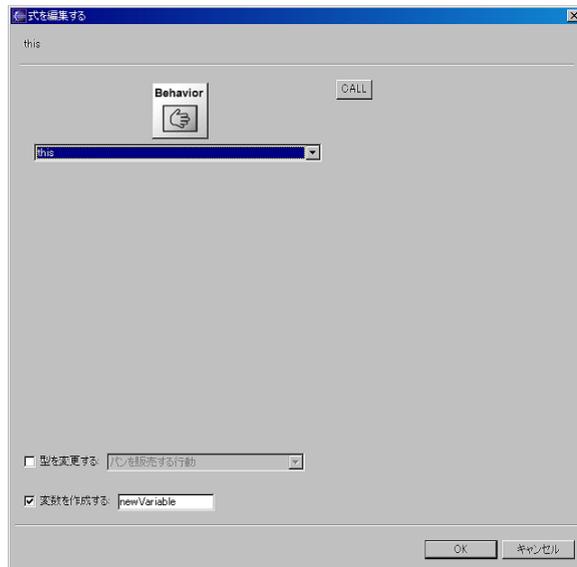


図 7.15: 変数宣言/代入における式を編集するダイアログ

7.4.4 四則演算

四則演算は、実数の値を式として設定し、その結果を変数として定義したいときに用います。算術式の中では、値だけではなく、定義済みのほかの変数を使うこともできます。

パレットから「四則演算」を選択して、配置したいメソッド上でクリックすると、数式を編集するダイアログが表示されます。

数式を編集するダイアログでは、算術式と宣言する変数名が設定できます。算術式のなかで、既に定義された変数を呼び出して利用したい場合は、「利用可能な変数:」プルダウン



図 7.16: 数式を編集するダイアログ

ンメニューから選択して追加します。なお、既に存在する変数と同じ名前を入力して作成した場合は、既に存在する変数に計算結果を代入することになります。

7.4.5 論理演算

論理演算は、論理式を作成して、条件分岐などの判定で利用される真偽値を定義したいときに用います。パレットから「論理演算」を選択して、配置したいメソッド上でクリックすると、論理式を編集するダイアログが表示されます。

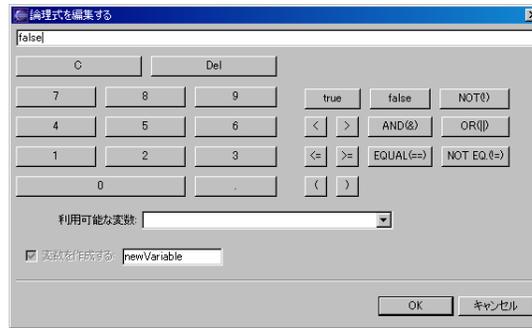


図 7.17: 論理式を編集するダイアログ

論理演算では「利用可能な変数:」から変数を選択して、論理式に追加することができます。全ての設定が終わったら、[OK] ボタンを押すと、論理式が配置されます。もし全ての設定が終わっているにもかかわらず、[OK] ボタンが押せない場合は、作成した論理式に構文エラーがあります。その場合は、以下のチェックポイントを参考に、ダイアログ上部に表示されている論理式を直接修正してみてください。

- カッコ () はきちんと対応しているかどうか？
- 変数名が続けて入力されているかどうか？

7.4.6 文字列

文字列は、アクション記述のなかで使用する文字列を、変数として定義したいときに用います。次の場合に必要になります。

- World に定義されているパラメータの値を取得します。
- ログなどを出力します。

パレットから「文字列」を選択して、配置したいメソッド上でクリックすると、文字列を編集するダイアログが表示されます。

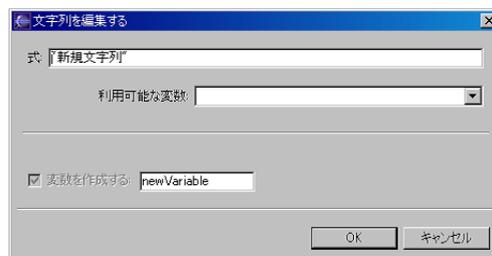


図 7.18: 文字列を編集するダイアログ

文字列を編集するダイアログでは、文字列の式と宣言する文字列の変数名が設定できます。文字列の式では「利用可能な変数:」から変数を選択して追加することができます。全ての設定が終わったら、[OK] ボタンを押すと、文字列が配置されます。

7.4.7 return 文

return 文は、メソッドの呼び出し元に対して、呼び出されたメソッドが返す値を決定して、メソッドの処理を中途脱出するための文です。戻り値が void ではないメソッドの処理の最後に、必ずつけなければなりません*4。

パレットから「return 文」を選択して、配置したいメソッド上でクリックすると、return 文が追加されます。戻り値が void ではないメソッドに追加した場合は、プルダウンメニューで、戻り値を選択します。



図 7.19: return 文の配置: 戻り値が void ではないメソッドに追加した場合

戻り値が void であるメソッドに追加した場合は、中途脱出するだけです。



図 7.20: return 文の配置: 戻り値が void であるメソッドに追加した場合

7.4.8 continue 文

continue 文は、繰り返される処理を途中で止めて、繰り返しの最初に戻って処理を続けたいときに用います。そのため、集合操作/回数繰り返し/条件付繰り返しの構造のなかでのみ用いることができます*5。

パレットから「continue 文」を選択します。もしパレット上に「continue 文」が見つからない場合は、「break 文」の右側の ボタンを押すと、「continue 文」が表示されます。選択した後に、配置したいメソッド上でクリックすると、continue 文が配置されます。配置された後の設定は必要ありません。

*4 この return 文は、プログラミング言語における return 文と同じものです。

*5 この continue 文は、プログラミング言語における continue 文と同じものです。

7.4.9 break 文

break 文は、繰り返される処理を途中で止めて、繰り返しを中途脱出したいときに用います。そのため、集合操作/回数繰り返し/条件付繰り返しの構造のなかでのみ用いることができます*6。

パレットから「break 文」を選択して下さい。もしパレット上に「break 文」が見つからない場合は、「continue 文」の右側の ボタンを押すと、「break 文」が表示されます。選択した後に、配置したいメソッド上でクリックすると、break 文が配置されます。配置された後の設定は必要ありません。

7.5 アウトラインビューで構造を把握する

作成したアクションブロック図の構造を鳥瞰したいときは、アウトラインビューで表示します。Eclipse のメニューバーの [ウィンドウ]-[ビューの表示]-[アウトライン] を選択すると、アウトラインビューが表示され、Action Designer で表示されているアクションブロック図の構造が、ツリー構造で表示されます。

7.6 ソースコードを生成する

作成したアクションブロック図から、シミュレーションのためのソースコードを生成するには、Action Designer 上のキャンバスを右クリックして表示されるメニューから [ソースコード生成] を選択します。プログレスバーが表示された後に、生成されたソースコードのファイルが表示されます。

7.7 型の再読み込み

アクションブロック図の作成後にモデルに追加された各種 Type() を利用するには、型を再読み込みする必要があります。Action Designer 上のキャンバスを右クリックして表示されるメニューから [型の再読み込み] を選択します。プログレスバーが表示された後に、アクションブロック図に型が読み込まれます。

7.8 独自のアクションパーツの作成と管理

用意されているアクションパーツのほかに、新しくアクションパーツを作成して追加することができます。追加されたアクションパーツは、作成時に設定されたクラスで利用することができます。

便利な機能ですが、初心者は必ずしも使用する必要はありません。

*6 この break 文は、プログラミング言語における break 文と同じものです。

7.8.1 アクションパーツの作成

アクションパーツ化したい構造や文を選択して右クリックし、表示されたウィンドウから [アクションパーツの作成] を選択します。アクションパーツの作成ダイアログが表示されます。アクションパーツの作成は、そのアクションパーツで利用したい引数を設定して、引数の型を選択した後に、名前やカテゴリ、表示などを設定するという流れで進みます。

(1) 引数を選択する

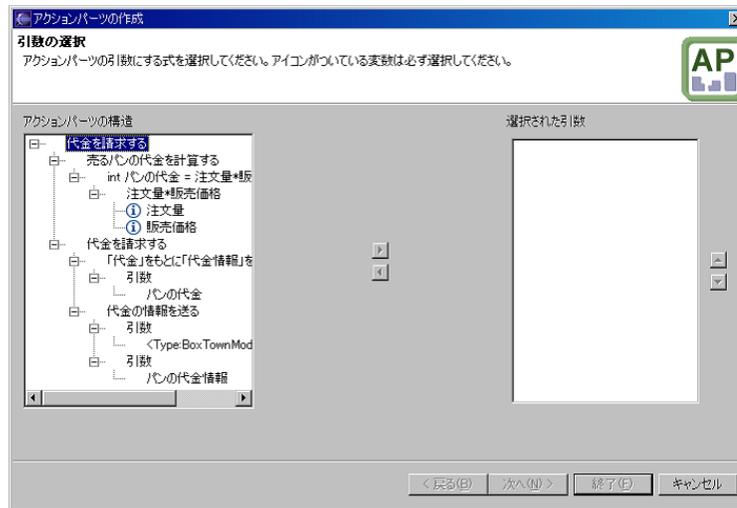


図 7.21: アクションパーツの作成ダイアログ: 引数を選択する

アクションパーツの作成ダイアログでは、はじめに、作成するアクションパーツを配置するときに設定したい変数を、引数として選択します。アクションパーツ化したい部分から外部の変数を参照している場合は、それらの変数には「!」アイコンが付加されて表示され、全て引数にする必要があります。同じ外部の変数を何度も参照していると、参照の回数だけ異なる引数として設定しなければなりません。

外部の変数を何度も参照している場合は、アクションパーツ化したい部分の先頭で、「変数宣言/代入」で新しく宣言する変数に代入しておき、その変数を参照するようにしましょう。そうすれば、いくつも引数を設定する必要はなくなります。引数としたい変数を選択し終わったら、[次へ] ボタンを押します。

(2) 選択した引数の型を設定する

引数としたい変数を選択したら、次はそれら引数となる型を設定します。引数とする変数の型を、プルダウンメニューから選択します。とくに型を変更する必要がなければ、[次へ] ボタンを押します。

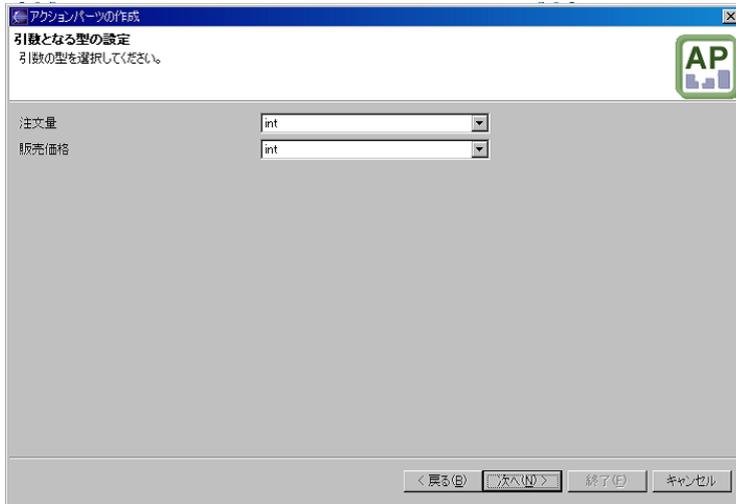


図 7.22: アクションパーツの作成ダイアログ: 選択した引数の型を設定する

(3) アクションパーツの名前、カテゴリなどを設定する

最後に、作成するアクションパーツの名前、追加カテゴリ、アクションパーツが利用できる型、アクションパーツ配置時の表示を設定します。

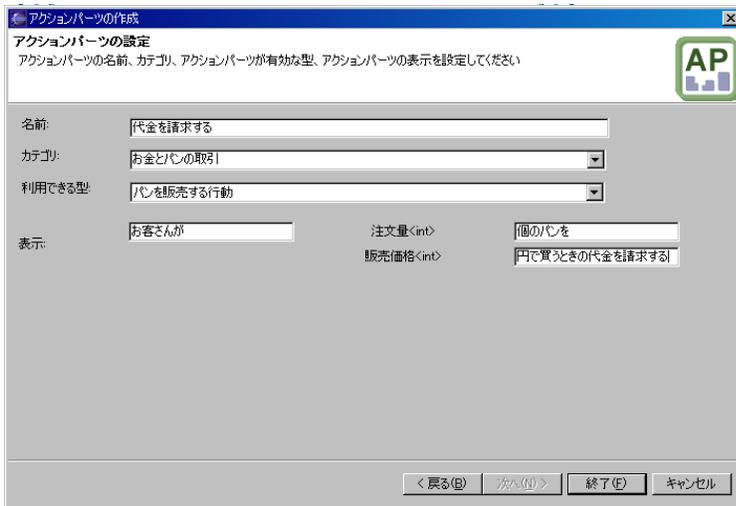


図 7.23: アクションパーツの作成ダイアログ: アクションパーツの名前、カテゴリなどを設定する

名前:

作成するアクションパーツの名前を入力します。

カテゴリ:

作成するアクションパーツを追加するカテゴリを設定します。既に存在するカテゴリに追加したい場合は、プルダウンメニューから選択します。新しくカテゴリを新

しく作成したい場合は、カテゴリの名前を入力します。

利用できる型:

作成するアクションパーツを利用できるクラスを設定します。汎用的なアクションパーツは、色々なクラスで使えることが望ましいですが、ある行動やエージェントに特化したアクションパーツは、その行動のみでしか使えないように制限することが望ましい場合があります。

表示:

作成したアクションパーツが配置されたときの表示を設定します。引数を挟んで文字列を入力することができます。アクションパーツの目的がわかるように設定して下さい。

7.8.2 アクションパーツの管理

作成したアクションパーツを削除したり、名前を変更したいときは、Action Designer上のキャンバスを右クリックして、表示されたウィンドウから [アクションパーツの管理] を選択します。アクションパーツの管理ダイアログが表示されます。



図 7.24: アクションパーツの管理ダイアログ

(1) アクションパーツを削除する

作成したアクションパーツを削除したい場合は、アクションパーツの管理ダイアログから削除したいアクションパーツを選択して [削除] ボタンをクリックします。

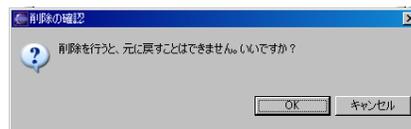


図 7.25: 削除の確認ダイアログ

削除の確認ダイアログが表示されるので、本当に削除してよければ [OK] ボタンを押す

と、アクションパーツが削除されます。なお、アクションパーツを削除しても、既に配置されたアクションパーツには影響はありません (配置・設定しているものは、消えないで残ります)。

(3) アクションパーツの名前を変更する

作成したアクションパーツの名前を変更したい場合は、アクションパーツの管理ダイアログから、名前を変更したいアクションパーツを選択して、[名前変更] ボタンをクリックします。名前を入力ダイアログが表示されるので、変更したい名前を入力して [OK] ボタンをクリックすると、アクションパーツの名前が変更されます。

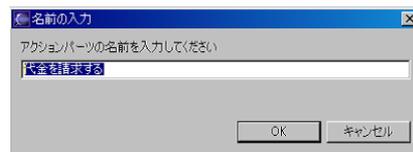


図 7.26: 名前を入力ダイアログ

なお、アクションパーツの名前を変更しても、変更前に配置されたアクションパーツには影響はありません。

第 8 章

World Composer

8.1 World ファイルを新規作成する

World Composer は、シミュレートする世界 (World) の初期設定を行うためのツールです。モデル作成プロセスにおいては、シミュレーションデザインフェーズで使用します。

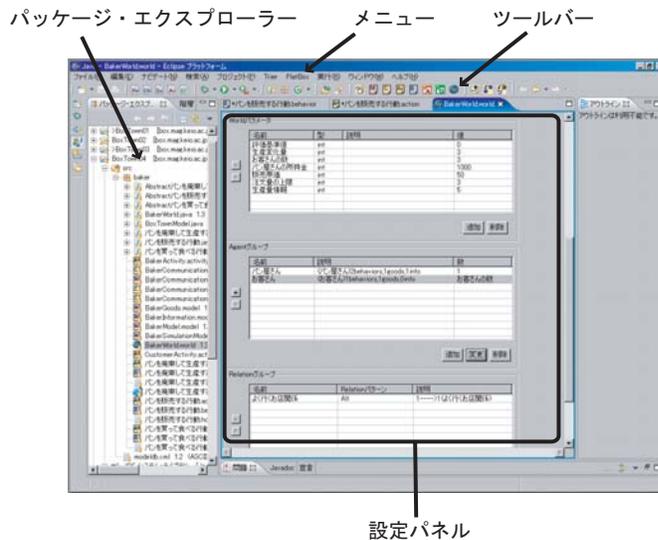


図 8.1: World Composer

8.2 World Composer を起動する

8.2.1 World を新規作成する

新しく World を作成するには、以下の手順で行います。

1. World を作成したいプロジェクトのパッケージを選択して、ツールバーから世界作成ボタン  をクリックします。
2. “World の新規作成” ダイアログが表示されるので、新しく作成したい World の名

前を入力して、[終了] ボタンを選択します。

3. 新しい.world ファイルが作られ、World Composer が起動します。

8.2.2 すでに作成した World を開く

すでに作成した World を開きたい場合は、パッケージ・エクスプローラー上で、World ファイルをダブルクリックします。もしくは、右クリックして [アプリケーションから開く]-[PlatBox World Composer] を選択します。World ファイルを読み込んだ World Composer が起動します。

8.3 World を設定する

8.3.1 モデルを読み込む

[モデルの読み込み] ボタンを押すと、Model Designer で作成した Type を読み込むことができます。この操作を行わないと、作成した Type が World Composer に反映されません。.world ファイルを作成した後や変更したりした後は、必ず最初にこのボタンを押してください。

8.3.2 World の基本設定

World の基本的な設定には、以下の項目があります。



表 8.1: World の設定

World の名前

World ファイルの名前が表示されます。

World の説明

作成する World についての説明を書きます。

Clock の設定

シミュレーション実行時に使用する時計を設定します。StepClock と RealClock の 2 つがありますが、通常、StepClock を使用してください。

乱数ジェネレータの設定

シミュレーションで使用する乱数ジェネレータ（呼び出すごとにランダムな数値を返してくれます）を追加します。最初から乱数ジェネレータは 1 つ設定されています。一様乱数以外のものや、新しく作成した乱数ジェネレータを使用したい場合は、ここで追加してください。

8.3.3 モデル全体が共有するパラメータの追加

World パラメータでは、モデル全体が共有するパラメータを設定します。ここで設定したパラメータは、World Composer の他の設定部分で参照することができます。また、PlatBox Simulator 上の World Initializer で、初期値を変更することができるようになります。

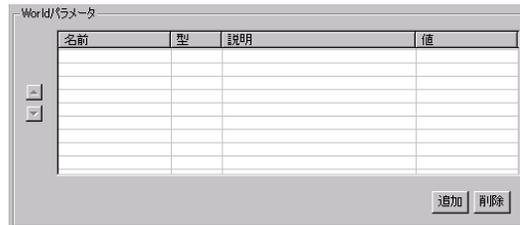


表 8.2: World パラメータの設定

名前

パラメータの名前を設定します。

型

パラメータの型を設定します。整数型の int、実数（浮動小数点数）型の double、long があります。

説明

パラメータの説明を書きます。この説明文は、PlatBox Simulator の World Initializer に表示されます。

値

パラメータの初期値を入力します。

8.3.4 Agent Group の追加

Agent グループとは、同じ AgentType, Behavior, Goods, Information をもつエージェントの集団のことです。Agent グループを、複数個設定することで、多様なエージェントが含まれるモデルを作成することができます。

名前	説明	数

表 8.3: Agent Group

Agent グループの [追加] ボタンを選ぶと、“Agent グループの設定” ダイアログが表示されます。

名前	説明	数
Bin ToyMode@AGENTTYPE_アサヒ		0
Bin ToyMode@AGENTTYPE_アサヒ		0

表 8.4: Agent グループの設定 ダイアログ

名前

すでに追加されている Agent Group の名前が表示されます。クリックすると、編集することができます。

説明

Agent Group に設定されている AgentType の名前と、Behavior, Goods, Information の種類の数が表示されます。

数

グループを構成するエージェントの数を設定します。詳細は、p.71 を参照してください。

BehaviorType

チェックボックスをオンにすると、その BehaviorType の Behavior がエージェントに追加されます。

GoodsType

チェックボックスをオンにすると、その GoodsType の Goods がエージェントに

追加されます。量の欄をクリックするとボタンが現れ、さらにそれをクリックすると、Goods の量を設定する別ダイアログが表示されます。詳細は、p.71 を参照してください。

InformationType

チェックボックスをオンにすると、その InformationType の Information がエージェントに追加されます。追加するときは、Information 実装クラスをクリックして、追加したい Information を選択してください。

8.3.5 Agent Group 間の関係の追加

どのエージェントとエージェントが関係を持つのかを設定します。Relation は、Agent Group ごとに結びます。クラス図でクラス間に関連を結んだのは、あくまでも関係をもつ可能性を記述したに過ぎません。具体的にどのように関係があるのかを、ここで設定するわけです。



表 8.5: Relation Group の設定

結び方は「双方向/ 一方向」の違いであり、Relation パターンは、エージェント同士をどのように関係づけるのかという 6 種類のが提供されています。

名前

Relation Group の名前を書きます。

Relation Type

使用する RelationType を選びます。

関係元の Agent グループ

関係元となる Agent Group を選びます。

関係先の Agent グループ

関係先となる Agent Group を選びます(Relation パターンが「Self」「Ring」「Small World」「Scale Free」の時は選べません)

方向

関係が双方向か一方向かを選びます(Relation パターンが「Self」「Ring」「Small World」「Scale Free」の時は選べません)

Relation パターン

Relation パターンを選択します。Relation パターンには、以下のものが用意されています。

	Relation パターン	説明
	All	全てに結ぶ
	Random	ランダムに結ぶ
	Self	自分自身にだけ結ぶ
	Ring	輪に結ぶ
	Small World	スモールワールド方式で結ぶ
	Scale Free	スケールフリー方式で結ぶ

8.3.6 Relation パターン別の設定

Random パターンで関係を結ぶ

Relation パターンが「Random」の場合には、以下の設定項目が追加されます。

- ・ 本数
結ぶ関係の本数を設定します。エージェントがランダムに選ばれて、関係を結びます。
- ・ 乱数ジェネレータ
使用する乱数ジェネレータを選択します。

Ring パターンで関係を結ぶ

Relation パターンが「Ring」の場合には、以下の設定項目が追加されます。

- ・ 距離
いくつ隣までを近隣としてリンクを張るかを指定します。
- ・ ループ設定
終端を「ループする」か「ループしない」かを選択します。特に問題がないかぎり、「ループする」にしておくことをおすすめします。
- ・ シャッフルのための乱数シード
Ring 上に並べる順番をシャッフルするための乱数のシード（種）を指定します。

Small World パターンで関係を結ぶ

Relation パターンが「Small World」の場合には、以下の設定項目が追加されます。

- ・乱数ジェネレータ
使用する乱数ジェネレータを選択します。
- ・つなぎなおす Relation の本数
Ring 型ネットワークからリンクを張りなおして Small World 型ネットワークを作る際に、何本つなぎなおすのかを指定します。
- ・シャッフルのための乱数シード
Ring 上に並べる順番をシャッフルするための乱数のシード（種）を指定します。

Scale Free パターンで関係を結ぶ

Relation パターンが「Scale Free」の場合には、以下の設定項目が追加されます。

- ・乱数シード
乱数のシード（種）を指定します。

8.3.7 Agent の数と Goods の量の設定

Agent の数や Goods の量の設定には、「固定値を入力する」「乱数ジェネレータを用いてランダムな値を入力する」「世界パラメータを参照する」という 3 通りの方法があります。

固定値を入力する

[値の直接入力] ラジオボタンを選んで、数値を入力します。

乱数ジェネレータを用いてランダムな値を入力する

[乱数の幅の入力] ラジオボタンを選んで、[最小値（以上）] と [最大値（未満）] を入力して、使用する乱数ジェネレータを選択します。

世界パラメータの参照する

[パラメータの選択] ラジオボタンを選んで、参照する世界パラメータを選択します。

8.4 作成した World を出力する

設定した World は、.java ファイルとして出力することで、PlatBox Simulator 上で実行できるようになります。World Composer の一番下にある [World の生成] ボタンを選択してください。“(.world ファイル名).java” というファイルが同じパッケージ内に出力されます。

8.5 シミュレーションの実行

8.5.1 シミュレーションの起動 (Component Builder 上での操作)

1. 実行する World の選択

パッケージ・エクスプローラ上で、実行したい World の java ファイルを選択します。

2. 実行

ツールバーにある 実行ボタンの右側の 部分を押し、実行メニューが表示されます。

実行メニューの中から [Run As...] [Java Application] を選ぶと、PlatBox Simulator が起動します。

8.5.2 シミュレーションの実行 (PlatBox Simulator 上での操作)

1. World の読み込み状態の確認

PlatBox Simulator の制御パネル上には、読み込まれた World の名前が表示されます。

2. ビューアーの表示

例えば、メニューから [ビューワ] [Communication Viewer] を選択して、Communication Viewer などを開きます。

3. シミュレーションの実行

シミュレーションの実行は、制御パネルの実行ボタンを押します。

8.5.3 PlatBox Simulator の終了

1. PlatBox Simulator の終了

PlatBox Simulator を終了するには、メインウィンドウの右上にある  ボタンを押すか、メニューから [ファイル] [終了] を選択します。

第9章

その他の補助ツールを使う

9.1 Boxed Economy プロジェクトからの移行を行う

PlatBox Simulator は、もともとは「Boxed Economy Simulation Platform」(BESP) という名称で呼ばれていたソフトウェアを発展・改名したものです。ここで問題になるのは、BESP 用に作成されたシミュレーションモデルは、そのままでは PlatBox Simulator 上では動作しないということです。そこで、現在の CB では、BESP 用に作成されたシミュレーションモデルを PlatBox 用のシミュレーションモデルに移行(変換)する機能が提供されています。

移行を行うには、対象となるプロジェクトを選択し、ツールバーから移行ボタンをクリックします。もしくは、ファイルメニューから [Plat Box]>[Immigrate to PlatBox] を選択します。

過去
に「Boxed
Economy
Simulation
Platform」
(BESP) を
利用していた
ユーザー向け
の情報です。

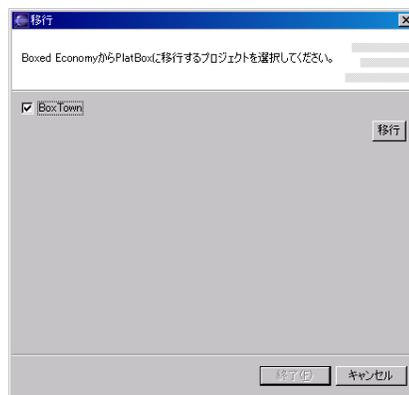


図 9.1: 移行ダイアログ

“移行”ダイアログで、PlatBox への移行を行うプロジェクトのチェックボックスにチェックをいれ、[移行] をクリックします。移行が完了したら、[終了] をクリックしてダイアログを閉じてください。

9.2 Type の名前を変更する

対象となるプロジェクトを選択し、ファイルメニューから [Plat Box]>[Rename Type] を選択します。名前を変えたい Type を選択し、名前を編集して [名前変更] をクリックします。Type の名前変更が完了したら、[終了] をクリックしてダイアログを閉じてください。



図 9.2: Type の名前変更ダイアログ

9.3 Template Action/GuardCondition を作成する

便利な機能ですが、初心者には必ずしも使用する必要はありません。

9.3.1 Template Action の新規作成

テンプレートアクションを新規作成するには、対象となるプロジェクトのパッケージを選択し、ツールバーからテンプレートアクション作成ボタンをクリックします。

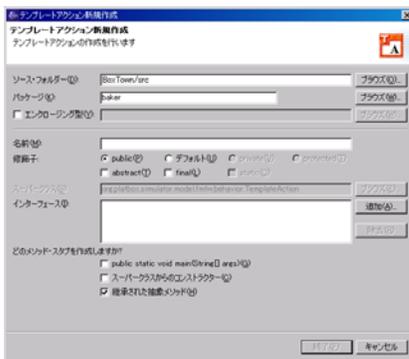


図 9.3: テンプレートアクション新規作成ダイアログ

「名前」を入力し、[終了] をクリックすると、指定したフォルダに、TemplateAction を継承したクラスの Java ファイルが新規作成されます。生成されたソースコードをもとに

実装したアクションテンプレートは、Behavior を作成する際に利用することができます。

9.3.2 Template Guard Condition の新規作成

テンプレートガードを新規作成するには、対象となるプロジェクトのパッケージを選択し、ツールバーからテンプレートガード条件作成ボタンをクリックします。

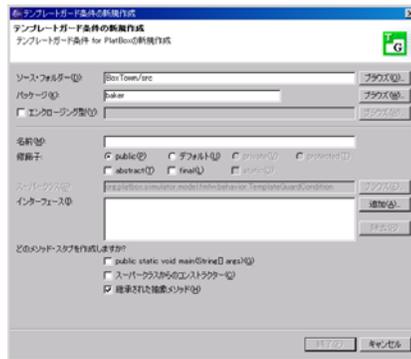


図 9.4: テンプレートガード条件新規作成ダイアログ

「名前」を入力し、[終了] をクリックすると、指定したフォルダに、TemplateGuardCondition を継承したクラスの Java ファイルが新規作成されます。生成されたソースコードをもとに実装したテンプレートガード条件は、Behavior を作成する際に利用することができます。

9.3.3 Behavior Designer でテンプレートを呼び出す

作成されたテンプレートは、Behavior Designer で呼び出すことができます。(Behavior Designer の詳しい操作方法は p.33 をご覧ください)。 Behavior Designer で、遷移をダブルクリックして “遷移のプロパティ” ダイアログを表示させます。

テンプレートアクション

テンプレートアクションを利用するには、利用可能なアクションの欄から選択します。もし表示されていない場合には、[利用可能なアクションの管理] からダイアログを表示させ、そこで [再読み込み] をクリックしてからダイアログを閉じます。

テンプレートガード条件

テンプレートガード条件を利用するには、ガード条件のプルダウンメニューから選択します。もし表示されていない場合は、[利用可能なガード条件の管理] からダイアログを表示させ、そこで [再読み込み] をクリックしてからダイアログを閉じます。

第 10 章

作成したモデルを公開する

10.1 図を印刷する

ツールバーから印刷ボタン  をクリックします。もしくは、[ファイル]>[印刷] を選択しても印刷することができます。

10.2 図を画像ファイルとして保存する

作成した図は、SVG 形式の画像ファイルとして保存できます*¹。画像を出力するためには、ツールバーから画像出力ボタン  をクリックします。もしくは、メニューバーから [画像出力]>[画像出力] を選択します。作成した図と同じフォルダに “.svg” と付いたファイルが作成されます。

10.3 作成中のモデルを共有する

複数のモデル作成者で、共同で作成している場合、そのモデルを共有したいことがあります。共有するためには、いくつかの方法がありますが、ここでは、圧縮ファイルとして出力するという方法を紹介します*²。

10.3.1 モデルを ZIP 形式でエクスポートする

対象となるプロジェクトを選択し、メニューバーから [ファイル]>[エクスポート] を開いてきます。次に、[ZIP ファイル] を選択し、[次へ] をクリックします。「To Zip File:」欄に出力先を指定し、[終了] をクリックします。指定した場所に、ZIP 形式の圧縮ファイルが出力されます。

*¹ SVG 形式とは、「Scalable Vector Graphics」(拡大縮小が可能なベクター方式のグラフィックス) 形式の略称で、拡大縮小・回転等をしてきれいなままの画像形式です。

*² 自分でサーバーが用意・設定ができるのであれば、CVS(Concurrent Versions System) を用いると、非常に便利です。

10.3.2 圧縮されているモデルをインポートする

まずインポートしたい ZIP ファイルを解凍します。解凍してできたフォルダを、workspace フォルダ (インストールガイドに従って設定していれば C:¥Program Files¥eclipseForPlatBox3.0¥workspace にあります) の中に移動します。

次に、Component Builder のメニューバーから [ファイル]>[インポート] を選択します。開いた “インポート” ウィンドウで、[既存プロジェクトをワークスペースへ] を選択し、[次へ] をクリックします。「プロジェクト内容:」の欄にある [ブラウズ] ボタンを押し、workspace 内に移動したインポートしたいフォルダを指定してください。[終了] をクリックすると、ファイルがインポートされ、パッケージ・エクスプローラーに表示されます。

10.4 完成したモデルを配布する

自分が作成したモデルを、自分や他の人が PlatBox Simulator 上で実行できるようにするためには、モデルを配布可能な形式で出力する必要があります。以下では、モデルを出力する方法と、その実行方法を説明します。

10.4.1 モデルを JAR 形式でエクスポートする

対象となる src を選択し、メニューバーから [ファイル]>[エクスポート] を開いてきます。次に、[JAR ファイル] を選択し、[次へ] をクリックします。エクスポート先を選択し、終了をクリックしてください。 .jar ファイルが生成されます。

10.4.2 PlatBox Simulator で実行する

変換した .jar ファイルを、実際に PlatBox Simulator で実行してみましょう。まず、PlatBox Simulator のフォルダ (おそらく、C:¥Program Files¥PlatBoxSimulator) にある plugins フォルダの中に、.jar ファイルをコピーします。

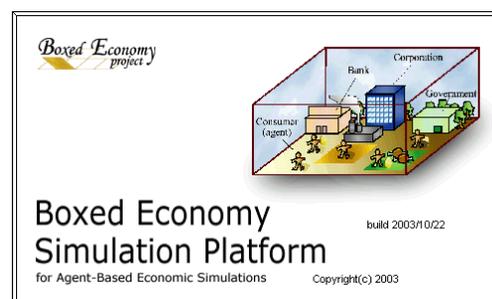
それから、PlatBox Simulator を起動し、PlatBox Simulator のメニューから [ファイル]>[世界を開く] を選び、実行したいファイルを選択して実行します。詳しくは、『PlatBox Simulator マニュアル』を参照してください。

PlatBox Project では、シミュレーション・プラットフォーム「PlatBox Simulator」およびシミュレーション作成支援ツール「Component Builder」を開発・提供しています。PlatBox Simulator は、シミュレーションを「動かす」ことによって、社会を理解するための道具です。これに加えて、私たちは、シミュレーションを「つくる」過程も社会を理解する助けになると考えています。「Component Builder」は、そのような理解を助けるモデル作成支援ツールです。この「動かすことでわかる」と「つくることでわかる」という二つのアプローチによって複雑な社会を理解する するための「新しい思考の道具」をつくるのが、私たちの目指すゴールです。

PlatBox Simulator は、Boxed Economy Project によって開発されてきた「Boxed Economy Simulation Platform」(BESP) の発展版です。Boxed Economy Project は、1999 年から、社会・経済のモデル化を支援する道具立てについて考えてきました。その後、私たちが提案するモデル化の枠組みは、経済分野に限定されたものではないと考え、2005 年にプロジェクト名を「PlatBox Project」に、ソフトウェアの名前を「PlatBox Simulator」に変更しました。

【Boxed Economy Project 時代の主なメンバー】

- 青山 希 (2002 年～現在)
- 浅加 浩太郎 (2001 年)
- 井庭 崇 (1999 年～現在) プロジェクトリーダー
- 海保 研 (2000 年～2002 年)
- 上橋 賢一 (2000 年～2002 年)
- 北野 里美 (2001 年～2002 年)
- 高部 陽平 (1999 年～2001 年)
- 武田 林太郎 (2003 年～2005 年)
- 田中 潤一郎 (2000 年～2002 年)
- 中鉢 欣秀 (2000 年～2001 年)
- 津屋 隆之介 (2000 年～2005 年)
- 広兼 賢治 (1999 年～2001 年)
- 松澤 芳昭 (2000 年～2004 年)
- 森久保 晴美 (2001 年～2002 年)
- 山田 悠 (2001 年～2005 年)



「Component Builder マニュアル」

執筆・編集者

PlatBox Project

青山 希

鈴木 祐太

津屋 隆之介

武田 林太郎

伊藤 智久

井庭 崇

Designers' Guide to Social Simulation, No.2

Component Builder マニュアル

2005 年 4 月 1 日 初版発行

2005 年 8 月 27 日 第 2 版発行

編著: PlatBox Project

〒252-8520 神奈川県藤沢市遠藤 5322 慶應義塾大学 井庭崇研究室

E-Mail: platbox@sfc.keio.ac.jp, もしくは iba@sfc.keio.ac.jp

Web: <http://www.platbox.org/>
